

AUTOMATISIERUNG KOMPLEXER CHEMISCHER SYNTHESVERFAHREN
AM BEISPIEL EINES OLIGOSACCHARID - SYNTHESAUTOMATEN

Dissertation zur Erlangung des akademischen Grades des
Doktors der Naturwissenschaften (Dr. rer. nat.)

eingereicht im Fachbereich Biologie, Chemie, Pharmazie
der Freien Universität Berlin

vorgelegt von

Frank Schuhmacher

aus Frankfurt am Main

2019

Die vorliegende Arbeit wurde in der Zeit vom Februar 2010 bis März 2017
unter der Leitung von Herrn Prof. Dr. Peter Seeberger
am Max-Planck-Institut für Kolloid- und Grenzflächenforschung
sowie in der Zusammenarbeit mit der Freien Universität Berlin angefertigt.

1. Gutachter: Prof. Dr. Peter H. Seeberger

2. Gutachter: Prof. Dr. Kevin Pagel

Disputation am 11.08.2020

*Standardisierung bedeutet die vollständige Beschreibung von Prozessen,
die unter gleichen Bedingungen, in gleicher Qualität,
zu 100% identisch sowie in beliebiger Menge und Häufigkeit reproduzierbar sind.*

Nur vollständig standardisierte Prozesse können automatisiert werden.

DANKSAGUNG

Besonders möchte ich an dieser Stelle Herrn Prof. Dr. Seeberger für die Überlassung dieses äußerst interessanten und spannenden Themas, der vielen Anregungen, Diskussionen und Bereitstellung der hervorragenden Arbeitsbedingungen am Institut, sowie sein mir entgegengebrachtes Vertrauen, danken. Herrn Prof. Dr. Seidel Morgenstern gilt mein Dank für die Möglichkeiten des fachlichen Austauschs und der Bereitstellung seiner Institutsinfrastruktur. Für die kurzfristige Übernahme des Zweitgutachtens geht mein aufrichtiger Dank an Prof. Dr. Pagel.

Allen Mitgliedern der Arbeitsgruppe Automation gilt mein Dank in besonderer Weise, durch die intensive Zusammenarbeit und die Vielzahl von Anregungen und konstruktiven Diskussionen erfahrene Unterstützung. Die zahlreichen Gespräche im Umfeld dieser Arbeit waren eine Bereicherung für den zukünftigen Lebensweg.

Einen großen Anteil an dem Erfolg dieser Arbeit tragen die Anwender der Systeme durch Ihre Bereitschaft Ihre Synthesen auch in der Entwicklungszeit der Synthesizer bereits hieran durchzuführen. Die hierbei vorrangig zu nennenden Kollegen wären: Dr. Jey Kandasamy, Dr. Matthan Hurevich, Dr. Fabian Pfrengle, Heung Sik Hahm, Dr. Chian-Hui Lai, Dr. Chien-Fu Liang, Dr. Richard Fair, Dr. Sandip Pasari, Dr. Ramesh Edupuganti, Dr. Lennart Lykke, Dr. Naresh Kottari, Dr. Max Bartetzko, Dr. Deborah Senf, Dr. Pietro Dallabernardina, Dr. Martina Delbianco und Dr. Priya Bharate.

Ein großer Dank geht auch an die netten Geister im Hintergrund, die erst die Realisierung in diesem Umfang ermöglichten. Dies sind insbesondere die mechanischen Werkstatt, sowie die umgesetzten Sonderwünsche im Zusammenhang mit Glasarbeiten, aber auch die Realisierung der elektrischen Besonderheiten und Zwischenversionen mit Teststellungen der IT Abteilung und der organisatorischen Zuarbeiten seitens des Büros.

Für eine Vielzahl an konstruktiven Gesprächen und informativen Austauschprozessen gebührt an dieser Stelle mein Dank an: Dr. Oliver Reich, Dr. Steffen Eller, Dr. Mark Schlegel, Dr. J. Cullen Klein, Dr. Markus Weishaupt.

Für die Flexibilität in Beschaffung- und Organisationsfragen ein großes Danke an Andreas Stockhaus, sowie bei technischen Belangen an Heiko Jung.

Inhalt

ABKÜRZUNGSVERZEICHNIS.....	I
PUBLIKATIONSLISTE	V
KURZZUSAMMENFASSUNG	IX
ABSTRACT	XI
1 EINLEITUNG UND ZIELSETZUNG	1
1.1 <i>Zucker - Der dritte Code des Lebens</i>	1
1.2 <i>Die Oligosaccharid-Synthese</i>	7
1.2.1 Die glykosidische Bindung	7
1.2.2 Glykosylierungsmethoden.....	11
1.3 <i>Oligosaccharidsynthese an der festen Phase</i>	14
1.3.1 Prinzip der Festphasenstrategie	14
1.3.2 Träger.....	17
1.3.3 Linker.....	19
1.3.4 Synthesestrategien	23
1.4 <i>Automation von Oligosaccharid –Synthesen</i>	26
1.4.1 Oligosaccharid-Synthesizer für die automatisierte Festphasen-Synthese.....	26
1.4.2 HPLC-unterstützte automatisierte Oligosaccharid-Synthese.....	33
1.4.3 Alternative Ansätze zur Automation von Oligosaccharid-Synthesen	35
1.5 <i>Zielsetzung der Arbeit</i>	38
2 AUSGANGSSITUATION & DETAILANALYSE DES EXISTIERENDEN SYSTEMS	52
2.1 <i>Kurze Marktübersicht für Peptid- und Oligonukleotid-Syntheseautomaten</i>	52
2.2 <i>Analyse des J-KEM – Systems</i>	53
2.2.1 Aufbau und prinzipielle Funktionsweise des J-KEM-Systems	53
2.2.2 Analyse der Gerätekommunikation	55
2.2.3 Analyse des Ansteuerungsverhaltens der Lösungsmittel-Ventile	60
3 KONZEPTIONIERUNG DER NEUEN GERÄTEGENERATION	61
3.1 <i>Grundkonzept des Projektaufbaus</i>	61
3.2 <i>Entwicklung des Gesamtsystems</i>	64
3.2.1 Entwicklungsplattform LabVIEW.....	64
3.2.2 Gerätesteuerung und Hardwareaufbau	74
3.2.3 Bediensoftware und Synthesesprache.....	86

4 DIE NEUE GENERATION DER SYNTHESEAUTOMATEN	89
4.1 <i>Überblick über das Gesamtsystem</i>	89
4.1.1 Anforderungen an den Synthesizer.....	89
4.1.2 Aufbau des Gesamtsystems.....	90
4.1.3 Transport von Lösungsmitteln und Zuckerbausteinen.....	93
4.2 <i>Aufbau und Entwicklung der Komponenten Hard- und Software</i>	95
4.2.1 Ventilsteuerung	95
4.2.2 Steuerung der Spritzenpumpen und Rotationsventile.....	110
4.2.3 Steuerung Druckverteilung.....	116
4.2.4 Steuerung des Fraktionssammlers.....	118
4.2.5 Steuerung der Temperatureinheit	121
4.2.6 Update der Arbeitssynthesizer	126
4.3 <i>Software der Benutzerschnittstelle</i>	129
4.3.1 Herstellen der Kompatibilität zu Syntheseskripten des Altsystems.....	129
4.3.2 Befehle, Module und Skripte.....	135
4.3.3 Prozesse	144
4.3.4 Synthesesteuerung mit der neuen Synthesizergeneration	148
4.4 <i>Synthesen an der neuen Gerätegeneration</i>	151
5 ZUKUNFTSAUSBLICK UND KONZEPTSTUDIE: „DAS CYBER-PHYSISCHE CHEMIE-SYSTEM“	170
5.1 <i>Einleitung.....</i>	170
5.2 <i>Weiterentwicklung der Hardware.....</i>	175
5.2.1 Zusatz-Module und hardwaretechnische Verbesserungen	175
5.2.2 Hardware-Modularisierung und Standardisierung	178
5.2.3 Prototyp eines druckgesteuerten Manifolds.....	181
5.2.4 Fotoreaktor – Prototyp für die fotochemische Synthese	184
5.2.5 Integration von Online-NMR, NMR Simulation, Validierung & Konsolidierungsprozesse	192
5.2.6 Makerboards, Internet der Dinge und Internet 4.0	197
5.3 <i>Weiterentwicklung der Synthesizer-Software.....</i>	204
5.3.1 Übersicht der Bedienebenen der Benutzerschnittstellen	204
5.3.2 Grafische Bedienoberfläche des Synthesizers	207
5.3.2.1 Schnellparametrier – Benutzeroberflächen	207
5.3.2.2 Der Building Block Selektor zur vereinfachten Dateneingabe.....	213
5.3.2.3 Chemiebasiertes Benutzer-Frontend.....	216
5.3.2.4 Molekül-Editor.....	222
5.3.2.5 Glykan-Editor.....	224

Inhaltsverzeichnis

5.3.3	Informationsverarbeitung und Wissensmanagement mit ELN und LIMS.....	227
5.3.4	Datenarchivierung & Datenmanagement	235
5.3.5	Datenbanken - Datenabgleich und Validierung	241
5.3.6	Format - Konverter & Kontrollen	246
5.3.7	Arbeitssicherheit.....	252
5.3.8	Syntheseplanung & Beschreibungssprache.....	255
5.4	<i>Expertensystem – „Maschinelles Lernen“, Simulation, Syntheseplanung</i>	259
5.4.1	Einleitung.....	259
5.4.2	Datenaufbereitung, Maschinelles Lernen & Modell-Entwicklung	261
5.4.3	Simulation & KI	266
ZUSAMMENFASSUNG.....		270
NACHWORT.....		273
LITERATURVERZEICHNIS		275
ABBILDUNGSVERZEICHNIS		304
TABELLENVERZEICHNIS		320
FORMELVERZEICHNIS.....		320
LEBENS LAUF.....		321
ANHANG		323

ABKÜRZUNGSVERZEICHNIS

6DOF	Six Degrees of Freedom (sechs Freiheitsgrade)
Ac	Acetyl CH ₃ CO
AcCl	Acetylchloride
AcN	Acetonitrile
AcOH	Essigsäure, acetic acid
Ac ₂ O	Essigsäureanhydrid, acetic anhydride
All	Allyl
API	application programming interface (Anwendungsprogrammierschnittstelle)
Araf	Arabinofuranose
Äq.	Äquivalent
Ber.	berechnet
BF ₃ *OEt ₂	Bortrifluorid-Diethyltherat
BML	Bipartite Local Model
Bn	Benzyl
Boc	tert-Butyloxycarbonyl
Bu	Butyl
Bz	Benzoyl
C	Konzentration
CID	Compound Identifier (in PubChem database)
CH	Cyclohexan
COSY	Correlated Spectroscopy Korrelationspektroskopie
CRN	Chemical Reaction Network
d	Dublett – NMR
DAQ	Data Acquisition
DC	Dünnschichtchromatographie
DCE	1,2-Dichlorethan
DCM	Dichlormethan
DDQ	2,3-Dichlor-5,6-dicyano-1,4-benzochinon
Dest.	destilliert
DMF	N,N-Dimethylformamid
DMS	Dimethylsulfid
DMSO	Dimethylsulfoxid
ELN	Elektronisches Labor-Notizbuch
ERT	extrem randomisierte Bäume
Et	Ethyl
EtOAc	Essigester(EE)
et al.	und anderen
EtOH	Ethanol
Eq	Äquivalente
Fmoc	9-Fluorenylmethoxycarbonyl
Fuc	Fucose
Gal	D-Galaktose
GalN	D-Galactosamin
GalNAc	N-Acetyl-D-galactosamin
GC	Gaschromatographie
GlcN	D-Glucosamin
GND	Chassis Ground (Elektronik: Bezugspotential)
GPI	Gycosylphosphatidylinositol
HELM	Hierarchical Editing Language for Macromolecules
HMDB	Human Metabolome Database

Abkürzungsverzeichnis

HMBC	Heteronuclear Multiple-Bond CH Correlation (NMR-Spektrum)
HPLC	High Pressure/Performance Liquid Chromatography (Hochdruck/Hochleistungs Flüssigkeits Chromatographie)
HSQC	Heteronuclear Single Quantum Correlation (NMR-Spektrum)
HTTP	Hypertext Transfer Protokoll
IdoA	L-Iduronsäure
IMS	Ion Mobility Spectrometer
InChI	International Chemical Identifier Code
InChIKey	Key of International Chemical Identifier
IR	Infrarotspektroskopie
IUPAC	International Union for Pure and Applied Chemistry
J	Kopplungskonstante
kNN	k-Nearest-Neighbor-Algorithmus
LC	Liquid-Chromatographie
Lex	Lewis-X
Lev	Levulinoyl(4-Oxopentansäure)
LG	Leaving Group, Abgangsgruppe AG
LIMS	Labor – Informationsmanagementsystem
m	Multiplett (NMR-Spektrum)
MALDI	Matrix Assisted Laser Desorption Ionisation (Matrixunterstützte Laserdesorptionionisierung)
Man	D-Mannose
Me	Methyl
MeOH	Methanol
MIRAGE	Minimum Information Required for A Glycomics Experiment
MIAPE	Minimum Information About a Proteomics Experiment
ML	Maschinelles Lernen
MQTT	Message Queuing Telemetry Transport
MS	Massenspektroskopie
MW	Mikrowelle
NFC	Near Field Communication (Nahfeldkommunikation)
NIS	N-Iod-Succinimid
NMR	Nuclear magnetic resonanz (Kernresonanz-Spektroskopie)
Nu	Nucleophil
NOESY	Nuclear Overhauser Effect Spectroscopy (NMR-Spektrum)
OAc	Acetat
OAIS	Open Archival Information System
PG	Protecting Group (Schutzgruppe SG)
Ph	Phenyl
PLN	Papier Labor-Notizbuch
PS	Polystyrol
PTFE	Polytetrafluorethylen
Py	Pyridin
R	Organischer Rest
RDF	Resource Description Framework
Rf	Retentionsfaktor
RF	Random-Forests
Rt	Retentionszeit
RFID	Radio-frequency Identification
RP	Reversed Phase (Umkehrphase)
RT	Raumtemperatur
s	Singulett (NMR-Spektrum)

Abkürzungsverzeichnis

SATPdb	Structurally Annotated Therapeutic Peptide database
SCUFL	Simple Conceptual Unified Flow Language
SMILES	Simplified Molecular Input Line Entry Specification
SPOS	Solid Phase Oligosaccharide/Organic Synthesis
SPS	Speicherprogrammierbare Steuerung
SubVI	Unterprogramm der grafischen Programmiersprache LabVIEW
SVM	Support Vector Machines
t	Triplett (NMR-Spektrum)
TFA	Trifluoressigsäure
THF	Tetrahydrofuran
TMOF	Trimethylortoformiat
TMSOTf	Trimethylsilyltrifluormethansulfonsäure
TMS	Trimethylsilyl
Tol	Toluol
TOCSY	Total Correlated Spectroscopy (NMR-Spektrum)
URL	Uniform Resource Identifier
UV	Ultraviolett
VBA	Visual Basic for Applications (Programmiersprache)
VI	Virtual Instrument (Funktionseinheit der grafischen Programmiersprache LabVIEW)
VIS	Visible Spectroscopy
WURCS	Web3 Unique Representation of Carbohydrate Structures
WWW	World Wide Web
XG	Xyloglucan
XML	Extensible Markup Language (Erweiterbare Auszeichnungssprache)

PUBLIKATIONSLISTE

Kandasamy, Jeyakumar; Schuhmacher, Frank; Hahm, Heung Sik; Klein, James C.; Seeberger, Peter H. (2014): **Modular Automated Solid Phase Synthesis of Dermatan Sulfate Oligosaccharides.**

In: *Chem. Commun.* 50 (15), S. 1875. DOI: 10.1039/c3cc48860h.

Bartetzko, Max P.; Schuhmacher, Frank; Hahm, Heung Sik; Seeberger, Peter H.; Pfrengle, Fabian (2015): **Automated Glykan Assembly of Oligosaccharides Related to ArabinoGalaktan Proteins.**

In: *Organic letters* 17 (17), S. 4344-4347. DOI: 10.1021/acs.orglett.5b02185.

Schmidt, Deborah; Schuhmacher, Frank; Geissner, Andreas; Seeberger, Peter H.; Pfrengle, Fabian (2015): **Automated Synthesis of Arabinoxylan-Oligosaccharides Enables Characterization of**

Antibodies that Recognize Plant Cell Wall Glykans. In: *Chem. Eur. J.* 21 (15), S. 5709-5713. DOI: 10.1002/chem.201500065.

Dallabernardina, Pietro; Schuhmacher, Frank; Seeberger, Peter H.; Pfrengle, Fabian (2016):

Automated Glykan Assembly of Xyloglucan Oligosaccharides. In: *Organic & biomolecular chemistry* 14 (1), S. 309-313. DOI: 10.1039/c5ob02226f.

Wilsdorf, M.; Schmidt, D.; Bartetzko, M. P.; Dallabernardina, P.; Schuhmacher, F.; Seeberger, P. H.; Pfrengle, F. (2016): **A Traceless Photocleavable Linker for The Automated Glykan Assembly of**

Carbohydrates with Free Reducing Ends. In: *Chem. Commun.* 52 (66), S. 10187-10189. DOI: 10.1039/C6CC04954K.

Heung Sik Hahm, Chien-Fu Liang, Chian-Hui Lai, Richard J. Fair, Frank Schuhmacher and Peter H. Seeberger (2016): **Automated Glykan Assembly of Complex Oligosaccharides Related to Blood**

Group Determinants. In: *The Journal of organic chemistry*, 81 (14), S. 5866-5877.

DOI: 10.1021/acs.joc.6b00554.

Bartetzko, Max P.; Schuhmacher, Frank; Seeberger, Peter H.; Pfrengle, Fabian (2017): **Determining Substrate Specificities of β 1,4-Endogalaktanases Using Plant Arabinogalaktan Oligosaccharides**

Synthesized by Automated Glykan Assembly. In: *The Journal of organic chemistry*, 82 (3),

S. 1842–1850. DOI: 10.1021/acs.joc.6b02745.

Dallabernardina, Pietro; Schuhmacher, Frank; Seeberger, Peter H.; Pfrengle, Fabian (2017):

Mixed-Linkage Glucan Oligosaccharides Produced by Automated Glykan Assembly Serve as Tools to Determine the Substrate Specificity of Lichenase. In *Chemistry (Weinheim an der Bergstrasse, Germany)*, 23 (13), S. 3191–3196. DOI: 10.1002/chem.201605479

Senf, Deborah; Ruprecht, Colin; Kruijff, Goswinus H. M. de; Simonetti, Sebastian O.; Schuhmacher, Frank; Seeberger, Peter H.; Pfrengle, Fabian (2017): **Active Site Mapping of Xylan-Deconstructing Enzymes with Arabinoxylan Oligosaccharides Produced by Automated Glycan Assembly**. In *Chemistry (Weinheim an der Bergstrasse, Germany)*, 23 (13), S. 3197–3205.

DOI: 10.1002/chem.201605902

Heung Sik Hahm, Mark K. Schlegel, Mattan Hurevich, Steffen Eller, Frank Schuhmacher, Johanna Hofmann, Kevin Pagel and Peter H. Seeberger (2017): **Automated Glycan Assembly Using the Glyconeer 2.1 Synthesizer**. *Proceedings of the National Academy of Sciences of the United States of America*, 114, E3385-E3389, DOI: 10.1073/pnas.1700141114

Andreas Geissner, Anika Reinhardt, Christoph Rademacher, Timo Johannssen, João Monteiro, Bernd Lepenies, Michel Thépaut, Franck Fieschi, Jana Mrázková, Michaela Wimmerova, Frank Schuhmacher, Sebastian Götzte, Dan Grünstein, Xiaoqiang Guo, Heung Sik Hahm, Jeyakumar Kandasamy, Daniele Leonori, Christopher E Martin, Sharavathi G Parameswarappa, Sandip Pasari, Mark K Schlegel, Hidenori Tanaka, Guozhi Xiao, You Yang, Claney L Pereira, Chakkumkal Anish, Peter H Seeberger (2019): **Microbe-focused Glycan Array Screening Platform**. *Proceedings of the National Academy of Sciences*, 116 (6), S. 1958-1967, DOI: 10.1073/pnas.1800853116

POSTER

Frank Schuhmacher and Peter H. Seeberger (2015): **New Generation of Synthesizers**: Flexibility modularity and cascading will change the automated chemical world.

Frank Schuhmacher and Peter H. Seeberger (2016): **Automated Oligosaccharide Syntheses meets Bioinformatics**. Chemdraws Objects defines Reaction Parameters, fills Database and allows Analysis.

Frank Schuhmacher and Peter H. Seeberger (2016): **A Photoreactor Extends the Synthesizer Functionality**, Flexible Light Modul for the new Generation of Oligosaccharid Synthesizers.

VORTRÄGE

„Automating Design and Synthesis of Oligosaccharides “

6th Glycan Forum Berlin, 19. - 21.03.2012

„Maximum Flexibility in the Automation of Laboratory Processes Through Modularization and Cascading of Components“

Ringberg Conference 2014, 21. – 25.09.2014

„Modular synthesis system as a tool to explore the Glycogalaxy“

4th Beilstein Glyco-Bioinformatics Symposium, 22. – 26.06.2015

„Quality Management – the Death of Scientific Freedom“

Beilstein Open Science Symposium, 22. – 24.05.2017

KURZZUSAMMENFASSUNG

Diese Arbeit beschreibt eine neue technische Methode zur Automatisierung komplexer chemischer Prozesse. Die vorgestellten Konzepte und Verfahren konnten durch den eigenverantwortlichen Aufbau von drei Oligosaccharid-Syntheseautomaten, deren Bereitstellung und dauerhaften Einsatz im Laborbetrieb bewiesen werden. Alle gerätetechnischen Komponenten wurden aus handelsüblichen Standards entwickelt, die zur Verwendung im Syntheseautomaten erweitert und angepasst wurden. Der Geräteentwicklung liegt dabei ein vollständig soft-, als auch hardwareseitig modularisiertes Konzept zugrunde. In der dezentralen Gerätesteuerung arbeiten die realisierten Zustandsmaschinen der einzelnen Gerätekomponenten autonom ihre übergebenen Aufgaben ab. Die übergreifende Kommunikation findet über Befehlswarteschlangen und dynamische Ereignisstrukturen statt. Dies führt zu extrem hoher Systemstabilität und Fehlertoleranz im Synthesebetrieb. Die fehlerfreie Ausführung von Langzeitsynthesen ermöglicht die wiederholte Herstellung komplexer, langkettiger Strukturen (20-mer, 30-mer, sowie 50-mer). Mit über 250 h kontinuierlicher Synthesezeit entstand das im Augenblick längste automatisiert synthetisierte Polysaccharid.

Die Systeme zeichnen sich aus durch einfache Bedienbarkeit, geringen Einarbeitungsaufwand, leichte Integration neuer Prozess-Schritte, sowie eine sichere und funktional geführte Synthesedefinition und Herstellung von Oligo- und Polysaccharid-Synthesen aus eingekoppelten Monomeren.

Schwerpunkte der technischen Realisation waren Optimierung der Syntheseeffizienz, Verbesserung der Reproduzierbarkeit, sowie die flexible Erweiterbarkeit des Gerätekonzepts. Es entstand eine Universalplattform, deren gesamtes Prozessmanagement nicht auf Oligo-/Polysaccharide beschränkt ist, sondern mit Anpassung der Steuerungsmodule und modularer Erweiterung der

Versorgungskomponenten, auch Peptide, DNA, etc. sowie Kombinationen davon synthetisieren kann. Verschiedene im System implementierte Schnittstellen ermöglichen übergreifende Interaktionen zu den Forschungsgebieten der Chemometrik, der physikalischen Chemie sowie der Chemoinformatik.

Die Weiterentwicklungsmöglichkeiten als multidisziplinäre, interoperable, vernetzte Plattform, welche anforderungsbezogen beliebige Moleküle synthetisieren kann, werden im Kapitel Zukunftsaussicht in der abschließenden Konzeptstudie beschrieben. Die dezentrale, autonome Steuerungsarchitektur, sowie die streng definierten Anwendungsebenen des realisierten Synthesizers, ermöglichen weltweit erstmalig die Hardwaresteuerung direkt aus einem Chemieprogramm (ChemDraw) durch Aufstellen der Reaktionsgleichung. Das Konzept des cyber-physischen „Chemie-Systems“ implementiert aktuelle technologische Entwicklungen zum Aufbau einer umfassenden Synthese- / Moleküldatenbank. Unter Verwendung des „maschinellen Lernens“, können Methoden und Verfahren zur Prozessoptimierung und kybernetischen Modellbildung für die Simulation und Vorhersage chemischer Reaktionen zum Einsatz kommen.

ABSTRACT

This thesis describes a new technical method for the automation of complex chemical processes. The presented concepts and procedures could be proved by the self-responsible construction of three oligosaccharide synthesis machines, their provision and permanent use in the laboratory. All device components were developed from commercially available standards that have been extended and adapted for use in the synthesizer. Device development is based on a completely soft- as well as hardware-based modularized concept. In the decentralized device control, the realized state machines of the individual device components work autonomously on their transferred tasks. The overall communication takes place via command queues and dynamic event structures. This leads to extremely high system stability and fault tolerance in the synthesis mode. The error-free execution of long-term syntheses allows the repeated production of complex, long-chain structures (20-mer, 30-mer such as 50-mer). With over 250 h of continuous synthesis time, the longest automated synthesized polysaccharide was produced on the automatic synthesizer.

The systems are characterized by simple operation, low training period, easy integration of new process steps, as well as a safe and functionally guided synthesis definition and production of oligo- and polysaccharide syntheses from monomers. The focus of the technical realization was optimization of the synthesis efficiency, improvement of the reproducibility, as well as the flexible extensibility of the device concept. The result is a universal platform whose overall process is not limited to oligosaccharides / polysaccharides. With adaptation of the control modules and modular extension of the supply components, the synthesis of peptides, DNA and combinations thereof is also possible. Various interfaces implemented in the system enable cross-functional interactions with the research areas of chemometrics, physical chemistry and chemoinformatics.

In the final concept study, the further development possibilities are described as a multidisciplinary, interoperable, networked platform that can synthesize any desired molecules. The decentralized, autonomous control architecture, as well as the strictly defined application levels of the realized synthesizer, enable at the first time a hardware control directly from a chemistry program (ChemDraw) by setting up the reaction equation.

The concept of the cyber-physical "chemistry system" implements current technological developments to build a comprehensive synthesis and molecular database. Using "machine learning", methods and procedures for process optimization and cybernetic modeling can be used to simulate and predict chemical reactions.

1 Einleitung und Zielsetzung

1.1 Zucker - Der dritte Code des Lebens

Komplexe Kohlenhydrate, die auch Glykane¹ genannt werden, vereinen in Form von Stärke, Viskose, Cellulose oder Chitin über 80% der weltweiten Biomasse auf sich. Der überwiegend wahrgenommene Teil der Kohlenhydrate ist für den strukturellen mechanischen Aufbau von biologischen Makrostrukturen und für die Energiespeicherung verantwortlich.

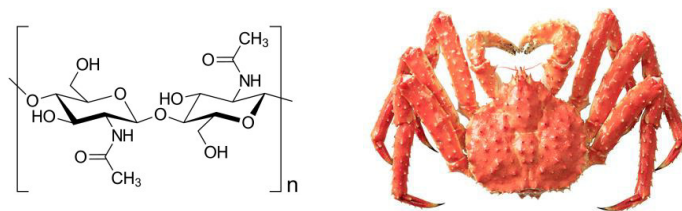


Abbildung 1: Chitin ist als eines der weitest verbreiteten Glykane für den strukturellen Aufbau von Pflanzen und Tieren verantwortlich, z. B. im Exoskelett der Riesenkrebse.

Glykane sind nach aktuellem Forschungsstand an vielen Zell-Zell-Interaktionen und an biologischen Prozessen^[1] beteiligt. Sowohl bei Entzündungen, Infektionen, Immunreaktionen aber auch bei der Entstehung neuen Lebens spielen Zuckermoleküle eine zentrale Rolle. Zahlreiche virale Krankheitserreger, z. B. Noro-, Rota-, Influenzaviren, nutzen Glykane als Rezeptoren. Dies gilt ebenso für das Verständnis über die Metastasenbildung in der Krebsforschung, als auch bei den Wirkmechanismen moderner Impfstoffe (z.B. für Streptokokken).

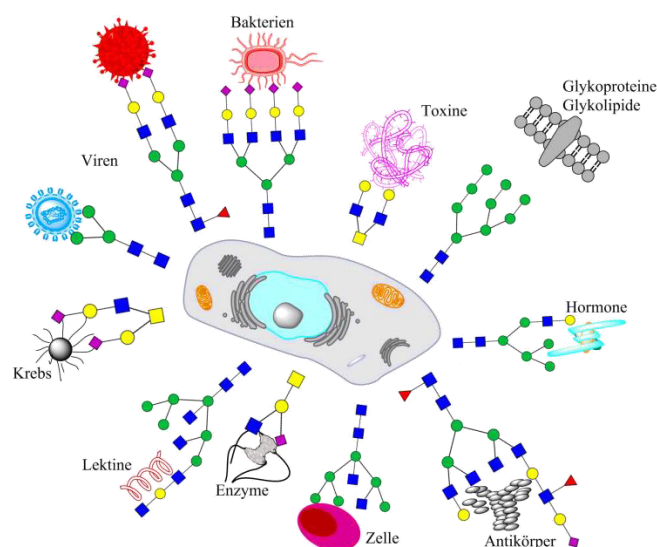


Abbildung 2: Schematische Darstellung verschiedener Zell-Interaktionen mit Beispielen von Zuckermolekülen^[2,3,4a,5,4b,6], die dabei eine Rolle spielen.

¹ Glykane werden auch als Polysaccharide bzw. Vielfachzucker bezeichnet.

Neben den Proteinen (Eiweiße), den Nukleinsäuren (DNA², RNA³) und den Lipiden (Fette) stellen die Kohlenhydrate eine eigenständige Klasse der wichtigsten Biomoleküle dar.^[1] Mit Ausnahme der Lipide liegen diese Stoffklassen als Polymere vor oder lassen sich als solche verknüpfen. Dabei kann aus einfachen, monomeren Bausteinen durch Kombination eine riesige Anzahl komplexer Makromoleküle aufgebaut werden.

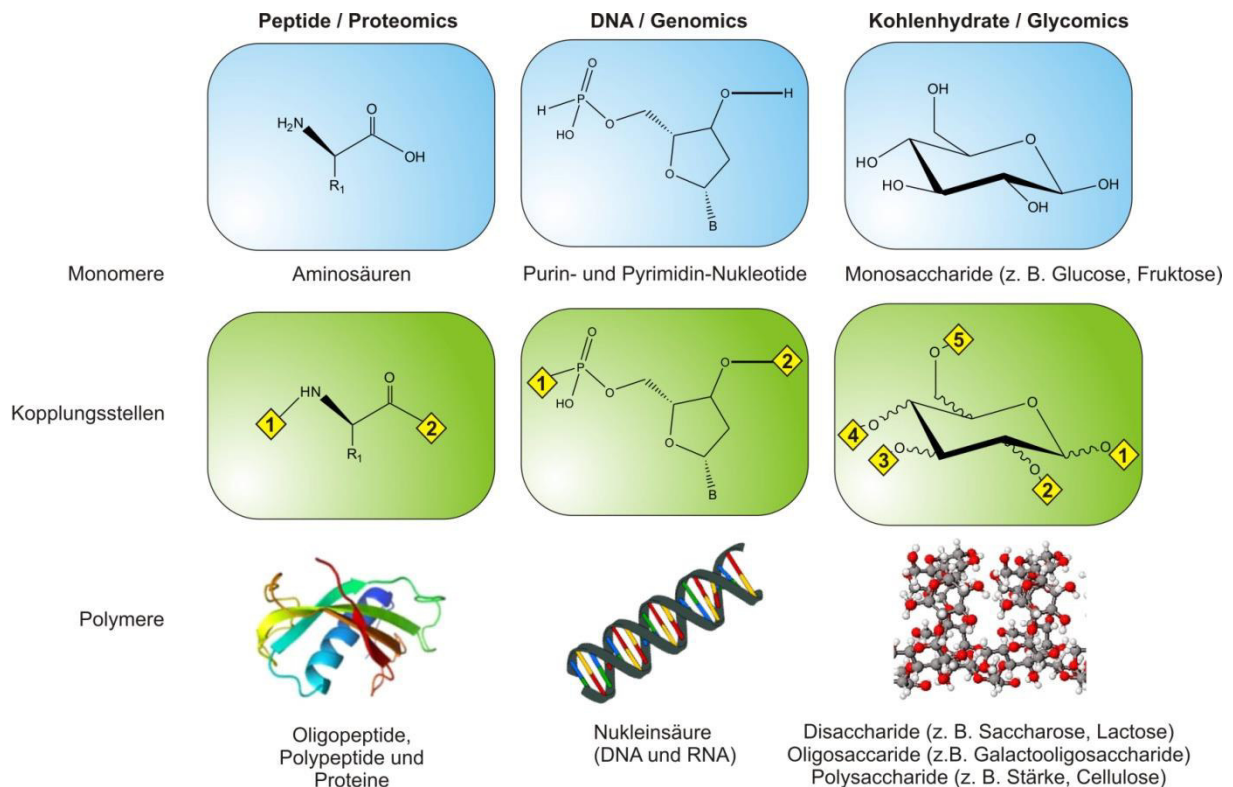


Abbildung 3: Vergleich der wichtigsten Biomoleküle und deren Ankopplungsstellen bei der Synthese von Biopolymeren. Bei den Zuckermolekülen erhöhen bereits die fünf möglichen Kopplungsstellen die Komplexität bei der Synthese, aber zusätzlich auch deren sterische Ausrichtung.

Das 1963 vorgestellte und damit das älteste Verfahren zur automatisierten Synthese von Biopolymeren ist die Festphasen-Peptidsynthese nach Merrifield^[7,8]. Jede Aminosäure verfügt über zwei Kopplungsstellen, mit einem N-terminalen oder einem C-terminalen Ende, aus denen sich durch lineares Andocken die einzelnen verschiedenen Aminosäuren Molekülketten aufbauen.

Aminosäureketten mit bis zu 10 Monomeren werden hierbei als Oligopeptide, längere Ketten als Polypeptide bezeichnet. Proteine sind Polypeptide mit über 100 Aminosäuren. Das größte bisher bekannte Protein Titin (Zellskelett) besteht aus einer Kettenlänge von mehr als 30.000 Aminosäuren und beinhaltet 320 stabile, meist kompakte Faltungsstrukturen, sogenannte Protein-Domänen^[9].

Proteine erfüllen wichtige strukturelle Funktionen, z. B. in Muskeln und Bindegewebe im menschlichen Körper und machen mit 18% den größten Anteil an einer menschlichen Zelle aus.

² DNA ist das Akronym des englischen „Deoxyribonucleic Acid“.

³ RNA ist das Akronym des englischen „Ribonucleic Acid“.

Basierend auf dem Vorbild der Festphasensynthese gelang 1981 die automatisierte Herstellung von Oligodeoxynukleotiden.^[8] Die Bausteine dieser Makromoleküle sind die sogenannten Nukleotide, die sich jeweils aus einer Phosphorsäure, einem Zucker mit fünf Kohlenstoffatomen (einer Pentose) und einer Nukleinbase zusammensetzen. Die Verkettung der Nukleotide zu einem Makromolekül erfolgt über eine Phosphordiester-Bindung zwischen Einfachzucker und Phosphorsäureester der jeweiligen Nukleotidbausteine. Jede Nukleinsäure hat ein freies 5'- und 3'-Ende. Die Verlängerung erfolgt biologisch immer linear am 3'-Ende.

Desoxyribonukleinsäure ist der Speicher der Erbinformation. Die drei Hauptklassen der Ribonukleinsäuren, die mRNA⁴, auch Boten-RNA genannt, die ribosomale RNA und die Transfer-RNA, sind an der Translation⁵ der genetischen Information aller lebenden Zellen bzw. Organismen beteiligt. Damit sind sie für die Übersetzung des Erbguts in den Aminosäuren-Code der Proteine verantwortlich, tragen zur Regulation der Gene bei oder besitzen – ähnlich einem Enzym - katalytische Funktionen.^[10]

Weitere drei Jahrzehnte später, im Jahr 2001, wurde die erste automatisierte Festphasen-Synthese von komplexen Kohlenhydratsystemen realisiert. Aufgrund zahlreicher reaktiver Hydroxylgruppen als mögliche Kopplungsstellen, stellt die Synthese komplexer Zuckermoleküle eine Herausforderung dar. Durch *glykosidische Bindungen* lassen sich nicht nur lineare, sondern auch verzweigte und mehrfach verzweigte Molekülstrukturen aufbauen. Neben Monosaccharid-Einheiten werden die Polymere als Disaccharide, Oligosaccharide mit drei bis zehn verbundene Monosacchariden und Polysaccharide bestehend aus mehr als zehn Monosacchariden, bezeichnet^[11]. Aufgrund ihrer unterschiedlichen glykosidischen Verlinkung aus Monomeren, werden so variantenreiche, biologische Makrostrukturen aufgebaut (Abbildung 1).

Die drei großen Klassen der Biopolymere sind also prinzipiell für drei unabhängige biologische „Codes“ zuständig. Am Aufbau von Proteinen im menschlichen Körper beteiligen sich 20 unterschiedliche Aminosäuren (proteinogene Aminosäuren). Weitere proteinogene Aminosäuren sind Selenocystein (U), das als Baustein einiger Enzyme zur Katalyse von Redoxreaktionen in menschlichen Proteinen vorkommt^[12], Pyrrolysin (O), das sich im katalytischen Zentrum der Proteine von Methan-produzierenden Bakterien befindet und für deren Energiegewinnungsprozesse essentiell ist^[13], sowie Formylmethionin (fMet), das als erste Aminosäure in alle Proteine bei der bakteriellen Proteinbiosynthese eingebaut wird.^[14]

⁴ Abkürzung: mRNA für engl. „messenger RNA“

⁵ Translation wird in der Biologie die Synthese von Proteinen in den Zellen lebender Organismen bezeichnet.

Peptide sind grundsätzlich linear aufgebaut. Zur Beschreibung der linearen Aminosäure-Sequenz bei der Herstellung eines Peptids oder Proteins ist ein einheitlicher Drei- oder Ein-Buchstaben-Code nach Empfehlung der IUPAC⁶/IUBMB⁷ - Nomenklaturkommission^[11] üblich.

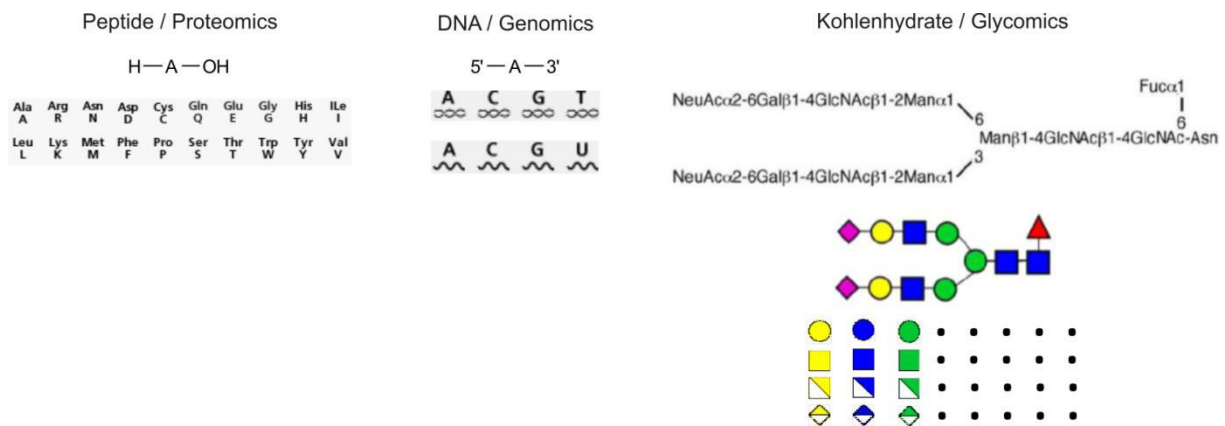


Abbildung 4: Peptide und Oligonukleotide (DNA/RNA) lassen sich aus einer begrenzten Anzahl von Grundbausteinen linear zusammensetzen. Zuckermolekülen werden aus einer Vielzahl von Bausteinen zusammengesetzt. Dafür existieren unterschiedliche Beschreibungs-codes. Die Symbolcodierung Cartoon - Symbolik ermöglicht die anschauliche Darstellung komplexer Zuckermoleküle.

Die Beschreibung der Primärstruktur von Oligonukleotiden erfolgt über einen einfachen linearen Buchstabencode der vier, jeweils in ihrem Basenanteil unterscheidbaren, Nukleotiden Adenin (A), Guanin (G), Cytosin (C), sowie Thymin (T) bei Desoxyribonukleinsäure (DNA) und Uracil (U) bei Ribonukleinsäuren (RNA). Je drei Basen bilden ein „Wort“, ein sogenanntes Codon, das meistens für eine Aminosäure steht. Da ein „Wort“ aus je drei Nukleotiden mit je vier möglichen Basen zusammengesetzt wird, lassen sich $4^3 = 64$ mögliche Kombinationen bilden. 61 Kombinationen codieren die proteinogenen Aminosäuren, die restlichen drei sind sogenannte „Stopp-Codons“ zum Beenden der Translation.^[15]

Der Komplexität der Zuckermoleküle wird eine rein lineare Beschreibung nicht gerecht. Bereits in der Strukturformel hat Emil Fischer^[16], der Pionier der Zuckerchemie, dafür eine eigene Schreibweise entwickelt. Mit der Fischer-Projektion kann die Raumstruktur einer linearen, chiralen chemischen Verbindung eindeutig zweidimensional abgebildet werden. Um auch komplexere Glykane in vergleichbarer Weise wie Peptide oder Nukleinsäuren zu beschreiben, entstand mit dem **LINUCS** Code^{8 [17]} ein Buchstabencode für die lineare Definition von Glykanen, aus denen die Zuckerbausteine (Building Blocks), deren Kopplungsart und evtl. Verzweigungen hervor gehen. Eine weitere Glykansprache veranschaulicht die Codierung komplexer Zucker mit Hilfe von Symbolen. Dabei existieren allein für die Hexosen schon acht verschiedene Varianten, die durch Kreise mit unterschiedlichen Farben symbolisiert werden. In Abhängigkeit der Substituenten, d. h. Kopplungen

⁶ IUPAC: International Union of Pure and Applied Chemistry

⁷ IUBMB: International Union of Biochemistry and Molecular Biology

⁸ LINUCS ist die Abkürzung von engl. "Linear notation for unique description of carbohydrate sequences".

an denen häufig anzutreffende funktionale Gruppen positioniert sind, ergeben sich unterschiedliche Symbolstrukturen. Aufgrund der Vielzahl möglicher Substituenten sind im Augenblick hauptsächlich die in der Natur anzutreffenden durch ein Symbol gekennzeichnet.

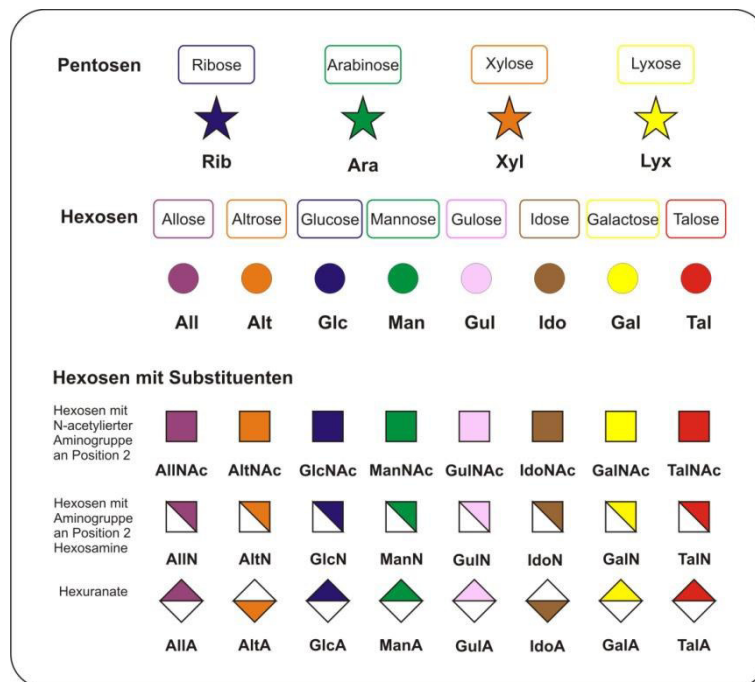


Abbildung 5: Symbole der wichtigsten Monomere mit biologischer Bedeutung^[18]

Die größte biologische Bedeutung besitzen Pentosen und Hexosen. Pentosen bestehen aus fünf und die Hexosen aus sechs Kohlenstoffen. Bei dieser Anzahl ist die Bildung von Ringstrukturen energetisch bevorzugt. Pentosen werden im Verlauf des Glukosestoffwechsels im Organismus intrazellulär gebildet und finden als wichtige Bestandteile von Koenzymen und Nukleinsäuren Verwendung. Bei den Hexosen ist Glukose der mengenmäßig bedeutendste Energielieferant des Organismus.^[19] Chitin (Abbildung 1) ist aufgebaut aus Acetylglucosamin-Bausteinen (GlcNAc), einer Hexose mit Substituenten.

Mit der glykosidischen Bindung entsteht ein neues Chiralitätszentrum.^[20-23] Durch die große strukturelle Vielfalt sind Kohlenhydrate zum Träger biologischer Informationen prädestiniert. Während bei Peptiden und Nukleinsäuren die Information einzig durch Anzahl und Sequenz der Monomer-Bausteine festgelegt ist, kann die Information bei Oligosacchariden außerdem durch Anlagerungen unterschiedlicher funktionaler Gruppen an den verschiedenen aktiven Stellen der Monomere, den Kopplungsmöglichkeiten untereinander, sowie durch Verzweigungsstellen definiert sein. Tabelle 1 veranschaulicht den Informationsgehalt von Di-, Tri-, Tetra- und Pentameren bei den Peptiden und Nukleinsäuren im Vergleich zu den Sacchariden anhand der möglichen Isomerenzahl.^[24] Die Menge an Information, die pro Baustein in einem Oligosaccharid enthalten sein kann, übersteigt bei weitem die der aktuell nahezu vollständig entschlüsselten der Proteomik und Genomik-Gebiete.

Homomer	Zusammensetzung	Peptide, Nukleinsäuren [a]	Saccharide [b]
Monomer	A	1	1
Dimer	AA	1	11
Trimer	AAA	1	120
Tetramer	AAAA	1	1424
Pentamer	AAAAA	1	17872

Heteromer	Zusammensetzung	Peptide, Nukleinsäuren [a]	Saccharide [b]
Monomer	A	1	1
Dimer	AB	2	20
Trimer	ABC	6	720
Tetramer	ABCD	24	34560
Pentamer	ABCDE	120	2144640

Tabelle 1: Isomeriemöglichkeiten bei Biopolymeren ohne D- und L- Formen. [a]3' → 5' verknüpfte Desoxyribo- oder Ribonukleinsäuren [b] bezogen auf Hexopyranosen (Glc, Man, ...).^[24]

Aufgrund der verschiedenen Verknüpfungsvarianten beinhalten Oligosaccharide weit mehr Information als Peptide oder Nukleinsäuren

Die fortschreitende Analyse zugrunde liegender Mechanismen der in Form von zellulären Glykokonjugaten⁹ an vielen biologischen Prozessen beteiligten Zuckerketten, hat gezeigt, dass Oligosaccharide Signalinformationen^[25] darstellen, deren Bedeutung durch Bindungspartner (Rezeptoren) gelesen und in biologische Reaktionen übersetzt werden kann. Damit stellen Monosaccharide die Buchstaben eines Kodierungssystems dar, das zur dritten Sprache des Lebens führt: dem „Glykocode“.^[26, 27]

Die Erforschung der strukturellen und funktionalen Vielfalt der Glykane hat sich zu einem der wichtigsten und vielversprechendsten Forschungsfelder entwickelt. Um jedoch die komplexen Wechselwirkungen der Biomoleküle untersuchen zu können, müssen den Forschern die Glykane in ausreichender Qualität und Quantität zur Verfügung stehen.^[2, 3, 5, 6, 28 - 30]

⁹ **Glykokonjugate** sind makromolekulare chemische Verbindungen aus Kohlenhydrat-Bausteinen und Proteinen, z. B. (Glykoproteine, Proteoglykane) oder Glykane verknüpft mit Lipiden (Glykolipide).

1.2 Die Oligosaccharid-Synthese

1.2.1 Die glykosidische Bindung

Glykosylierung ist die Schlüsselreaktion bei der Herstellung von Oligosacchariden und Glykokonjugaten. Die glykosidische Bindung wird durch nukleophile Substitution einer Abgangsgruppe (L) zwischen dem anomeren Kohlenstoff eines Monosaccharids und der Hydroxygruppe einer anderen Zuckereinheit gebildet. Das Produkt der Glykosylierungsreaktion, das Glykosid, liegt nach der Reaktion mit einem Nukleophil als Vollacetal vor. Das Monosaccharid, das den Glykosylrest zur Verfügung stellt, wird als Glykosyl-Donor, die aufnehmende Verbindung als Glykosyl-Akzeptor bezeichnet. Im Allgemeinen erfolgt die Reaktion in Gegenwart eines Aktivators (Promotor oder Katalysator), der den Weggang, der anomeren Abgangsgruppe unterstützt und zur Bildung eines Oxocarbenium Ions führt.

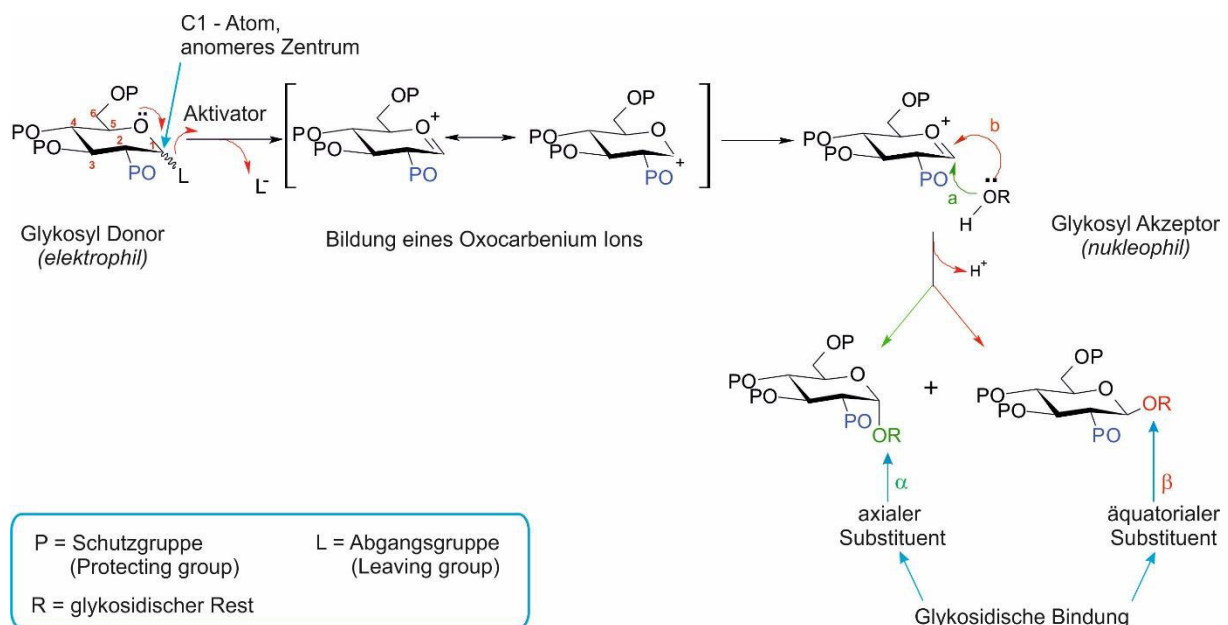


Abbildung 6: Prinzip der Glykosid-Synthese, bei dem der Zucker in zwei verschiedenen Konfigurationen, als α - und β -Glykosid, entstehen kann.

Prinzipiell gibt es vier Arten glykosidischer Bindung: 1,2-*trans* oder 1,2-*cis*-Glykosiden (Abbildung 9) und deren beiden Diastereomere, die sich an der anomeren Position als stereogenes Zentrum bilden und als α - bzw. β -Glykosid bezeichnet werden (Abbildung 6 und Abbildung 9). Die Kontrolle der Stereoselektivität während der Glykosylierung kommt eine große Bedeutung zu, denn die atomar identischen, jedoch in ihrer sterischen Ausrichtung unterscheidbaren Moleküle haben meist biologisch unterschiedliche Wirkungen. Erfolgreiche Glykosylierungsreaktionen erfordern damit eine hohe Regio- und Stereoselektivität, um vorzugsweise nur ein reines Anomer zu erhalten. Bei einem α -Anomer ist der Substituent axial, bei einem β -Anomer äquatorial am C-1 Atom ausgerichtet. 2-O-Alkyl-Schutzgruppen wie z. B. Benzyl- oder Allylether, führen aufgrund des

anomeren Effektes zur bevorzugten Bildung des thermodynamisch stabileren α -Glykosids. Generell lässt sich der anomere Effekt bei allen Molekülen beobachten, bei denen zwei oder mehr Heteroatome an ein tetraedrisches Zentrum gebunden sind.^[31]

Der anomere Effekt wird mit zwei Haupttheorien erklärt, der $n\text{-}\sigma^*$ -Hyperkonjugation (Abbildung 7a) und der Dipol-Dipol Wechselwirkung (Abbildung 7b). In beiden Fällen spielen die nicht-bindenden Elektronenpaare des *endo*-cyclischen Sauerstoffatoms eine große Rolle. Dies zeigt sich besonders, wenn sich ein elektronegativer Substituent (X) an der anomeren Position befindet (Abbildung 7c).

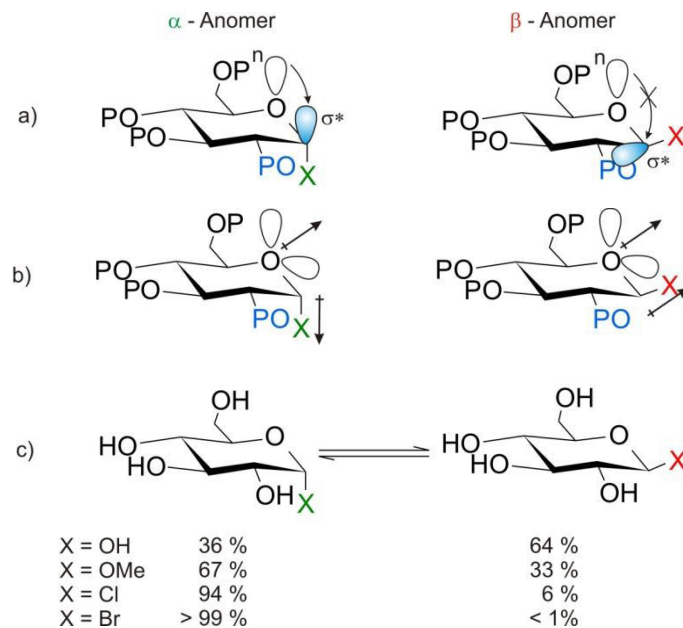


Abbildung 7: Der anomere Effekt: a) $n\text{-}\sigma^*$ -Hyperkonjugation, b) Dipol-Dipol Wechselwirkung, c) Verhältnis der Konfigurationen in Abhängigkeit vom Substituenten am Beispiel D-Glucose

Liegt als Beispiel das α/β -Verhältnis von unsubstituierter D-Glucose (X = OH) in wässriger Lösung noch bei 36/64, so verschiebt sich die Gewichtung mit zunehmender Elektronegativität des Substituenten stärker in Richtung der α -Anomere und erreicht bei X = Br einen Anteil von über 99 %.

Die am meisten favorisierte und akzeptierte Erklärung für diesen Effekt ist die Wechselwirkung zwischen einem freien Elektronenpaar, das "axial" in einem Molekülorbital (n) an O5 lokalisiert ist, und einem unbesetzten, anti-bindenden Molekülorbital (σ^*) der C1-X-Bindung. Dabei verteilt sich die Elektronendichte auf zwei Atome. Die Delokalisierung dieser Elektronen ermöglicht die Stabilisierung des elektronenarmen anomeren Zentrums durch Hyperkonjugation. Aufgrund der Orbitalsymmetrien ist diese Wechselwirkung nicht möglich, wenn sich X in einer äquatorialen Position befindet.

Während Hyperkonjugation zu einer stabilisierenden Wechselwirkung das α -Anomers führt, lassen sich einige energetische Effekte durch das resultierende Dipolmoment der sp^3 -hybridisierten Elektronen erklären. Beim β -Anomer weisen die *exo*-cyclischen Ionenpaare eine stark abstoßende elektrostatische Wechselwirkung mit den O5-Ionenpaar-Elektronen auf. Die parallel

zueinanderstehenden Dipolmomente wirken additiv und führen zu einem energetisch ungünstigen Zustand. Befindet sich X in axialer Position, zeigen die Dipole in verschiedene Richtungen und reduzieren die resultierende destabilisierende Wechselwirkung drastisch.^[32]

Die elektrostatische Abstoßung wird als weiteres Merkmal dem anomeren Effekt zugeschrieben, welche zur Bevorzugung von *synclinalen* (gauche) oder antiperiplanaren (anti) Konformationen wie im C5-O5-C1-X Segment (Abbildung 8) führt.

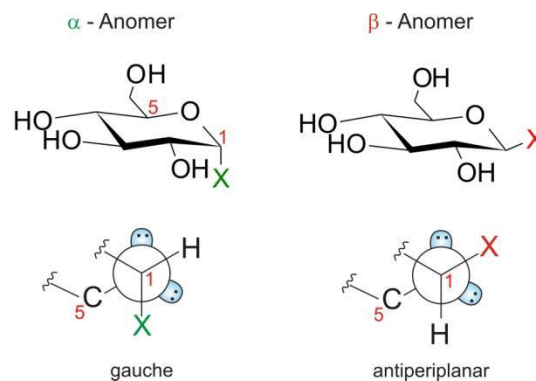


Abbildung 8: Im antiperiplanaren β -Anomer findet eine größere elektrostatische Abstoßung zwischen den Ionenpaaren des endocyclischen Sauerstoffs und dem elektronegativen anomeren Substituenten statt, als in der *synclinalen* Konformation des α -Anomers.

Das elektronegative Heteroatom X in der C1-O5 Bindung ist im *anti*-Konformer zwischen zwei Ionenpaaren platziert, was zu einer stärkeren elektrostatischen Abstoßung führt und deshalb energetisch unvorteilhafter ist. Diese Hypothese wird als **Antiperiplanare Lone Pair Hypothese** oder ALPH bezeichnet.^[33]

Zwischen der stabilisierenden anomeren Wirkung, die das α -Anomer begünstigt und anderen Faktoren, wie Lösungsmittelleffekte und sterische Effekte, welche das β -Anomer begünstigen können, besteht ein Gleichgewicht.^[34]

Die Wahl der geeigneten Schutzgruppe an der C2 - Position des Glykosylierungsmittels und dessen Zusammenspiel mit dem C1 Atom ist für das stereochemische Ergebnis einer Glykosylierung von entscheidender Bedeutung. Die Bezeichnung *trans/cis* formuliert dabei die räumliche Beziehung der glykosidischen Bindung zweier Substituenten bezüglich einer Referenzebene. Stehen die Hydroxylgruppen am C1 und C2 Atom, wie in Abbildung 9 dargestellt, *trans* zueinander, so zeigen die Hydroxylgruppen in gegensätzliche Seiten der Referenzebene, z. B. oben/unten beim β -D-Glucopyranosid und unten/oben beim α -D-Mannopyranosid. Bei einer *cis*-Anordnung befinden sich die benachbarten Substituenten auf der gleichen Seite der Referenzebene, d. h. beide zeigen nach unten (α -D-Glucopyranosid) oder beide nach oben (β -D-Mannopyranosid).

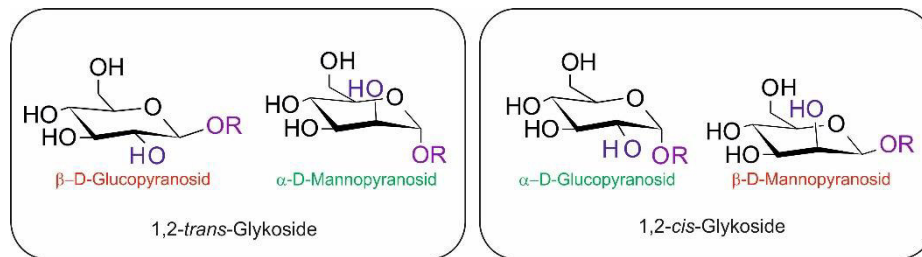


Abbildung 9: Beispiele für 1,2-trans und 1,2-cis-Glykoside

1,2-trans-Glykoside sind durch den Nachbargruppeneffekt, z. B. durch Verwendung von 2-O-acetylierten Glykosyl-Donoren, relativ leicht herzustellen. Die Estergruppe stabilisiert das nach der Aktivierung entstehende Carbokation durch die Nachbargruppenbeteiligung, wodurch ein cyclisches Carboxonium-Ion resultiert (Abbildung 10). Prinzipiell kann der Angriff des Alkohols sowohl von der Oberseite als auch von der Unterseite erfolgen. Befindet sich jedoch am benachbarten C2-Atom ein nachbargruppenaktiver Substituent, der intramolekular das Kation angreifen kann, so wird die α -Position abgeschirmt und der Alkohol erhält nur noch von oben Zugriff. Damit wird ausschließlich das 1,2-trans-Glykosid hergestellt. Jedoch kann als Nebenreaktion aufgrund des beidseitig nukleophil reagierenden Charakters der cyclischen Zwischenstufe ein durch Angriff des Alkohols auf das verbrückende C-Atom entstehende Orthoester als Konkurrenzprodukt zum Glykosid resultieren. Leicht saure Glykosylierungsbedingungen können die Akkumulation der Orthoester-Zwischenprodukte unterbinden.

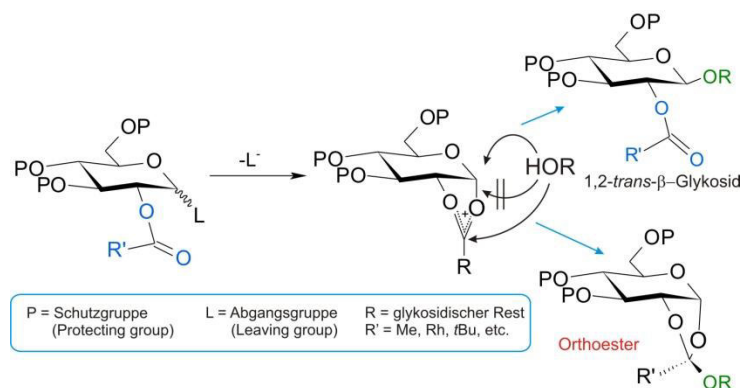


Abbildung 10: Einfluss des Nachbargruppeneffektes auf die Glykosylierung eines 1,2-trans-Glykosids.

Die selektive Herstellung von 1,2-cis- α -glykosidische Bindungen gestaltet sich schwieriger. Dazu dürfen sich keine aktiven Nachbargruppen im Molekül befinden, was eine ausgereifte Schutzgruppenstrategie und vielfältige Optimierung der Reaktionsbedingungen, z. B. Katalysatoren, Lösungsmittel, Konzentration, erforderlich macht. Die Herstellung von 1,2-cis-Bindung kann beispielsweise unter Verwendung von 2-O-alkylierten Glykosyl-Donoren wie Thioglykosiden oder Trichloracetimidaten erfolgen.

1.2.2 Glykosylierungsmethoden

Es gibt zahlreiche Glykosylierungsverfahren, die sich durch die Abgangsgruppe und den verwendeten Aktivator unterscheiden (Tabelle 2). Meist werden die Verfahren nach dem beteiligten Glykosyl-Donor benannt. Die Ausnahme stellen hier das Königs-Knorr-Verfahren, das Glykosyl-Halogenide als Donoren verwendet, und die Fischer-Glykosylierung, die reduzierende Zucker nutzt.

Für die Festphasenmethode eignen sich nicht alle Glykosylierungsreaktionen. Die ersten Studien zur Kohlenhydrat - Festphasensynthesen verwendeten Glykosyl-Halogenide (vorwiegend Per-O-Acetylglykosylbromide) unter Bedingungen der Königs-Knorr- oder der Helferich-Reaktion^[35] sowie der Orthoester-Glykosylierungsmethode.^[36] Die Aktivierung durch Schwermetallsalze erschwert jedoch die Nutzung dieser Reagenzien auf festen Trägern und findet deshalb nur noch selten Verwendung. Fortgeschrittene Modifikationen verwenden Glykosyl-Fluoride als Donor-Verbindung. Diese lassen sich unter Bedingungen aktivieren, in denen die meisten Schutzgruppen stabil bleiben. Die Trichloracetimidat-Methode ist ein universelles Glykosylierungsverfahren, welches die Verwendung von Schwermetallsalzen vermeidet und sich hinsichtlich Reaktivität und Selektivität als besonders effizient erwies.^[37] O-Glykosyl-Trichloracetimide wurden erstmals 1991 als neue Glykosyl-Donoren an löslichen PEG-Trägern für die Synthese eines Disaccharids erfolgreich eingesetzt.^[38] Trichloracetimide sind leicht herstellbar, ausreichend stabil und können mit katalytischen Mengen an Lewis-Säuren wie TMSOTf und anderen Triflaten für die Glykosylierungsreaktionen aktiviert werden. Dieses Verfahren liefert ausreichend hohe Ausbeuten für eine Vielzahl von Monosacchariden.

Sulfoxid als Glykosyl-Donor wurde 1989 von Kahne eingeführt und ermöglicht die effiziente Glykosylierung auch von sehr unreaktiven Hydroxy-Gruppen. Dies führte zur Herstellung von Oligosacchariden, sowie einer codierten kombinatorischen Bibliothek von 1300 Disacchariden und Trisacchariden.^[39] Die Reaktion beinhaltet den Sulfoxid-Donor, ein Aktivierungsmittel (üblicherweise Triflic-Anhydrid (Tf₂O)), eine gehinderte, nicht-nukleophile Base (2,6-Di-tert-butyl-4-methylpyridine, DTBMP) und einen nukleophilen Akzeptor (am häufigsten ein Alkohol).^[40] Mittlerweile nehmen Sulfoxide viele weitere Rollen in der Glykosylierungsreaktion ein. Sie sind Aktivatoren für Thioglykoside, Halbacetale und Glycale, dienen als Vorstufen für Glykosyl-Triflate, die für die stereoselektive β -Mannosidsynthese essenziell sind, und bicyclische Sulfoniumionen, die die stereoselektive Synthese von α -Glykosiden ermöglichen.^[41]

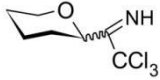
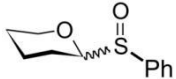
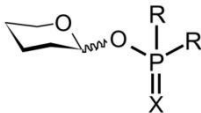



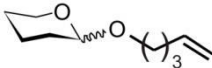
Methoden	Aktivierung	Literatur
Trichloroacetimidate 	$\text{BF}_3 \cdot \text{Et}_2\text{O}$, TMSOTf, $\text{Sn}(\text{OTf})_2$, AgOTf, $\text{ZnCl}_2 \cdot \text{Et}_2\text{O}$	[42 - 46]
Glykosyl Sulfoxide 	$\text{R}'\text{OH}$, DTBMP, Tf_2O	[39 - 41, 47, 48]
Glykosyl Phosphate  R = Alkyl, O-Alkyl X = O, S	TMSOTf, CH_3CN	[42, 49 - 54]
1,2-Anhydrozucker  R = OR' , SR' , CN	DMDO, CH_2Cl_2	[55 - 57]
Thioglykoside Fischer Helferich 	$\text{R}'\text{OH}$, $\text{Hg}(\text{Oac})_2$ NBS, DMTST, IDCP, NIS/TfOH	[58 - 62]
Glykosyl Halogenide Königs-Knorr  L = F, Cl, Br	Ag(I)-Salze, Hg(II)-Salze, Bu_4NBr	[63 - 79]
n-Pentenyl Glykoside 	NIS, Et_3SiOTf , IDCP, TMSOTf	[61, 71 - 73]

Tabelle 2: Beispiele von Glykosyl-Donoren bzw. Akzeptoren für die Synthese an der festen Phase

Glykosyl-Phosphate sind extrem reaktive Glykosylierungsmittel, die bei niedrigen Temperaturen innerhalb von Minuten in sehr hohen Ausbeuten zu sterisch anspruchsvollen Verbindungen, wie z. B. β -(1 \rightarrow 4) -gebundenen Glukosiden aktiviert werden.^[74]

Mit der Glycal-Assembly-Methode wurde Anfang der 1990er Jahre die Festphasensynthese von Oligosacchariden unter dem Donor-gebundenen Paradigma umgesetzt. Dabei wurden trägergebundene Glycale durch Epoxidierung mit Dimethyldioxiran (DMDO) in die entsprechenden 1,2-Anhydrozucker umgewandelt und aktiviert. Danishefsky erzeugte mit 1,2-Anhydrozucker aus Glycalen eine Vielzahl glykosidischer Bindungen, darunter unter anderem das Lewis-B-Hexasaccharid, eine Blutgruppendeterminante. Die Stereoselektivität der Epoxidierung hängt stark von der Art der Schutzgruppen und der sterischen Hinderung der Substituenten ab.^[55, 74]

Thioglykoside können leicht aus anomeren Acetaten oder 1,2-Anhydrozucker in großem Maßstab hergestellt werden, lassen sich über verlängerte Zeiträume sogar bei Raumtemperatur lagern und können selektiv mit einer Reihe von thiophilen Promotoren wie Dimethylthiosulfoniumtriflat (DMTST), Methyltriflat oder NIS / Trifluormethansäure aktiviert werden. Trotz der hohen Toxizität der Aktivatoren, haben Thioglykoside eine weite Verbreitung in der Festphase gefunden.

Die Verwendung von n-Pentenylglykoside (NPG) als Glycosyl-Donoren wurde 1988 von Fraser-Reid untersucht und auch auf die Festphasen-Oligosaccharidsynthese ausgedehnt. Die Aktivierung der Abgangsgruppe durch elektrophile Reagenzien wie NIS / TESOTf beruht auf einer elektrophilen Addition an die Doppelbindung des Aglycons, gefolgt von einer intramolekularen Verdrängung durch den Ringsauerstoff und dem eventuellen Ausstoß der Pentenylkette zu einer Oxoniumspezies. Das Abfangen mit einem Glykosyl-Akzeptor führt zum gewünschten Glykosid. Mit der Akzeptor-gebundenen Strategie konnten hohe durchschnittliche Kopplungsausbeuten von mehr als 90% und ausgezeichnete α - oder β -Selektivität erreicht werden, wie durch Gelphasen-¹³C-NMR gezeigt wurde.^[72]

1.3 Oligosaccharidsynthese an der festen Phase

1.3.1 Prinzip der Festphasenstrategie

Die Festphasenstrategie^[75], für die Merrifield 1984 den Nobelpreis für Chemie erhielt, revolutionierte das Gebiet der chemischen Synthese von Oligomeren. Unter Verwendung von unlöslichen Polymerträgern^[76] beschleunigte die Festphasensynthese den Herstellungsprozess und wurde in großem Umfang bei der Herstellung vieler organischen Moleküle^[77], Oligopeptide^[78] und Oligonukleotiden^[8] verwendet.

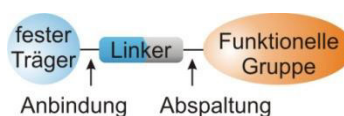


Abbildung 11: Prinzip des Festphasensystems

Bei der Festphasentechnik (Solid-Phase-Technik) ist mindestens ein Reaktionspartner immobilisiert über einen Linker als zukünftige „Sollbruchstelle“ an den festen Träger gekoppelt. An diesem reagiert der erste Kopplungsbaustein mit einer kovalenten Bindung. Damit vereinfacht sich die Reaktionsführung, da zeitaufwendige Reinigungs- und Isolierungsschritte durch einfache Filtrationen ersetzt und die im Überschuss eingesetzten Reagenzien am Ende der Synthese zusammen mit der Lösung vom polymergebundenen Produkt getrennt werden. Durch geeignete Prozessführung werden weitere Reaktionspartner schrittweise angekoppelt. Die wachsende Kette bleibt dabei am Träger kovalent fixiert und ist während der gesamten Synthesesequenz unlöslich.

Dem frühen Beispiel der Festphasen-Peptidsynthese durch Merrifield^[75], folgten in den 1970er Jahren erste Studien zur Festphasen-Oligosaccharidsynthese von Fréchet und Schuerch. Das verwendete Monosaccharid für die Synthese von Di- und Trisacchariden war über die anomere Position mit Allylalkohol-funktionalisiertem Merrifield-Harz verbunden.^[79] Anderson beschrieb eine thioglykosidische Bindung am festen Träger, der das reduzierende Ende des Oligosaccharids in der Form des Lactols freisetzt.^[80] Dabei wurde ein Ester-Linker verwendet, um die C6-Hydroxygruppe von Glucosamin mit Polystyrol zu verbinden. Diese Bindung war gegenüber Glykosylierungen stabil, wurde jedoch leicht mit Natriummethoxid gespalten. Zusätzlich zu Studien mit funktionalisiertem Polystyrol (Merrifield-Harz), fand auch poröses Glas¹⁰ als nicht quellender anorganischer Träger in der Festphasen-Oligosaccharidsynthese von Schuerch Anwendung.

¹⁰ Engl.: Controlled Pore Glass (CPG); Akzeptierter Standard bei den Festphasensynthese von DNA und RNA. Eigenschaften: hohe Oberfläche, effizienter Stoffübergang und chemische Inertheit.

Seit den ersten Versuchen entwickelte sich die Polymer-gestützte Oligosaccharidsynthese zu einem effizienten Werkzeug für die schnelle Synthese von Oligosaccharid-Sequenzen,^[81a-c, 82] für die zahlreiche Synthesestrategien entwickelt wurden. Die effizienteste und häufigste Strategie für die Oligosaccharid-Festphasensynthese besteht im Allgemeinen in der Bindung der ersten vollständig geschützten Kohlenhydrateinheit, die an ihrer anomere Position über eine funktionale Ankergruppe, dem Linker, an einen polymeren Träger gekoppelt wird.

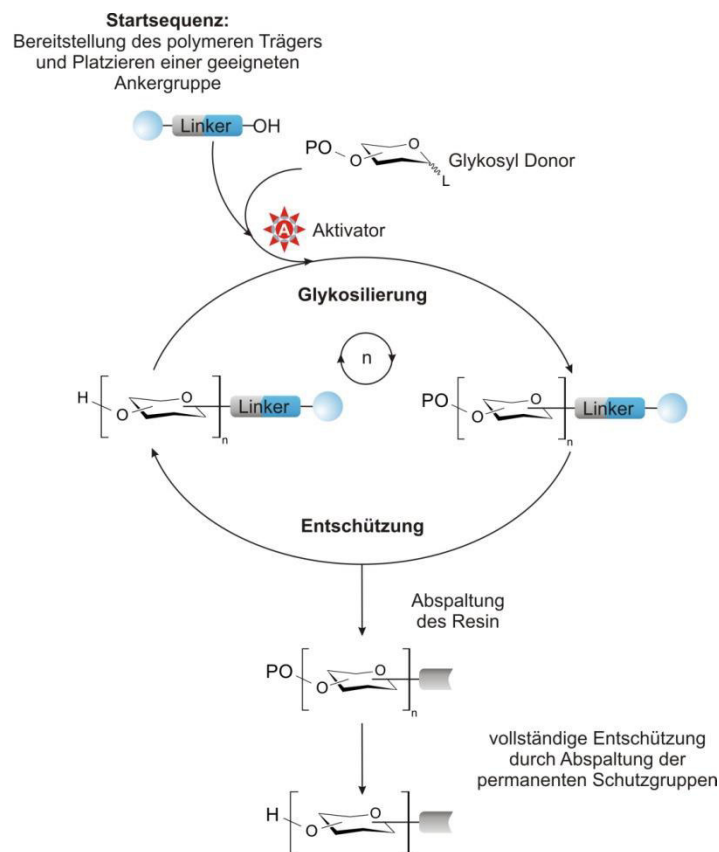


Abbildung 12: Schematischer Ablauf einer automatisierten Zuckersynthese

Nach der Anlagerung wird der erste Kohlenhydratanteil je nach gewünschtem Verzweigungsmuster mit orthogonalen Schutzgruppen versehen. Die permanenten Schutzgruppen verbleiben während der gesamten Synthese im Molekül, während die temporären Schutzgruppen der Kettenverlängerung dienen. Die am meisten verwendeten Schutzgruppen sind Acetat- und Benzoatester, sowie Benzylether. Zur Kettenverlängerung an der Festphase dienen Mono- oder auch Disaccharid-Bausteine. Der resultierende Glykosyl-Akzeptor wird anschließend mit einem geeigneten Donor glykosyliert, der seinerseits einer Schutzgruppenmanipulation unterzogen wird, um einen Glykosyl-Akzeptor zu erhalten. Erst die endgültige Abspaltung vom Linker setzt das Oligosaccharid vom festen Träger frei. Der repetitive Zyklus von Glykosylierungen und Schutzgruppenmanipulationen eignet sich zur Automatisierung der Oligosaccharid-Festphasensynthese.

Gegenüber der klassischen Synthese in Lösung weist die Festphasensynthese einige Vorteile auf:

- **Vereinfachte Reaktionsführung:**

Alle Reaktionen können im gleichen Reaktionsgefäß durchgeführt werden. Komplexe Aufarbeitungs- und Reinigungsschritte reduzieren sich auf einfaches Waschen des Trägers am Ende einer jeden Reaktion. Die im Überschuss eingesetzten Reagenzien werden dabei zusammen mit der Lösung vom polymergebundenen Produkt getrennt.

- **Höhere Ausbeute:**

Die Möglichkeit mit hohen Überschüssen des gelösten Reaktionspartners zu arbeiten, führt zu einer Beschleunigung der Reaktion und zu nahezu vollständiger Umsetzung.

- **Reduzierung intermolekularer Reaktionen:**

Mit der Immobilisierung des Reaktionspartners an den Bindungsstellen kommt das Verdünnungsprinzip zu tragen. Die isolierte Stellung der reaktiven Gruppe erschwert unerwünschte Nebenreaktionen zwischen immobilisierten Produkten, wie z. B. Quervernetzungen und Mehrfachkupplungen bei polyvalenten Reagenzien.

- **Einfache Automatisierung von Reaktionsschritten:**

Bei der Synthese von Oligo- und Polymeren kann die Automatisierung zu einem hohem Zeitgewinn führen. Damit ist die Bereitstellung einer großen Probeanzahl möglich.

Entscheidend für eine erfolgreiche Festphasen-Oligosaccharidsynthese ist die richtige Auswahl der drei funktionalen Anforderungen, die miteinander in Wechselwirkung stehen:

- fester Träger (unlösliches Polystyrol oder lösliches Polyethylenglykol (PEG))
- Linker (die als harzgebundene Schutzgruppe betrachtet werden kann)
- Funktionelle Gruppe (erfordern eine chemische Schutz / Entschützungs-Strategie der reaktiven Gruppen)

Zur Überwachung des Fortschritts in der Synthese wurden verschiedene analytische Techniken entwickelt, wie hochauflösende ein- und zweidimensionale NMR-Spektroskopie^[56], Gate-entkoppelte ¹³C-NMR-Spektroskopie mit ¹³C-angereicherten Schutzgruppen^[83], fluorierte Schutzgruppen und ¹⁹F-NMR-Spektroskopie^[84], FT-IR^[85], MALDI-TOF MS^[86] und kolorimetrische Tests.^[87, 88]

1.3.2 Träger

Bei der Auswahl des Trägermaterials ist die Robustheit gegenüber den eingesetzten Verknüpfungs- und Entschützungsreaktionen wichtigste Voraussetzung. Trotz der chemischen Inaktivität in einer Reaktionssequenz besitzen die Eigenschaften des polymeren Materials großen Einfluss auf die Reaktionsabläufe.^[89]

Theoretisch kommt als Träger jede Materialvariante mit großen Oberflächen in Betracht. Das am häufigsten genutzte Trägermaterial sind jedoch Kunstharze in Form mikroporöser Polystyrol-Kügelchen¹¹ (PS), vernetzt mit 1 – 5% Divinylbenzol, einer Teilchengröße von ca. \varnothing 20 μm - 200 μm und einer Beladungskapazität von bis zu 5 mmol/g Polymer.

Nur ein vernachlässigbarer (< 1%) Teil der Reaktionen verläuft direkt an der Oberfläche des Resins. Bis zu 99% der aktiven Stellen befinden sich im Inneren des Polymers.^[90] Dadurch herrscht eine weitestgehend hydrophobe Reaktionsumgebung, welche für die Synthese essentiell ist. Die eigentlichen Anknüpfungspunkte der Reaktionsstellen liegen räumlich weit voneinander entfernt, wodurch eine direkte Querreaktion nahezu ausgeschlossen ist. Durch die Vermeidung der direkten Kontaktmöglichkeit, lassen sich chemische Konstrukte, wie z. B. sterische Anordnungen und Ringe, realisieren, die in Lösung unter vergleichbaren Reaktionsbedingungen nicht möglich wären.

Durch Solvatisierung (Quellen) in den verwendeten Lösungsmitteln entsteht eine Gelphase. Dabei wird das polymere Träger-Resin für die Reagenzien permeabel. Gute Quelleigenschaften des Resins bedeuten zugleich eine gute Zugänglichkeit der Reagenzien an den reaktiven Zentren. Die Volumenzunahme beträgt hierbei ein Vielfaches des Ausgangsvolumens im Trockenzustand. Eine hohe Dichte an reaktiven funktionellen Gruppen im Verhältnis zum Harzvolumen¹² ist von großem praktischen Vorteil und führt zu geringerem Lösungsmittelverbrauch. Meist werden Harze mit einer Beladungsdichte von ca. um 1 mmol/g verwendet.

Grundlegende Verbreitung in der Praxis haben Polystyrol-DVB-Copolymere wie das Merrifield Harz, sowie Polystyrol-PEG-Harze, z. B. TentaGel™, erlangt.

In der automatisierten Oligosaccharid Festphasensynthese findet das Merrifield Harz^[75] aufgrund seiner chemischen und mechanischen Stabilität am häufigsten Verwendung.

Das Merrifield Harz ist ein chlormethylisiertes Copolymer aus Styrol und 1 – 2 % Divinylbenzol (Polystyrol-DVB-Copolymere). Das Harz ist relativ preiswert, besitzt eine relativ hohe Beladungskapazität (1,2 mmol/g), ist jedoch bei der Anwendung auf Lösungsmittel wie DCM, DMF, THF, NMP, Pyridin und Dioxan beschränkt.

Beim TentaGel™ - Harz von E. Bayer^[91] sind die jeweiligen Kopplungsstellen auf dem vernetzten

¹¹ Die Polystyrol-Kügelchen werden auch als „Beads“ bezeichnet.

¹² Das Verhältnis reaktiver, funktioneller Gruppen zum Harzvolumen wird als „Beladungsdichte“ bezeichnet.

Polystyrol-Harz durch lange Polyethylenglykolketten (Polystyrol-PEG-Harze) ausgeformt. Dadurch werden die großen Abstände der Reaktionsstellen erreicht. Das TentaGel™ - Harz ist kompatibel mit Wasser, besitzt jedoch niedrigere Beladungseigenschaften (0,2 - 0,3 mmol/g) und einen höheren Preis. Vermutlich aufgrund verbesserter Quelleigenschaften hat sich dieses Harz in speziellen Anwendungen dem Merrifield Harz überlegen gezeigt, z. B. für die Bibliothekssynthese ohne mechanisches Mischen.^[92]

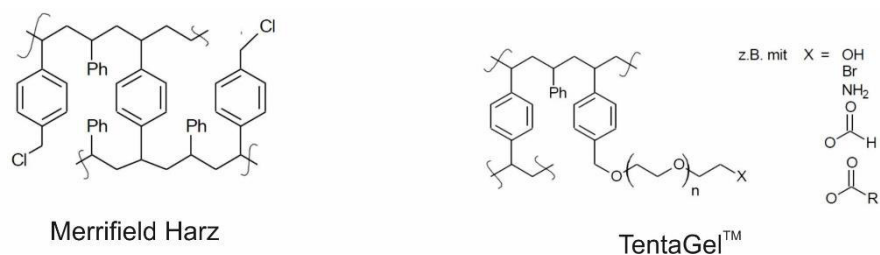


Abbildung 13: Struktur von Merrifield Harz und TentaGel™

Shimizu und Nishimura nutzten nanostrukturierte goldbasierte Träger, um die Oligosaccharidkette über einen Polyethylenglykol (PEG) - basierten Thiollinker an die Oberfläche zu binden.^[93] Diese Methode ermöglichte die Analyse der Reaktion mit dem MALDI-TOF, ohne zuvor den Zucker vom festen Träger zu trennen. Auch war es möglich aufgrund des PEG-Linkers eine globale Entschützung des Produkts, einem Disaccharid, durchzuführen, um ein Glykonanopartikel zu erhalten. Für die oberflächengebundene iterative Kohlenhydratsynthese (STICS) verwendeten Stine und Demchenko poröses Gold mit großer Oberfläche als festen Träger.^[94] Das Verfahren benötigt eine geringere Menge an Lösungsmittel als die herkömmliche Festphasensynthese, auch kann die Festphase am Ende der Synthese wiederverwendet werden. In Verbindung mit einem Split-and-Pool¹³-Ansatz, einer Schlüsselmethode der kombinatorischen Chemie, ermöglicht der Prozess eine einfache Bibliothekserzeugung.

Neben den unlöslichen festen Trägern werden Oligosaccharid-Synthesen auch an löslichen Polymeren durchgeführt. Unlösliche Träger erfordern oft eine umfangreiche Reaktionsentwicklung, um Verfahren, die für die Lösungsphasen-Synthese entwickelt wurden, für die Festphasenstrategie zugänglich zu machen. Lösliche Polymerträger kombinieren die Vorteile des Lösungsphasenstrategie mit der einfachen Aufarbeitung der Festphasensynthese. Während alle chemischen Umwandlungen in homogener Lösung durchgeführt werden, wird das Polymer nach jedem Schritt ausgefällt, um die Entfernung überschüssiger Reagenzien durch einfache Filtration sicherzustellen. Ein potenzieller Nachteil ist der Materialverlust während des Ausfällungsschritts nach jeder Kopplung, was die Gesamtausbeute bei der Montage großer Strukturen verringert.

¹³ Split-Pool-Synthese: Verfahren der kombinatorischen Chemie zur Erzeugung von Substanzbibliotheken auf polymerem Trägermaterial mittels Festphasen-Synthese.

Der einfachste lösliche Träger ist Polyethylenglycol (PEG), in dem eine der terminalen Hydroxygruppen methyliert und die andere frei ist und dazu dient, das erste Monomer - entweder direkt oder über einen Linker - zu immobilisieren.^[95] Im Handel erhältliches PEG ist in vielen Lösungsmitteln leicht löslich. Polymere vom Hybridtyp, die sowohl hydrophobe als auch hydrophile Fragmente enthalten, umfassen Copolymere von Acrylamid oder dessen N-substituierten Derivaten mit Polyethylenglycol (PEGA) mit endständigen Aminogruppen^[96], ein Copolymer von Polystyrol mit Polyethylenglycol (TentaGel™)^[97] und Methoxypolyethylenglykol (MPEG)^[98].

1.3.3 Linker

Der Linker stellt eine reversible, stabile Bindung zwischen Resin und Zuckermolekül bereit, hält die wachsende Molekülkette vom Träger fern, um den Einfluss der festen Oberfläche auf den Reaktionsverlauf zu minimieren, und ermöglicht nach der Synthese das selektive Abspalten des gewünschten Zielmoleküls. Der Linker definiert dabei die Struktur der funktionellen Gruppe, die bei der Spaltung am Saccharid verbleibt. Idealerweise erlaubt ein geeigneter Linker eine direkte Weiterverwendung der synthetisierten Oligosaccharide, die auf Oberflächen von Kohlenhydratmikroarrays gekuppelt oder mit Proteinen und anderen Makromolekülen zu Glykokonjugaten verknüpft werden können.^[99] Zur Bereitstellung konjugationsfertiger Verbindungen sind weitere Spaltungsschritte, wenn der Linker mehrfach spaltbar ist^[100], oder Modifikationen an der anomeren Einheit nötig.

Die chemische Natur dieses Ankers bestimmt alle anderen Schutzgruppen- und Kupplungsmanipulationen, die während der gesamten Synthese durchgeführt werden können. Dabei kann der Linker selbst als Schutzgruppe^[101] angesehen werden, die unter den Reaktionsbedingungen stabil und unter milden orthogonalen Spaltungsbedingungen labil sein muss, um die quantitative Freisetzung des Produkts zu ermöglichen.

Linker werden nach ihren Spaltungsbedingungen^[102,103], anhand ihrer funktionalen Gruppe (z. B. Ester-Linker, Silyl-Linker) oder nach speziellen Eigenschaften (z. B. Safety-Catch- und Traceless-Linker) klassifiziert. Für die Festphasensynthese steht eine große Anzahl von Linkern bereit, jedoch ist nur ein Teil für die Oligosaccharidsynthese geeignet. Beispiele von Linker-Systemen, die speziell für die Anforderungen in der automatisierten Oligosaccharid-Festphasensynthese entwickelt wurden, werden in Tabelle 3 vorgestellt.

Bereits in der ersten automatisierten Oligosaccharid-Festphasensynthese fand ein 4-Octenediol Linker in Verbindung mit dem Merrifield Resin Verwendung. Octendiol- und Butendiol-Linker^[104] (Tabelle 3, Nr. 3), die eine einzige Doppelbindung enthalten, nutzen die Olefin-Kreuzmetathese (oder Basenhydrolyse und Kreuzmetathese) zur Abspaltung des Alkene-basierten Linkers vom festen

Träger, wobei n-Pentenyl-funktionalisierte Oligosaccharide entstehen. Der Linker ist entweder durch Ether^[51] (Tabelle 3, Nr. 1) oder eine Esterbindung^[105] (Tabelle 3, Nr. 2) an den festen Träger gebunden, orthogonal zu allen in der Festphasensynthese verwendeten Standardmanipulationen und kann selektiv am Ende der Synthese abgespalten werden.^[106] Bei der Abspaltung wird ein Grubbs' Katalysator der ersten Generation (Grubbs' 1) in Gegenwart von Ethylen verwendet.

Das resultierende vollständig geschützte n-Pentenylglykosid (Octendiol) oder Allylglykosid (Butendiol) kann zu einem freien reduzierenden Zucker hydrolysiert, oder aktiviert und dann glykosyliert werden. Die Kohlenstoff – Kohlenstoff - Doppelbindung lässt sich modifizieren, um Verknüpfungen für Glykokonjugate herzustellen.

Diese Linker besitzen jedoch einige Nachteile. Der Grubbs-Katalysator, der für die Abspaltung vom Harz erforderlich ist, ist teuer und oft schwierig aus den Endreaktionsprodukten zu entfernen. Die Herstellung von Glykokonjugaten erfordert weitere Funktionalisierung, wodurch sich die Gesamtausbeute des fertigen Oligosaccharidprodukts verringern kann. Als Hauptnachteil erweist sich jedoch die Doppelbindung am Octendiol-Linker, welche die Verwendung von Elektrophilen ausschließt, wie sie für die Aktivierung von Thioglykosiden erforderlich sind.

Eine Weiterentwicklung, der die Nachteile von Octendiol- und Butendiollinkern angeht, ist ein bifunktionaler Linker basierend auf einem 4-Hydroxymethylbenzyl-N-(5-hydroxypentyl)-N-Benzylcarbammat-Spacer (Tabelle 3, Nr. 4). Er ist über eine Esterbindung an den festen Polystyrolträger gebunden und setzt nach Abspaltung die Glykane als carbamatgeschützte Aminoglykoside frei.^[107] Der bifunktionale Linker ist vollständig kompatibel mit den häufigsten Glykosylierungsbedingungen, einschließlich der elektrophilen Bedingungen, die für die Aktivierung von Thioglykosiden erforderlich sind. Die Spaltung des Esters unter Verwendung einer basenkatalysierten Hydrolyse ist kostengünstig und erfolgt unter milden Bedingungen. Das terminale Amin wird unter Standard-Hydrogenolyse - Bedingungen an Palladiumkatalysatoren freigesetzt und kann als Anlagerungspunkt für die Synthese von Glykoarrays und Glykokonjugaten verwendet werden kann. Als Hauptnachteil dieses Linkers erweist sich der Ausschluss von Acetatgruppen als temporäre Schutzgruppen, da die zur Entfernung erforderlichen Reaktionsbedingungen auch zur Abspaltung des Oligosaccharids aus dem Harz führen würden.

Der Succinamic-Linker^[108] (Tabelle 3, Nr. 5) wurde für die HPLC assistierte automatisierte Oligosaccharidsynthese entwickelt und verwendet. Dieser Linker, der über eine Amidbindung mit dem festen Träger (TentaGel™) verbunden ist, kann auch durch basenkatalysierte Hydrolyse abgespalten werden. Der Succinamic-Linker bietet ähnliche Vorteile wie der bifunktionale Linker und könnte im Prinzip die elektrophilen Bedingungen unterstützen, die für die Thioglykosid-Aktivierung

erforderlich sind. Der Hauptnachteil bei dem Succinamic-Linker ist, dass die Alkohol-Gruppe, die bei der Hydrolyse aus dem Harz erzeugt wird, für die Konjugation weiter funktionalisiert werden muss.

Fotospaltbare Linker ermöglichen eine orthogonale Spaltung des Linker-Systems in Gegenwart einer beliebigen Schutzgruppe. Der gebräuchliche ortho-Nitrobenzylether wurde bereits 1973 von Patchornik auf die Oligosaccharidsynthese an einem festen Träger angewendet.^[109] Der für die automatische Festphasen-Oligosaccharidsynthese entwickelte fotospaltbare Linker^[110] ist über eine Etherbindung mit dem festen Träger verbunden und kann durch UV-Bestrahlung effizient im kontinuierlichen Fluss^[111] gespalten werden (Tabelle 3, Nr. 6). Aufgrund der Verringerung der Lichtstreuung durch die Harzkügelchen und der Abnahme der Lichtintensität mit dem Abstand von der Lichtquelle verbessert dieser synthetische Aufbau die Spaltungseffizienz gegenüber Batch-Reaktionen.

Der Linker ist sowohl in basischen als auch in sauren Medien stabil und verträglich mit allen Klassen von Glykosylierungsmitteln. Nach der Spaltung setzt die Entfernung der Benzyloxycarbonyl-Schutzgruppe durch Hydrogenolyse die endständige Aminogruppe zur weiteren Funktionalisierung frei.

Durch Modifikation des etablierten fotospaltbaren Linkers, entstand im Seeberger-Labor 2016 ein Traceless Linker (Tabelle 3, Nr. 7). Die reduktiv labile Funktionalität im Linker toleriert alle gängigen Reagenzien und Protokolle für die automatisierte Oligosaccharid-Festphasensynthese, einschließlich Glykosylierungen mit Glykosylphosphaten und Thioglycosiden, die Entfernung von Schutzgruppen wie Fmoc, Lev und Nap, sowie die Verkappung von freien Hydroxylgruppen als Benzylester. Der Linker stellt eine Ergänzung für den fotospaltbaren Linker bei der Synthese von Aminoalkyl-funktionalisierten Glykanen dar. Vor allem die Möglichkeit große Mengen verwandter Glykane mit freien reduzierenden Enden herzustellen, birgt ein großes Potenzial für die Erstellung von Standards für LC / MS - und Ionenmobilitäts-Massenspektrometrieexperimente.^[112]

Einleitung und Zielsetzung

Nr.	Linkerstruktur	Abspaltung	Produkt	Linkertyp
1				Octendiol - Linker spaltbar durch Olefin Metathese ^[51]
2		Grubbs' Katalysator $\xrightarrow{\text{H}_2\text{C}=\text{CH}_2}$		Octendiol - Linker spaltbar durch Olefin Metathese ^[105]
3				Butendiol - Linker spaltbar durch Olefin Metathese ^[104]
4		a) $\xrightarrow{\text{NaOMe, MeOH}}$ b) $\xrightarrow{\text{Pd/C, H}_2}$		Bifunktionaler Linker ^[107]
5		$\xrightarrow{\text{NaOCH}_3 \text{ in } \text{CH}_3\text{OH}/\text{CH}_2\text{Cl}_2}$		Basenlabiler Succinamic Linker ^[108]
6		a) $\xrightarrow{h\nu}$ b) $\xrightarrow{\text{Pd/C, H}_2}$		Fotolabiler Linker ^[110]
7		a) $\xrightarrow{h\nu}$ b) $\xrightarrow{\text{Pd/C, H}_2}$		Fotolabiler traceless Linker [113]

Tabelle 3: Beispiele für Linker in der automatisierten Oligosaccharid-Festphasensynthese. Bis auf Linker Nr. 4, sind alle an ein Merrifield – Harz gebunden.

1.3.4 Synthesestrategien

Die Synthese von Oligosacchariden an fester Phase unterscheidet sich von Peptid- und Nukleotidsynthesen durch die Besonderheiten und hohe Komplexität der Kohlenhydratchemie. Saccharide sind typischerweise eher verzweigt, als rein linear aufgebaut, Monosaccharideinheiten können durch α - oder β -Verknüpfungen verbunden sein und es sind mehrere selektive Schutz- und Entschützungs-schritte bei der Synthese erforderlich. Gerade die letzte Anforderung erweist sich als Herausforderung, denn in einer Glykosylierungsreaktion enthalten sowohl Donoren als auch Akzeptoren viele ähnliche funktionelle Gruppen, die differenziert und selektiv berücksichtigt oder geschützt werden müssen.^[114] Erst die Fortschritte in der Lösungsphasensynthese dieser Moleküle in den 1990er Jahren, ermöglichten durch vielseitige Schutzgruppenstrategien^[115] und zunehmend leistungsfähige und selektive Glykosylierungsmittel die Grundlage für den Zugang zu komplexen Oligosacchariden durch polymergestützte Synthesen.^[102,116]

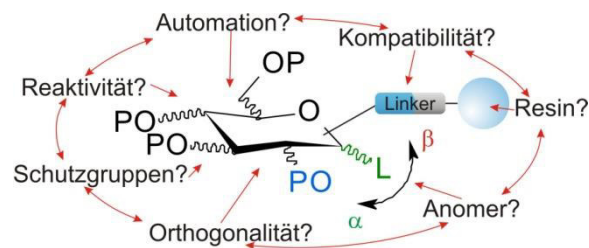


Abbildung 14: Herausforderungen einer Oligosaccharid-Synthese

Aus dem zentralen Aspekt jeder glykosidischen Bindung, der Vereinigung eines Glykosyl-Donors und eines Glykosyl-Akzeptors, ergeben sich zwei grundsätzliche Synthesestrategien.

Bei der Donor-gebundenen Strategie ist das „nichtreduzierende“ Ende der ersten Kohlenhydrateinheit über eine Ankergruppe (Linker) an den festen Träger gebunden (Abbildung 15). Die Kettenverlängerung wird durch Bereitstellung des Akzeptorbausteins unter Anwesenheit eines Aktivators im Überschuss vollzogen. Da der Donor in einer typischen Glykosylierungsreaktion als "reaktives Zwischenprodukt" fungiert, führen unproduktive Nebenreaktionen (Eliminierung, Hydrolyse usw.) aufgrund der Verknüpfung an den festen Träger zu einem direkten Verlust in der Gesamtausbeute. Nach jeder Glykosylierung ist die Umwandlung der anomeren Position am reduzierenden Ende in bestimmte Abgangsgruppen erforderlich, was im Oligosaccharid-Stadium keineswegs einfach ist.^[117]

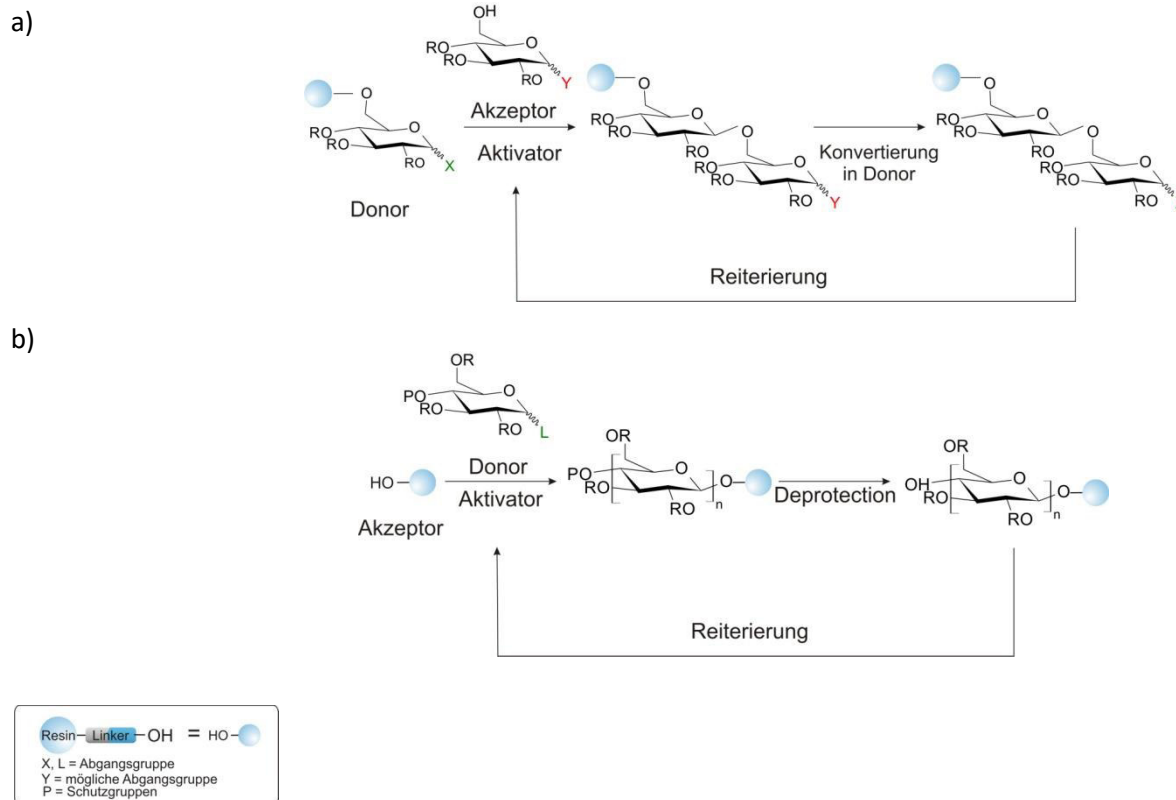


Abbildung 15: a) Donor-basierte Synthesestrategie b) Akzeptor-basierte Synthesestrategie

Im Allgemeinen wird deshalb die Akzeptor-gebundene Strategie als vorteilhafter angesehen, die einen Überschuss an Donor in Lösung verwendet und den Akzeptor an das Resin bindet. Sowohl das polymergetragene Aglycon als auch der Glykosyl-Donor sind reaktiv genug, um die Vervollständigung aller Glykosylierungen sicherzustellen. Das selektive Entfernen der Schutzgruppe, um die Hydroxygruppe für die nächste Glykosylierung freizusetzen, kann quantitativ durchgeführt werden, wodurch ein nahezu homogenes Produkt erhalten werden kann. Dies entspricht der Strategie von Merrifield für die Peptidsynthese, bei der die reaktive Spezies in Lösung und im Überschuss vorliegt. Hierbei werden die unerwünschten Nebenreaktionen zusammen mit nicht umgesetztem Donor einfach gewaschen.^[57,88]

Aus der Kombination der beiden grundsätzlichen Ansätze wurde 1998 mit der Synthese eines Trisaccharids die bidirektionale Strategie vorgestellt.^[118] Der polymergebundene Zuckerbaustein wird in zwei aufeinanderfolgenden Glykosylierungen abwechselnd als Donor und als Akzeptor eingesetzt. Für eine Kettenverlängerung in zwei Richtungen wird eine geeignet differenzierte Akzeptorstelle und eine anomere Donorfunktion benötigt. Die latente Donoreinheit am reduzierenden Ende des Saccharids, welches anfänglich als Glykosyl-Akzeptor dient, muss gegenüber den Kopplungsbedingungen, die für die Verlängerung der Akzeptor-gebundenen Verzweigung verwendet

werden, vollständig inert sein. Dieser Ansatz eignet sich besonders für die Herstellung verzweigter Strukturen.^[102, 119]

Je nach gewählter Synthesestrategie unterscheiden sich die Schutzgruppen und Manipulationen in jeder Stufe der Festphasensynthese. Wird die Synthese über die Akzeptor-gebundene Strategie durchgeführt, sollte im terminalen Monosaccharid des Glykosyl-Akzeptors der Schutz der Hydroxygruppen verändert oder selektiv entfernt werden. Bei der Donor-gebundenen Strategie ist es dagegen notwendig das glykosidische Zentrum im terminalen Monosaccharid des Glykosyl-Donors zu aktivieren. Auch die in Lösungen vorhandenen Reagenzien, ob als Glykosyl-Donor oder Glykosyl-Akzeptor müssen ebenfalls geeignete Schutzgruppen aufweisen.^[120]

Die Komplexität der Schutzgruppenmanipulation steigt mit jeder zusätzlichen glykosidischen Verknüpfung. Fortgeschrittene Strategien zur Herstellung größerer Oligosaccharide konzentrieren sich auf die Minimierung von Reinigungs- und Zwischenreaktionsschritten, z. B. Fraser-Reids „armed–disarmed“ Ansatz,^[121,71] Nicolaou's selektive Aktivierung,^[122] Danishefskys „Glykal-basierte“ Synthese,^[123] Kahnes „One-Pot“-Synthese,^[39] Roy's^[124] und Boons'^[125] „latent-aktives“ Konzept, Ogawa's orthogonale Strategie,^[126] Huang's Präaktivierungskonzept,^[127 – 129] usw.

Die Komplexität der Schutzgruppenmanipulation bleibt jedoch erhalten und erfordert die Verwendung einer umfangreichen orthogonaler Schutzgruppenstrategie.^[130]

Stereoselektive Glykosylierungen und Schutzgruppenmanipulationen mit hohen Ausbeuten sind essentiell für eine erfolgreiche Festphasen-Oligosaccharidsynthese. Aufgrund dieses Problems gibt es derzeit keinen einzigen schrittweisen synthetischen Ansatz, der auf die Synthese aller Oligosaccharide oder sogar nur der 0,15 Millionen möglichen Tetrasaccharide anwendbar ist, die aus den neun üblichen Monosacchariden zusammengesetzt werden können, die in der menschlichen Biologie eine entscheidende Rolle spielen. Im Gegensatz dazu beinhaltet die Festphasensynthese von Peptiden und Oligonukleotiden hauptsächlich eine variierte Schutzgruppenmanipulation im iterativen Prozess.^[114]

1.4 Automation von Oligosaccharid-Synthesen

1.4.1 Oligosaccharid-Synthesizer für die automatisierte Festphasen-Synthese

Die Grundlagen moderner Synthese-Automation legte Merrifield bereits 1970 mit seinem Syntheseapparat (Abbildung 16). Dessen struktureller Aufbau gilt auch heute noch und besteht aus folgenden Segmenten: Lösungsmittelzufuhr, Teilbereich der Aktivatoren und Entschützungsreagenzien sowie Zufuhr der Building-Blocks, die über Rotationsventile in den Reaktionsraum transportiert werden können.

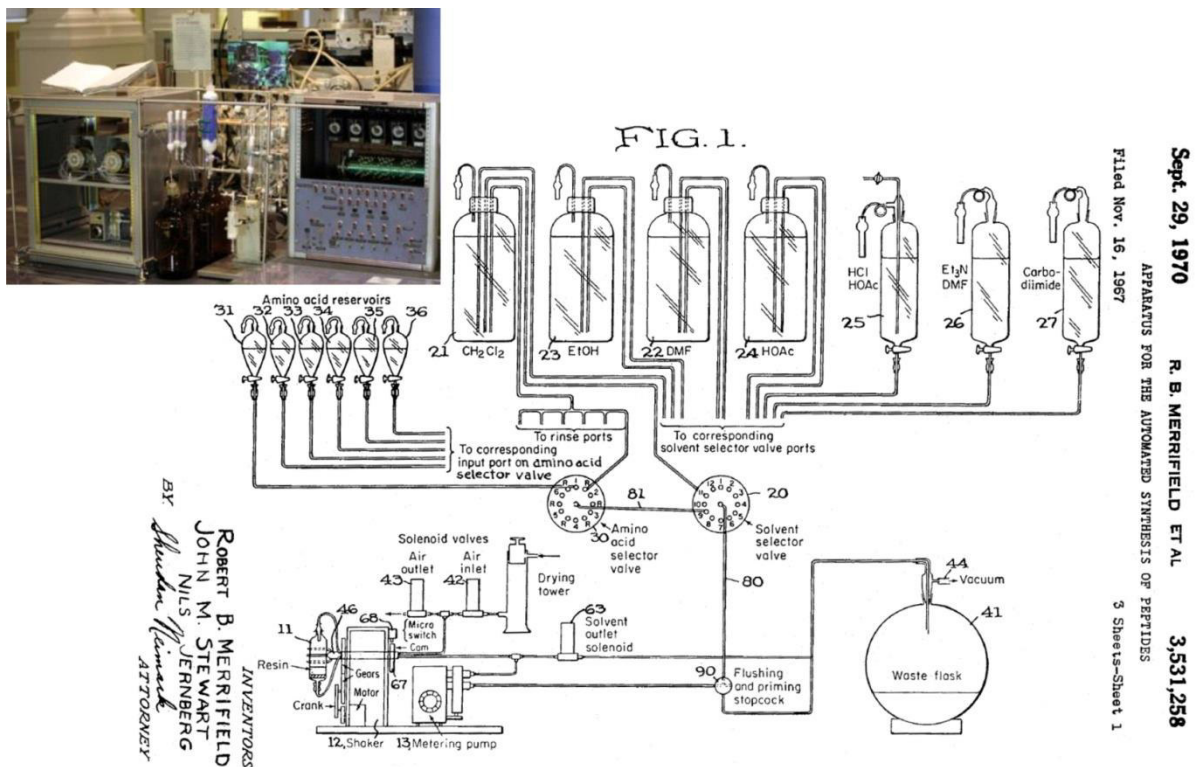


Abbildung 16: Nachbau des ersten automatischen Peptidsynthesizers und schematische Zeichnung des automatischen Syntheseapparates für Peptide aus der Patentschrift von Bruce Merrifield ^[131]

Der Flüssigkeitstransport erfolgt durch Unterdruck und Überdruck in separaten Systemabschnitten und ermöglicht den Transport von überschüssigen Reaktanten aus dem Reaktionsgefäß. Mit diesem Merrifield - Automaten sind bereits komplexere Peptide automatisiert herstellbar. Das Besondere an diesen ersten Geräten war eine durch Steckwalzen ausgelegte Programmierung als mechanische Steuerung.^[132]

Marvin H. Caruthers gelang 1985 die Umsetzung der automatisierten Festphasensynthese zur Herstellung von Oligonukleotiden mit Hilfe eines Synthese-Automaten.^[8] Die dazu methodischen Grundlagen der Phosphotriester-Methode, als heute gängigstes Verfahren der DNA-Synthese, wurde 1975 von Letsinger^[133] erforscht und von Matteucci und Caruthers^[134] im Jahre 1981 für die

Anwendung an der Festphase erweitert. Genau wie Peptid-Synthesizer sind auch DNA-Synthesizer heute weit verbreitete standardisierte Laborgeräte.

In Anlehnung an die Festphasen-Peptidsynthese realisierte die Seeberger-Gruppe 2001^[135] die beschleunigte Synthese von komplexen Zuckermolekülen. Grundlage für die Entwicklung des ersten automatischen Oligosaccharid-Synthesizers stellte der Umbau eines Peptidsynthesizers des Typs ABI 433 dar, der mit einer individuellen Temperierung ausgestattet und für die Synthese von Oligosacchariden optimiert wurde.^[136,137] An der Apparatur wurde ein vollständig geschütztes Hexa- β -Glukan **4** gebunden an ein Octendiol-funktionalisiertes Merrifield – Harz aufgebaut (Abbildung 17).

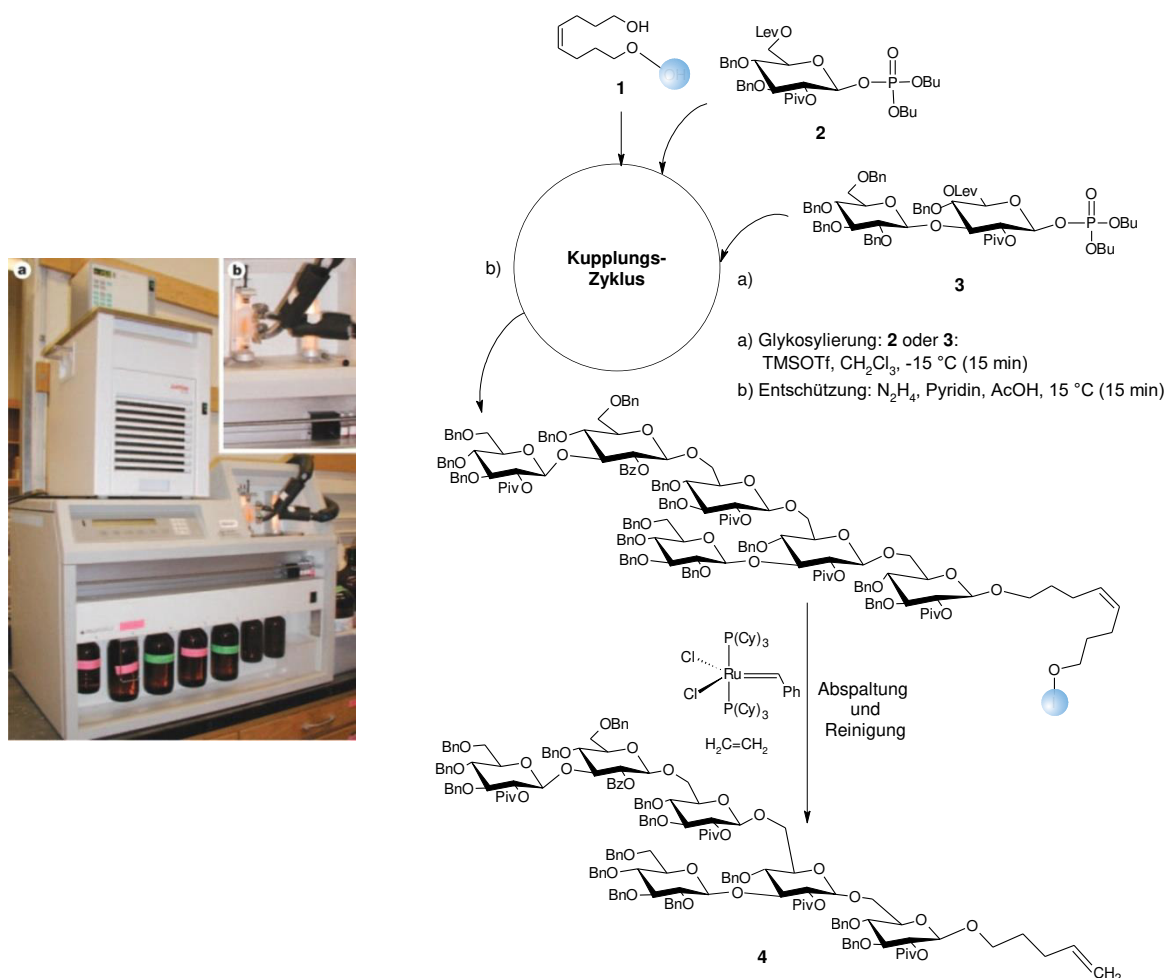


Abbildung 17: Erste automatische Festphasen-Synthese eines Hexasaccharids

Bei der ersten automatisierten Festphasen-Synthese kamen zwei Glykosylphosphat-Bausteine zum Einsatz: Monosaccharid **2** und Disaccharid **3**, die alternierend für die Kupplung durch Aktivierung mit Trimethylsilyltrifluormethansulfonat (TMSOTf) verwendet wurden. Als temporäre Schutzgruppe an beiden Bausteinen wurde Levulinylester genutzt, welches nach Entfernung mit Hydrazin das Nucleophil für den nächsten Glykosylierungsschritt lieferte. Nach Herstellung des Hexasaccharids, erfolgte die Abspaltung vom festen Träger durch Kreuzmetathese. Das vollständig geschützte

Hexasaccharid **4** konnte mit einer Gesamtausbeute von 89%, bestimmt durch HPLC, hergestellt werden. Das terminale Olefin lässt sich weiter funktionalisieren, um einen nukleophilen Konjugationspunkt bereitzustellen.^[138]

Nachdem die grundsätzliche Machbarkeit zur Automatisierung der Oligosaccharid-Synthese bewiesen war, arbeitete die Forschergruppe Seeberger an der Weiterentwicklung dieser vielversprechenden Technologie, um die Herstellung von Oligosacchariden mit anderen wichtigen Resten und Bindungen wie Aminozucker,^[139a,82,139b] Sialinsäuren,^[140] 1,2-cis-Glykosiden^[141] und verzweigten Oligosaccharidsequenzen zu ermöglichen.^[136]

Unter Verwendung der im Wesentlichen gleichen Automatisierungsplattform, berichtete Codée und Mitarbeiter über die Synthese von β -Mannosiden.^[142] Das erworbene Wissen wurde auf eine sehr effektiven Synthese von Globo-H-Hexasaccharid übertragen. Dieses Oligosaccharid wird auf Oberflächen verschiedener Arten von Krebszellen exprimiert und dies stellt ein Schlüsselziel für die Entwicklung von Antikrebs-Impfstoffen und -Therapeutika dar.^[143]

Aufgrund der großen biologischen Bedeutung von Globo-H, wurde das Antigen mit verschiedenen Methoden synthetisiert.^[128,144a,141,144b-d]

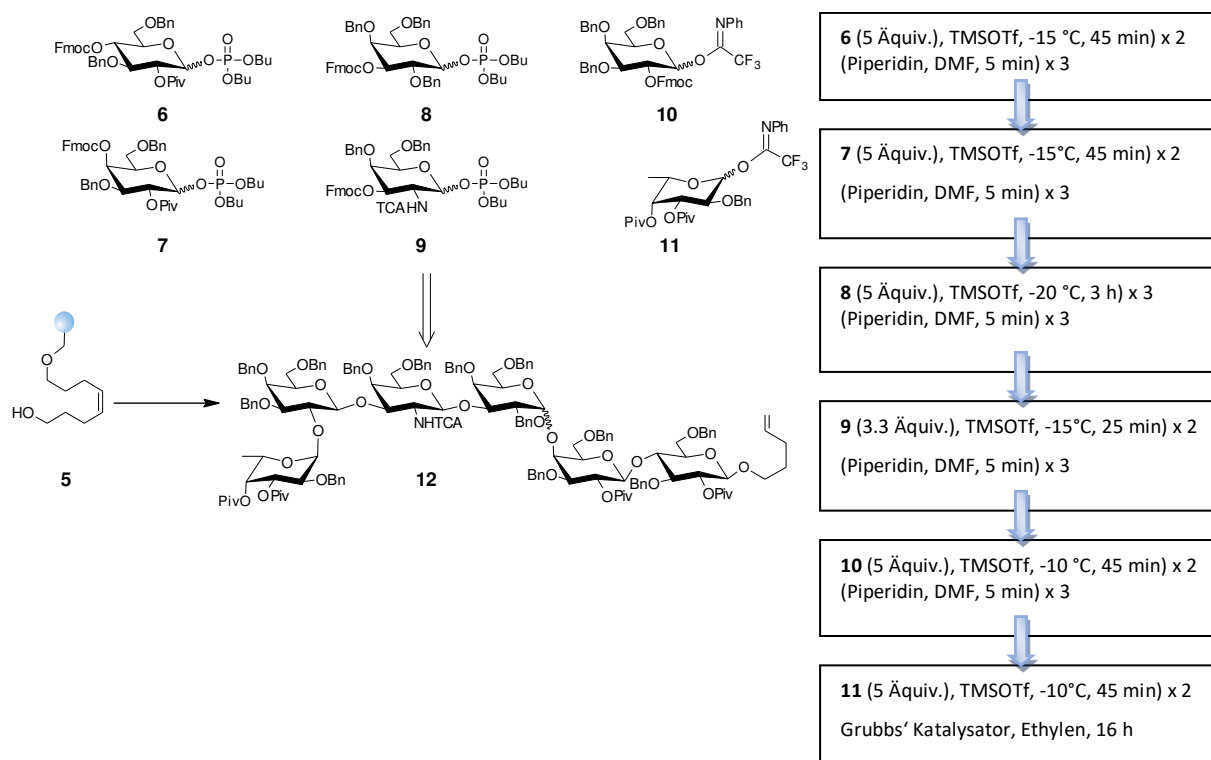


Abbildung 18: Automatische Synthese eine Globo-H Hexasaccharids mit 30% Ausbeute

Seeberger gelang mit der Herstellung zweier tumorassoziierter Kohlenhydrat-Antigenen, die Herausforderungen der 1,2-cis-glykosidischen Bindungsbildung mithilfe eines automatisierten Ansatzes zu lösen.^[141] Sorgfältige Verfeinerungen der Reaktionsbedingungen ermöglichten die

1,2-cis-Galaktosylierung in Dichlormethan-Ether. Für die Herstellung des geschützten Globo-H-Hexasaccharid **12** waren sechs Bausteine (**6 – 11**) erforderlich (Abbildung 18). Mit Ausnahme der letzten Fucoseeinheit **11**, enthielt jedes Monomer Fluorenylmethoxycarbonyl (Fmoc) als temporäre Schutzgruppe. Die Auswahl der Fmoc-Gruppe erfolgte aufgrund ihrer hohen Säurestabilität und leichten Spaltbarkeit in Gegenwart einer schwachen Base wie Piperidin. Zudem stellt Fmoc nach der Spaltung ein Dibenzofluoren-Addukt zur Verfügung, das zur Überwachung der Effizienz der Entschützung über einen quantitativen kolorimetrischen Probenansatz verwendet werden kann.^[105] Zunächst wurde der Glykosylphosphat-Donor **6** über Glykosylierung unter Verwendung von TMSOTf (eine Wiederholung) als Promotor an das Harz **5** gebunden. Es folgte die Entschützung des Fmoc-Substituenten mit Piperidin (zwei Wiederholungen), um einen polymergebundenen Akzeptor bereitzustellen. Das allgemeine synthetische Protokoll bestand aus sich wiederholenden Zyklen der Glykosylierung unter Verwendung von entweder Glykosylphosphat (**6 - 9**) oder Glykosyl-N-phenyltrifluoracetimidat-Donoren (**10 und 11**), gefolgt von der Entschützung mit Piperidin. Die Abspaltung des Endprodukts **12** vom Polymerträger erfolgte durch Olefin-Kreuzmetathese in Gegenwart vom Grubbs'-Katalysator^[145] und Ethylen. Die Gesamtausbeute betrug 30%.

Die Gruppe Seeberger stellte 2012 den ersten vollständig automatischen Festphasen-Oligosaccharid-Synthesizer vor.^[146] Die eigens für die Oligosaccharidsynthese entwickelte Apparatur kombiniert folgende Komponenten: Spritzenpumpen, Magnetventile, ein Kryostat mit Temperaturkontrollsystem für Reaktionstemperaturen zwischen -50 ° C und 90 ° C und einen Fraktionssammler. Das Reaktionsgefäß ist mit einem porösen Glasfilter ausgestattet, der die Zufuhr des Inertgases zur Aufrechterhaltung wasserfreier Reaktionsbedingungen ermöglicht. Das Gas wird auch zur schnellen Entfernung der Reagenzien und Lösungsmittel in der Lösungsphase verwendet. Das Glykosylierungs- und Entschützungprotokoll kann unter Verwendung eines Computerprogramms eingestellt werden. Dies ermöglicht die vollständige Automatisierung von Reaktionen, Temperaturkontrolle, Spaltung und Sammlung des freigesetzten Oligosaccharidprodukts. Im Gegensatz dazu machte der früher verwendete Peptidsynthesizer eine manuelle Temperatureinstellung bei jedem Glykosylierungszyklus und manuelle Sammlung von Reaktionsgemischen oder gespaltenen Produkten erforderlich. Zudem war die Anzahl der Gefäße für Reagenzien begrenzt, wodurch die in einer gegebenen Synthese mögliche Chemie eingeschränkt wurde.

Die Vielseitigkeit dieser neuen Automationsplattform demonstrierten Seeberger und Mitarbeiter durch eine Kombination von drei verschiedenen Glykosylierungsmethoden innerhalb der automatischen Synthese eines verzweigten Glucans vom mannosereichen Typ. Die Herausforderung bei der Herstellung des Kernpentasaccharids **18**, als gemeinsamer Bestandteil aller N-Glykane, bestand in der schwierigen β (1- \rightarrow 4)-Mannosidbindung, die mit dem Disaccharid Chitobiose verknüpft ist, und den beiden Verzweigungen an der zentralen Mannose.^[147]

Mit einem neuen Typ Linker **13**, der unter den gebräuchlichsten Glykosylierungs- und Schutzgruppenentfernungsbedingungen stabil ist, wurde die Durchführbarkeit der vollständigen Automation sichergestellt. Der Aufbau des Chitobioseteils erfolgte durch zwei Glykosylierungs-Entschützungs-Zyklen mit dem Glykosyl-Donor **14**. Die anspruchsvolle β -Mannosid-Bindung wurde mit Donor **15** hergestellt, der mit einer 2-(Hydroxycarbonyl)-benzyl-Abgangsgruppe ausgestattet war.^[148] Nach Behandlung mit TBAF, wurde durch selektive Öffnung des Benzylidenacetals die C6-Hydroxylgruppe unter Verwendung von Boran freigelegt. Danach folgte unter Verwendung des Mannose-N-Phenyltrifluoracetimidat-Bausteins **16** und Aktivierung mit Trimethylsilyltriflat bei 10 °C, eine Bis-Mannosylierung zur Herstellung der verzweigten Pentasaccharid-Struktur. Die Abspaltung vom festen Träger in Gegenwart von MeONa ergab das Produkt **17** im Verhältnis α / β = 1/3. Mittels präparativer HPLC wurde das unerwünschte α -verknüpfte Anomer als Nebenprodukt abgetrennt, danach die resultierende diastereomerenreine Verbindung durch Hydrierung entfernt, um das Zielpentasaccharid **18** in einer Gesamtausbeute von 3,5% zu erhalten (Abbildung 19).

Aus der weiteren Optimierung des vollautomatischen Synthesizers erster Generation, entstand im Wesentlichen mit der gleichen Technologie eine zweite Gerätegeneration, die als Ausgangspunkt zur Fortentwicklung des neuen Gerätekonzepts in dieser Arbeit dient. Das als „J-KEM“-Synthesizer betitelte Gerät wird in nachfolgenden Kapiteln ausführlich beschrieben.

Am Synthesizer zweiter Generation wurde mit der Herstellung eines α - (1,6) verknüpften linearen 30-mer-Mannosids das bis zu diesem Zeitpunkt längste, durch chemische Synthese gewonnene, Polysaccharid hergestellt.^[149] Die Gesamtausbeute von 1% entsprach einer durchschnittlichen Ausbeute von 96% pro Syntheseschritt (Abbildung 20).

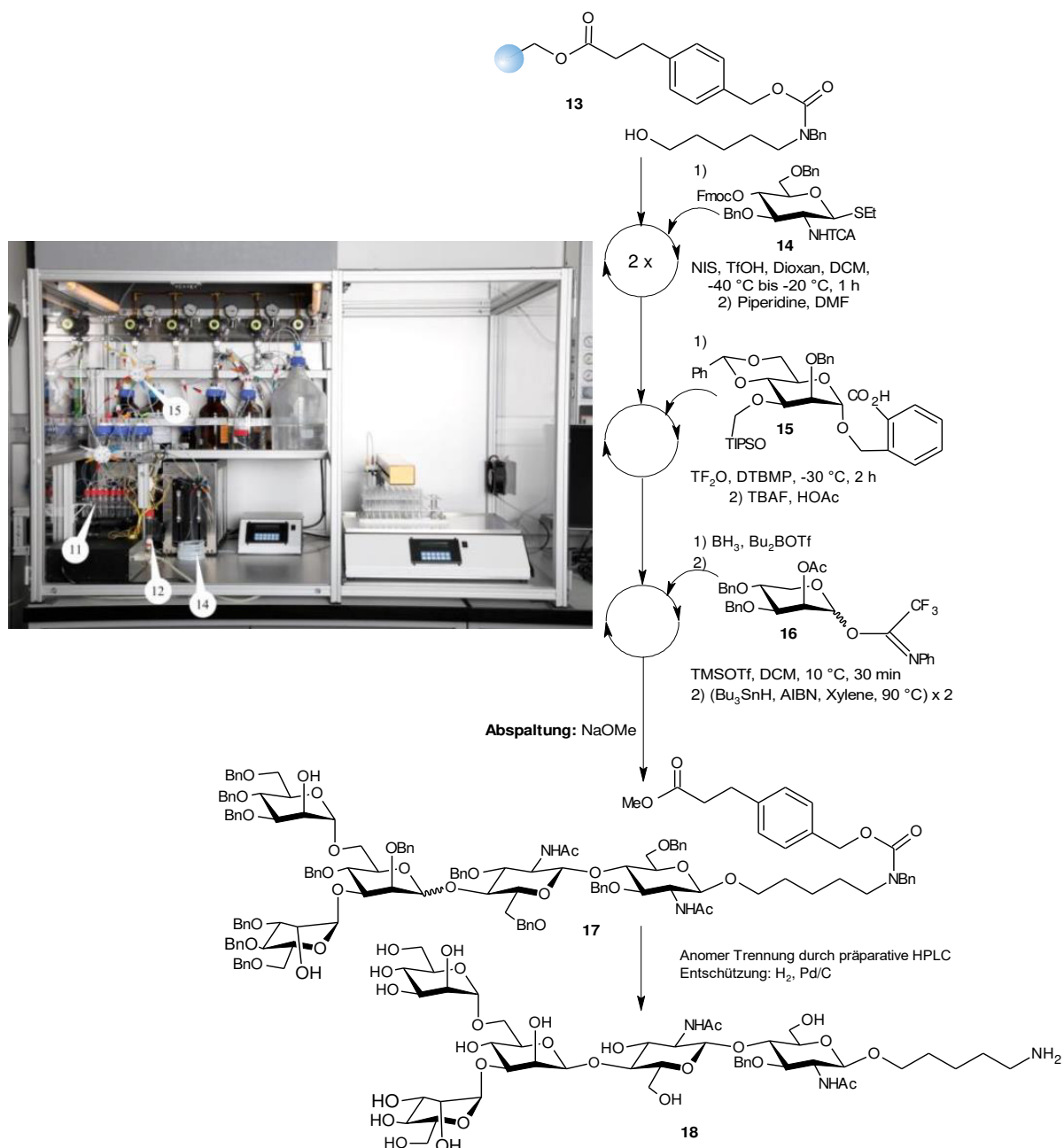


Abbildung 19: Automatische Synthese einer N-Glykan Zielstruktur unter Verwendung des ersten vollautomatischen Oligosaccharid-Synthesizers

Zur Durchführung der Synthese wurde ein modifiziertes Merrifield Harz **19** mit einem fotospaltbaren p-Nitrophenyl-Linker ausgestattet. Jede Glykosylierung wurde unter Verwendung von drei Äquivalenten des Phosphat-Donors **20** mit Trimethylsilyltriflat als Aktivator dreimal bei -15°C wiederholt. Die Verkappung der nicht umgesetzten Hydroxylgruppen in **21** wurde durch Acetylierung mit Ac₂O in Gegenwart von Pyridin beeinflusst. Mit der Entschützung des 6-O-Fmoc-Substituenten mit Piperidin wurde der Glykosyl-Akzeptor **22** für den nächsten Glykosylierungsschritt hergestellt. Um eine hohe Stabilität, Stereoselektivität und Ausbeute während der Synthese sicherzustellen, wurden an den anderen Positionen Benzoylschutzgruppen verwendet. Nach Durchführung von 29 Glykosylierungszyklen mit den vorläufigen Schutzgruppenentfernungsschritten, wurde das

resultierende 29-mer mit dem Donor **23** glykosyliert. Letzteres ist mit einer Spacer-Komponente ausgestattet, um eine effektive „cap-and-tag“ Reinigungstechnik zu ermöglichen.^[150]

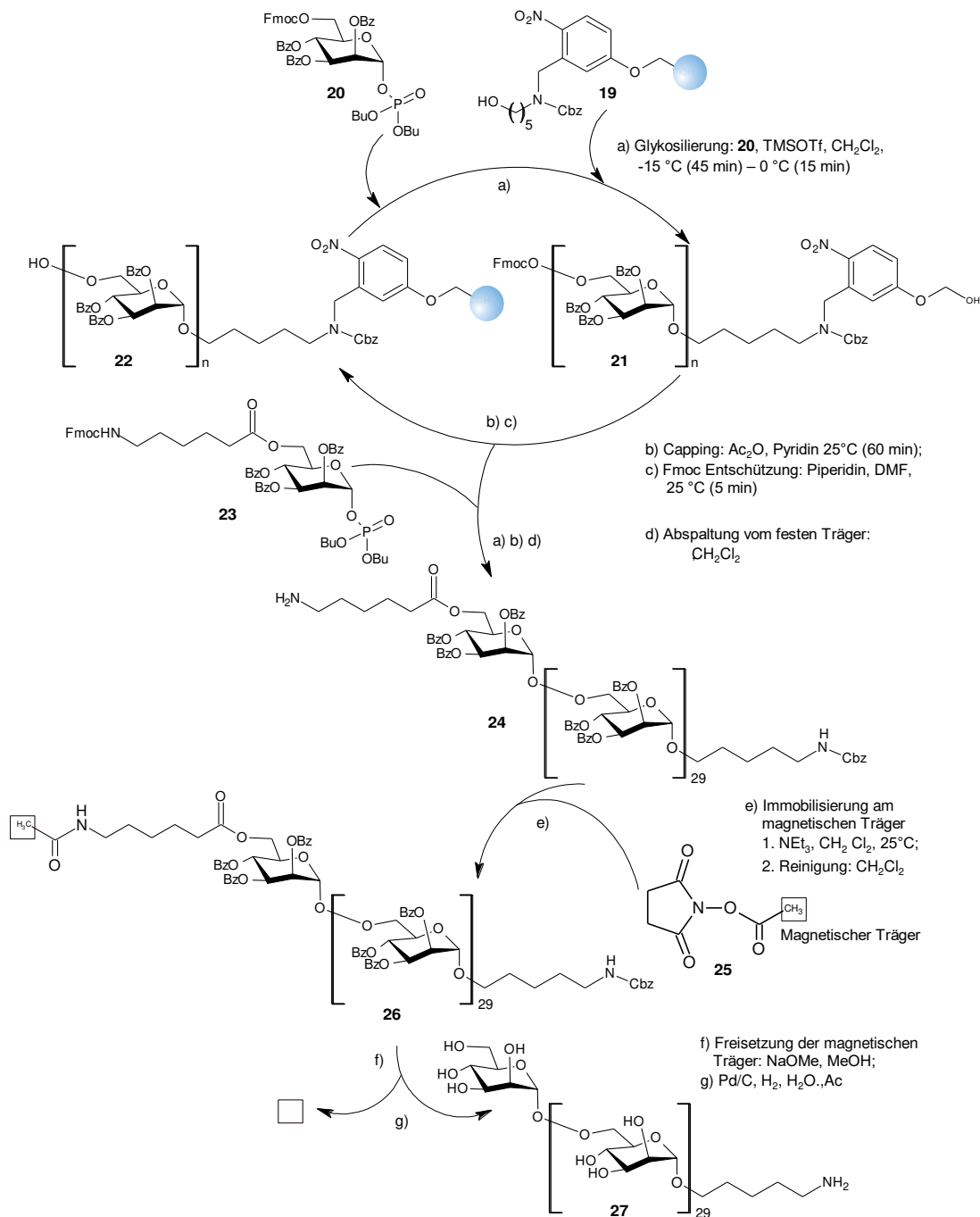


Abbildung 20: Automatische Festphasen-Synthese eines linearen 30-mer α -(1,6)- Oligomannosids unter Anwendung der Catch-Release Reinigung

Nach der Freisetzung aus dem Polymerträger unter Verwendung von UV-Licht, wurde das resultierende 30-mer **24** über die Aminogruppe, die mit einem ϵ -Aminocapronester-Spacer ausgestattet war, an den magnetische Träger zur terminalen Mannoseeinheit konjugiert.

Dieser Ansatz ermöglichte die magnetische Trennung des markierten Oligosaccharids **26**, gefolgt von einfachem Waschen mit Dichlormethan und Methanol. Die anschließende Freisetzung aus den magnetischen Trägern und die gleichzeitige Debenzoylierung mit Natriummethoxid in Methanol, gefolgt von einer Hydrierung zur Entfernung der Cbz-Gruppe, führte zu dem vollständig ungeschützten α -(1,6)-Oligomannoside 30-mer **27**.

1.4.2 HPLC-unterstützte automatisierte Oligosaccharid-Synthese

Basierend auf einem pumpengetriebenen HPLC-System, entwickelte die Gruppe Demchenko eine automatisierte Oligosaccharid-Synthese,^[108] die eine Kombination von Festphasensynthese und kontinuierlichem Fluss darstellt.

Hierzu wurde eine Omnifit-Chromatographie-Säule mit dem TentaGelTM-NH₂ Harz beladenen Glykosyl-Akzeptor **29** gepackt. Die Säule wurde mit dem HPLC-System verbunden, das aus einer Kolbenpumpe mit drei Kammern, einem Detektor mit variablem UV-Bereich und einem Computer mit installierter Standard-HPLC-Betriebssoftware bestand (Abbildung 21). Nach dem Spülen der Säule mit Lösungsmittel, erfolgte die gleichzeitige Anlieferung zweier getrennter Lösungen, die den Glykosyl-Donor und den Promoter (TMSOTf) enthielten. Der Glykosyl-Donor **28** war mit einer 2-O-Benzoylgruppe ausgestattet, um die Stereoselektivitätskontrolle sicherzustellen, und mit 6-OFmoc als selektiv entfernbare temporärer Substituent. Das Mischen erfolgt durch den kontinuierlichen Fluss der Reagenzien über den festen Träger, wobei sie ihn durchdringen, wenn sie durch die Kartusche geleitet gehen. Nach einer relativ kurzen Reaktionszeit, typischerweise 30 bis 60 Minuten, wurden die beiden Pumpen gestoppt und das System über eine separate Pumpe mit Dichlormethan gespült. Mit Piperidin in DMF erfolgte eine Fmoc-Entschützung. Danach wurde das System mit Dichlormethan gespült und die alternierenden Entschützungs- und Glykosylierungsschritte wiederholt, bis das gewünschte Oligosaccharid hergestellt war. Das Syntheseprodukt wurde unter Verwendung einer rezirkulierenden Lösung von NaOMe in Methanol-Dichlormethan vom Polymerträger abgespalten. Resultierend wurde Pentasaccharid **30** in 62% Ausbeute in wesentlich kürzerer Zeit, als bei einem manuellen Syntheseansatz gewonnen.

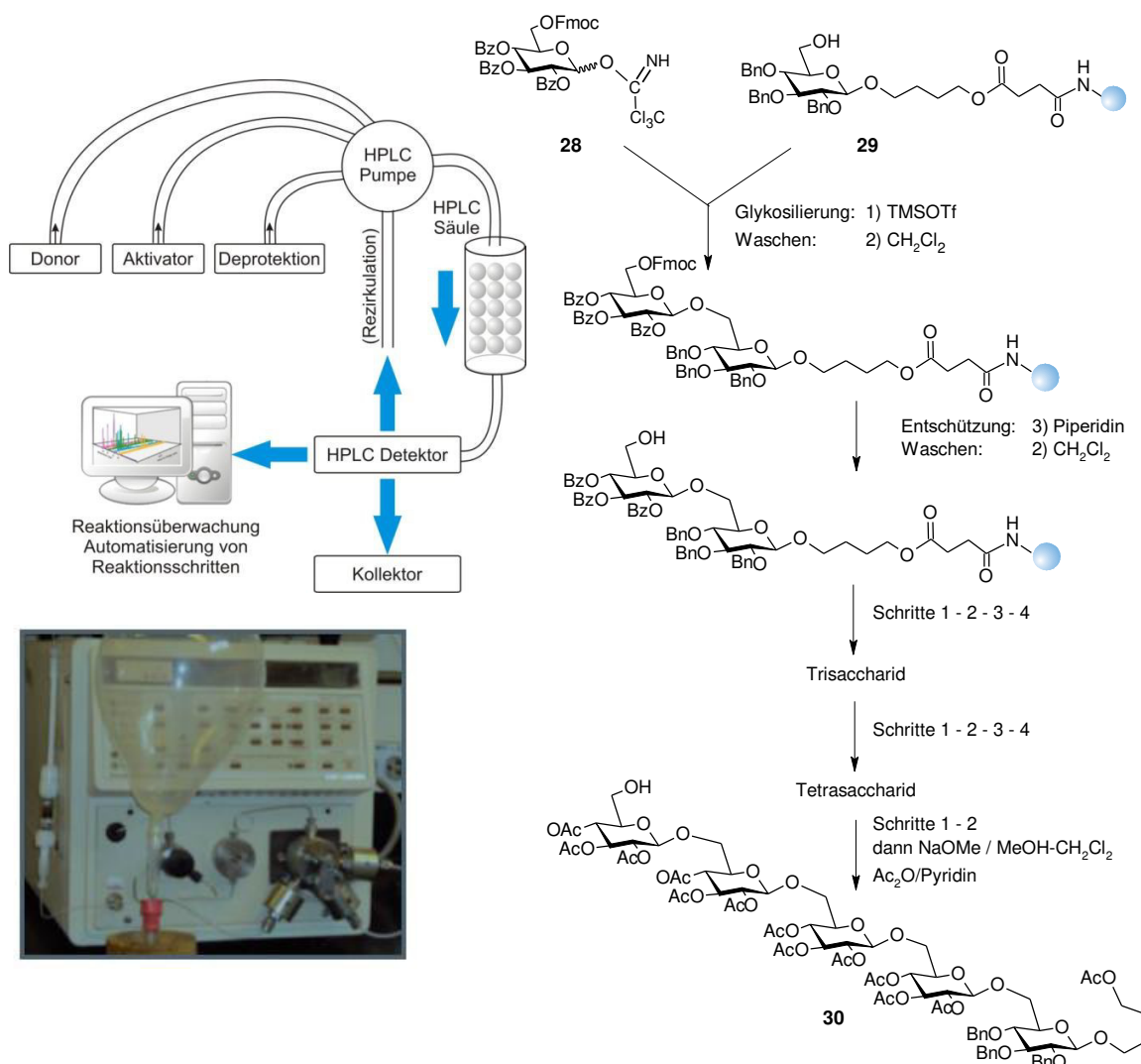


Abbildung 21: HPLC-assistierte automatische Oligosaccharid-Synthese

Die HPLC-unterstützte Synthese kann unter Verwendung einer zugänglichen Ausrüstung durchgeführt werden, wobei die Reaktanten und Reagenzien von der Pumpe oder dem Autosampler abgegeben werden und die Reaktionen durch den UV-Detektor überwacht werden können, das Veränderungen der UV-Absorption der aus der Säule eluierten Lösung aufzeichnet.^[108]

Gegenüber der manuellen Festphasen-Oligosaccharidsynthese bietet die HPLC-unterstützte Synthese folgende Vorteile: schnellere Reaktionszeiten, Echtzeit-Reaktionsüberwachung mit einem HPLC-Detektionssystem, Schritte und Sequenzen können auch mit der Standard-HPLC-Software automatisiert werden. Neuartige O-Benzoxazolyl-Imidate wurden als vielversprechende Glykosyl-Donoren für HPLC-basierte Anwendungen gefunden.^[151]

1.4.3 Alternative Ansätze zur Automation von Oligosaccharid-Synthesen

Mit der Festphasenstrategie wurden bemerkenswerte Fortschritte bei der Herstellung von Oligosacchariden erzielt. Die leistungsfähige und moderne Technik bietet für die Automation viele Vorteile. Jedoch laufen Glykosylierungsreaktionen nicht immer effizient, wodurch oft mehrere Kupplungszyklen für jede Verknüpfung notwendig sind, um entsprechend hohe Ausbeuten zu erzielen. Dies hat zur Entwicklung alternativer Träger für die Oligosaccharidsynthese geführt.^[152] Dazu gehören mit Methoxypolyethylenglykol (MPEG) als polymeren Träger^[95,119,153] unterstützte, mit ionischen Flüssigkeiten^[154] erweiterte und Fluor-Tag basierte Oligosaccharid-Synthesen.^[155] Alle diese Methoden verwenden Substrate mit einem chemischen Aufbau, welche das Substrat unter Reaktionsbedingungen vollständig löslich machen, und danach eine selektive Isolierung des Produkts vom Substrats durch Extraktion oder Ausfällung ermöglichen. Da die Träger unter Glykosylierungsbedingungen löslich sind, muss der Donor nicht im großem Überschuss bereitstehen. Auch sind keine zusätzlichen Kupplungszyklen zur Glykosylierung erforderlich, um eine vollständige Reaktion sicherzustellen. Ansätze aus der Lösungsphasenchemie zur stereoselektiven Oligosaccharidsynthese, lassen sich damit gut auf Automatisierungs-Plattformen übertragen.^[152]

Die Fluor-Tag unterstützte Synthese ist eine relativ neue Strategie zur Gewinnung von Oligosacchariden. Sie beruht auf den Löslichkeitseigenschaften einer wachsenden Oligosaccharid-Kette durch eine Fluormarkierung.

Die Seeberger-Gruppe verwendete für die Synthese des β -(1 \rightarrow 6)-D- Glukopyranosid - Homotetramers **36** einen lösungsbasierten Silizium-Mikrofluidikreaktor (Abbildung 22), den sie für die Fluor-Tag unterstützte Synthese von Oligosacchariden anpassten.^[156] Hierbei ermöglicht der perfluorierte Linker am reduzierenden Ende der Oligosaccharide die effiziente Reinigung durch fluorige Festphasenextraktion (FSPE). Für die Glykosylierungsreaktion in der mikrofluidischen Vorrichtung wurden drei separate Lösungen in Dichlormethan hergestellt: Glykosyl-Akzeptor, Phosphatdonor und der Aktivator (TMSOTf), die jeweils über Spritzenpumpen dem Einlass 1, 2 und 3 zugeführt werden. Durch diffusionskontrollierte Probenahme wird jede Lösung in die Mischzone eingeführt. Die Reaktionsparameter der Komponenten werden durch die Konzentration in der Spritze und durch die Fließgeschwindigkeit jedes Stroms gesteuert. Nach dem Mischen findet die Reaktion innerhalb der Reaktionszone statt. Die Reaktion kann unter Verwendung verschiedener Arten von Instrumenten überwacht werden, die am Auslass des Mikroreaktors angeschlossen sind, einschließlich UV-VIS-, IR- oder Massenspektrometer.

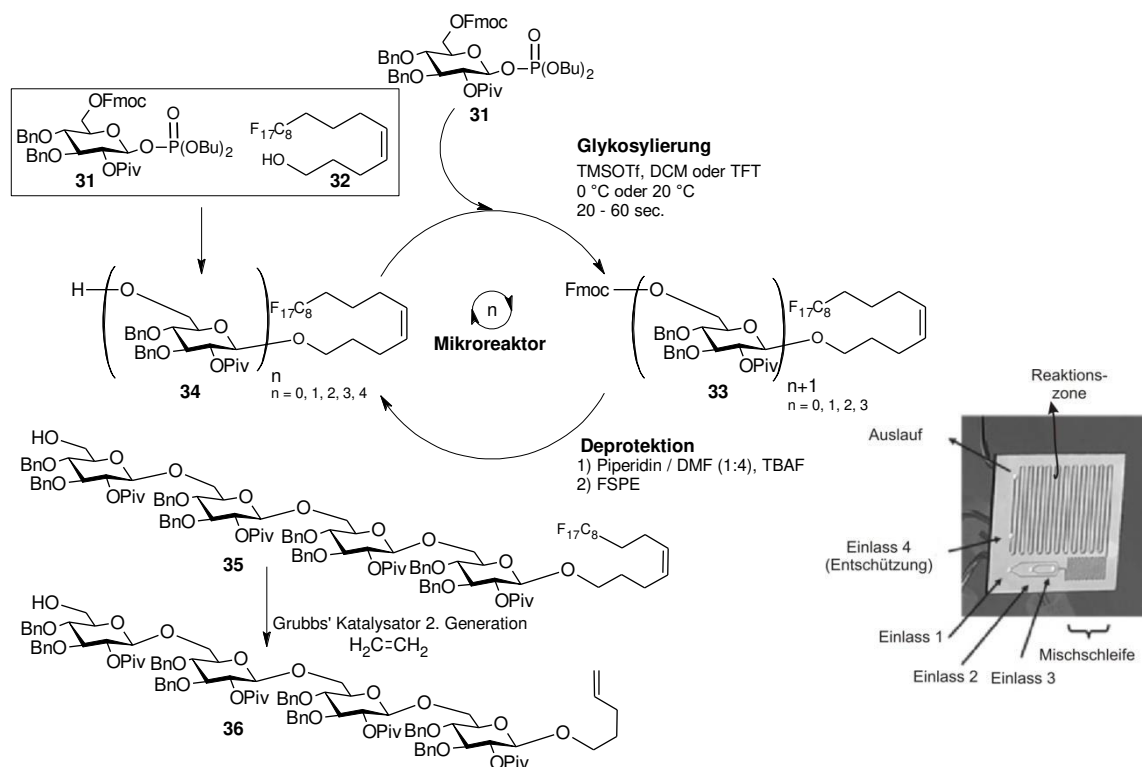


Abbildung 22: Fluor-Tag unterstützte Synthese im Mikroreaktor

Zuerst wurde der Glykosylphosphat-Donor **31** mit einem fluorigen Linker **32** in Gegenwart von TMSOTf glykosyliert.^[156] Nach dem Glykosylierungsschritt folgte zur Herstellung des Fluorglykosyl-Akzeptors **33** die Entfernung der Fmoc-Gruppe mit Piperidin und TBAF. TBAF wurde eingesetzt, um 6-O-TMS zu entfernen, das als Ergebnis konkurrierender Silylierung gebildet wurde. Danach folgten wiederholend die Glykosylierung mit dem Glykosyl-Donor **31** und anschließende Entschützung, bis das Tetrasaccharid **35** hergestellt war. Die Abspaltung vom fluorhaltigen Träger mittels Grubbs-Katalysator der zweiten Generation zur lieferte als Produkt das β -(1 \rightarrow 6)-D- Glukopyranoside Homotetramer **36**.

Die Reaktionszeiten für die Glykosylierungen betragen 20 Sekunden für die Bildung des Disaccharids und jeweils 60 Sekunden für die Tri- und Tetrasaccharide. Die Ausbeuten für die Reaktionen nach der Reinigung betragen jeweils 97, 90 und 95% für Di-, Tri- und Tetrasaccharide.

Zu den Vorteilen der Verwendung von Mikrofluidreaktoren gehören die Einfachheit der Maßstabsvergrößerung, leichte Optimierung von Reaktionsbedingungen durch Variierung von Fließgeschwindigkeit und Reagenzien-Mengen, bessere Kontrolle der Reaktionstemperatur und die Möglichkeit mit verschiedenen Analysetechniken und Automatisierungsplattformen zu arbeiten. Zu den Systemnachteilen gehören: mögliche längere Reaktionszeiten, wenn eine geringe Fließgeschwindigkeit erforderlich ist, und die Inkompatibilität mit Feststoffen.

Pohl entwickelte unter Verwendung des Festphasenextraktionsprotokolls eine Fluor-Tag unterstützte Glykosylierungsstrategie.^[157] Der fluorhaltige Linker **37** war am reduzierenden Ende des Glykosyl-Akzeptors **39** angebracht und glykosylierte unter Verwendung eines Überschusses an Mannosetrichloracetimidat **38** durch Aktivierung mit Trimethylsilyltriflat. Danach folgte eine Acetatspaltung an C2 zur Kettenverlängerung. Nach jedem Syntheseschritt wurde das Zwischenprodukt unter Verwendung eines automatisierten dreistufigen Fluor-Festphasenextraktionsprotokolls gereinigt. Bei der Filtration auf der FSPE-Säule, eluieren alle fluorfreien Moleküle unter Verwendung einer 20%-igen Lösung von Wasser in Methanol. Danach können fluorhaltige Moleküle, die auf der Säule zurückgehalten werden, unter Verwendung einer fluorophilen Wäsche, wie Methanol oder THF, freigesetzt werden. Das resultierende α -(1,2)-Tetramannosid **40** konnte so mit einer Gesamtausbeute von 79% hergestellt werden. Diese Methode wurde für die Synthese von verzweigten Oligomannosiden erweitert.

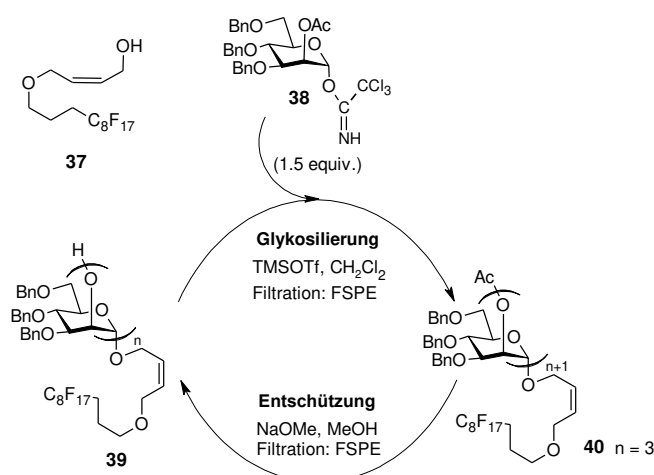


Abbildung 23: Automatische Synthese von linearen Mannose-Oligosacchariden unter Verwendung von Fluor-Tag Unterstützung und FSPE Filtration

Der spezielle Linker ist während des Glykosylierungszyklus stabil und bewahrt die Löslichkeit der wachsenden Oligosaccharid Kette sowohl in organischen Lösungsmittel, die während der Glykosylierung Verwendung finden, als auch in wässrig-organischen Mischungen, welche zur Reinigung durch FSPE eingesetzt werden. Die damit identischen Reaktionslösungsbedingungen und Reinigungsprotokolle zwischen Glykosylierungs- und Entschützungsreaktionen machen die Synthese der Automation zugänglich. Die Linkerstruktur ist eindeutig entscheidend für das Design einer zuverlässigen Lösungsphasen-Automatisierungsstrategie. Diese Ergebnisse erweisen sich als vielversprechend für einen automatisierten Fluorphasen-Ansatz zur Oligosaccharidsynthese.^[158]

1.5 Zielsetzung der Arbeit

Überblick

Zu Projektbeginn wurden im Exposé^[159] zur vorliegenden Arbeit die grundlegende Ausrichtungen, funktionale Anpassungen & Erweiterungen, sowie aktuell zu überarbeitende Teilgebiete dokumentiert. Diese Themen werden hier in ihren Hauptpunkten zusammengefasst und aufgezeigt:

- Einfache Erlernbarkeit der Synthesedurchführung
- Standardisierung der Synthesen
- Reproduzierbarkeit der Synthesen
- Modularisierung der Ansteuerungsprogramme
- Skalierbarkeit der Synthesevorgänge
- Kalibrierungs-, Wartungs-, und Test- Durchläufe
- Verwaltung & Organisation von Syntheseabläufen
- Verwendung von Standardkomponenten

Einfache Erlernbarkeit der Synthesedurchführung

Bei der Übernahme der Arbeit im Jahr 2010 ist die Ansteuerung des J-KEM-Synthesizers, im Folgenden auch als Altsystem bezeichnet, durch Zeilenkommandos innerhalb einer speziellen Anwendungssoftware realisiert. Diese werden zur Erstellung einer Ablauf-Befehlsliste manuell in einen Texteditor kopiert. Es besteht bisher keine Möglichkeit Code-Sequenzen zusammenzufassen und diese in anderen Abläufen zu integrieren. Da die Anzahl der Schleifenebenen auf eins begrenzt ist, sind sinnvolle Kaskadierungen nicht programmatisch umsetzbar. Für einen neuen Anwender ergeben sich aus diesem Grund extreme Hürden bezüglich Erlernbarkeit und flexibler Umsetzung neuer Synthesen am Syntheseautomaten. Bereits einfache Prozeduren erfordern ein tiefgehendes Systemverständnis, welches neben der chemischen Umsetzung der Synthese zu erlernen ist. Leider liegen weder Schaltpläne, Konstruktionszeichnungen, noch Ablaufdiagramme oder Pflichtenhefte als Dokumentationsgrundlage vor. Dieser Mangel vergrößert die Einarbeitungszeit erheblich und lässt den notwendigen Wissensstand undefiniert. Der sporadische Benutzer übersieht hier leicht aufeinander aufbauende Schrittketten. Bei Wartungen und Reparaturen werden oftmals Funktionalitäten außer Kraft gesetzt, die erst im Nachgang in ihren Auswirkungen in Erscheinung treten. Komplexere chemische Ansteuerungen werden durch mehrfaches manuelles Aneinanderkopieren realisiert. Hierdurch kommt es vermehrt zu Fehlern, da Abschnitte in Code Sequenzen zwar ähnlich aussehen, sich jedoch innerhalb weniger Parameterwerte (z.B. Ventilnummer, Haltezeit, usw.) funktional unterscheiden. Der verwendete Editor erlaubt keine Suche und Ersetzung von kompletten Code- Abschnitten. Sobald innerhalb einer Zeile noch

zusätzlicher Kommentar vorliegt wird diese nicht mehr als identisch angesehen und somit auch nicht mit den neuen Parameter korrekt ausgetauscht. Hierdurch liegen eine Vielzahl von Dateien auf den Synthese-Rechnern, die unterschiedlichen Entwicklungszustand aufweisen, jedoch nur von dem eigentlichen Ersteller eindeutig in ihrer Funktionalität und Verwendbarkeit temporär zuzuordnen sind. Die Verwendung dieser "Bibliotheken" ist nur Anwendungsexperten geläufig. Für Neueinsteiger und Nichtexperten ist die Einarbeitungszeit kaum zu reduzieren. Funktionierende Abläufe werden entsprechend innerhalb der jeweiligen Arbeitsgruppe individuell weiter gepflegt. Dadurch driften auch ehemals identische Codesequenzen individuell auseinander und lassen sich nicht mehr gemeinsam warten.

Zielsetzung:

- Einfache Erlernbarkeit der Synthesedurchführung durch die effiziente Benutzbarkeit für Nichtexperten.

Innerhalb der speziell unter Visual Basic erstellten Applikationssoftware fehlen jegliche Hinweise oder übliche Hilfen, die eine Einarbeitung erleichtern oder Fehler zu vermeiden helfen würden. Es liegt am Institut kein Quellcode oder funktionale Beschreibungen vor. Beides wird auch nicht auf Nachfrage zu akzeptablen Konditionen vom Hersteller bereitgestellt. Somit ist einer grundlegende Neuentwicklung und Konzeption der Benutzersoftware notwendig. Hierbei sollen dann selbsterklärende Oberflächen die Einarbeitung erheblich intuitiver gestalten. Auch sollten chemisch relevante Informationen direkt eingebbar werden. Da in der Software keinerlei Plausibilitätskontrollen erfolgen, kam es wiederholt zu Überläufen im Abfallsammelbehälter und in den Probenaufnahmegefäßen. Einige Synthesen mussten abgebrochen werden, nachdem es zu Transportproblemen durch Mindermengen und Fehlansaugungen gekommen ist. In solchen Fällen sind die vorliegenden Teilreaktionsprodukte nur unter erheblicher Mehrarbeit wieder aufzureinigen und vorzubereiten, dass eine Synthesefortführung erfolgen kann. Somit wären softwareseitige Hinweise und Hilfen für die Syntheseplanung und Durchführung hilfreich. Neben Links zu den Veröffentlichungen bereits durchgeführter Synthesen, können Funktionen zum aktiven Support und Informationsunterstützung mit in die Software integriert werden.

Zielsetzung:

- Einfache Erlernbarkeit der Synthesedurchführung durch erhebliche Vereinfachung in der softwareseitigen Bedienung.

Da die Synthesen jeweils von unterschiedlichen Anwendern durchgeführt wurden, gibt es keine sinnvolle allgemeingültige Benennung, nach welcher ein Neueinsteiger schnell und direkt zur ausbaubaren Beispielabläufen gelangen könnte. Die bisherigen erfolgreichen Synthesen stellen

jedoch eine sinnvolle Ausgangsbasis für die Bewertung und Einschätzung der notwendigen Funktionalität dar. Somit sollten in einem ersten Schritt die bisherigen Ablaufprogramme gesichtet, sortiert und strukturiert werden. Hierzu werden zu den jeweiligen Publikationen die durchgeführten Ablaufprogramme von den Anwendern angefragt. Die hieraus resultierenden Informationen können dann direkt über die Software den Anwendern bereitgestellt werden. Durch eine geführte Auswahl bekommt der Anwender dann die ehemalige Ansteuerungssequenz in den Zugriff. Hierdurch erhöht sich sowohl die Reproduzierbarkeit wie auch die Effizienz in der Systemnutzung.

Zielsetzung:

- Einfache Erlernbarkeit der Synthesedurchführung durch den direkten Zugriff auf bisherige Synthesen.

Die Einweisung neuer Anwender erfolgt mündlich und in den meisten Fällen parallel im Umfeld einer aktuell durchgeführten Synthese. Bei diesen Syntheseabläufen sind oftmals Sonderfälle, die jetzt untersucht werden sollen, integrierter Bestandteil. Dadurch werden Zwischenschritte und gerade interessierende Sonderfälle besprochen, die in der Normalsynthese nicht enthalten sind. Um die Möglichkeiten der Fehlbedienung zu reduzieren, sollen die zu beachtenden Systemabläufe durch protokollierte Schrittketten und / oder Checklisten unterstützt werden. Im ersten Schritt kann hierzu die Erstellung papiergebundener Ablaufprotokolle erfolgen, da hierdurch sowohl eine Standardisierung der Arbeitsschritte als auch die Grundlage zu einer nachfolgenden Automatisierung erfolgen kann. Diese Standardisierung kann weiterführend um die im Umfeld zur Synthese ablaufenden Vorbereitungen (Gefäßreinigung, Vorbereiten oder "Trocknen" von Lösungsmitteln, usw.) ergänzt werden. Da hierdurch dem jeweiligen Benutzer alle für diese Arbeitsebene relevanten Informationen vorliegen, ist sowohl die Fehlbedienung reduziert wie auch die Systemsicherheit erhöht. Es ist hierdurch eine unterschiedliche Tiefe in der Einarbeitung möglich, da den Anwendern immer alle notwendigen Informationen bereitstehen sich aufgabenbezogen detaillierter einzuarbeiten. Langfristig sollen hierüber auch die Auswertungen und Analysen mit in diese Standardisierungsverfahren integriert werden.

Zielsetzung:

- Einfache Erlernbarkeit der Synthesedurchführung durch die Bereitstellung von Schrittketten und Checklisten.

Standardisierung der Synthesen

Für eine Vielzahl von Prozessen benötigt man immer wieder die gleichen Grundfunktionalitäten. Diese können die Zulieferung von Volumina, das Spülen von Systemabschnitten,

Temperierungskurven oder Positionierung des Fraktionssammlers abbilden. Anpassungen oder Erweiterungen können im Altsystem vom jeweiligen Anwender nur in seiner aktuell verwendeten Gesamt-Ansteuerungssequenz vorgenommen werden und sind somit nicht allgemeingültig austauschbar. Eine effizientere Nutzung würde sich aus der Aufteilung der Gesamtsequenzen in kleinere funktionale Teilkomponenten ergeben. Diese sind dann sowohl von allen Anwendern direkt verwendbar. Durch die Übersichtlichkeit lassen sich solche Komponenten erheblich einfacher warten und anpassen.

Zielsetzung:

- Standardisierung der Synthese durch allgemein verwendbare Module.

Durch die geplante Modularisierung der einzelnen Funktionsschritte können die Synthesen in wiederverwendbare Standardelemente aufgeteilt werden. Diese für sich eigenständig testbaren Teilelemente bilden dann die Grundlage aus denen auch komplett neue Synthesen direkt zusammengestellt werden können. Da diese Elemente ihre Funktionalität bereits mehrfach reproduzierbar unter Beweis gestellt haben, müssen die jeweiligen Ansteuerung nicht real ausprobiert werden. Dadurch ist eine erhebliche Reduktion der Vorlaufzeit an den jeweiligen Systemen möglich, da zur eigentlichen Entwicklung der Synthesen kein Hardwarezugriff benötigt wird. Somit werden die Automatisierungssysteme nur noch für die Durchführung der Synthesen verwendet. Dieser Demomodus hat auch den Vorteil, dass der Ablauf einer Synthese im Zeitraffer simuliert werden kann. Hierdurch lassen sich eine Vielzahl von Problemen bereits im Vorfeld erkennen und beseitigen. Durch die Anpassungen in den neu abgeleiteten Modulen lassen sich alle gewünschten Flexibilisierungen realisieren und sobald diese als funktional erkannt wurden als neuer Standard etablieren.

Zielsetzung:

- Standardisierung der Synthese durch Wiederverwendbarkeit und Entwicklungsvereinfachung.

Weiteres Verbesserungspotenzial ergibt sich bei der Ressourcen- & Energienutzung. Das zur Verfügung stehende Gyrostat von JULABO wird aktuell gegen Raumtemperatur gekühlt. Da in den Laboren Kühlsole zur Verfügung steht, ist es sinnvoll hier diese Art der Umlaufkühlung zu verwenden. Die jeweiligen Kühlzeiten sollten sich dadurch erheblich reduzieren lassen. Auch wurde bei der Analyse wieso die Temperierungszeiten so lang sind, festgestellt, dass die Lüftungsgitter erheblich verschmutzt sind (6-8 mm geschlossene Flusendecke) was zu einer extremen Verschlechterung der Wärmeabfuhr führt. Des Weiteren sind die Regelungsparameter auf den werksseitigen Defaultwerten verblieben. Diese entsprechen nicht dem vorliegenden Sachverhalt des existierenden

Regelkreises und sollten somit auf die aktuellen Werte parametrisiert werden.

Im Altsystem werden die in den Zulieferschläuchen bei der Dosierung angelieferten Buildingblockvolumina durch den nachfolgenden Spülprozess in den Abfallbehälter verworfen. Die Rückführung der Buildingblockvolumina aus den Zulieferschläuchen wäre eine kosteneffizient Anpassung der Module.

In der bisherigen Ablaufsteuerung sind nur rudimentäre Befehle für die verwendete Kloehn -Pumpe realisiert. Um die gesamte Funktionalität nutzen zu können, ist eine Flexibilisierung in der Pumpenansteuerung notwendig. Gerade was die Zuführungsdosierung bei kleinen Schrittzahlen angeht, um ein tropfenweises Dosieren zu ermöglichen muss die Parametrierungsmöglichkeit erheblich erweitert werden. Gleiches mit umgekehrten Vorzeichen gilt für die Spritzenbefüllung, da hier das System mit wesentlich höheren Geschwindigkeiten gefahren werden kann.

Bisher sind bei den seriellen Kommunikationen nur die minimal notwendigen Steuerleitungen kontaktiert. Dies reicht zwar für eine grundlegenden Interaktion aus, sorgt jedoch bei längeren Sequenzen häufig für Pufferüberläufe und daraus folgend für das Einfrieren des Systems. Somit sollten alle Kabel um die Hardware-Handshake-Verwendung bei der Kommunikation mit den seriellen Komponenten erweitert werden.

Zielsetzung:

- Standardisierung der Synthese durch effizientere Automatisierung.

In den J-KEM-Systemen kommt es bei längeren Synthesen (ab 6 Stunden Gesamtlaufzeit) gehäuft zum Versagen von Teilkomponenten des Automatisierungssystems. Hierbei sind unterschiedliche Effekte zu beobachten. Die Kloehn-Pumpen fangen bei Positionierungen mit der aktuell niedrigsten Geschwindigkeit an zu brummen und fahren nicht auf die vorgegeben Positionen. Da es keine Möglichkeit gibt Sprungmarken einer Gesamtsynthese zu setzen, kann bei solchen Sachverhalten, trotz vielleicht erfolgreicher Abhilfe der Fehlersituation, eine teilabgearbeitete Synthese nicht an der aktuellen Position erneut gestartet werden. Die Synthese muss komplett neu aufgesetzt oder umprogrammiert werden. Vom eingesetzten Materialverlust und der investierten Arbeitszeit ganz zu schweigen. Somit ist die Vermeidung von Ausfällen und Systemabstürzen eine grundlegende Anforderung an das neue Automatisierungskonzept.

Der Fraktionssammler lässt sich nach einer Fehlpositionierung nicht mehr verfahren und die ausgekoppelten Syntheseproben werden alle in das gleiche Auffanggefäß exportiert. Durch das begrenzte Volumen kommt es hier nicht nur zu einer Vermischung, wodurch keine Analysemöglichkeit mehr besteht, sondern auch zur Kontaminierung des Gesamtsystems.

Vereinzelnde Ventile hängen fest und lassen sich nicht mehr schalten. Hierdurch kommt es zu undefinierten Zulieferungen in das Reaktionsgefäß welche die Synthesen negativ beeinflussen. Selbst

wenn dieser Effekt nur bei Spülungen auftritt hat dieser zur Folge, dass durch das Überschreiten der real existierenden Volumina Gas in das System eingetragen wird und somit Dosiervolumina nicht mehr stimmen.

Da ein vorliegender Wunsch, die Syntheseführung in Teilbereichen bei Temperaturen unter -60 °C durchzuführen, gehäuft zu Abstürzen führte, wurde auch hierzu eine umfangreiche Analyse durchgeführt. Diese ergab ein Temperierungsproblem mit Julabo bei tieferen Temperaturen, da hierbei das verwendete Kühlmittel aufgrund Voluminaabnahme dem Kompressor den angestrebten Druckaufbau nicht ermöglichte. Auffüllen der Kühlmittelmenge löst zwar das Tieftemperatur-Problem, führt jedoch bei Raumtemperatur zum Überlauf, da nun zu viel Kühlmittel vorhanden ist. Somit ist ein Überlaufgefäß mit Reservoir- Funktion vorzusehen, um den Gesamttemperaturbereich abzudecken.

Zielsetzung:

- Standardisierung der Synthese durch Fehlerreduktion, Fehlervermeidung und Fehlermanagement

Bei den individuellen Anpassungen und Änderungen und Reparaturen werden unterschiedlichste Varianten der Lösungsfindung verwendet, z. B. kommen beim Austausch von Schläuchen unterschiedliche Durchmesser zum Einsatz. Aufgrund der abweichender Schlauchdurchmesser stimmen die jeweiligen Totvolumina nicht mehr mit den hinterlegten Prozessdaten überein. Damit ist die ins Reaktionsgefäß eingetragene Menge der Reaktanten nicht stimmig. Auch beim Ventilwechsel sollte auf gleiche Funktionalität (Öffner / Schließer), sowie gleiche elektrische Leistungsdaten geachtet werden, um vergleichbares Schaltverhalten zu erhalten.

Hieraus ergeben sich Anforderungen für die sinnvolle Lagerhaltung von Ersatzteilen und Hinweise für die Einbau und Schrittkette bei Funktionstest.

Zielsetzung:

- Standardisierung der Synthese durch Wartbarkeit und Reparaturunterstützung.

Reproduzierbarkeit der Synthesen

Damit keine der bisher gewonnenen Informationen zu den Synthesen verloren gehen, wäre die vollständige Weiterverwendung der bisher erarbeiteten Synthese- Abläufe trotz veränderter Hardware und neuem Softwarekonzept anzustreben. Hierzu sollten die bisherigen Ansteuerungsprogramm ohne weitere manuelle Anpassung, das heißt 1:1- Übernahme in der neuen Konzeption verwendbar sein. Somit ist entweder ein Konverter-Programm, mit dem die alten

Funktionen auf die neuen Funktionalität gemappt werden oder ein reduzierter Befehlssatz identisch zu implementieren.

Zielsetzung:

- Um eine Reproduzierbarkeit der Synthesen zu erreichen sollen die bisherigen Syntheseabläufe 1:1 und möglichst direkt verwendbar sein.

Da die im Jahre 2010 am Institut vorhandenen beiden Systeme unterschiedlichen Software- und Hardwareausbauzustand aufweisen, lassen sich aktuell keine Synthesen geräteübergreifend miteinander vergleichen. Somit wäre es sinnvoll sowohl die Hardware wie auch die Software und die Ansteuerungssequenzen universell zu standardisieren. Dies würde es ermöglichen Synthesen parallel an mehreren Systemen gleichzeitig durchzuführen, um Aussagen zur Reproduzierbarkeit zu erhalten.

Zielsetzung:

- Reproduzierbarkeit der chemischen Synthesen unabhängig von dem jeweilig verwendeten Automatisierungssystem

Aufgrund der wissenschaftlichen Relevanz komplexer Glykan- Strukturen, lässt sich bereits abschätzen, dass die zu synthetisierenden Makromoleküle immer komplexer und vor allem immer größer werden. Hieraus resultiert automatisch eine längere Laufzeit der Systeme, wenn diese durch Monomere oder funktionale Buildingblocks hergestellt werden. Somit darf in solchen Fällen nicht der Automat das begrenzende Medium darstellen. Einige der zu eliminierenden Probleme der aktuellen Systeme sind nachfolgend aufgelistet:

- Fehlermeldungen des Temperierungsgerätes
- Festfahren des Fraktionssammlers
- Aufhängen des Controllers
- Gasbildung in der Spritze
- Druckinstabilitäten in der Versorgung

Diese Effekte sind nach dem aktuellen Stand die Hauptursachen kompletter Syntheseausfälle.

Zielsetzung:

- Um eine Reproduzierbarkeit der Synthesen zu erreichen ist eine zwingende Voraussetzung die Langzeitstabilität des Automatisierungssystems

Modularisierung der Ansteuerungsprogramme

Bei den bisherigen Ansteuerungsprogrammen ist die Wiederverwertbarkeit erheblich eingeschränkt. Für jede neu durchzuführende Synthese wird ein komplett neu zusammengestelltes Ablaufprogramm generiert. Da sich jedoch Teilbereiche wiederholen, wäre eine entsprechende Mehrfachverwendungsmöglichkeit der Module realisierbar. Hierzu müssen die monolithischen Steuerprogramme in funktionale Einzelbestandteile aufgeschlüsselt werden. In diesen sind dann auch individuelle Zeitoptimierungen möglich, da viele Codebereiche "Altlasten" beinhalten. Auch wäre eine Formalisierung von Komponenten sinnvoll, da einige Strukturen und Abläufe sich mehrfach im System befinden.

Die systematische Standardisierung der Steuerungsprogramme stellt hierzu die Grundlage, denn die vielen Individualentwicklungen generieren ein erhebliches Fehlerpotential.

Zielsetzung:

- Modularisierung der Ansteuerungsprogramme, um eine Wiederverwendbarkeit der funktionalen Module zu erreichen.

Wird der Ansteuerungscode auf überschaubare Module herunter gebrochen, ist auch eine nachweisbare und systematische Optimierung der Prozessketten möglich. Bisher wurden mehrere Anpassungen an unterschiedlichen Stellen im Code vorgenommen, um in einem Schritt das gewünschte Ergebnis zu erhalten. Somit waren die Effekte nicht wirklich eindeutig auf die jeweiligen Änderungen rückführbar. Der Anwender braucht sich in dem modular aufgebautem Code nur noch mit der Strukturebenen befassen, die aktuell zielorientiert überarbeitet werden soll. Damit ist eine erhebliche Reduktion der zu erlernenden Komplexität erreichbar, welches sich auch in der Einarbeitungszeit und somit in der Arbeitseffizienz niederschlägt. Da als Ziel jedes Modul eine definierte Funktionalität aufweist, sind dann die Kombinationen von Prozessen möglich. Hieraus lassen sich somit auch die komplexesten Syntheseverfahren überschaubar abbilden.

Zielsetzung:

- Die autarken Komponenten können in einer Kaskadierbarkeit der Module Einsatz finden.

Eine der Herausforderungen ist die Reduktion der Synthesezeiten aufgrund der steigenden Komplexität der Verfahren. Eine der effizientesten Möglichkeiten hierzu wäre die Kaskadierbarkeit der Systemmodule. Auch die Parallelisierung von Verfahrensmodulen könnte den Zeitaufwand erheblich reduzieren. Zum Beispiel kann während einer Temperierung bereits das Spülen der Systemkomponenten erfolgen. Auch das Aufziehen von Reaktanten in den Spritzen, kann neben

anderen Aktionen autark verlaufen. Das Verfahren vom Fraktionssammler als eigenständige Funktionalität würde wiederum eine parallele Abarbeitung erheblich vereinfachen.

Zielsetzung:

- Durch die Modularisierung ist es möglich eine Parallelisierung von Prozessabläufen zu erreichen.

Um auf die unterschiedlichen Wünsche der Anwender reagieren zu können, muss sich die Anwendungssoftware erheblich verändern. Es gibt User, die nur vorhandene Synthesen mit unterschiedlichen Buildingblocks wiederholt abarbeiten wollen. Andere versuchen die jeweilig enthaltenen Prozessparameter anzupassen, um eine erhöhte Ausbeute zu erreichen. Auch die komplette Neuerstellung von Synthesen oder Synthesemodulen soll über die Software möglich sein. Da zukünftig mit erweitertem Einsatz neuer Hardwaremodulen zu rechnen ist, müssen auch diese entsprechend programmatisch im Zugriff stehen. Es wäre somit hilfreich für die unterschiedlichen Anforderungen auch eine sinnvolle Bedienung bereitzustellen. Eine Variante wären die Einbindung von Funktionen, in denen Experten Oberflächen erstellen können, die für die nachfolgenden Benutzerebenen dann nur noch zu parametrieren sind. Dann kann auf den unterschiedlichen Ebenen jeweils die benötigte Funktionalität realisiert werden. Jeder Anwender ist dann in der Lage seine Zielvorstellung entsprechend zeitnah umzusetzen.

Zielsetzung:

- Bereitstellung von Expertenmodulen und unterschiedlicher Benutzerebenen zur flexiblen Anpassung auf die vorliegenden Synthesen- Anforderungen.

Skalierbarkeit der Synthesevorgänge

Im Altsystem liegen alle Syntheseabläufe als statische Textdateien vor. Sollen unterschiedliche Mengen, Konzentrationen, Temperaturen oder Dosiergeschwindigkeiten zur Anwendung kommen, müssen diese per Hand editiert werden. Da jedoch reine Zahlenwerte in den Textdateien ohne jegliche Bedeutung vorliegen, bedeutet dies eine manuelle Fleißarbeit, die auch entsprechend fehleranfällig ist. Eine umsetzbare Herangehensweise wäre die Verwendung von Platzhaltervariablen die unterschiedliche funktionale Bedeutungen abbilden können.

Zielsetzung:

- Skalierbarkeit der Synthesevorgänge durch Bereitstellung von Platzhalter-Konstrukten.

Gerade bei neuen Synthesen liegen keinerlei Erfahrungen zur Automatisierung vor. Somit müssen alle relevanten Syntheseparameter iterativ durchgetestet werden, bis eine Kombination vorliegt, bei der die Ausbeute sich mit den angestrebten Erwartungswerten deckt. Mit Hilfe zuvor festgelegter

Wertebereiche könnten auch automatisiert Parametervariationen in den festgelegten Intervallen zu eigenständigen Syntheseprogrammen erstellt werden.

Zielsetzung:

- Skalierbarkeit der Synthesevorgänge durch die Automatisierung von Prozess- Parametern.

Einer der Hauptnachteile der Applikationsumgebung des Altsystems ist die Begrenzung in der Schleifeniteration. Vor allem bei Synthesen mit identischer Mehrfachkopplung wären flexiblere Schleifen-Konstrukte wünschenswert. Bei einem entsprechenden programmatischen Aufbau ist die Anzahl der Schleifenebenen nur über die Speichertiefe begrenzt und sollte somit für alle denkbaren Synthesen bei weitem ausreichen. Damit eine noch weiterführende Flexibilität erreicht werden kann, wären zusätzliche Schalter / Flags und Bedingungen eine sinnvolle Ergänzung der Programmierkonstrukte.

Zielsetzung:

- Skalierbarkeit der Synthesevorgänge durch die Flexibilisierung der Schleifenebenen und Bedingungs-Konstrukten.

Kalibrierungs-, Wartungs-, Test- Durchläufe

Im J-KEM-System sind die zugeführten Volumina jeweils nur über die Öffnungszeiten der Ventile vorgebar. Hier sollte eine Rückführung auf die aktuelle Werte der Umgebungsparameter: Versorgungsdruck, Temperatur usw. ermöglicht werden. Hierfür sind geeignete Korrekturkurven aufzunehmen und zyklisch zu verifizieren, um eine Volumina-Kontrolle der unterschiedlichen Lösungsmittel zu ermöglichen. Damit die Synthesen sich langfristig reproduzieren lassen, sind Möglichkeiten zur Mengenkontrolle an verschiedenen Stellen im System eine sinnvolle Ergänzung zur Kalibrierung und Funktionskontrolle. Durch die große Abhängigkeit der Syntheseeffizienz von der Temperaturführung, bietet sich die Bereitstellung unterschiedlicher Temperaturprofile mit den möglichen Zeitverläufe an. Gerade die Informationen zur den Regelparameter sind hierbei von entscheidender Bedeutung, da es durch Überschwinger zu ungünstigen Temperierungs- Situationen kommen kann. Aufgrund der unterschiedlichen Innendurchmesser der Teflon- Schläuche, verändern sich auch die möglichen Dosisierungsgeschwindigkeiten. Gerade bei erhöhter Außentemperatur ist auch beim Aufziehen der Spritzen die Siededruck der Lösungsmittel eine nicht zu vernachlässigender Grenzwert. Solche Informationen sollten aktiv in der aktuelle Synthese automatisiert einfließen.

Zielsetzung:

- Bereitstellung von Kalibrierungsläufen

In den Altsystemen sind Reparaturen meist nur von den Experten durchführbar. In vielen Fällen werden trotzdem auftretenden Fehler und Unzulänglichkeiten erst bei der nächsten durchgeführten Synthese offensichtlich. Um hierbei einen verlässlichen Ablauf mit überprüfbaren Resultaten zu erreichen, sollten entsprechende Testmodule bereitgestellt werden. Diese Module erleichtern dann eine Systeminbetriebnahme und Wartungsarbeiten erheblich, da hiermit Systembereiche gezielt auf ihre jeweilige Funktion und Wiederherstellung getestet werden können:

- Befüllungslauf
- Blasen-/ gasfreie Schlauchbefüllung
- "Ausheizen" des Reaktionsgefäß
- Stickstoffspülung des Gesamtsystems
- Abschnittsweise Dichtigkeitskontrolle
- automatisierte Spülprozesse von Segmenten und kompletten Lines

Zielsetzung:

- Bereitstellung von Wartungsmodulen.

Um den Anwendern die jeweiligen Funktionalitäten des Systems jederzeit überprüfbar zur Verfügung zu stellen, wäre es hilfreich verschiedene Teilprogramme als eigenständige Einheiten auszukoppeln:

- Waschprogramme des Resins
- Transfer der Zuckerbaustein- Volumina in die Gefäße
- Rückspülen der Zuckerbausteinen aus den Leitungen
- Positionierungsgenauigkeit des Fraktionssammlers bei Mehrfachpositionierung

Zielsetzung:

- Bereitstellung von Testläufen und Funktionskontrollen.

Verwaltung & Organisation von Syntheseabläufen

Die Einarbeitung für neue User gestaltet sich sehr unterschiedlich, in Abhängigkeit der zeitlichen Auslastung und Motivation der Experten. Hier wäre es hilfreich eine Möglichkeit zum autodidaktischen Einarbeiten innerhalb des Systems bereitzuhalten, welches durch die Anwender kontinuierlich ergänzt und erweitert werden kann.

Die Hauptpunkte einer solchen Funktionalität sind nachfolgend aufgelistet:

- Dokumentation innerhalb der Module
- Hinweise auf Besonderheiten und möglichen Erweiterungen
- EDV-technisch verwendbar
- Links zu Publikationen und zum elektronischen Laborbuch, usw.

Zielsetzung:

- Organisation von Syntheseabläufen durch aussagekräftige Namen-/Ordnerstruktur und integriertes Versionsverwaltungsmanagement.

Durch die zuvor aufgezeigten Punkte lassen sich bereits eine Vielzahl von benötigten Dateien als Zubringer für das Automatisierungssystem ableiten. Diese werden oft auf externen USB- Sticks transferiert. Damit kann keine effektive Versionskontrolle oder ein Nachverfolgungssystem für Änderungen stattfinden. Wenn Änderungen auffallen, ist nicht nachvollziehbar durch wen diese vorgenommen wurden. Aus welchem Grund und mit welcher Zielsetzung ist im Nachgang nicht mehr nachvollziehbar. Die Realisation unterschiedlicher Benutzerebenen wäre ein effiziente Ergänzung des Dateiverwaltungssystems. Vergleichbar zu einem Versionsverwaltungssystem, sollten die für das System notwendigen Dateien über die Applikation im Zugriff stehen.

Zielsetzung:

- Verwaltung von Syntheseabläufen durch Zugriff auf alle relevanten Systemdateien über die Applikationsebene.

Verwendung von Standardkomponenten

In den Altsystemen wurden mehrere Spezialkomponenten verbaut. Hierdurch ist bei Fehlern in den Abläufen kein einfaches Austauschen von Komponenten möglich, um möglichst effizient die Fehlerursache eingrenzen und ermitteln zu können. Da auch keinerlei Entwicklungstools bereitgestellt sind, ist keine Möglichkeit vorhanden, über ein Mitprotokollieren oder Debuggen hier die jeweiligen Zustände zu verifizieren. Auch bei lokal auftretenden Probleme oder bei Reparaturen können die Komponenten nicht direkt ausgewechselt werden, da diese von der Steckerbelegung und Firmware-Versionen her nicht kompatibel sind. Nachfolgend sind die existierenden Kritikpunkte an den aktuellen Automatenkomponenten zusammengefasst:

- keine Erweiterungsmöglichkeit der Hardware
- komplexe, undokumentierte Schrittketten müssen eingehalten werden

- keine Parallelisierung / Kaskadierung von Prozessen möglich
- keine sinnvollen Halte-/Stoppmöglichkeiten außer harter Syntheseabbruch
- keine Möglichkeit plausible Systemzustände als Synthesestopp/ -pause zu definieren oder nach einem Abbruch gezielt anzufahren, um den Syntheseablauf neu aufzusetzen
- keine Kontroll-/ Messkanäle zur situationsbedingten Prozessanpassung
- keine Debugger-/ Simulationsmöglichkeit der Synthese

Zielsetzung:

- Eliminierung des Spezialkontrollers / Spezialhardware durch die Verwendung von Standardmodulen

Durch die unterschiedlichen Bauformen sind die verschiedenen Druckversorgungseinheiten untereinander nicht kompatibel. Somit besteht keine Möglichkeit des direkten Vergleiches der Systeme untereinander. Da diese Druckversorgungseinheiten als Sonderbaugruppe extern gefertigt und dem Institut als Gesamtbaugruppe ausgeliefert wurden, sind keine detaillierten Informationen über den Aufbau und die verwendeten Komponenten vorhanden. Gerade für die geplanten Erweiterungen im Sinne der Parallelisierung und Kaskadierung sind die Informationen zu maximal möglichen Volumenströmen und verwendeten Materialien notwendig. Somit müssen diese Komponenten vollständig überarbeitet und angepasst werden.

Zielsetzung:

- Test und Verwendung standardisierter Druckversorgungseinheiten.

Der Syntheseautomat für die Glykansynthese steht immer noch in seinen Ausbaumöglichkeiten am Anfang. Dieser Zustand ist vergleichbar zu den Peptid-Synthesizern in den 70iger Jahren. Daraus resultiert, dass sowohl ständige Änderungen und Anpassungen sowohl hardwareseitig wie auch in der Software vorgenommen werden müssen. Um hierbei eine möglichst große Flexibilität den Anwendern bereitzustellen und diese in ihren Wünschen und Anregungen kaum einzuschränken sollen nachfolgende Anpassungen am Systemdesign / Komponenten vorgenommen werden:

- Profilrahmensystem
- Türen
- Abzug
- Kamin
- Abfallbehälter
- Schubladen für Lösungsmittel
- Flaschenhalter (Aktivatoren & Reaktanten)

- Modularisierung der Hardware
- Verwendung von 19" Gerätekomponenten
- Standardisierte Einschübe und Überlaufwannen
- Einheitliche Probensammelgefäße

Zielsetzung:

- Mechanische und konstruktive Standardisierung

Definition der Anforderung an das Synthesystem

Da bisher keinerlei umfassende Pflichten-/ Lastendefinition für des chemische Automatisierungssystem vorhanden ist, wird nachfolgend eine Analyse der existierenden Systeme aus den Bereichen der Peptid- und DNA- Synthesizer vorgenommen. Unter der Berücksichtigung der bei manuellen Synthesedurchführung im Glykanbereich aufgezeigten Zusatzinformationen aus den Supporting - Informationen, ergeben sich nachfolgende allgemeingültige Anforderungen an den Oligosaccharid-Synthesizer:

- flexibel einstellbare/verfahrbare und genau geregelte Reaktionstemperatur
- gute Durchmischung der Reaktionssubstrate
- effiziente Abscheidung in den Abfallbehälter (Spülvorgänge)
- Produktsammlung in frei definierbaren Probengefäßen/Positionen
- Vermeidung von Lufteintrag (Schutzgasatmosphäre)
- Vermeidung und Entfernung von Feuchtigkeit im System
- Dosierter / reproduzierbarer Substanzeintrag ins Reaktionsgefäß
- Schnelles Spülen des Reaktorgefäßes
- Drucküberwachung des Schutzgases im gesamten System
- Substrat- Reaktanten Zuführung ohne Kontamination oder Nebenreaktion
- direktes Spülen individueller Leitungsabschnitte
- Austauschbarkeit von Komponenten
- sichere und nachvollziehbare Verwaltung der Reaktionsprodukte

2 Ausgangssituation & Detailanalyse des existierenden Systems

2.1 Kurze Marktübersicht für Peptid- und Oligonukleotid-Syntheseautomaten

Die Festphasensynthese von Peptiden wurde bereits in den 1960er Jahren grundlegend automatisiert.^[78] Heute hat der Chemiker die Auswahl unter einer Vielzahl unterschiedlicher kommerzieller Synthesegeräte, die als Vorbild für die Entwicklung eines Oligosaccharid-Syntheseautomaten dienen können. Um von den Erkenntnissen der automatisierten Peptid- und Oligonukleotid-Synthese zu profitieren, analysierten wir die vorhandenen Konzepte und erarbeiteten einen Überblick über die vorhandenen Möglichkeiten der kommerziell verfügbaren Systeme.

Technisch unterscheiden sich die Geräte nur marginal voneinander, wobei je nach Preisklasse jedoch z.B. Mikrowellenunterstützung, Parallelsynthese und unterschiedliche Pumpenzulieferung erhältlich sind. Alle kommerziellen Peptid- und Oligonukleotid-Syntheseautomaten werden als eigenständige Gerätevarianten mit zentraler Steuerung (vgl. Seite 55) vertrieben. Jegliche Veränderungen und Erweiterungen sind damit Spezialentwicklungen auf Kundenwunsch, was individuelle Geräteanpassungen teuer, zeitaufwendig und daher fast unmöglich macht. Dies gilt vor allem für kurzfristig auszutestende Ideen in der aktuellen wissenschaftlichen Fragestellungen im Hinblick auf Syntheseführung oder Kontrollvarianten. Die Bediensoftware aller Geräte ist jeweils als eigenständige Lösung umgesetzt, die keine weitreichenden programmiertechnischen Schnittstellen für Softwareerweiterungen ermöglicht. Die meisten kommerziellen Anbieter legen keine Softwareschnittstelle (API) offen. Sogar bei den modularen Systemplattformen wird das Einbinden neuer Komponenten nicht oder nur mit hohem finanziellen Aufwand seitens des Anwenders ermöglicht.

Fazit: Kommerziell verfügbare Gerätekonzepte und Softwareplattformen erlauben es nicht, die angestrebte Flexibilisierung, Erweiterbarkeit und benutzerfreundliche Bedienphilosophie zeitnah umzusetzen.

Die Rotationsventile haben die Aufgabe verschiedene Schlauchabschnitte miteinander zu verbinden. Sie erlauben die Entnahme exakt definierter Volumina aus den Vorratsgefäßen, die Zwischenpufferung in den Schleifen und die Zuführung der Substanzen z. B. in das Reaktionsgefäß durch Umschalten auf den entsprechenden Ausgang. Desweiteren ist für lokale Spülprozesse an jedem Rotationsventil ein direkter Ausgang zum Abfallsammelbehälter vorgehalten. Die in den Schlauchabschnitten vorhandenen Flüssigkeiten können mit Schutzgas beaufschlagt werden, was entweder zum Zurückdrücken oder zum Bereitstellen einer Gasblase zwischen Reaktanz und Transportlösungsmittel verwendet werden kann.

Die Temperatur der Glykosylierungsreaktion wird mit Hilfe eines elektrischen Temperiergerätes eingestellt. Während der Synthese können Proben der Reaktionslösung aus dem Reaktionsgefäß entnommen und in den Sammelgefäßen des Fraktionssammlers aufgefangen werden.

Für Wasch- und Spülprozesse ist die Genauigkeit der Spritzenzulieferung nicht notwendig. Somit sind für diese Prozesse schnellere, über Spulventile steuerbare, druckgeregelte Flüssigkeitszuführungen vorgehalten.

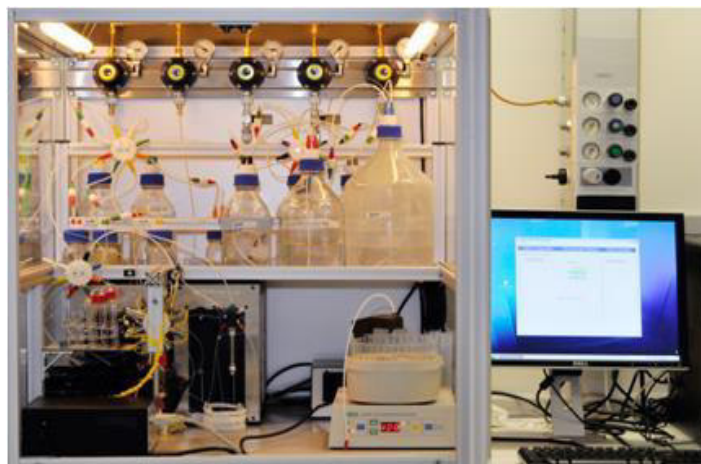


Abbildung 25: J-KEM System Oligosaccharid-Syntheseautomat.^[160]

2.2.2 Analyse der Gerätekommunikation

Das J-KEM-System kommuniziert über eine zentrale Steuereinheit (Controller), welche Befehle sequenziell an die entsprechenden Gerätekomponenten übermittelt. Die Anwenderschnittstelle ist ein Visual Basic (VB) Programm (s. a. Abbildung 27, Seite 56) auf einem Rechner, in das der Benutzer die gewünschte Befehlsliste eingibt.

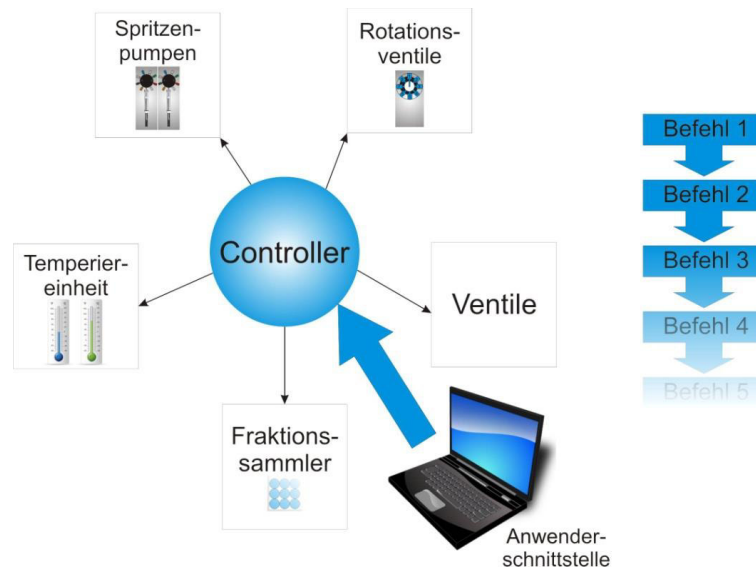


Abbildung 26: Gerätesteuerung des J-KEM Oligosaccharid-Syntheseautomaten mit dem Controller als zentralem Element und steuerungsbezogener Abarbeitung

Die einzelnen Hauptgerätekomponenten sind hochspezialisierte, kommerzielle Geräte. Die Temperatureinheit wurde von Julabo zugekauft, während der Fraktions-sammler, die Spritzenpumpen und die Rotationsventile von Kloehn erworben wurden. Diese Komponenten müssen jeweils über unterschiedliche Schnittstellen und Softwareprotokolle angesprochen werden, wobei für die Geräteprogrammierung keine Standardsoftware-Lösung zur Schnittstellenkommunikation existiert. Die Julabo Temperatureinheit wie auch der Fraktions-sammler und die Kloehn – Pumpen besitzen jedoch eine eigenständige Benutzersoftware zur individuellen Einzelgerätesteuerung.

Der Kommunikationsfluss des Controllers findet stets in einer Richtung statt (Abbildung 26) und wirkt lediglich steuernd auf die verschiedenen Gerätekomponenten. Es werden also keine funktionalen Geräteinformationen an die Steuereinheit zurückgeführt, womit eine Statusüberwachung der Systemkomponenten nicht möglich ist. Auch existieren keine Schnittstellen zwischen den Gerätekomponenten, die logische Systemabhängigkeiten miteinander verknüpfen könnten. Die Umsetzung von Bedingungsstrukturen in der Programmierung der Synthesesteuerung oder ereignisorientierte Abarbeitung von Befehlen ist mit diesem Prinzip der Gerätesteuerung nicht möglich.

Das Visual Basic Programm für den Controller übergibt eine reine Befehlsliste (vgl. Abbildung 28, Seite 57). Höhere Programmierstrukturen, wie verschachtelte Schleifen, Bedingungen oder höhere Protokollstrukturen werden nicht unterstützt. Die Zusammenstellung der Kommandolisten erfolgt per Handeingabe oder durch Kopieren aus bestehenden Listen. Dabei fehlt eine Plausibilitätskontrolle für die eingegebenen Kommandos, entsprechend leicht kann es zu Schreibfehlern, Zuweisungsfehlern von Ports, Eingabefehlern in der Ventilreihenfolge oder bei der Voluminavorgabe kommen.

Die Bedienung des J-KEM Syntheseautomaten erfordert eine genaue Kenntnis über den Aufbau des Gerätes, wie die Zuordnung der Ventilnummern zu den entsprechenden Chemikalien oder den Ventilnummern zur Schaltung der Transportwege. Eine interaktive Benutzerführung und informative Hilfedatei fehlen. Dies führt zu einer vergleichsweise langen Einarbeitungszeit in die Benutzung des Systems und engt den Anwenderkreis stark ein.

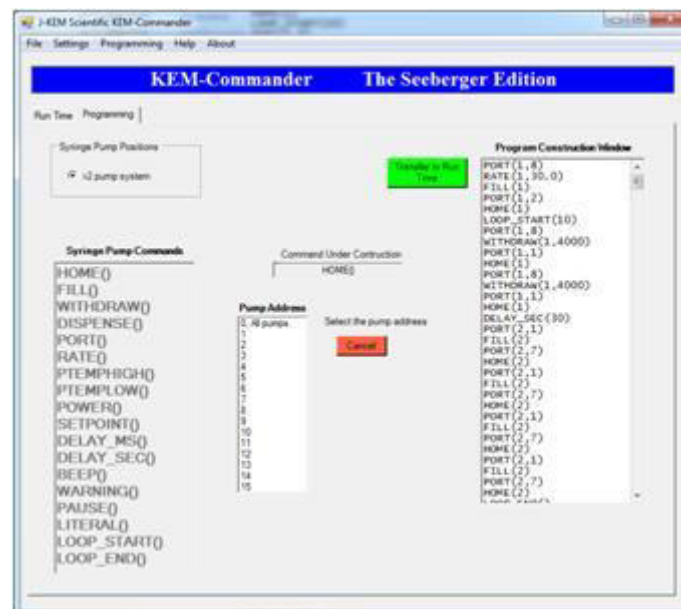


Abbildung 27: J-KEM Benutzeroberfläche

Die beiden vorhandenen J-KEM Synthesizer unterscheiden sich sowohl Soft- als auch Hardware-technisch voneinander. Über die an diesen Geräten durchgeführten Synthesen liegen unabhängige und nicht harmonisierte Textprotokolle und Support- Informationen vor. Syntheseprotokolle sind damit unter den J-KEM-Geräten nicht austauschbar. Eine Vereinheitlichung der Informationen würde zu einer geräteübergreifenden Reproduzierbarkeit der Synthesen führen und wäre ein wesentlicher Schritt hin zur universellen, standardisierten Austauschbarkeit der Syntheseprotokolle.


```

RATE (2,10)
DISPENSE (2,1000)
DELAY_SEC (2)
PORT (2,6)      activator 1 into loop
PORT (6,2)
RATE (2,1)
WITHDRAW (2,1000)
DELAY_SEC (2)
PTEMPLOW (-19)
VALVE (1,ON)    open outlet for bubbling
VALVE (16,OFF)
VALVE (14,ON)
DELAY_MS (100)
PORT (2,6)      350 uL of activator 1 into rv
PORT (6,6)
RATE (2,1)
DISPENSE (2,500)
DELAY_SEC (2)
PORT (2,6)      empty rv line
PORT (6,6)
RATE (2,1)
WITHDRAW (2,500)
DELAY_SEC (2)
PORT (2,6)      flush loop with DCE
PORT (6,7)
RATE (2,10)
HOME (2)
DELAY_SEC (2)
PORT (2,2)      Resting positions
PORT (6,1)
DELAY_SEC (2)
PORT (2,1)      fill organic sy with DCE
RATE (2,10)
WITHDRAW (2,4500)
DELAY_SEC (2)
PORT (2,6)      activator 1 to storage
PORT (6,8)
RATE (2,1)
WITHDRAW (2,25) Ar bubble to separate activator 1 and DCE
DELAY_SEC (2)
PORT (2,6)
PORT (6,2)
RATE (2,3)
DISPENSE (2,225)
DELAY_SEC (2)
PORT (2,6)      empty sy and flush loop
PORT (6,7)
RATE (2,10)
HOME (2)
DELAY_SEC (2)
PORT (2,2)      Resting positions
PORT (6,1)
DELAY_SEC (60) bubbling rct for 1 min
VALVE (14,OFF) stop bubbling and solution to fraction collector
VALVE (1,OFF)
VALVE (15,ON)
VALVE (16,ON)
VALVE (11,ON)

```

Abbildung 28: Beispiel eines J-KEM Programmtextes mit farblicher Codierung der Befehle, Kurzbeschreibung zur besseren Lesbarkeit und Vergleichbarkeit

Da die Steuereinheit des J-KEM Systems eine Spezialentwicklung ist, obliegt die jeweilige Optimierung und Realisierung zusätzlicher Hardware dem externen Zulieferer. Dieser nimmt beim Auftreten von lokal nicht lösbaren Problemen oder Wünschen zielorientiert Anpassungen vor. Die Bereitstellung neuer Kommandos für die Steuereinheit oder die Möglichkeit mehrere Kommandos zu eigenen Befehlen zusammen zu fassen, sowie das Einbinden alternativer Gerätekomponenten in den Synthesizer, stellt somit eine kostenpflichtige Dienstleistung des externen Anbieters dar. Aufgrund des Aufwands, den jede Veränderung an den Systemen bedeutet, erfolgte noch keine hinreichend qualitative Überprüfung der durchgeführten Optimierungen, um daraus alternative Lösungskonzepte zu erarbeiten.



Abbildung 29: Rückseite der J-KEM Controllereinheit mit aufgestecktem Schnittstellenadapter zur Protokollierung der gerätespezifische Befehls- und Ansteuerungsprozeduren.

Die Weiterentwicklung des Oligosaccharid-Syntheseautomaten und die Umsetzung der geforderten Flexibilität, bedürfen deshalb einer herstellerunabhängigen, eigenständigen Geräteentwicklung, bei der das gesamte Projektmanagement, der Gerätebau und die Softwareerstellung in kompletter Eigenverantwortung verbleiben.

Der erste Schritt zur Umsetzung des neuen, flexiblen Gerätekonzeptes ist deshalb die Entschlüsselung der Gerätekommunikation zwischen Controller und den einzelnen Gerätekomponenten. Aufgrund fehlender Dokumentationen zur Programmierung der Mikrocontrollersteuerung und zum Aufbau der Hardware (Schaltpläne), wurde mit Hilfe selbstentwickelter Adapterstecker (Abbildung 30, Seite 58) ein Zwischenabgriff und Protokollierung der notwendigen Informationen ermöglicht.

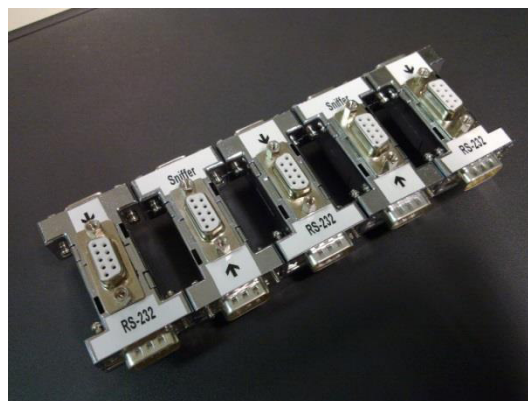
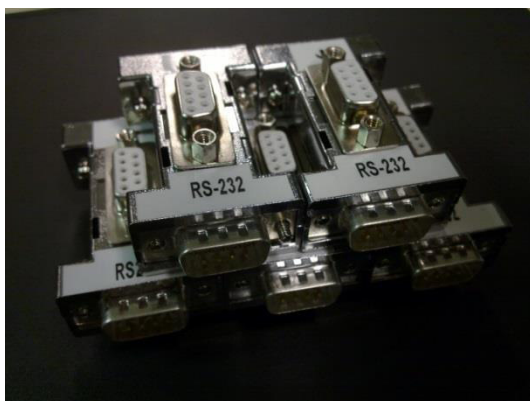


Abbildung 30: Selbst entwickelte Adapterstecker zum Protokollieren der RS232 Kommunikation

Die Hardware wurde durch die Verwendung von Standard D-Sub 9poligen Selbstbaugehäusen möglichst vereinfacht konstruiert. Dies stellt eine erhebliche Minimierung der räumlichen Abmessungen im Gegensatz zu kommerziell verfügbaren Varianten dar, wobei zusätzlich die gesamte Verkabelung innerhalb des Gehäuses untergebracht wurde. Die Adapter ermöglichen ein effizientes und störungsfreies Arbeiten sowohl lesender als auch schreibender Kommunikationszugriffe auf das Bussystem der überwachten Geräte. Hierdurch ist zur frühzeitigen Erkennung störungs- oder

problembehafteter Situationen eine flexible Testsituation aufbaubar. Da keine zusätzliche Elektronik verbaut ist, können die für das verwendete Betriebssystem aktuellen Komponenten (=> Vermeidung der Treiberproblematik) eingesetzt werden. Die reale RS232 Kommunikation findet über Hauptstecker in Pfeilrichtung statt, während der protokollierende Abgriff auf den Nebensteckern erfolgt. Damit immer von außen sichtbar die gleiche Hardwarepositionierung erreichbar ist, wurden die Sniffer mit einer Vorzugsrichtung (Pfeil) erkennbar gekennzeichnet.

Als Anwendungsbeispiel (Abbildung 29, Seite 58) ist die Verwendung an der Kontrollereinheit von J-KEM gezeigt. Wenn mehrere RS232/USB-Einzeladapter gleichzeitig zur Anwendung kommen und an einem Rechner ausgewertet werden, können alternativ USB auf 8x seriell Adapter z. B. LOGILINK AU0033/DELOCK 61860 im Vertrieb bei Reichelt Elektronik zum Einsatz kommen, um den Verkabelungsaufwand im Gegensatz zur Einzeladaption zu reduzieren.

Durch die individuell realisierte Sniffer-Software war es möglich den lokalen Ablaufcode mit den Steuerungsanweisungen nachzubilden. Hierdurch konnten sowohl die aktuellen Funktionen wie auch eine Vielzahl von Sachverhalten, die bisher zu nicht reproduzierbaren Instabilitäten geführt hatten, nachvollzogen werden. Dies war möglich, da das jeweilige Zeitverhalten (Zeitstempel) der Geräte, sowohl lesend wie auch schreibend am RS485 und dem RS232 parallel über alle Schnittstellen protokolliert wurden.

2.2.3 Analyse des Ansteuerungsverhaltens der Lösungsmittel-Ventile

Die messtechnische Auswertung des Ventilsteuerungs-Verhalten ohne die existierende Verkabelung bleibend zu beeinflussen, erfolgte mit Hilfe eigens gefertigter Adapterkabel, die sich temporär in die vorhandene Verkabelung einklinken lassen. Hierdurch können zwischenzeitlich Tests zum Zeitverhalten und zur Ansteuerung durchgeführt werden. Ein vollwertiger Rückbau auf den Urzustand ist jederzeit möglich, um alle Effekte reproduzierbar zuzuordnen. Zur Vermeidung elektrischer Beeinflussung der Ventilansteuerung wurden die Informationsabgriffe über Optokoppler realisiert.

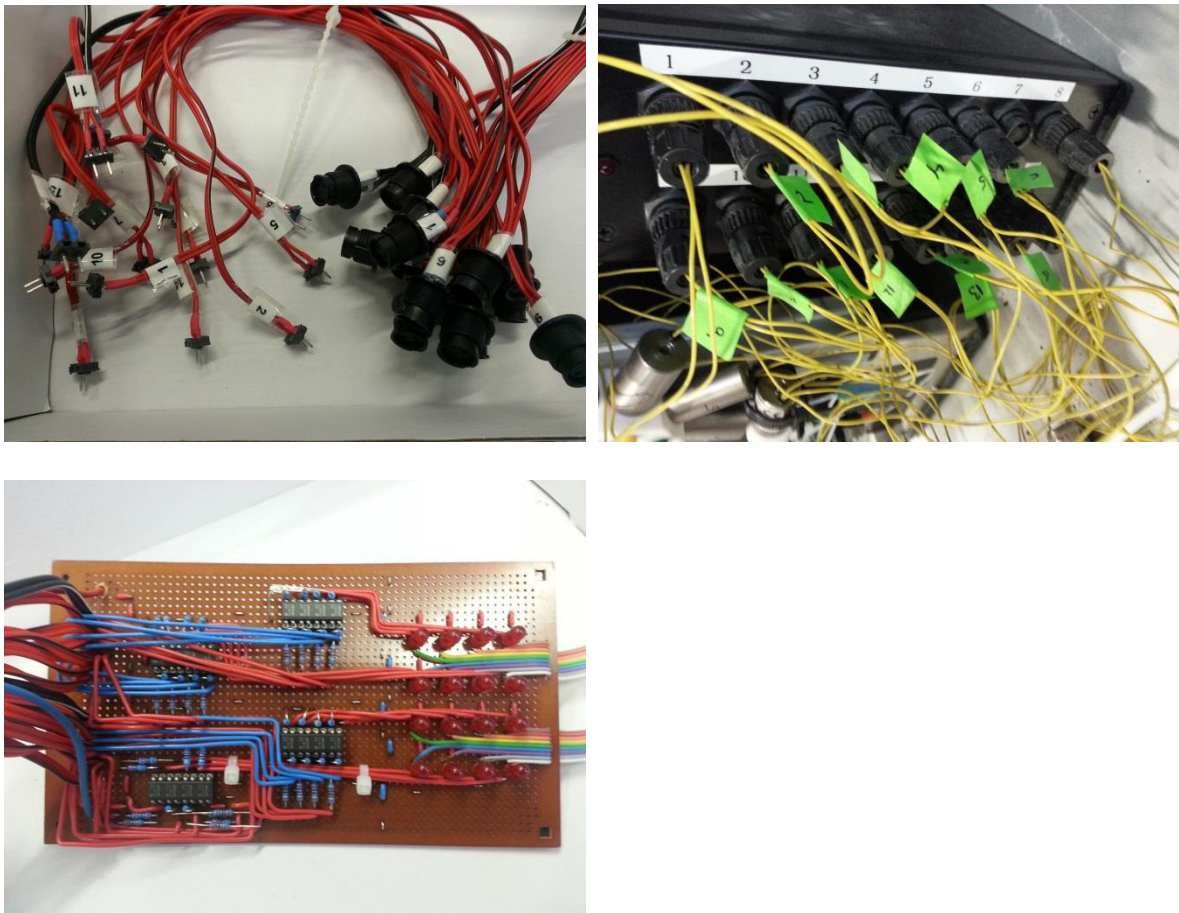


Abbildung 31: Adapterkabel (o. li.) zum Informationsabgriff der Ventilansteuerung (o. re.) über selbst konzipierte und aufgebaute Optokopplerschaltung/Adapterboard (u.)

3 Konzeptionierung der neuen Gerätegeneration

3.1 Grundkonzept des Projektaufbaus

Eine umfangreiche Analyse der am Institut vorhandenen Oligosaccharid-Syntheseautomaten, die mit Hilfe der Controller der Firma J-KEM System von Dr. Lenz Kröck^[161,160] gebaut wurden und die Erfahrungen in der Nutzung dieses Gerätes bildete die Grundlage für die neue Konzeption. Da in der Vergangenheit lokal aufgetretene Probleme jeweils punktuell gelöst und gerätetechnische Änderungen nicht mit chemischer Effizienz verglichen wurden, erfolgte nun eine umfassende Neuentwicklung. Prinzipiell standen für diese Geräteneukonzeption zwei grundlegend verschiedene Herangehensweisen zur Wahl: eine Projektentwicklung nach dem **Top-Down** oder dem **Bottom-up** Prinzip.

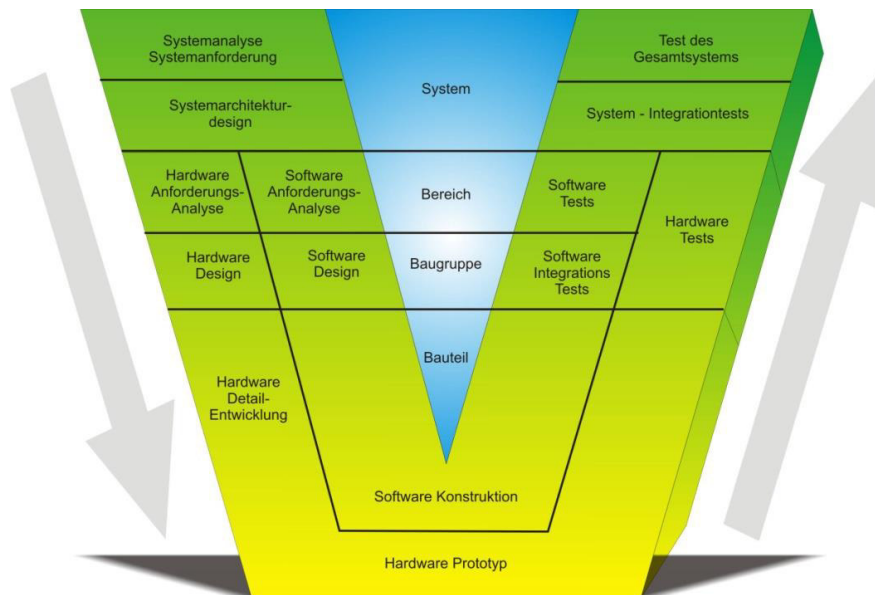


Abbildung 32: Konzeption und Umsetzung der Geräteweiterentwicklung erfolgt im Bottom-up Prozess. Die für die Geräteentwicklung notwendigen Projektschritte sind als V-Modell dargestellt.

Bei der **Bottom-up** Entwicklung wird auf Basis einer fundamental ausgetesteten Grundlage, durch immer abstraktere und allgemeingültigere Zwischenebenen, iterativ die Anwenderplattform generiert. Somit wächst das Konzept von unten nach oben. Der Vorteil dieser Vorgehensweise ist die Stabilität und die Wiederverwendbarkeit der einzelnen Schichten, da bei der grundlegenden Bereitstellung nicht die Zielsetzung, sondern die technischen Möglichkeiten die Vorgabe sind.

Im Extremfall der reinen **Top-Down**-Entwicklung wird ein anwenderdefiniertes Konzept zielgerichtet umgesetzt. Der Vorteil hier ist die sehr schlanke und zielgerichtete Bereitstellung von Funktionalitäten, womit eine zeitliche Optimierung der Projektentwicklung möglich ist. Dabei besteht die Notwendigkeit einer eindeutigen Spezifikation der Anforderungen in einem Lastenheft im Vorfeld. Dabei ist weitreichendes Erfahrungswissen erforderlich, welche Funktionalitäten zwingend

notwendig sind, um mit dem Endprodukt effizient arbeiten zu können. Ein enger Austausch mit dem Anwender ist in der Erstellung des Pflichtenheftes unabdingbar.

Der **Bottom-up** Ansatz ist ideal für umfangreiche Grundlagenforschung, da hierbei maximale Flexibilität besteht und im Bedarfsfall implementiert werden kann. Bei der Hardwareauswahl müssen Kostenüberlegungen in die Spezifikation eingehen, da sonst nur multifunktionale und somit die teuersten Geräte und Softwaretools zum Einsatz kämen. Grundlegende Hardwarespezifikationen im Synthesizer-Projekt sind Exaktheit der Volumendosierung, Anzahl der Zuführungskanäle, maximaler Temperaturhub und Temperaturexaktheit. Diese Vorgaben waren von Seiten der Chemiker noch nicht eindeutig definiert und konnten nicht exakt eingegrenzt werden. Die Projektumsetzung basierte auf einem **Bottom-up** Prozess, wobei eine **Top-Down** Entwicklung berücksichtigt wurde, um innerhalb der finanziellen Grenzen zu bleiben und die Kosten/Nutzen-Effizienz zu gewährleisten.

Im Gesamtkonzept des neuen Synthesizers bildet die unterste Ebene der funktionalen Gerätesteuerung (vgl. Abbildung 66, Seite 87) die Grundlage und damit die erste Abstraktionsebene der zu entwickelnden Systemsoftware- und Hardwarekonstruktion. Diese für die Stabilität der Prozessführung entscheidende Entwicklungsstufe ist für die Kommunikation und Interaktion zwischen den einzelnen Gerätekomponenten verantwortlich. Die Erfahrungen mit dem vorhandenen Synthesizer zeigten, dass die Gerätesteuerung mittels Spezial-Controller von einem externen Dienstleister (vgl. Abbildung 26, Seite 55) das begrenzende Objekt in der Systementwicklung ist. Bei der Neukonzeption wurde auch das Konzept der zentralen Gerätesteuerung selbst, wie sie in aktuellen kommerziellen Geräten üblich sind, in Frage gestellt.

Zentrale Gerätesteuerungen sind in der Entwicklungsphase wesentlich kostengünstiger und schneller umzusetzen und werden daher in der Produktion von Endgeräten mit festgelegten Funktionen verwendet. Ändern sich die Geräteanforderungen jedoch nachträglich häufig, so ist dies vielfach mit aufwendigen Neuentwicklungen verbunden. Eine zentrale Steuerung ist deshalb nicht als flexibles, ausbaufähiges Gerätekonzept einsetzbar, sondern nur für kommerzielle und in Serie zu produzierende Synthesizer-Varianten mit klar definiertem Funktionsumfang sinnvoll. Das Konzept des neuen Oligosaccharid-Synthesizers wurde deshalb auf einer dezentralen Gerätesteuerung (vgl. Abbildung 33, Seite 63) aufgebaut. Anstelle der Spezialhardware des J-KEM Controllers wurden handelsübliche PC-I/O-Karten von National Instruments eingesetzt, die zusammen mit den Standard-PC-Schnittstellen die Kommunikation mit den verschiedenen Gerätekomponenten des Synthesizers ermöglichen.

Im Gegensatz zur sequenziellen Befehlsabarbeitung können nun mehrere Befehle gleichzeitig abgearbeitet werden (Multitasking). Auch lassen sich korrespondierende Gerätekomponenten, wie Kloehn Spritzenpumpe und Kloehn Rotationsventile, untereinander direkt verknüpfen, um

hardwaretechnische Bedingungsstrukturen umzusetzen. Durch die Auswertungen von Rückmeldungen der angesprochenen Geräte lassen sich höhere Programmierstrukturen realisieren. Aus der reinen Gerätesteuerung kann sich somit eine geregelte Prozessführung entwickeln. Die Applikationsebene der dezentralen Gerätesteuerung bleibt dabei stets offen für die flexible Integration von Soft- und Hardwareerweiterungen. Diese Möglichkeiten der Gerätekommunikation mussten jedoch zuerst softwaretechnisch durch die Entwicklung von eigenständigen Gerätetreibern erschlossen werden, da diese nicht von den jeweiligen Herstellern zur Verfügung stehen. Dezentrale Gerätesteuerungen sind aufgrund ihrer Komplexität wesentlich zeit- und kostenintensiver in der Projekt- und Entwicklungsphase umzusetzen. Für den Aufbau der neuen Synthesizer-Prototypen boten sie jedoch maximale Flexibilität für wechselnde Geräteanforderungen durch die Anwender.

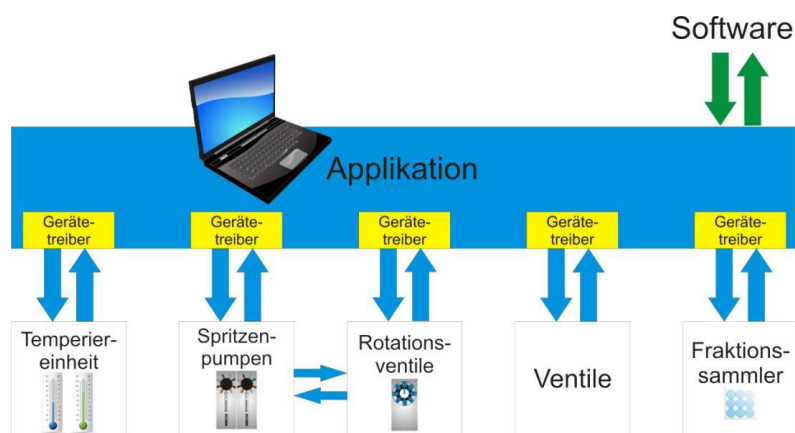


Abbildung 33: Dezentrale Gerätesteuerung

Der größte Teil der Entwicklungsarbeit für den neuen Synthesizer konzentrierte sich deshalb auf die Ebene der Gerätekommunikation, um die Grundvoraussetzungen für einen stabilen Prozessablauf während der Synthese zu legen. Konzeptions- und Umsetzungsfehler auf dieser Ebene schränken die nachfolgenden Funktionsebenen im Synthesizer-Gesamtkonzept (vgl. Abbildung 66) ein.

3.2 Entwicklung des Gesamtsystems

3.2.1 Entwicklungsplattform LabVIEW

Die programmatische Umsetzung des dezentralen Steuerkonzeptes des Synthesizers erforderte die grundsätzliche Entscheidung für die geeignete Software-Entwicklungsplattform. Konsequenterweise sollte das Konzept der Flexibilisierung und Standardisierung von Gerätekomponenten und Funktionalitäten verfolgt werden. Ebenso sollte die Programmumgebung eine universelle Gestaltungsmöglichkeit der Benutzeroberfläche, die auch für den chemischen und biologischen Bereich optimiert werden kann, vorhalten.

Die Entwicklungsumgebung LabVIEW von National Instruments wurde wegen der einfachen Erlernbarkeit, weitreichender Unterstützung der Schnittstellen industriellen Standards, sowie des vorhandenen Zugangs am Max-Planck-Institut zu den Entwicklungs- und Lizenzversion ausgewählt.

LabVIEW ist ein grafisches Programmiersystem, das hauptsächlich für die Mess-, Regel- und Automatisierungstechnik entwickelt wurde und in Industrie und Forschung weit verbreitet ist. Der große Vorteil gegenüber anderen Entwicklungsumgebungen ist die relativ einfache Programmierung paralleler Abläufe (Multithreading), die für das dezentrale Steuerkonzept erforderlich sind. Dabei werden zahlreiche Kommunikationsprotokolle und Schnittstellen unterstützt, was die Einbindung und Ansteuerung der unterschiedlichen Gerätekomponenten des Synthesizers erleichtert. Durch die intuitive Bedienung, der einfachen Handhabung von Unterprogrammen (SubVIs), sowie der umfangreichen Funktionsbibliothek ist es möglich, innerhalb kürzester Zeit lauffähige Programmteile zu erstellen und zu testen. Stark verkürzte Entwicklungszeiten und damit Zeit- und Kosteneinsparung sind bei hoher Qualität möglich. In der geplanten Umsetzung wird es dem Anwender ermöglicht, individuelle Benutzeroberflächen für die Synthesizer-Applikation zu erstellen, die ohne Programmierkenntnisse in die Synthesizer-Applikation integriert werden können. Maximale Flexibilität ist trotz reduzierter Einarbeitungszeit garantiert.

LabVIEW Programme werden als VI (virtuelle Instrumente) bezeichnet und bestehen immer aus zwei Komponenten, der Benutzerschnittstelle (Frontpanel) und dem grafischen Programmiercode (Blockdiagramm). Für beide Komponenten existieren unterschiedliche, umfangreiche Standard-Panelemente zur Erstellung professioneller Anwendungen, die sich individuell anpassen lassen. Interessant im Hinblick auf die zukünftige Benutzeroberfläche ist jedoch die Möglichkeit, auch eigene Elemente zu definieren oder aus anderen Komponenten abzuleiten. Die Eingabe- und Anzeigeelemente des Frontpanels werden per Drag & Drop von der Elemente-Palette auf der grafischen Benutzeroberfläche platziert. Diese Bedien- und Anzeigeterminals haben immer ein korrespondierendes Element im Blockdiagramm.

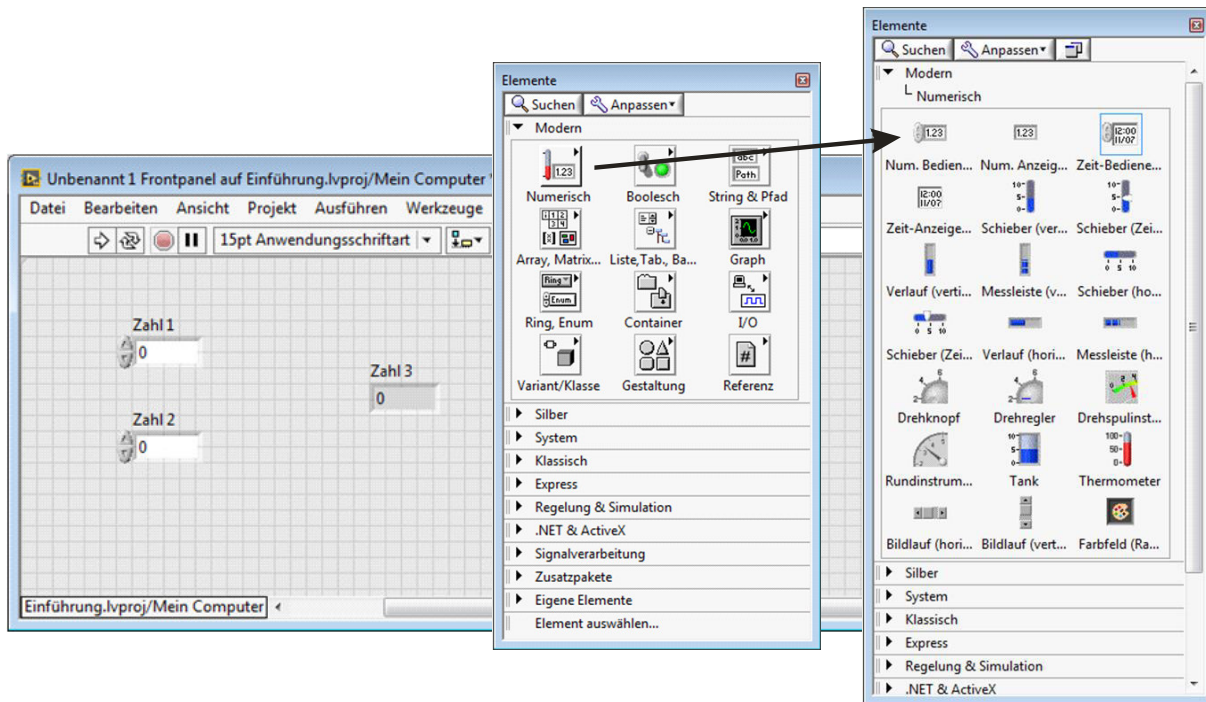


Abbildung 34: Frontpanel eines LabVIEW Virtual Instruments (VI) mit zwei numerischen Bedienelementen und einem numerischen Anzeigeelement, die sich aus einer umfangreichen Elemente-Palette per Drag & Drop auf der Programmfläche positionieren lassen.

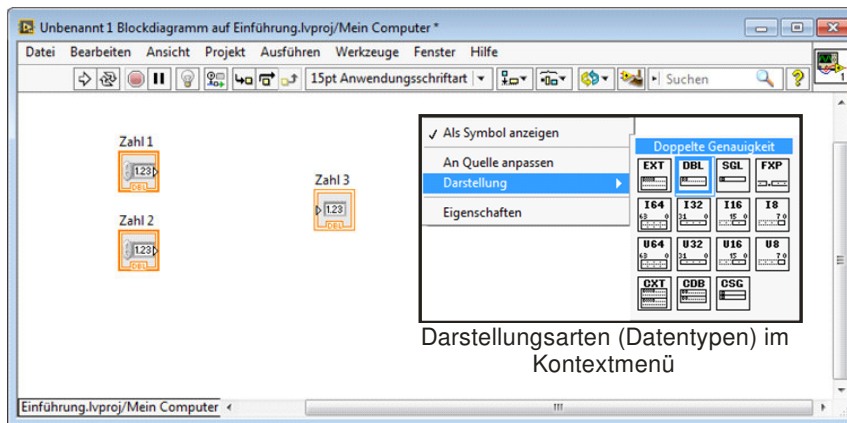


Abbildung 35: Darstellungsarten im Blockdiagramm. Datentypen sind über Rahmenfarbe und Symbol gekennzeichnet.

Hierdurch ist das Blockdiagramm, das den grafischen Programmiercode enthält, direkt mit den Elementen des Frontpanels verknüpft. Eingabelemente lassen sich am ausgefüllten Rahmen und dem mit einem Pfeil symbolisierten Ausgang zur rechten Seite erkennen, Anzeigeelemente am einfachen Rahmen und dem Eingang linksseitig. Die Rahmenfarbe (z. B. Orange) gibt Hinweis auf den Datentyp, ebenso die Bezeichnung DBL für den Datentyp Double (Genauigkeit von 15 Stellen, 64 Bit reell).

Programmiert wird in LabVIEW nach dem Datenfluss-Prinzip^[162]. Der Programmcode besteht aus verschiedensten Elementen, die jeweils für eine bestimmte Funktion stehen. Dazu gehören unter anderem Eingabe- und Anzeigeelemente (Controls und Indikatoren), sowie Konstanten, Funktionen

und Operatoren, Ablaufstrukturen (For- und While-Schleife, Case-Anweisung, ...), sowie Unterprogramme (SubVI's).

Diese Elemente werden an den Anschlussterminals miteinander verbunden und bestimmen so den Datenfluss. Dies wiederum definiert die Programmfunktion.

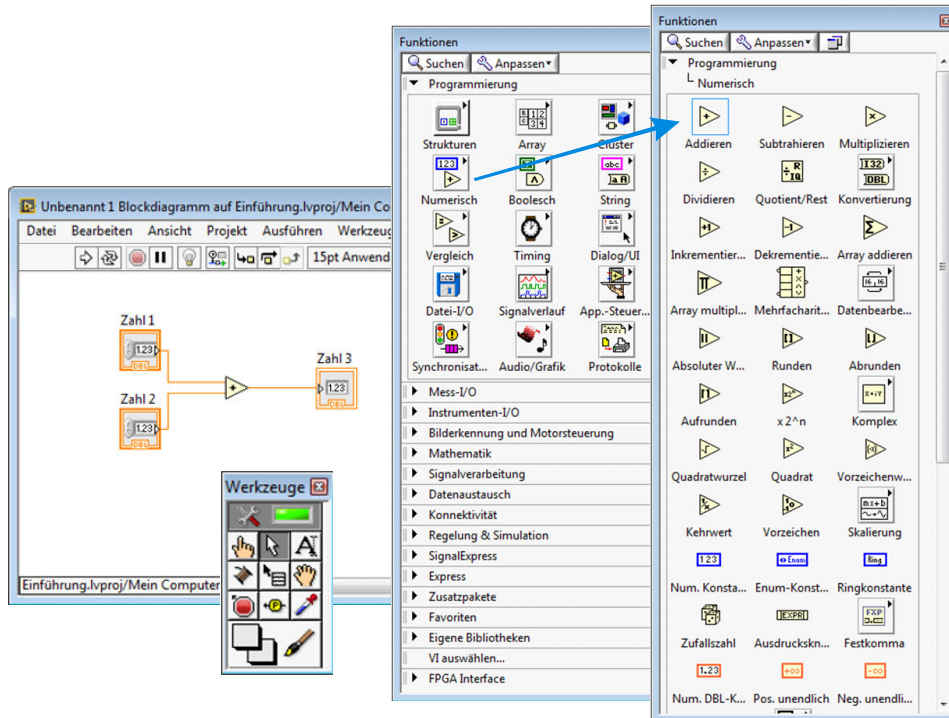


Abbildung 36: Programmcode einer einfachen Addition. Die Funktion wird per Drag & Drop im Blockdiagramm positioniert. Nach dem Datenflussprinzip werden die Eingabelemente mit den Eingängen des Additionsobjekts verbunden und das Ergebnis durch Verbinden mit der Kabeltrommel (Werkzeugpalette) ans Anzeigeobjekt übergeben.

Das Datenfluss-Prinzip lässt sich im Blockdiagramm durch die Highlight-Funktion veranschaulichen. Die Durchführung der Rechenoperation erfolgt nach Programmstart, sobald alle Daten der Eingabelemente zur Verfügung stehen. Im Highlight – Modus werden die lokalen Verbindungswerte im Blockdiagramm auf den jeweiligen Kabeln kurzzeitig angezeigt. Diese Funktion ist hilfreich, um in komplexeren VIs die datenflussorientierten Prozesse nachzuvollziehen wie z. B. bei der Fehlersuche oder zur Einarbeitung in vorhandene Programme.

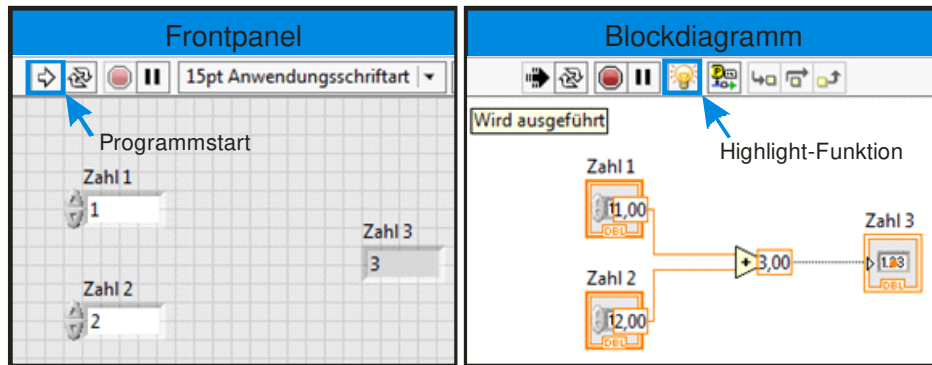


Abbildung 37: Nach Eingabe der Zahlen auf der Benutzeroberfläche kann der Programmcode zur Ausführung gebracht werden. Im Blockdiagramm lässt sich mit der Highlight-Funktion parallel dazu der Datenfluss veranschaulichen.

Mit der Definition des Anschluss terminals wird das LabVIEW VI für andere Programme nutzbar. Bei einfachen Programmen findet dazu normalerweise ein Standard-Muster Verwendung. Dabei werden die in der Programmerstellung genutzten Ein- und Ausgänge dem zugehörigen Anschluss terminal zugewiesen.

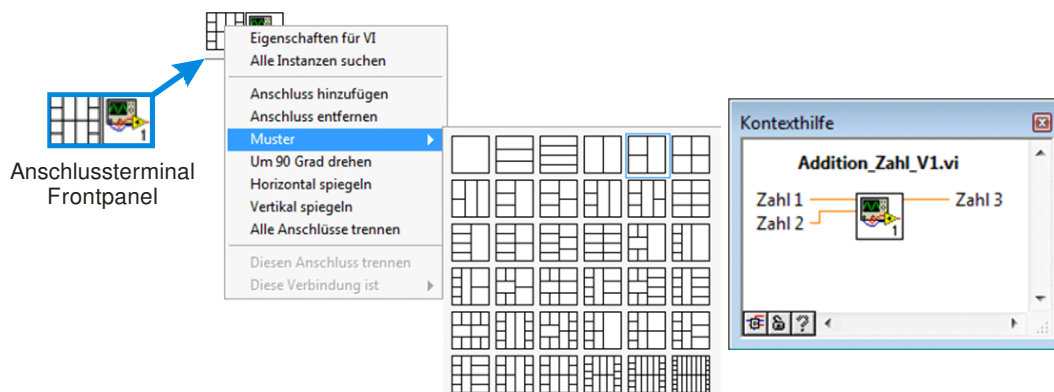


Abbildung 38: Dem VI wird ein Muster für den Anschluss terminal zugewiesen. Aufgrund des Anschluss terminals ist für dieses VI eine Beschreibung in der Kontexthilfe entstanden.

Mit Erstellung des Anschluss terminals entsteht eine Beschreibung der Anschlussbelegung des erstellten VIs in der Kontexthilfe. Das erstellte VI (*Addition_Zahl_V1.vi*) lässt sich jetzt in einem neuen Blockdiagramm nutzen. Dazu werden die Eingänge über das Kontextmenü mit Bedienelementen ausgestattet und mit den entsprechenden Eingabe- und Anzeigeelementen verbunden. Die Auswahl der zur Verfügung stehenden Bedienelemente ist abhängig von der Definition im Ausgangs-VIs.

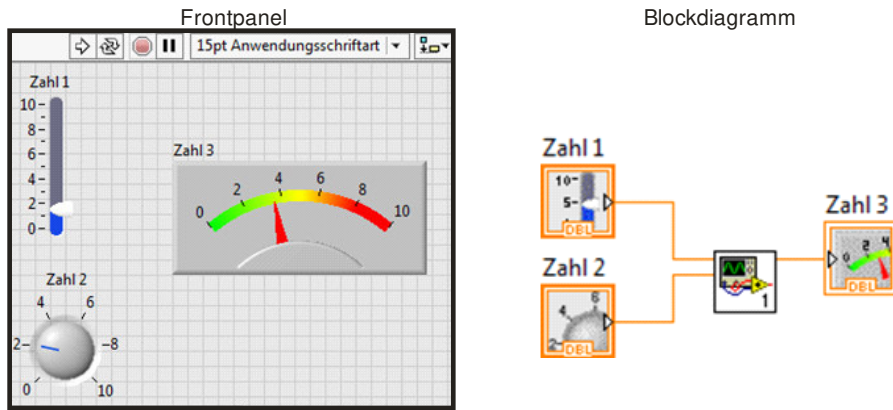


Abbildung 39: Das Beispiel VI wird in einem neuen Blockdiagramm mit beliebig wählbaren Eingabe- und Anzeigeelementen.

Eingangswerte, die bei Erzeugung des VIs vorhanden waren, stehen nun als Standardwerte im neuen Element bereit. Individuelle Anpassungen der Eingabe- und Anzeigeelemente sind durch direkte Selektion und Edition möglich, z. B. um Anzeigebereiche zu skalieren oder das Farbdesign zu verändern. Die Anzeigeelemente lassen sich aber auch durch eigene funktionale Objekte ersetzen, ganz ohne Änderung des Programmcodes.

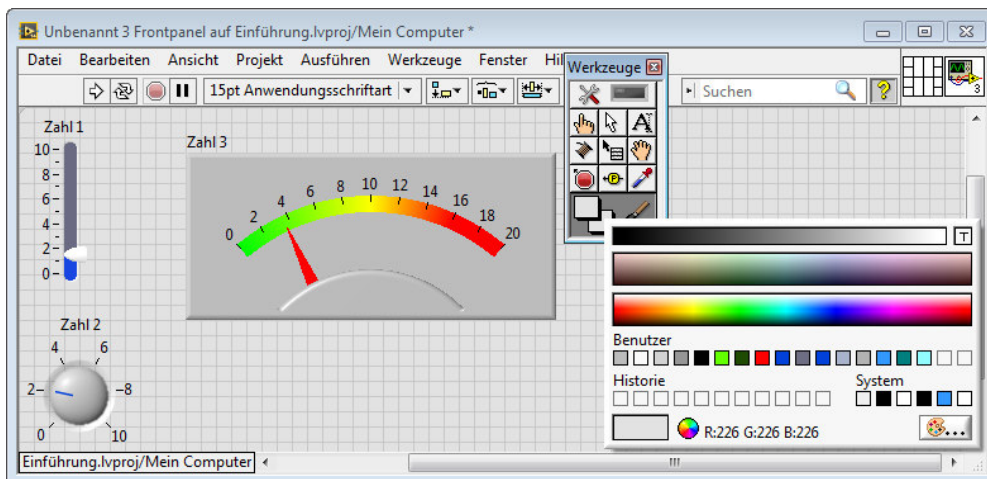


Abbildung 40: Eingabe- und Ausgabeelemente können individuell verändert, ausgetauscht oder durch individuelle Anzeigeelemente ersetzt werden. Der Programmiercode bleibt dabei unverändert.

Das VI wird im Blockdiagramm durch den Standard-VI-Icon dargestellt. Dieses VI ist damit lediglich numerisch eindeutig von anderen VIs mit Standard-Icon unterscheidbar. Zur Übersichtlichkeit und Erhöhung der Lesbarkeit komplexer Programmstrukturen ist jedoch eine eindeutige Programmsymbolik für das geschriebene VI erforderlich. Dazu steht ein eigener Editor zur Verfügung, der durch Doppelklick auf das VI-Icon startet.

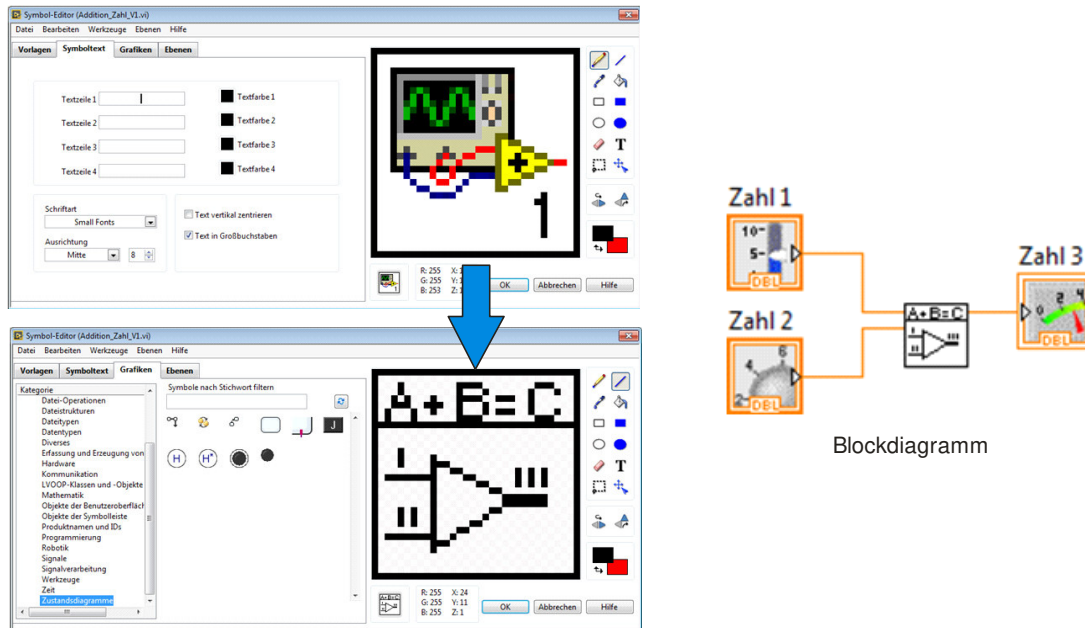


Abbildung 41: Für die Gestaltung der VI Icons steht der Symbol-Editor zur Verfügung. Aussagekräftige VI – Icons erhöhen die Lesbarkeit komplexer Programmstrukturen.

Zu einem VI Symbol gehören ein klassenübergreifender Headertext, sowie ein aussagekräftiges Symbol. Daneben ist für die Wartung und Erweiterung der Programmierung eine Dokumentation der erstellten VIs unerlässlich. Über die VI-Eigenschaften lässt sich dem Element ein Beschreibungstext zuordnen, der das Kontextmenü entsprechend verändert.

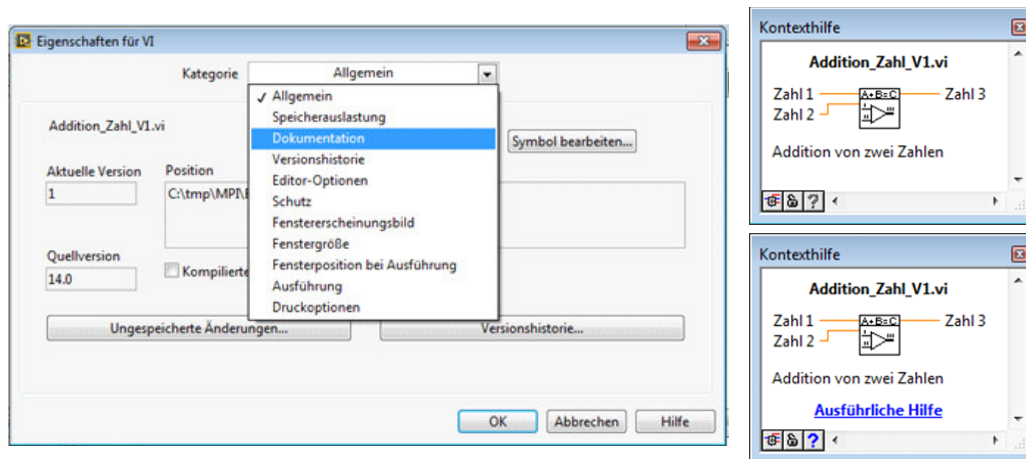


Abbildung 42: LabVIEW verfügt über umfangreiche Dokumentationsmöglichkeiten, die sowohl lokal in der Kontexthilfe als auch global als Online – Hilfe Bestandteil der Entwicklungsumgebung werden und damit Anwender und Entwickler unterstützen.

Dem Programmierer steht ein komplettes Online Hilfesystem für die aktuelle Programmierung und die Verwendung in zukünftigen Erweiterungen zur Verfügung. Die Erklärungen der Dokumentation werden so Bestandteil der Entwicklungsebene und helfen zukünftigen Entwicklern die Programmierung nachzuvollziehen. Die als *.html Datei angelegte Online Hilfe kann beim Ausdrucken der VI-Dokumentation durch individuelle Einstellungen spezifiziert werden. In der Dokumentation kann auf die *.html Datei und andere moderne Formate verwiesen werden. Das Kontextmenü

erweitert sich damit um den Link [Ausführliche Hilfe](#), hinter dem die jeweilige Erklärung zum Quellcode hinterlegt ist.

Das so erstellte VI besitzt nur eine Funktion: es kann zwei Zahlen addieren. Dabei ist die Funktion der numerischen Addition universell angelegt, so können beispielsweise auch Zahlenarrays in einem neuen Blockdiagramm (*Addition_Array_V1.vi*) miteinander verrechnet werden, wenn diese Anschlüsse des Zahlenarrays im Anschlussterminal des Frontpanels als Ein- und Ausgänge definiert werden.

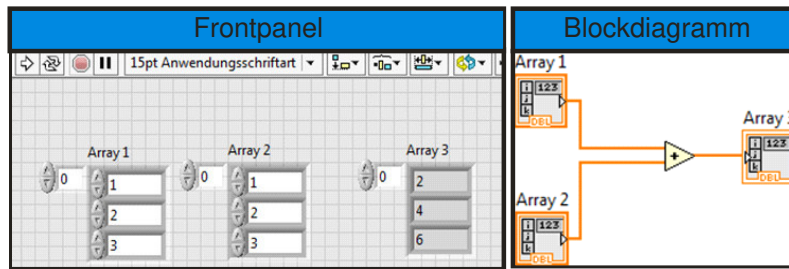


Abbildung 43: Mit demselben Programmcode im Blockdiagramm lassen sich auch Arrays addieren, wenn diese als Ein- und Ausgabeelemente definiert sind.

Diese Funktionalität steht dem VI (*Addition_Zahl_V1.vi*) jedoch nicht zur Verfügung, da in diesem nur Einzelwerte erlaubt wurden. Im Blockdiagramm kann das Zahlenarray als Eingabeelement aufgrund des erlaubten Datentyps deshalb nicht direkt mit dem ersten Beispiel-VI (*Addition_Zahl_V1.vi*) verbunden werden, obwohl intern dieselbe Funktion verwendet wird.

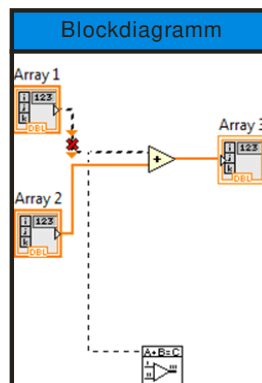


Abbildung 44: Das Beispiel-VI kann nicht mit dem Zahlenarray als Eingabeelement verbunden werden, da diese Operation nicht im VI definiert ist, intern wird jedoch dieselbe Programmstruktur genutzt.

Wie können Datentyp unabhängige Sub-VI's realisiert werden? In einem weiteren logischen Schritt wird eine Textaddition in einem weiteren neuen Blockdiagramm (*Addition_Text_V1.vi*) durchgeführt. Dazu ist die Anwendung einer neuen Funktion erforderlich.

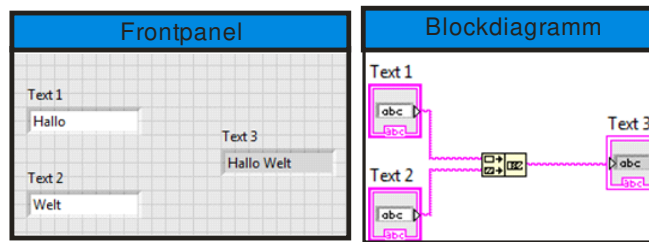


Abbildung 45: Für eine Textaddition wird eine neue Funktion genutzt, die mit entsprechend des Datentyps korrespondierender Text-Ein- und Anzeigeelemente verkabelt sind.

Die Funktion wird mit den entsprechend des Datentyps erlaubten Eingabe- und Anzeigeelementen verbunden. Alle diese Beispiele entsprechen demselben Anschlussmuster und derselben abstrakten Programmstruktur: Durchführung einer Berechnung / Operation mit zwei Eingabeelementen und einem Anzeigeelement für das Ergebnis. Die Programm-VIs sind dennoch nicht miteinander kompatibel, da alle erstellten VIs streng definierte Anwendungsregeln besitzen.

Die Funktionen dieser drei VIs lassen sich in einem neu erstellten polymorphen VI¹⁴ vereinen. Dazu werden dem VI die anderen Instanzen¹⁵ (VI zur Zahlenaddition, Array-Addition, Text-Addition) zugewiesen. Für jede einzelne Instanz lassen sich Symbole hinterlegen, deren Icon abhängig von den angeschlossenen Werten wechselt und deren Instanz in der Lage ist damit zu rechnen. In der Grundinitialisierung liegt die erste Instanz aller zugewiesenen Instanzen oben. Dieses neue VI wird unter dem Namen Addition (*Addition_V1.vi*) abgespeichert.

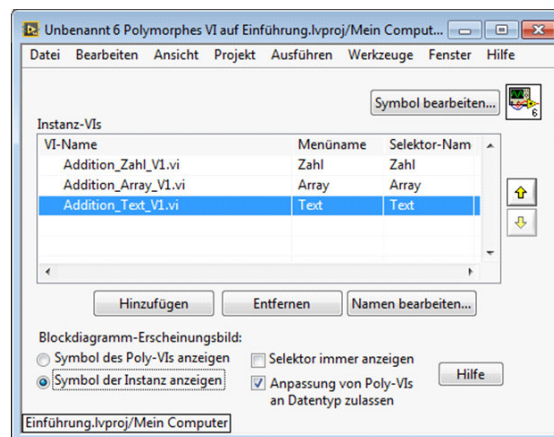


Abbildung 46: Im polymorphen VI sind mehrere Funktionalitätsinstanzen ineinander vereint. Das VI erkennt den Datentyp der angeschlossenen Werte und wechselt entsprechend zu der Instanz, die damit rechnen kann.

Bei der Entwicklung verwenden die meisten Programmierer nur Anschlüsse, die während der Entwicklung für sie erforderlich sind. Anhand des einfachen Beispiels lässt sich bereits erahnen, dass es sinnvoll sein kann auch weitere, in der Entwicklungsphase nicht zwingend benötigte Anschlüsse, im Anschlussterminal vorzuhalten, vor allem wenn weitere Datentypen erforderlich sind oder

¹⁴ Polymorphe VIs vereinen unter einem Anschlusselement weitere VIs (Programme), die jeweils mit unterschiedlichen Datentypen rechnen können.

¹⁵ Instanz VI bezeichnet das VI, das jeweils für einen Datentyp im Polymorphen VI steht.

zusätzliche Eingabelemente benötigt werden. Damit stehen dem Programmcode die notwendigen Ressourcen zur Verfügung hocheffiziente VIs zu generieren, die abstrakte, umfangreiche Funktionalitäten auf eine einfache, lesbare, hocheffiziente Struktur gebündelt abbilden können. Dies dient auch zukünftigen System-Weiterentwicklungen, da nachträgliches Einfügen fehlender Anschlüsse mit hohem Wartungsaufwand verbunden ist.

Komplexere VIs besitzen immer einen Fehlercluster. So kann es z. B. bei der Addition von Signalen durch unterschiedliche Abtastzeiten oder Signallängen zu Fehlern kommen.

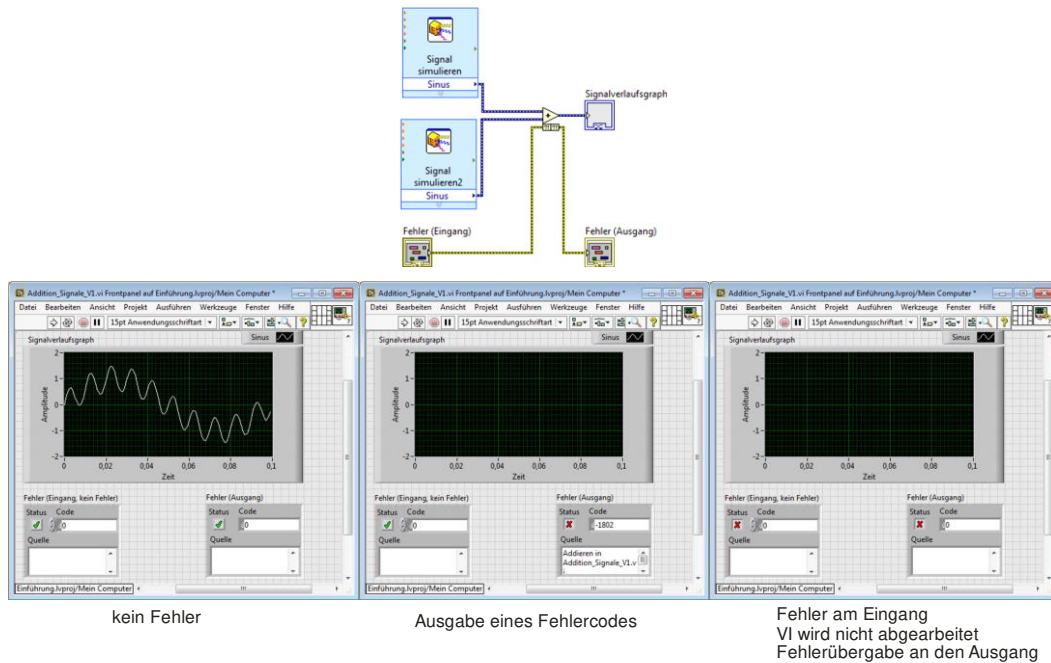


Abbildung 47: Das Fehlercluster definiert das Verhalten des VIs im Fehlerfall.

Auch in anderen, „einfachen“ VIs können Fehlercluster vorgesehen werden, um das Verhalten im Fehlerfall zu definieren. So erweitert sich der Programmcode im Blockdiagramm für das Beispiel-VI (*Addition_Zahl_V1.vi*) mit Fehlerverarbeitung um den Error-Cluster-Eingang (bestehend aus Status, Code und Text), dessen Status als Selektor für eine CASE-Struktur dient, sowie den Error-Cluster Ausgang. Im einfachsten Fall der Fehlerverarbeitung wird der Code nur ausgeführt, wenn kein Fehler am Eingang anliegt.

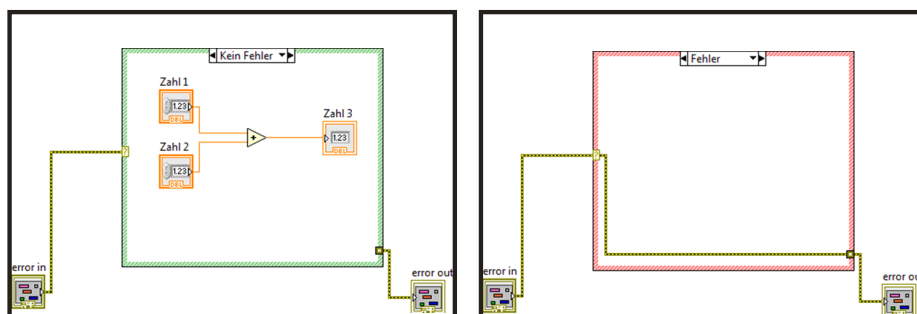


Abbildung 48: Das Beispiel VI wird um einen Fehlercluster mit CASE – Struktur erweitert. Die Programmausführung findet nur statt, wenn kein Fehler am Eingang vorliegt.

Für komplexeres Fehlermanagement wird der Fehlercluster mit Anweisungen zur Reaktion auf unterschiedliche Fehlersituationen befüllt, Prozeduren zur Fehleranalyse hinterlegt und unter Umständen durch unterschiedliche Programme individuelle Fehlerbehandlungsroutinen zur Behebung des Fehlers bereitgestellt. Die Fehlerbehandlung kann beispielsweise über Zahlencodes, unterschiedliche Bedingungsschleifen mit dem Ziel der Fehlerbereinigung abarbeiten. Wenn das Element ordnungsgemäß abgearbeitet wurde, erfolgt die Löschung des Status und das Zurücksetzen des Fehlertextes. Zur weiteren Funktionserweiterung wird eine CASE Struktur innerhalb des Beispiel-VIs mit einem Textelement ENUM (vorzeichenloses Word) verbunden.

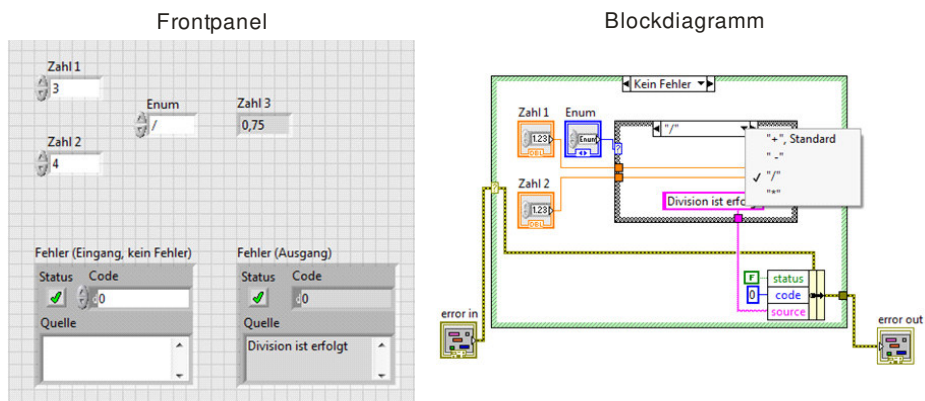


Abbildung 49: Erweiterung des Beispiel-VIs um weitere Mathematikfunktionen über ein ENUM-Element.

Dieses ENUM Objekt kann mit Text befüllt werden: + - / *. Dieser Text dient als Selektor für die CASE – Struktur. Diese Funktionalität hat bei Auswahl von + in dem ENUM Feld eine Addition zur Folge. Für die anderen Texte (+,-, /, *) wird in den dazugehörigen CASE Strukturen die jeweilige Funktionalität bereitgestellt, also Subtraktion, Division , Multiplikation. Auch innerhalb dieser Case Struktur ist wiederum eine Aktualisierung des Fehlercluster sinnvoll.

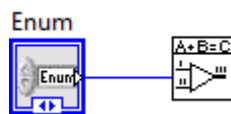


Abbildung 50: Verknüpfung des ENUM Elements mit dem VI stellt die erweiterte Funktionalität allen Instanzen im VI bereit.

Die Idee bei der LabVIEW-Programmierung ist immer entsprechend einfache wiederverwendbare SubVIs zur Verfügung zu stellen, deren Funktionalität direkt über die anschließbaren Terminals offengelegt ist, sowie deren Dateneingänge bereits mit sinnvollen Standardwerten definiert sind. Darüber hinaus ist ein geeignetes Fehlermanagement in Abhängigkeit der jeweiligen Funktionalitäten hilfreich, um möglichst universellen Anwendungen gerecht zu werden.

3.2.2 Gerätesteuerung und Hardwareaufbau

Die Funktion des Synthesizer-Gesamtsystems basiert auf der Qualität der unteren funktionalen Schicht. Fehler in dieser Umsetzungsebene erschweren spätere Hardwareweiterentwicklungen. Umgekehrt, eine sorgfältige Entwicklung nach dem **Bottom-up** Prinzip, bei dem der Funktionsumfang der Geräte uneingeschränkt vollständig erschlossen wird, ermöglicht ein flexibles, modulares System, in dem sich später effizient Geräteeerweiterungen und gerätetechnische Umbauten umsetzen lassen. Dieser vergleichsweise hohe Zeitaufwand bei der Entwicklung amortisiert sich mit jeder weiteren Synthesizer-Variante, die keiner weiteren programmatischen Anpassung der Gerätesteuerung mehr bedarf.

Die einzelnen Gerätekomponenten des Synthesizers verwenden unterschiedliche Arten der seriellen Kommunikation, welches Expertenwissen vom Anwender voraussetzt. Um dies zu vermeiden, werden Zwischenebenen im programmatischen Zugriff realisiert. Dabei kommen Methoden und Anwendungen von **Black Box** Elementen zum Einsatz.

Eine typische Anwendung für ein **Black Box** Element ist ein Gerätetreiber, der ein vollständiges, virtuelles Gegenstück der Hardware-Komponente in der Applikation darstellt. Jeder Befehl der existierenden Gerätehardware wird vollständig administriert. Die Schnittstelleninteraktion basiert auf dem digitalen / analogen Hardwarezugriff in Verbindung mit dem individuellen Geräteprotokoll, dem Fehlermanagement und dem hardwarenahen Handshakeprotokoll. Dabei ist auch die Interpretation von Rückmeldungen der realen Gerätekomponente an die virtuelle Maschine korrekt zu analysieren und gegebenenfalls der übergeordneten Programmstruktur mitzuteilen. Dieses virtuelle Gegenstück der Gerätekomponente, der vollwertige Gerätetreiber, wird als **Black Box** Element ^[163] bezeichnet.

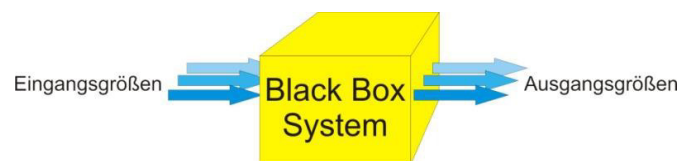


Abbildung 51: Gerätetreiber werden als Black Box Elemente umgesetzt. Der Anwender muss sich nur mit der Eingabe der Eingangsgrößen beschäftigen, wobei Kenntnisse über die innere Funktionalität des Black Box Systems nicht nötig sind, um die angestrebte Funktionalität zu nutzen.

Für den Anwender bedeutet dies, dass er keine Kenntnisse über die Funktionsweise des Systems benötigt. Er kommuniziert mit dem Gerät über die bereitgestellte Eingabe-Schnittstelle, die für ihn alle tieferen funktionalen Schritte zur Gerätesteuerung übernimmt.

Bei der Entwicklung des Gerätetreibers nach dem **Black-Box-**Prinzip^[164] werden die jeweiligen Gerätekomponenten zur analytischen Verifizierung in verschiedene, eindeutig definierte Ausgangszustände geführt oder diese zumindest erfasst und protokolliert. Nach Initiierung des **Black**

Box Elements werden die Ausgangssituationen wiederum beobachtet, protokolliert und analysiert. Lässt sich eine eindeutige Korrelation zwischen der Änderung einer Eingangsgröße und einem Effekt in der Ausgangsgröße erkennen, kann der Versuchsaufbau zur Vermeidung von Kreuz-Nebeneffekten gezielt variiert werden. Aus dem funktionalen Zusammenhang zwischen Eingangs- und Ausgangsgröße ergeben sich die Kausalitätsbeschreibungen, aus denen sich die abstrahierten Funktionen der Gerätetreiber-elemente ableiten.

Diese Herangehensweise erfordert jedoch Qualitätssicherung in der Funktionskontrolle der **Black Box** Elemente. Ein häufiger Fehler ist hierbei ist das Koppeln falscher Kenngrößen an die Qualität, wodurch zwar die jeweiligen Kennwerte erfüllt werden, das System aber dennoch nicht wirklich funktioniert. Eine systematische, kontinuierliche Prozessoptimierung leitet sich hier aus den formalen Methoden der Norm ISO 9000 ^[165] für Qualitätsmanagementsysteme ab. Ziel ist hierbei die innere Funktionalität des jeweiligen **Black Box** Systems in seinem kontextbezogenen aktuellen Optimum zu stabilisieren. Allgemein und auf die einfachste Ebene herunter gebrochen bedeutet dies:

Ein Prozess nimmt keine Fehler an,
ein Prozess macht keine Fehler
und ein Prozess gibt keine Fehler weiter.

=> 0% Fehlertoleranz

Diese Prozessstabilität kann nur mit einem ausgereiften Fehlermanagement gewährleistet werden.

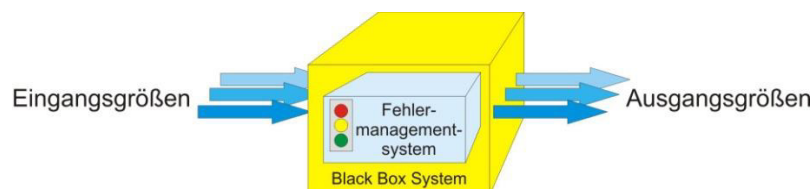


Abbildung 52: Innerhalb des Gerätetreibers sorgt ein komplexes Fehlermanagementsystem für definierte Systemzustände.

Das Fehlermanagementsystem ist ein integraler Bestandteil der Gerätetreiber. Im Hintergrund wird somit eine aktive Überwachungsstruktur implementiert, die auch auf Störfunktionen nicht mit Blockade und Abschottung reagiert, sondern mit der Anpassung der internen Regelparameter auf den neuen Sachverhalt. Diese Überwachungsstruktur wird als Zustandsmaschine^[166, 167] realisiert.

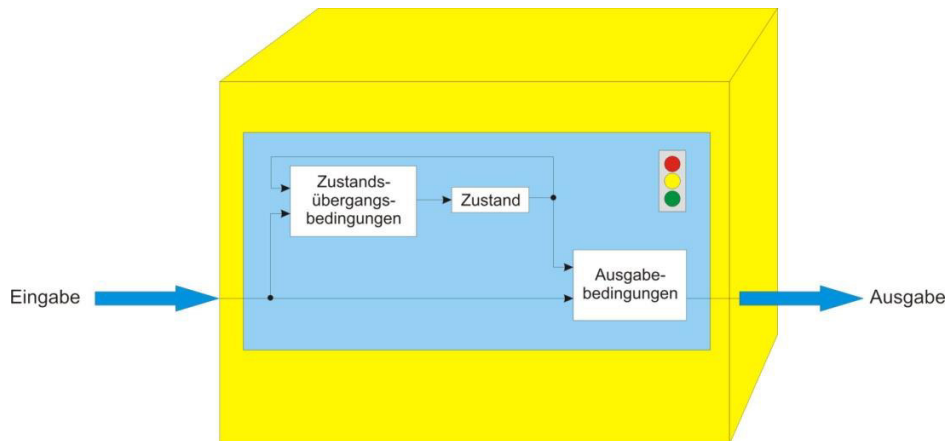


Abbildung 53: Das Fehlermanagement ist in Form einer Zustandsmaschine realisiert.

Eine Zustandsmaschine modelliert das Verhalten eines Systems, das aus Zuständen, Zustandsübergängen und Aktionen besteht. Ein Zustandsübergang wird durch logische Bedingungen beschrieben, die für die Zustandsänderung erfüllt werden müssen. Eine Aktion ist eine Reaktion (Ausgabe) auf einen definierten Zustand. Der Zustand protokolliert in gewissem Umfang seine Vergangenheit, d. h. die Änderungen seit Systemstart bis zum aktuellen Zeitpunkt.

Dieser komplexe Sachverhalt lässt sich an einem der einfachsten Zustandsautomaten, einer Ampel, in Wechselwirkung mit dem Verkehrsteilnehmer anschaulich erklären. Die angezeigten Lichtsignale sind definierte Zustände, die zu einem eindeutigen Verhalten beim Verkehrsteilnehmer führen sollen. Rot steht für Halt, Grün für Fahrt. Interessant wird das Verhalten beim Lichtzeichen Gelb. In dieser Zustandsübergangsbedingung sind mehrere logische reguläre Konsequenzen möglich: scharfes Abbremsen vor der Ampel oder schneller über die Kreuzung fahren, aber auch irreguläre Verhaltensweisen: normal weiterfahren (Fehler=> geblitzt werden/Unfall). Bis zu einem gewissen Grade liegt eine Freiheit in der Interpretation des visualisierten Signals beim Empfänger. Aber auch der Ausfall des Lichtsignals sollte zu keinem undefinierten Verhalten führen. In dem Fall gilt die Straßenverkehrsordnung/Schilder oder eine Verkehrsregelung durch die Polizei. Das Verhalten bei Zuständen, Übergangszuständen und Störfällen ist bei einer Zustandsmaschine also zu jedem Zeitpunkt klar definiert.

Das Beispiel verdeutlicht aber auch die eigentliche Komplexität solcher Systeme. Nicht nur die Ampelschaltung als solche ist ein Black Box Element, in dem eine Zustandsmaschine wirkt. In der Regel interagieren mehrere Ampeln miteinander, allesamt eigene Zustandsautomaten, die unter einem Master – Element agieren. Streng genommen operiert der Verkehrsteilnehmer von außen betrachtet ebenso als eine Zustandsmaschine in der Wechselwirkung mit den Ampel-Elementen, aber auch mit anderen Verkehrsteilnehmern.

Die Ampel kann allerdings nur eine Anforderung verarbeiten: bei Bedarf (durch Drücken des Schaltknopfs, durch ein Signal der Verkehrsüberwachung, durch eine Zeitregelung) auf Grün zu

schalten. Die im Synthesizer betriebenen Einzelgerätekomponenten jedoch sind in der Lage viele verschiedene gerätetechnische Anforderungen wie z. B. Initialisieren und Kalibrieren bis zur Terminierung auszuführen.

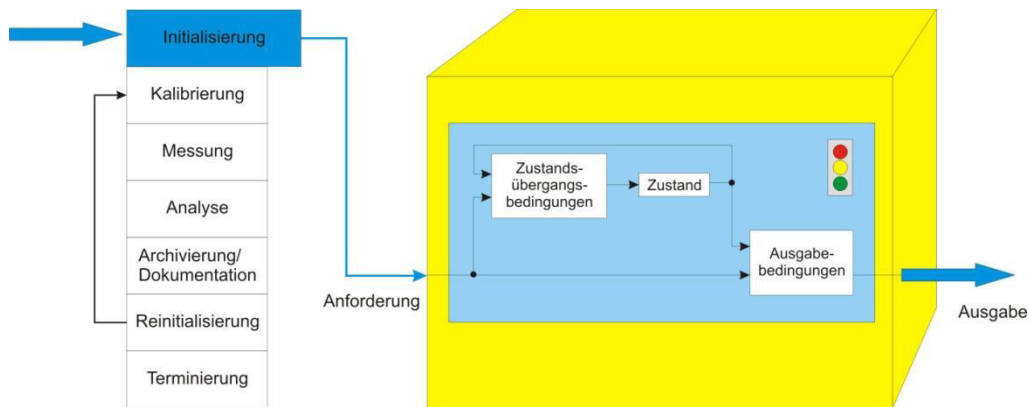


Abbildung 54: Befehle werden von der übergeordneten Applikationsebene in eine Warteschlange geschrieben, die linear abgearbeitet wird. Priorisierte Anforderungen können an die aktuelle Stelle der Warteschlange rücken (z. B. Not-Aus).

Die Kommunikation zwischen diesen eigenständigen Zustandsmaschinen erfolgt über ereignisorientierte Warteschlangen. Jede Interaktion mit diesen Warteschlangen ist ein Wunsch an die jeweilige autarke Zustandsmaschine, die über die Möglichkeit der Abarbeitung oder der dazu notwendigen Vorarbeit entscheidet. Die Abarbeitung der Warteschlange erfolgt im Normalfall linear. Durch Prioritätsvergabe kann eine Anforderung aber auch an höherer Stelle der Liste positioniert werden, wo sie bevorzugt abgearbeitet werden wie dies z. B. bei Betätigung eines Not-Aus Knopfes der Fall ist. Die nachfolgende Tabelle listet als Beispiel die allgemeinen gerätetechnischen Anforderungen auf, die jede Gerätekomponente verarbeiten muss.

gerätetechnische Anforderung	Umsetzung / Ausgabe
Initialisierung	<ul style="list-style-type: none"> Reservierung der Hardware (blockiert die Mehrfachverwendung) Spezifizierung der Schnittstellenparameter Aufbau einer grundlegenden Kommunikation Auslesen gerätespezifischer Konfigurationsfiles Reset-Position / Wartungsposition
Kalibrierung	<ul style="list-style-type: none"> Hinterlegen von Sensordaten Skalierung physikalischer Größen Umrechnungstabellen Linearisierungsobjekte z. B. lösungsmittelspezifische Druckkurven, Durchflussraten, Füllzustände
Messung	<ul style="list-style-type: none"> Auslesen gerätespezifischer Daten wie z. B. <ul style="list-style-type: none"> beim Temperiergerät: Drücke beim Rotationsventil: Ventilposition bei den Spritzen: aktuelle Stempelposition

Abbildung 55: Allgemeine Anforderungen an die Gerätetreiber VIs und deren interne Umsetzung bzw. Ausgabefunktionalität

gerätetechnische Anforderung	Umsetzung / Ausgabe
Analyse	<ul style="list-style-type: none"> z. B. Spektrumsvergleich bzw. Hüllkurven und Amplitudenauswertung
Archivierung / Dokumentation	<ul style="list-style-type: none"> Protokollierung der jeweilig ausgeführten Anforderungen mit aktualisierten Werten, möglichen Warnungen oder lokal auftretenden Fehlern mit eventueller Fehlerbehandlung
Reinitialisierung	<ul style="list-style-type: none"> Zurücksetzen des Gerätes auf Ausgangswerte (wie nach Initialisierung) Freigeben von Puffern Anfahren der Initialisierungswerte / Reset-Positionen
Terminierung	<ul style="list-style-type: none"> Freigeben der Hardware

Abbildung 55: Allgemeine Anforderungen an die Gerätetreiber VIs und deren interne Umsetzung bzw. Ausgabefunktionalität

Jede Gerätekomponente des Gesamtsystems ist zu jedem Zeitpunkt über den eigenen Zustand bzw. Status informiert. Dieser Zustand kann mittels einer übergeordneten Kontrollinstanz auch von den anderen Teilnehmern des Systems abgefragt werden, um geräteübergreifend logische Abhängigkeiten miteinander abzugleichen und Prozesse zu synchronisieren.

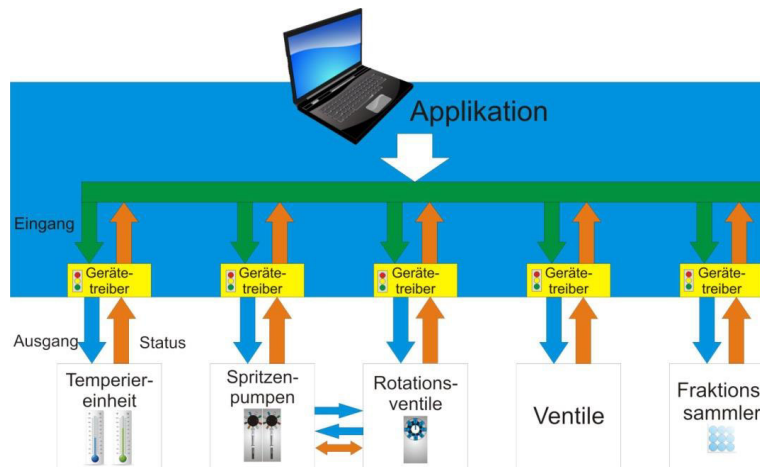


Abbildung 56: Geräteübergreifende Funktionskontrolle der einzelnen Systemkomponenten durch Statusmeldungen der Gerätetreiber als eigenständige Zustandsmaschine.

Da keine gegenseitigen direkten Zugriffe erlaubt sind, mit Ausnahme der Spritzenpumpe und Rotationsventile, die direkt eine herstellerbedingte Zustandsmaschinen-Kommunikation innerhalb deren lokalen Controller-Baugruppe führen können, sind hier auch keine fehlerhaften Werte oder Anweisungen möglich, die zur lokalen Irritation durch Mehrdeutigkeit oder Schrittkettenverstoß führen. So lässt sich eine geräteübergreifende Funktionskontrolle, z. B. beim automatisierten Spülen von kompletten Lines durch die Spritzen- / Ventilsteuerung oder bei der Herstellung von sinnvollen Grundzuständen wie der Reset-Positionen der Ventile, realisieren. Hierdurch wird ein dezentralisiertes Abarbeiten der Prozessanforderungen ermöglicht.

Diese Systemkonstruktion garantiert damit die hohe Prozessstabilität und Fehlertoleranz, die für die Durchführung langwieriger, komplexer, chemischer Synthesen erforderlich ist. Das Konzept ermöglicht eine Prozessautomatisierung, die in der Lage ist, Entscheidungen in gewissen Grenzen auszuführen und vor allen Dingen auf auftretende Fehler individuell und eigenverantwortlich zu reagieren.

Führt man diese Entwicklung auf Ebene der Gerätesteuerung konsequent auch für die höheren Funktionsebenen der Gerätekontrolle durch, so wird eine vollständige Modularisierung innerhalb jeder der aufeinander aufbauenden Schichten möglich. Modulare Funktionselemente auf gleicher Ebene besitzen dann die Fähigkeit mit hierarchisch gleichwertigen Funktionselementen zu kommunizieren. Jede Ebene der gerätesteuerungstechnischen Umsetzung lässt sich so als eine in sich geschlossene Regelkreisstruktur darstellen, die wieder für sich einer Zustandsmaschine im Black Box Element entspricht. Damit wird mit dieser Form der Gerätesteuerung im neuen Synthesizer eine funktionale Grundstruktur etabliert, die sich am ISO 7–Schichten-Modell^[168] anlehnt.

Der Benutzer interagiert dabei hierbei auf der Mensch – Maschinen – Schnittstelle mit der Leitebene (Applikationsebene). Diese Leitebene ermöglicht das Anzeigen der wichtigsten systeminternen Informationen, das grundlegende Bedienen des Synthesizers und übernimmt die wichtigsten Überwachungsfunktionen neben der eigentlichen Systemsteuerung.

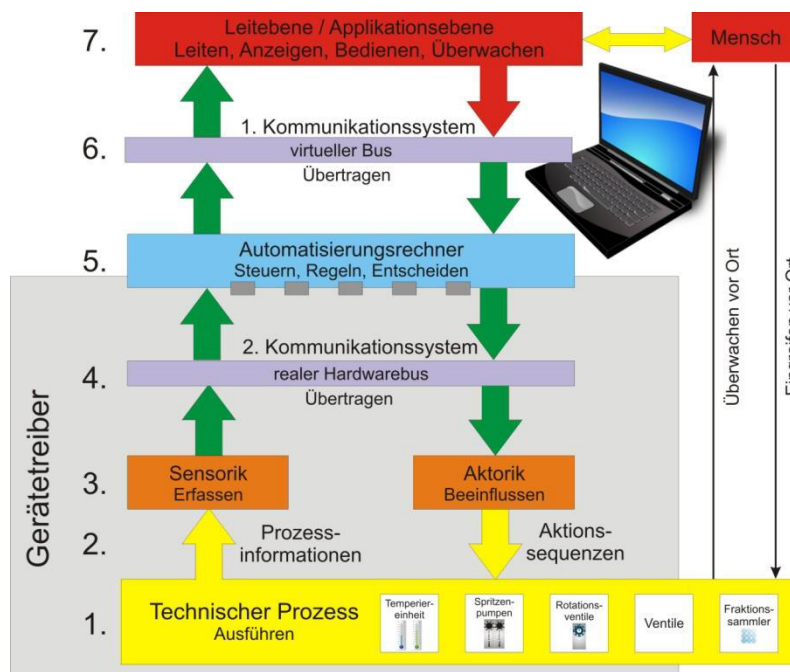


Abbildung 57: Die funktionale Grundstruktur der gesamten Synthesizer-Systems orientiert sich am ISO 7 Schichtenmodell.

Von hier aus wird mit einer interaktiven Warteschlange über ein virtuelles Bussystem, dem ersten Kommunikationssystem, kommuniziert. Ausschließlich über diese virtuelle Ebene werden

Befehlsanforderungen und Systeminformationen an die jeweiligen Systemkomponenten über die Gerätetreiber ausgetauscht, die für sich autonome Zustandsmaschinen darstellen. Jedes virtuelle Gerät trifft für sich allein Entscheidungen, die entweder zu Steuer- oder Regelprozessen führen. Aus dieser virtuellen Aufteilung entstehen Anforderungen, die über die jeweilige Ausgabeschnittstelle des betreffenden Gerätetreibers auf den realen Hardwarebus (zweites Kommunikationssystem) übertragen werden. Von hier aus wird somit die Sensorik, welche Prozessinformationen erfasst, ausgelesen oder die Aktorik zur Umwandlung von Signalen in physikalische Größen zur Steuerung vorgegeben. Diese wirken auf den jeweils beeinflussten technischen Prozess. Als Beispiel dienen hierzu die in Abbildung 58 dargestellte sensorischen (mechanische oder optischen) Kontrollmöglichkeiten zum Überprüfen des ordnungsgemäßen Lösungsmitteltransports ins Reaktionsgefäß. Hierzu wird u. a. durch Überwachen des Stromflusses im schaltenden Ventil, des Flüssigkeitstransports im Schlauch und durch eine Füllstandskontrolle im Reaktionsgefäß die Funktionalität plausibilisiert.

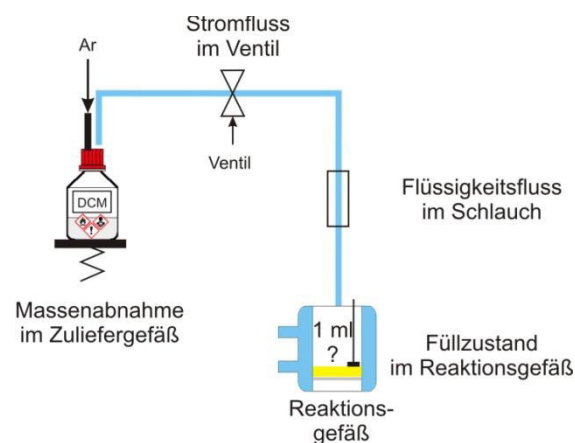


Abbildung 58: Sensorische (mechanische und optische) Überprüfungs­möglichkeiten stellen fest, ob eine Systemanforderung wie z. B. ein Lösungsmitteltransport in das Reaktionsgefäß ordnungsgemäß erfolgt ist. In Abhängigkeit ob die Anforderung erfolgreich umgesetzt wurde oder ein Fehler erkannt ist, greifen Fehlervermeidungsstrategien zur Syntheseoptimierung.

Der Benutzer kann in Gefährdungssituationen einen Not-Aus betätigen, der parallel am normalen Prozessablauf vorbei die Ausführung des technischen Prozesses zeitnah beendet. Auf der Ebene der geplanten Arbeitssynthesizer ist diese Zugriffsmöglichkeit wichtig, um weitreichende Flexibilität in der Entwicklungsphase zu sichern. Bei kommerziellen Endgeräten ist von dieser Art der Umsetzung jedoch abzusehen, da es in ungünstigen Systemzuständen zu aufwendigen Wiederherstellungsprozessen wie dem Spülen der Leitungen, der Re-Initialisierung der Pumpen, Positionsermittlung der Rotationsventile und aktuelle Schrittkettenposition in der Syntheseabarbeitung führen kann.

Diese erhöhte Bedienerfreundlichkeit wird durch die eigene Kontroll-Ebene, eine übergeordnete Funktionseinheit erreicht, die zunächst im Automatisierungsrechner (Schicht 5) mit Hilfe der

Software-SPS realisiert wird. In der weiteren Geräteoptimierung ersetzt dann eine Hardware-Lösung mittels SPS (speicherprogrammierbare Steuerung) diese übergeordnete Funktionalität. Dies dient der Gerätesicherheit, da bei Störfällen wie einem Stromausfall oder eines betriebssystembedingten Systemabsturzes im Automatisierungsrechner, diese eigenständige Hardware durch einen batteriegepufferten Stromkreis immer noch in der Lage ist, den Synthesizer in einen definierten, für den Anwender und die laufende chemische Synthese, sicheren Betriebszustand zu fahren.

Die übergeordnete Funktionseinheit gibt zwar die Anforderungen an das Gerät ab, jedoch entscheidet die SPS auf Basis hinterlegter Fehlervermeidungsstrategien, wann die Anforderungen umgesetzt werden. Der Gerätetreiber umfasst mehrere Funktionsschichten im Steuerungskonzept der Systemhardware des Gesamtsystems (Abbildung 57). Diese umfassende Funktion wird in LabVIEW durch eine konzeptionelle, allgemeingültige Programmstruktur beschrieben.

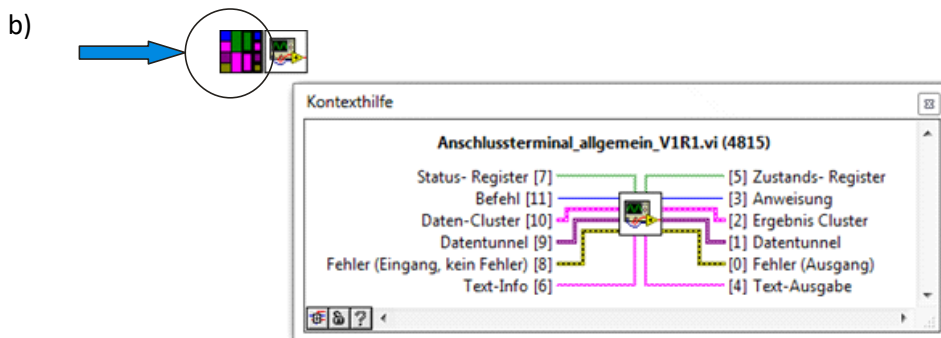
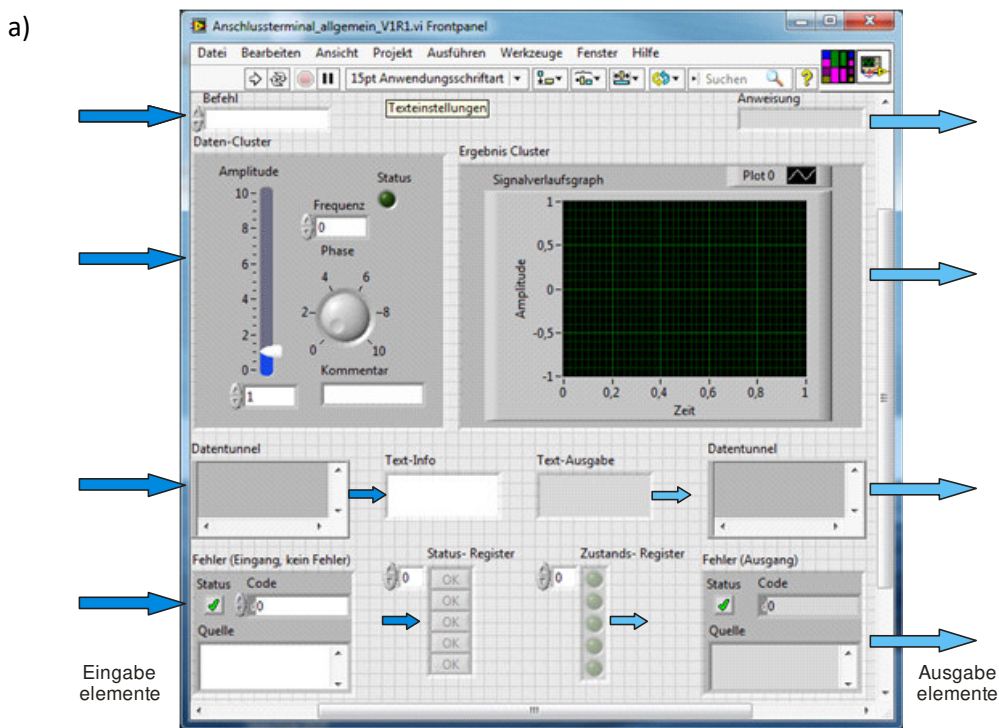


Abbildung 59: a) Beispiel für die konzeptionelle Umsetzung eines allgemeinen Gerätetreiber VIs als Black-Box-Elemente mit interner Zustandsmaschine (Frontpanel) b) Kontexthilfe zu a) zeigt die erstellten Anchlusselemente, die für das allgemeine, konzeptionelle Gerätetreiber-VI angelegt sind.

Beim Design der Frontends hilft eine klar strukturierte Gestaltung, um die Lesbarkeit komplexer VIs im Hinblick auf Wartbarkeit und Fehlersuche zu erhöhen. Die verwendeten Frontendelemente sollten hierbei die maximal zu erwartende Komplexität abbilden, damit bei zukünftigen funktionalen Erweiterungen kein verkabelungstechnischer Mehraufwand entsteht. Die hier beschriebene systematische Gestaltung hat sich als optimale VI Vorlage für die Gerätetreiber VIs bewährt und liegt in objektindividuellen Ausbaustufen jedem im Projekt erstellten VI und SubVI zugrunde.

Die Anschluss terminals der Gerätetreiber-VIs entsprechen immer demselben Schema. Linksseitig befinden sich sechs Eingänge und rechtsseitig sechs Ausgänge, die von ihrer grundlegenden Funktionalität ähnlich belegt sind. So befindet sich der Fehlereingangs-Cluster stets unten links, der entsprechende Fehlerausgangs-Cluster analog dazu unten rechts. Die Funktionsanforderung (Befehl) steht der Befehlsweiterleitung (Anweisung) auf der rechten Seite gegenüber. Im Dateneingang (Daten – Cluster) auf der Eingangsseite werden die zur jeweiligen Befehlsanforderung notwendigen Parameter und Daten bereitgestellt. Analog dazu wird im Datenausgangs-Cluster (Ergebnis-Cluster) auf der rechten Seite die entsprechend der Befehlsabarbeitung veränderte Datenbasis zur Verfügung gestellt.

Für die individuelle Übergabe verschiedenster Datentypen befindet sich unterhalb des Daten-Clusters auf der Eingangsseite ein Variant-Cluster (Datentunnel), dessen Interpretation aufgrund der Befehlsanforderung in unterschiedlichster Art möglich ist. Hier können innerhalb des VIs komplexe Datentypen für die nachfolgende Codesequenz zur Verfügung gestellt werden. Auch auf der Ausgangsseite ist ein Cluster vom Datentyp Variant (Datentunnel) entsprechend bereitgestellt. Obwohl die Grundfunktion der jeweiligen VIs mit der Definition der Standardwerte bei der Codeentwicklung bereits zugrunde gelegt ist, erhält der Anwender durch ein Status-Cluster die Möglichkeit hier flexibel programmatisch zuzugreifen. Um jedoch für weitreichende Änderungen bereits relativ zukunftsicher Schnittstellen zur Verfügung zu stellen, existiert als letztes Anschluss terminal die Übergabemöglichkeit von Textarrays, deren funktionale Auswertung in Abhängigkeit der Befehlsanforderung sowohl Zugriffe auf Verzeichnisstrukturen als auch aktiv genutzten Skriptcode abbilden kann.

Durch diese Gestaltung der Eingangs- und Ausgangsobjekte wird eine weitreichende Flexibilität für die Verwendung dieser VIs gewährleistet. Funktionale Erweiterungen innerhalb der Befehlsstruktur haben jeweils eine interne Bedingungsselektion zur Folge, wodurch ein tiefer gelagertes Sub-VI zur Ausführung kommt. Dadurch wird prinzipiell ein ähnlicher Aufbau abbildet, wie der bereits besprochene universelle Gerätetreiber-VI.

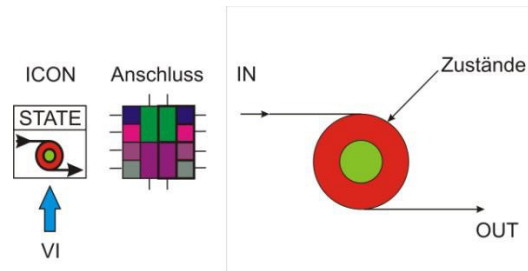


Abbildung 60: Jedes VI in der Programmoberfläche besteht aus einem visuell aussagekräftigem Programm-ICON und einem funktionalen Anschlussterminal.

Als LabVIEW Anwendungsbeispiel umgesetzt, ist nachfolgend ein allgemeines Gerätetreiber-Haupt-VI mit sechs statisch aufeinander folgenden Schrittkettenelementen dargestellt (Abbildung 61). Für jedes Schrittkettenelement existiert ein eigenes Unterprogramm (Sub-VI), das die jeweiligen Zustände wie z. B. Initialisierung durch Bereitstellung der Verbindung zur seriellen Schnittstelle mit Übergabe von Baudrate¹⁶, Datenlänge und Austauschparameter oder Kalibrierung zur Durchführung der gerätespezifischen Referenzmessung, herstellt.

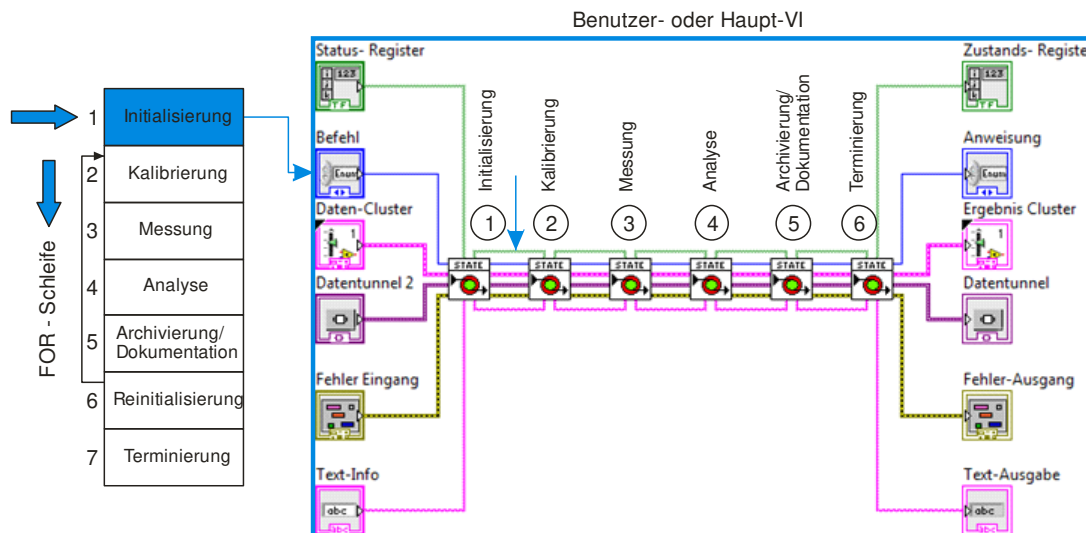


Abbildung 61: LabVIEW Programmbeispiel: Blockdiagramm einer Zustandsmaschine aus sechs statisch aufeinanderfolgenden Schrittkettenelementen

Normalerweise werden diese sechs Zustände mit sechs verschiedenen Codes erstellt und über eine Schleife abgearbeitet. In der vorliegenden Arbeit wird jedoch ein anderer Ansatz verfolgt: die Objekte halten alle zukünftigen Zustände bereits bei der Entwicklung vorrätig. Sie kennen die Schrittkette, um die Zustände herbeizuführen, zu stabilisieren und diese vorschriftsmäßig abzuarbeiten. Diese Programm-VIs werden der Reihe nach abgearbeitet. Das Sub-VI Initialisierung übergibt die Anforderung zur Kalibrierung erst an das nachfolgende Unterprogramm, wenn der angeforderte Initialisierungszustand erreicht wurde. Dabei hat das VI aber auch immer Kenntnis vom Zustand des Gesamtsystems, da das Statusregister mit angeschlossen wurde und dieses auch den weiteren Sub-

¹⁶ Baudrate ist eine Einheit in der Kommunikationstechnik, welches die Schrittgeschwindigkeit (Symbolrate) darstellt. Bei einem Baud wird ein Symbol pro Sekunde übertragen.

VIs zur Verfügung steht. Das Error-Cluster teilt dem nachfolgenden VI mit, ob es zu einer Fehlermeldung kam. So kann eine Kalibrierung nicht durchgeführt werden, wenn die Initialisierung fehlgeschlagen ist. Innerhalb der VIs werden dazu Fehlermanagementroutinen verankert, die auf die Statusrückmeldungen entsprechend reagieren.

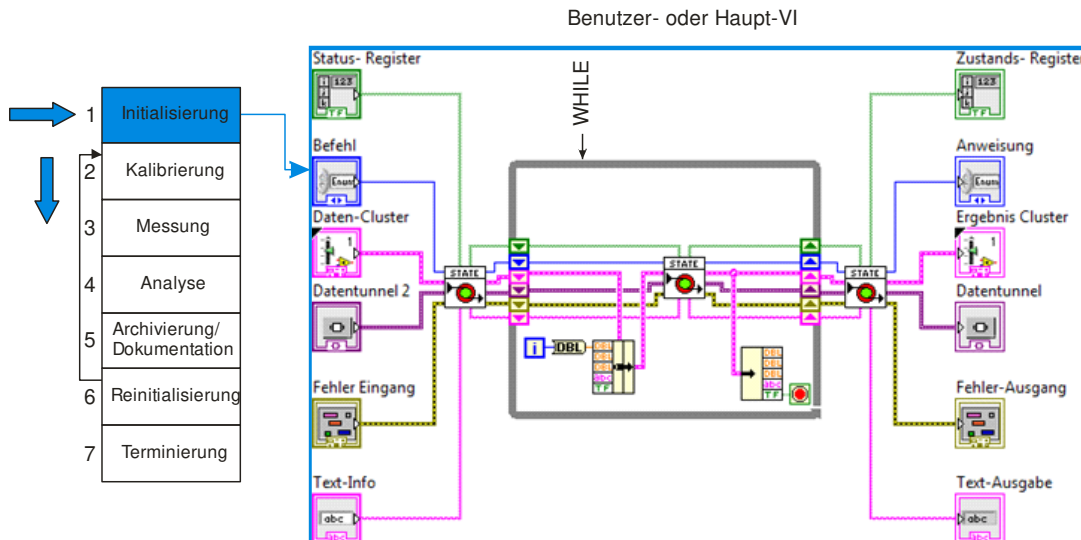


Abbildung 62: LabVIEW Programmbeispiel: Blockdiagramm einer Zustandsmaschine mit Zugriff über Schieberegister

Da aufgrund des hier verfolgten Konzeptes zum Aufbau der VIs die Übergabeobjekte identisch aufgebaut sind, können diese auch bei wesentlich komplexeren Schrittketten aufeinandergestapelt werden. Die Schrittkettenelemente sind im Programmbeispiel (Abbildung 62) bis auf das Initialisierungs- und Terminierungs-VI übereinandergeschichtet. Sowohl am Eingang als auch am Ausgang der SubVIs wird der Datenfluss über ein Schieberegister gesteuert. Das Schieberegister hat die Eigenschaft, die übergebenen Datenobjekte an den Anfang der Schleife zurückzuführen, damit sie für den nächsten Schleifendurchlauf als Eingangsobjekte zur Verfügung zu stehen. Im einfachsten Fall geschieht dies über eine FOR – Schleife(wenn es sich um eine reine Ablaufsteuerung handelt). In diesem Beispiel (Abbildung 62) findet eine WHILE – Schleife Verwendung, die solange durchläuft, bis die Abbruchbedingung erreicht ist. Je nach lokalen Bedingungen kann in dieser Programmkonstruktion wieder auf andere Zustände zurückgesprungen werden. Die Überprüfung eines Prozesses auf Plausibilität und als daraus resultierender Konsequenz die Rückführung an eine bestimmte Stelle in der Schrittkette, führt zur Entwicklung selbstheilender VIs. Der Vorteil dieser Programmstruktur ist die einfachere Wartbarkeit. Auch benötigen individuelle Funktionserweiterungen keinen umfangreichen Eingriff in die Gesamtprogrammierung. Die zusätzlichen SubVIs werden einfach mit einer neuen Bedingung in der WHILE – Schleife gestapelt.

Abbildung 63 zeigt die erstellten Sub-VIs für den Gerätetreiber der Kloehn-Pumpe als eine der Gerätekompenten im Synthesizer – Gesamtsystem. Jede, der in den SubVIs umgesetzten

Funktionen, besteht aus einer Vielzahl von Sub-VIs, in der die notwendigen Schrittketten (Abbildung 62) abgearbeitet werden.

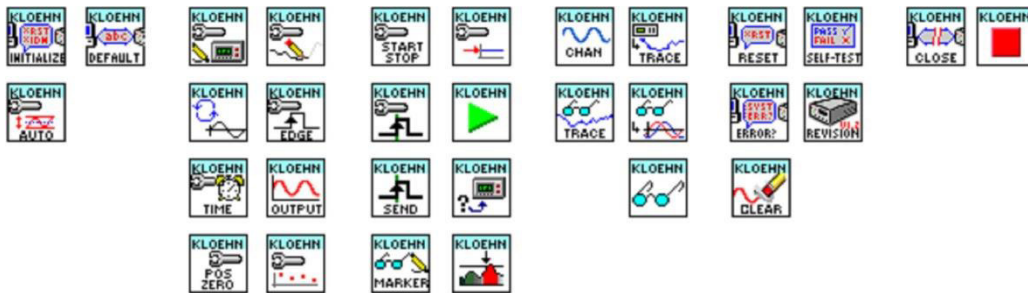


Abbildung 63: Für die Kloehn – Pumpe erstellte VIs, die aus verschiedenen SubVIs mit komplexen Schrittkettenanweisungen bestehen

Zusammengefasst gilt für die Gerätetreiber als eigenständige Black – Box – Elemente: auf die Anforderung (= Eingabe) erfolgt eine definierte Reaktion (= Ausgabe), der innere Prozess der Entscheidungsfindung wird für den Nutzer des Systems durch eine Zustandsmaschine umgesetzt. Die Zustandsmaschinen können untereinander ihren Status abfragen und logische Entscheidungen damit verknüpfen. Das Gesamtsystem interagiert miteinander auf einem virtuellen Kommunikationsbus. Eine übergeordnete Entscheidungsstruktur entweder softwareseitig (Software – SPS) oder im weiteren Verlauf der Geräteentwicklung als Hardware – SPS umgesetzt, überführt das System immer in stabile Zustände, wenn es zu äußeren Störfällen wie einem Stromausfall, einem Not-AUS oder einem Absturz des Betriebssystems des Systemrechners kommt. Die Gerätekommunikation orientiert sich strukturell am ISO-7-Schichtenmodell.

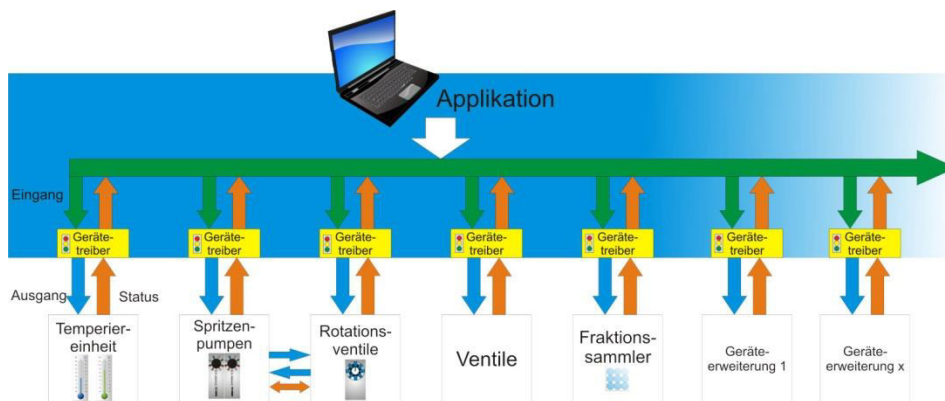


Abbildung 64: Geräteerweiterungen sind beim dezentralen Steuersystem relativ leicht in das Gesamtsystem zu integrieren. Neben dem Hardwareanschluss muss der neue Gerätetreiber zum Betrieb in der Synthesizer-Applikation eingebunden werden.

Das Gesamtkonzept zur Umsetzung der Gerätesteuerung sichert eine hohe Prozessstabilität und Fehlertoleranz während der Abarbeitung der Synthesen. Dadurch wird die Gerätebedienung insgesamt vereinfacht und es ist kein tieferes Expertenwissen zur Hardware mehr notwendig. Der Hardwarezugriff mit den verschiedenen Protokollen und unterschiedlichen Kommunikationsmechanismen mit den jeweiligen Endgeräten erfolgt über eine Schnittstelle, die so

konzipiert ist, dass mit einem rutimentären Befehlswoortschatz die Grundfunktionalität abgedeckt wird. Bei der Umsetzung wird mit NAMUR^[169] die aktuell am weitest konzeptionierte Variante an chemischen EDV-Standardschnittstellen unterstützt. Aus diesem Grund werden die zukünftigen gerätetechnischen Module und Einzelplatzgeräte, die diesen Standard unterstützten, bevorzugt behandelt. Die Erweiterungen des Gesamtsystems mit neuen Gerätekomponenten, z. B. einem Fotoreaktor, sind nach diesem Konzept leicht möglich. Neben dem Hardwareanschluss, muss dann nur der neue Gerätetreiber in die Synthesizer-Applikation eingebunden werden. Die Steuersoftware selbst bleibt unverändert.

3.2.3 Bediensoftware und Synthesesprache

Damit der neue Synthesizer einem größeren Anwenderkreis zugänglich wird, sind erhebliche Vereinfachungen, sowohl in der Bedienoberfläche als auch in der Benutzerführung erforderlich. Dabei ist zu unterscheiden zwischen der Hardwaresteuerung (siehe 3.2.2 Gerätesteuerung) und der Anwender-Applikation, an welcher der Benutzer die notwendigen Eingaben zur Synthesesteuerung vornimmt. Unabhängig auf welcher Ebene die „Sprachentwicklung“ für den Synthesizer im folgenden Konzept beschrieben wird, auf die zugrundeliegende Gerätesteuerungssoftware hat dies keinen Einfluss mehr. Die Steuerungssoftware bildet auch als Gesamtsystem eine Black-Box-Struktur ab, in der alle hardwaretechnisch umgesetzten Funktionen – auch in beliebiger Interaktion der einzelnen Gerätekomponenten untereinander - auf Anforderung bereitsteht.

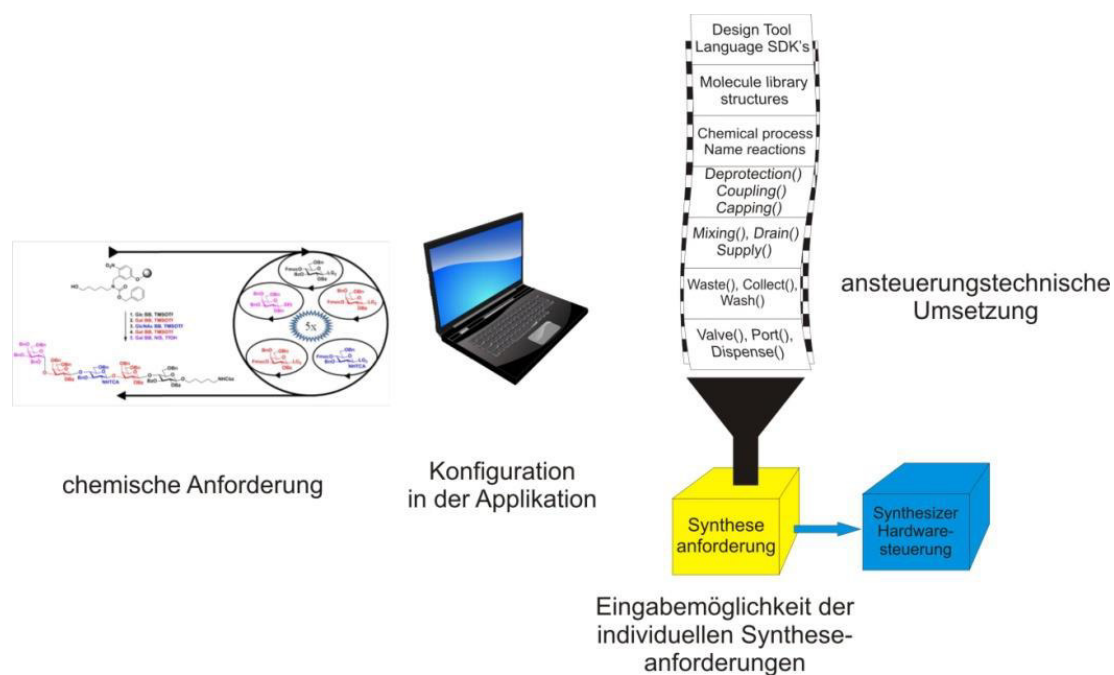


Abbildung 65: Hardwaresteuerung und die Bediensoftware ergeben zusammen die Gesamtapplikation. Die Hardwaresteuerung ist ein Black Box System, an das die Bediensoftware Anforderungen stellt, die sich aus der Synthesevorschrift ergeben. Die Abstraktionstiefe ist von der Stufe der Sprachentwicklung, die innerhalb der Bediensoftware verwendet wird, abhängig.

Der Anwender erhält Zugriff auf alle Grundfunktionen in Basiseinheiten, die zur gerätetechnischen Steuerung notwendig sind. Nach außen hin wechselwirken diese über eine Bandbreite von definierten Einzelbefehlen. Dies ist die funktionale Schnittstelle zur zweiten Komponente der Gesamtapplikation: der Anwender-Applikation. Die nachfolgende Übersichtsgrafik zur Abstraktion der Synthesprache baut sich auch hier nach dem Bottom-up Verfahren von unten nach oben auf. Je höher ein Funktionsblock in der Darstellung implementiert ist, desto weniger Kenntnis muss der Verwender von den geräte-/prozessspezifischen Funktionalitäten haben.

Auf der Ebene der physikalischen Prozesse erfolgen die Benutzereingaben befehlsorientiert. Die dazu nötigen Kommandos leiten sich - historisch bedingt - aus dem Befehlsvorrat der bisherigen Synthesizer ab, da alle bisher erarbeiteten Funktionalitäten und Steueranweisungen der letzten verwendeten Synthesizer-Version weiter genutzt werden sollen. Die Gerätesteuerung erfolgt jedoch bereits komplett auf Basis der neuen Synthesizer-Applikation unter LabVIEW. Auf dieser Ebene ist die Benutzerführung noch stark hardwareorientiert und der Anwender benötigt Kenntnisse über die Abfolge der Syntheseschritte in Bezug auf Abläufe in der Steuerungskette der einzelnen Gerätekompenten.

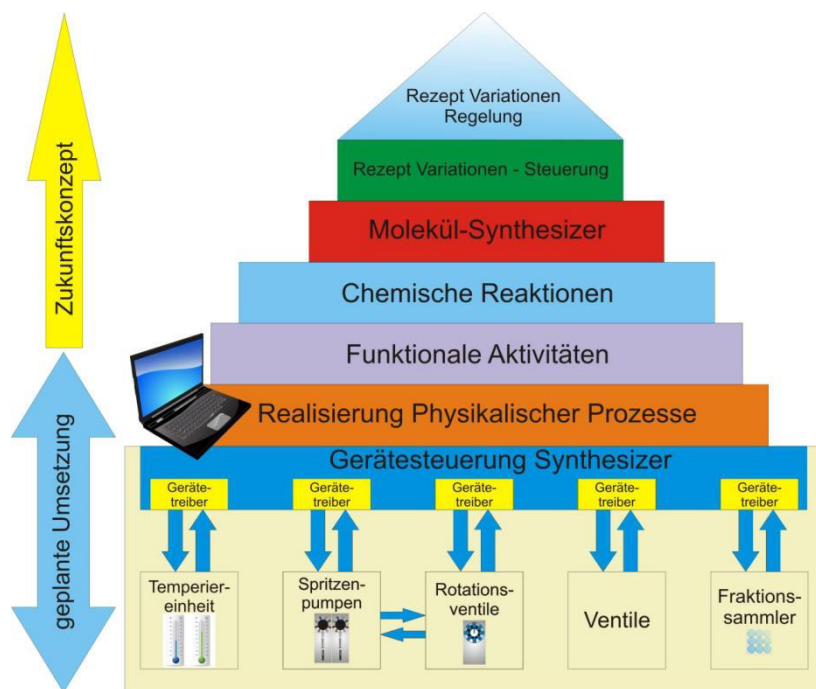


Abbildung 66: Abstraktionsebenen zur Entwicklung eines „universellen“ Synthesautomaten. Mit jeder höheren Abstraktionsebene wächst die Komplexität der zur Verfügung stehenden Funktionalitäten. Zugleich nimmt die Komplexität in der Bedienung für den Anwender in jedem Abstraktionslevel ab.

In der nachfolgenden Abstraktionsebene werden Einzelbefehle zu geräteunabhängigen Funktionen und physikalischen Parametrierungen durch funktionale Aktivitäten abstrahiert. Dabei fassen Funktions-Module komplexe, gerätekompentenübergreifende Funktionalitäten, wie z. B. Spülen der Schläuche, Waschzyklen und Temperatursteuerung, zusammen.

In den Verfahrensablauf – Modulen auf der nachfolgenden Ebene ist die Definition von zusammenhängenden technischen Anweisungen, Abhängigkeiten, Zeitverläufen, Schleifen und Temperaturgradienten-Profile in Form von umfangreichen Skripten abbildbar. Ziel bei der Software-Entwicklung der Benutzeroberfläche ist eine flexible Anwendungsebene in Richtung von chemischen Verfahren. Auf dieser Ebene kann durch verständliche Anweisungen bereits eine effiziente Benutzerführung stattfinden. In Form von Prozessen werden modulare chemische Reaktionen, Bezeichnungen und Einzelzyklen umgesetzt. Die Prozessebene ermöglicht dem Benutzer ein individuelles Parametrieren des Syntheseverlaufes durch individuell angepasste Bedienungsoberflächen.

Ziel des Gesamtkonzepts der Anwendersoftware ist die Entwicklung einer flexiblen Synthesesprache, die über mehrere Abstraktionsebenen alle bisherigen Syntheseprotokolle mit wenig Aufwand weiterhin nutzbar lässt. Intuitive Software-Bedienung durch die Weiterentwicklung der ursprünglich kommandogeführten Bedienstruktur in Richtung einer chemisch basierten Struktur kommt den Bedienerwünschen entgegen. Dieses Bedienkonzept aus abstrakten Funktionen, die sich aus den gewonnenen Erkenntnissen der automatisierten, chemischen Verfahren ergeben, steht am Anfang der angestrebten Entwicklungskette. Die Protokolle aller durchgeführten Synthesen werden zur systematischen, strukturierten Analyse in einer geeigneten chemischen Datenbank bereitgestellt, um die im Gesamtkonzept vorgesehene Schaffung der rudimentären Voraussetzung zum Aufbau eines Expertensystems zu ermöglichen.

4 Die neue Generation der Oligosaccharid-Syntheseautomaten

4.1 Überblick über das Gesamtsystem

4.1.1 Anforderungen an den Synthesizer

Die Umsetzung des neuen Oligosaccharid-Syntheseautomaten erforderte eine Vielzahl geräte-, material-, sowie steuerungstechnischer und baulicher Maßnahmen, die während der Projektdurchführung synchronisiert werden mussten. Dabei waren die gesetzlichen Vorschriften zur Konstruktion und Betrieb elektrischer Laborgeräte zu beachten.^[170,171] Im Verlauf des Projektes wurden drei Syntheseautomaten neuester Generation realisiert, im Folgenden auch als „Arbeitssynthesizer“ bezeichnet, die den Chemikern bereits während der Entwicklungszeit direkt als Arbeitsgeräte zur Verfügung gestellt wurden. Allen drei Synthesizern liegt die gleiche geräte- und steuerungstechnische Konzeption zugrunde, obwohl sie teils mit unterschiedlicher Hardware aufgebaut und in Funktion gebracht wurden.

Da bereits bei Projektbeginn zwei J-KEM-Synthesizer am Institut existierten, bekam das erste vollständig neu aufgebaute System die Bezeichnung *Arbeitssynthesizer III*. Dieses System wurde im März 2012 nach ausführlichen Tests mit der ersten Vollsynthese^[172] in die alltägliche Laborpraxis übernommen.

J-KEM-Synthesizer II wurde im Rahmen einer Reparatur auf das neue Gerätekonzept mit der National Instrument Plattform zum *Arbeitssynthesizer II* umgebaut. Dies ermöglichte die systemübergreifende reproduzierbare Verwendung des gleichen Ansteuerungscodes und führte zur stabilen Ausführbarkeit auch längerer Synthesesequenzen auf dem Gerät.

Im nächsten Projektschritt wurde sowohl Ansteuerung, als auch Mechanik des verbliebenen J-KEM-Synthesizer I vollständig auf das im Konzept beschriebene aktuelle technische Niveau gebracht und als *Arbeitssynthesizer I* in Betrieb genommen.

Zu den Hauptanforderungen bei der Projektrealisation zählte die verbesserte Temperierung. Das Temperiergerät sollte zur effektiven Reduzierung der Gesamtsynthesezeit schneller und exakt die gewünschte Zieltemperatur erreichen, insbesondere bei Temperaturen unter -20° C, und über den eingestellten Reaktionszeitraum durch Regelung stabil die Reaktionstemperatur halten. Mit einer gleichmäßigen, kontinuierlichen Zuführung von Gasblasen zur optimalen berührungsfreien Durchmischung der Syntheselösung im Reaktionsgefäß sollte eine Erhöhung der Reaktionseffizienz erreichbar sein. Durch softwareseitige Erweiterungen der implementierten Fraktionssammler, die der Entnahme und Sammlung von Zwischen- und Endprodukten auch während der Synthese dienen, wurde die Gerätesteuerung den Bedürfnissen der automatisierten Synthese angepasst. Der Betrieb unter Schutzgasatmosphäre erfordert umfangreiche geräte- und materialtechnische

Voraussetzungen an die Dichtigkeit von Verbindungen und gleichmäßiger Druckverteilung im System. Durch die Drucküberwachung der Synthesizer wurde eine Kontrollmöglichkeit für die systembedingte Reaktionsumgebung installiert. Das Gesamtsystem erforderte konstruktive Lösungen zur Vermeidung bzw. der Entfernung von Feuchtigkeit in den Synthesewegen, da diese die Reaktivität der Synthesereaktion beeinträchtigen. Die exakte Dosiergenauigkeit und Dosiergeschwindigkeit der Spritzenpumpen bei Zugabe der Zuckerbausteine, sowie der Aktivierungs- und Entschützungsreagenzien entsprechend der Synthesevorschriften, stellte eine weitere steuerungstechnische Anforderung dar. Zur Vermeidung von Kreuzkontaminationen und daraus resultierender Nebenreaktionen waren konstruktive und softwareseitige flexible Lösungen zum Spülen des Gesamtsystems bzw. von Teilsegmenten erforderlich. Dies galt auch für das automatische Spülen der Reaktanten im Reaktionsgefäß, sowie das Ausleiten nicht gekoppelter Substanzreste in den Abfallbehälter. Rein softwareseitig galt es eine effiziente Datenverwaltung der Synthesen und deren Reaktionsprodukte in einer sinnvollen Datenstruktur zu Dokumentationszwecken und als Systemressource aufzubauen.

4.1.2 Aufbau des Gesamtsystems



Abbildung 67: Schematischer Aufbau des neuen Arbeitssynthesizers mit dezentraler Steuerung durch den Automatisierungsrechner und Standardsystemkomponenten.

Der Betrieb des Syntheseautomaten findet unter Argon Schutzgasatmosphäre statt. Alle in die Apparatur eingebrachten Glasgefäße für die Zuckerbausteine, Lösungsmittel, Reagenzien müssen deshalb vorher mit Lösungsmittel gespült, ausgeheizt und mit eigener Schutzgasfüllung versehen werden. Die Lösungsmittel sind vor Verwendung im Synthesizer durch Absolutierung von Wasser zu befreien. Die Anlage wird mittels Druckverteiler unter Überdruck betrieben. Rückschlagventile verhindern die Kontamination der Zuleitungen durch Substanzen, die aufgrund von Gasentwicklung aus dem Reaktionsgefäß zurückgedrückt werden könnten. Die benötigten Drücke sind bei der ventilgesteuerten Zuführung durch die Spulenventile aufgrund unterschiedlicher Viskositäten von Reagenzien und Lösungsmitteln individuell anzupassen, um jeweils gleiche Volumina pro Zeiteinheit bereit zu stellen.



Druckverteiler

Rotationsventile

Spritzenpumpe

DIO Box

Abbildung 68: Einbaupositionen der wichtigsten Synthesizer-Komponenten am Neubau des Arbeitssynthesizer III

Die gesamte Verschlauchung innerhalb des Synthesizers besteht aus PTFE Schläuchen. Diese sind ebenso wie Peek und FEP chemisch resistent gegenüber den verwendeten Lösungsmitteln. Der Transport exakt zu dosierender Reaktanten und Zuckerbausteine erfolgt mit 1/16" Schlauchdurchmesser. Bei der Lösungsmittelzuführung über die Spulenventile, sowie des Schutzgases, kommen 1/8" Schläuche zur Anwendung.

Elektrische Kabel innerhalb des Synthesizers wurden mit Teflonschläuchen ummantelt und deren Enden mit Teflonschrumpfschläuchen versiegelt, um die elektrischen Leitungen im Falle einer Fehlbedienung vor Lösungsmittelkontakt zu schützen. Metallbereiche, die durch Lösungsmittel beeinträchtigt werden können, wurden mit selbstklebenden Teflonfolien versiegelt. Der Syntheseautomat wird unter einer PE-Haube betrieben, der über ein Stellventil an den Gebäude-Abzug angeschlossen ist und damit den Apparat von möglicher Gasentwicklung befreit. Die Lösungsmittel sind in einem separaten Teilbereich des Synthesizers untergebracht.

Nach dem zwischenzeitlichen Umzug der Arbeitssynthesizer in neue Räumlichkeiten wurden sie den dort vorhandenen begehbaren Abzügen angepasst und auf/umgebaut. Der Betrieb findet nun vollständig hinter Schutzglas statt. Am neuen Standort werden die Lösungsmittel in einem externen Lösungsmittelschrank gelagert. Damit ist die Einhaltung der Laborrichtlinienverordnung^[171] bezüglich der erlaubten Gebindegröße, sowie erweiterte Schutzbestimmungen wie die elektrische Erdung gewährleistet. Auch für den Feuerfall sind aufgrund von selbstverlöschenden Rohren und Schutz gegen Wärmeeintrag die aktuellen Sicherheitsbestimmungen umgesetzt.



Abbildung 69: Gesamtrahmenaufbau vom Arbeitssynthesizer III mit lokaler mobiler integrierter Werkstatt

Der fertige Syntheseapparat setzt sich aus ca. 500 Einzelteilen zusammen und erforderte umfangreiche konstruktive und elektronische Aufbauarbeiten. Aufgrund der vorgesehenen Verwendung oder der gewünschten Positionierung im System, mussten eine Vielzahl von Einbaukomponenten zusätzlich mechanisch bearbeitet werden. Damit die zahlreichen konstruktiven mechanischen Anpassungen, elektronische Verschaltungen, sowie anfallende Wartungsarbeiten, wie Wechsel oder Reinigung von Filterelementen und Rückschlagventilen, schnell und effizient vor Ort umgesetzt werden konnten, ist unterhalb des Synthesizers eine mobile Werkbank mit entsprechendem Werkzeug integriert (Abbildung 69).

Hinter der Werkbank befindet sich bei diesem Arbeitssynthesizer III der Steuerungs-PC, das Auffanggefäß der Lösungsmittelabfälle, sowie die Überlaufleinheit.

4.1.3 Transport von Lösungsmitteln und Zuckerbausteinen

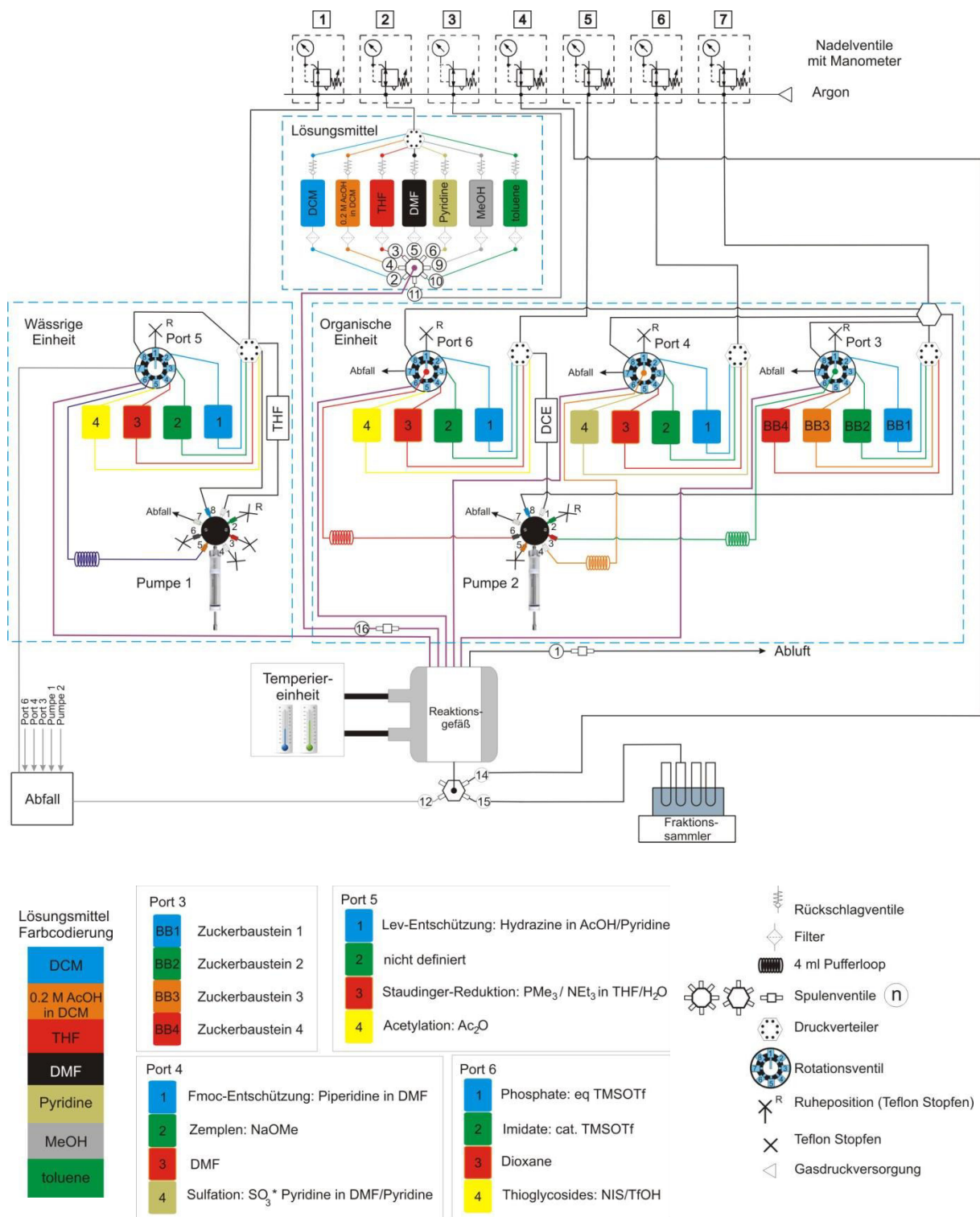


Abbildung 70: Flüssigkeitstransport von Lösungsmitteln, Reagenzien und Zuckerbausteinen im Synthesizer.

In Abbildung 70 ist das Gesamtsystem der Transportleitungen für Zuckerbausteine, Lösungsmitteln, Entschützungs- und Aktivierungsreagenzien, sowie der Druckluftleitungen dargestellt. Die Anschlüsse der Transportleitungen wurden farblich codiert, um Verwechslungen der Verschraubungen beim Einbringen der Substanzen möglichst auszuschließen.

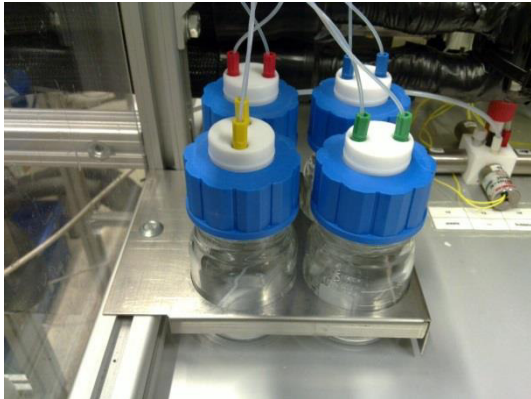


Abbildung 71: Farbcodierung zur vereinfachten Zuordnung der Flüssigkeitsführung an den Vorratsflaschen der Aktivatoren und den Entschützungsreagenzien

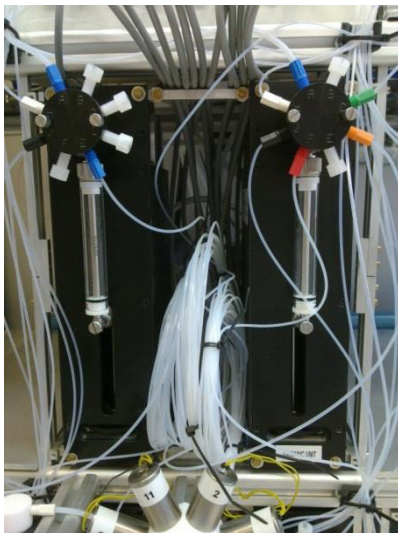
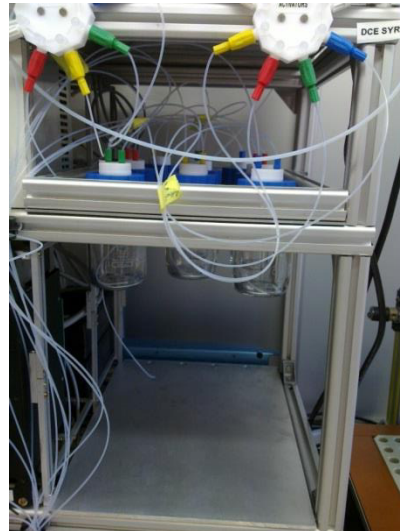


Abbildung 72: Pufferloops der spritzenbasierten Flüssigkeitszufuhr und die dazu notwendigen Kloehn-Pumpen

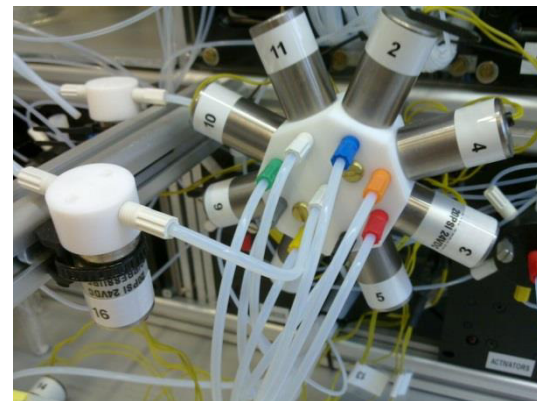


Abbildung 73: Ventilkonzept bei druckbasierter Lösungsmittel-Zuführung (Spulenventil) mit Zentralverschluss (Ventil 16) des Reaktors

Um Kontaminationen der Spritzenpumpen mit den jeweilig transportierten Lösungen zu vermeiden, wurden Pufferloops eingebaut. In diesen Schlauchabschnitten werden die zu dosierenden Substanzen für die Synthesereaktion zwischengespeichert. Diese Loops sind mit 4 mL so zu bemessen, dass sie mit einem vollständigen Spitzenvolumen (bei 5 mL Spritzen) sicher gespült werden können. Zusätzlich kann bei der Befüllung durch eine entsprechende Ventilschaltung eine Gasblase als Trennschicht zwischen Transportlösung und Reagenz eingebracht werden, die eine direkte Kontaktierung verhindert.

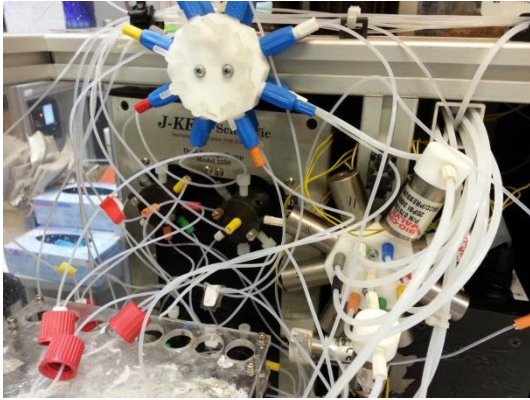


Abbildung 74: Flüssigkeitsführung für Bausteine

Die Glasgefäße mit den Zuckerbausteinen (Buildingblocks) wurden in direkter Nähe zu den Rotationsventilen positioniert. In den Transportleitungen verbliebene Restmengen an Zuckerbausteinen werden vor dem Spülen der Schläuche zurück in die jeweiligen Gefäße befördert. Aufgrund des ökonomischen Umgangs mit den Reaktanten, durch möglichst gering gehaltene Transport-Volumina, konnte der systembedingte Verbrauch auf ein Minimum reduziert werden.

4.2 Aufbau und Entwicklung der Komponenten Hard- und Software

4.2.1 Ventilsteuerung

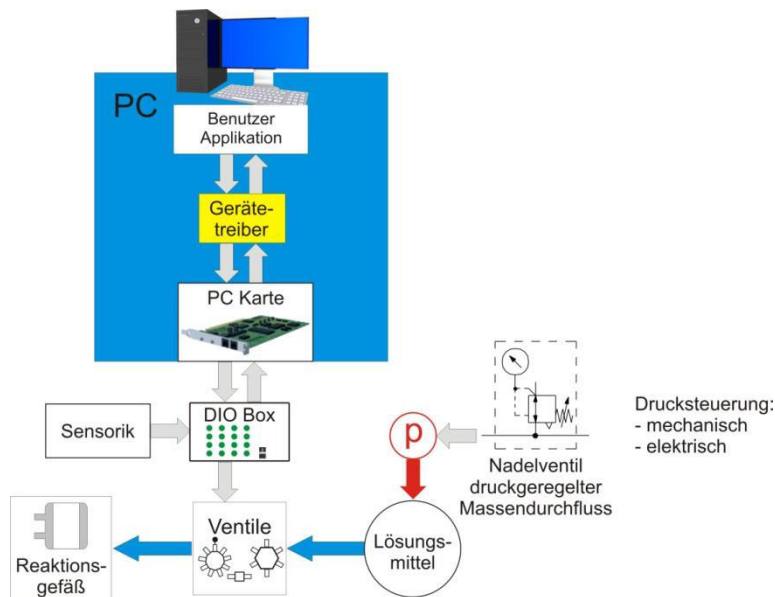


Abbildung 75: Schematischer Aufbau der Ventilsteuerung zum Lösungsmitteltransport. Die Schaltung der Spulenventile erfolgt über den Ventil-Gerätetreiber mit Hilfe einer PC-Karte, sowie einer sich anschließenden Verstärkerbox (DIO Box).

Der Lösungsmitteltransport im System erfolgt druckbasiert über Spulenventile, welche die Steuerung der Transportwege ins Reaktionsgefäß bzw. den Abfallbehälter ermöglichen. Die elektrischen Ventile ermöglichen die schnelle Zulieferung der Substanzen in den Reaktor. Aufgrund der bei Raumtemperatur gelagerten Lösungen, kommt es bei der ventilgesteuerten Flüssigkeitszuführung im Reaktor zu einer lokalen Temperaturänderung. Der Verfahrensaufwand für zwei voneinander unabhängige Flüssigkeitssteuerungen ist durch das modulare Konzept überschaubar.

Die Ventile benötigen wegen der im Labor verwendeten Lösungsmittel eine vollständig mit Teflon

ummantelte Funktionseinheit. Aufgrund der Lebensdauer und der möglichen chemischen Anwendungsbereiche, wurden die Spulenventile der Firma BIO-CHEM ausgewählt.



Abbildung 76: Die lösungsmittelresistenten, vollständig mit Teflon ummantelten Spulenventile ermöglichen die Steuerung des schnellen Flüssigkeitstransports im Synthesizer

Im neuen Syntheseautomat übernimmt eine Digital I/O PC-Karte von National Instruments (*NI -PCI 6519*) die Steuerung der chemischen Ventile. Diese Karte ist im Automatisierungs-PC des Synthesizers zur Bereitstellung der Steuergrößen eingebaut und wird vollständig von dem Datenerfassungs- (DAQmx¹⁷) Treiber durch LabVIEW unterstützt. Die ausgewählte PC-Karte verfügt sowohl über 16 Eingänge als auch 16 galvanisch getrennte Ausgänge (mit 5 V und 125 mA je Kanal). Die Eingänge sind aktuell noch nicht vollständig belegt und werden für die zukünftige Auswertung von Sensordaten, z. B. zur Füllstandsüberwachung der Lösungsmittelgefäße, vorgehalten. Zur Schaltung benötigen die Spulenventile eine 24 V Spannungsversorgung mit 100 mA Stromspeisung. Da die Digital I/O Karte nur 5 V liefert, wurde mit der DIO-Box eine entsprechende Verstärkerbox realisiert. Diese stellt, mittels Trenntrafospeisung und Optokoppler, 16 getrennt galvanische Kanäle zur Verfügung.



Abbildung 77: DIO Box Prototyp für die Arbeitssynthesizer I + II mit Spezialsteckeranschluss zur Ventilsteuerung. Auf der Geräterückseite finden sich die Anschlussbuchsen für das Schnittstellenkabel zum PC, Spannungsversorgung, sowie Status-LEDs zur lokalen Kontrolle der aktiven Ventilschaltung.

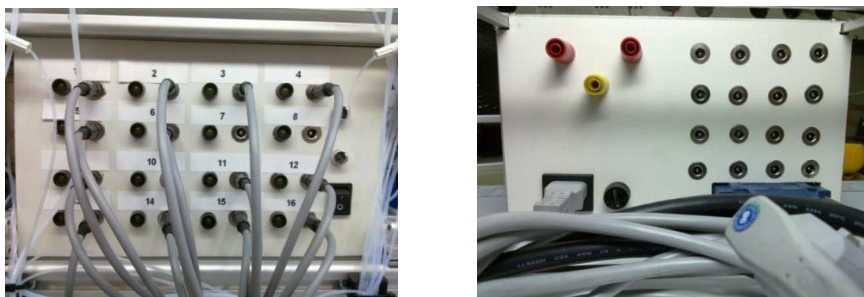


Abbildung 78: DIO Box für den Arbeitssynthesizer III. Anschlüsselemente für Spulenventile auf der Vorderseite sind durch Standardniederspannungsstecker ersetzt und mit Status-LEDs und Hauptschalter komplettiert. An der Rückseite finden sich die PC – Anschlüsse, Spannungsversorgung(5V, 24V), sowie und Netz-Sicherung.

¹⁷ DAQ = engl. data acquisition

Bei der Auswahl aller erforderlichen Komponenten wurden etablierte Industriestandards bevorzugt, um die schnelle, kostengünstige Austauschbarkeit der Bauteile zu gewährleisten.

Der Anschluss an die NI-Schnittstellenkarte erfolgt über einen 25-poligen D-Sub-Stecker. Die DIO Box nutzt einen Standard Kaltgerätestecker für den 220 V Anschluss. Lediglich für die Niedervoltverbindung der ersten beiden DIO-Box-Prototypen kommt, aus Kompatibilitätsgründen zu den Vorgänger-Systemen, noch der dort verwendete Spezialstecker zum Einsatz. Dieser wird in der DIO-Box für den Arbeitssynthesizer III durch Standard-Niederspannungsstecker ersetzt.

Da die intern verbauten Niederspannungstrafos eine größere Leistung bereitstellen, als zur maximalen Ventilansteuerung benötigt wird, befinden sich auf der Rückseite der DIO Box eine 24 V / 5 V Versorgung mit GND¹⁸ zur individuellen Verwendung für den nachträglichen Einbau von Sensorik (z. B. Schalt- und Sicherungselemente), sowie die digitalen Eingänge der Erfassungskarte und die Kontaktierungsbuchsen zur NI-Karte. Auf der Vorderseite der DIO Boxen befinden sich die Ventil-Anschlüsse, sowie über die LEDs eine lokale Funktionsanzeige der jeweiligen Schaltzustände. Die Verstärkerbox des Arbeitssynthesizer III (Abbildung 79) wurde direkt für den Einbau in ein 19“-Industrierahmen-Standardsystem als platzsparende Gerätevariante konzipiert. Der Netzschalter befindet sich hier an der Vorderseite, damit er auch nach dem Einbau im Rahmensystem bequem erreichbar bleibt. Nach Fertigstellung der DIO – Boxen wurde das Schaltverhalten jeder Ansteuerungsschaltstelle geprüft und ausführlich getestet. Die Geräte-LEDs signalisieren dem Anwender direkt den aktuellen Schaltzustand der entsprechenden Ventile und dienen so der effizienten Fehlersuche.

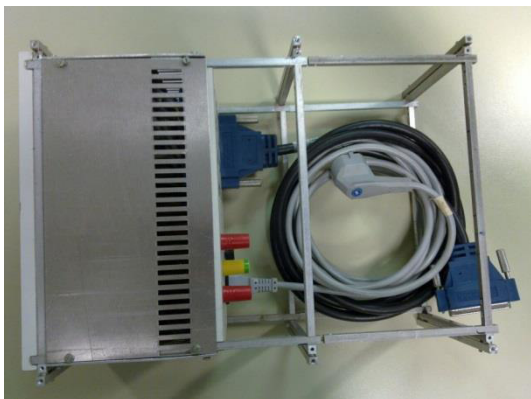


Abbildung 79: DIO Box für Arbeitssynthesizer III als modulare Komponente für den 19“ Einbaurahmen in der Draufsicht.

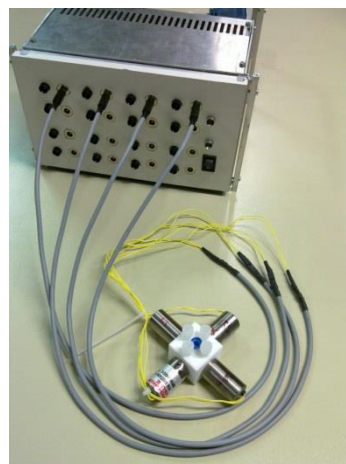


Abbildung 80: Funktionskontrolle an einem „Vierer“- Spulenventil.

¹⁸ GND =engl. engl. chassis ground, bezeichnet das Bezugspotential (i. d. R. Potential = 0V) für alle Signal- und Betriebsspannungen.

Die Spulenventile werden im Betrieb jeweils einzeln über den Gerätetreiber gesteuert. Der Gerätetreiber übersetzt die Anforderungen des Gesamtsystems in Handlungsanweisungen an den Datenerfassungs – Treiber (DAQ) in LabVIEW, der damit die eingebaute Digital I/O-Karte steuert. Diese übernimmt die Umwandlung der digitalen Daten in ein analoges Steuersignal, welches durch die Verstärkerbox auf die notwendige Schaltspannung von 24 V transformiert wird. Die dazu notwendige DAQ-Treiber Softwareinstallation ist im *Anhang 1.1.* beschrieben.



Abbildung 81: Die Ventilsteuerung erfolgt mittels Gerätetreiber über eine Digital I/O Karte. Die Karte kann in LabVIEW direkt über die Treibersoftware angesprochen werden. Das Signal am Datenausgang der Karte muss für die Ventile auf 24 V verstärkt werden und spricht damit die Aktorik an. Die PC-Karte ist ebenfalls in der Lage parallel Messdaten weiterer Sensorik aufzunehmen.

Die Ventilschaltung über die Datenerfassungskarte, die neben der Aktorik auch die Signalerfassung ermöglicht, bietet ausreichend Komplexität, um an dieser Stelle exemplarisch die prinzipielle Vorgehensweise bei der Umsetzung eines Hardwarezugriffs zu erläutern, wie er für alle realisierten Gerätetreiber und auch für die Software der Modulerweiterungen zur Anwendung kam.

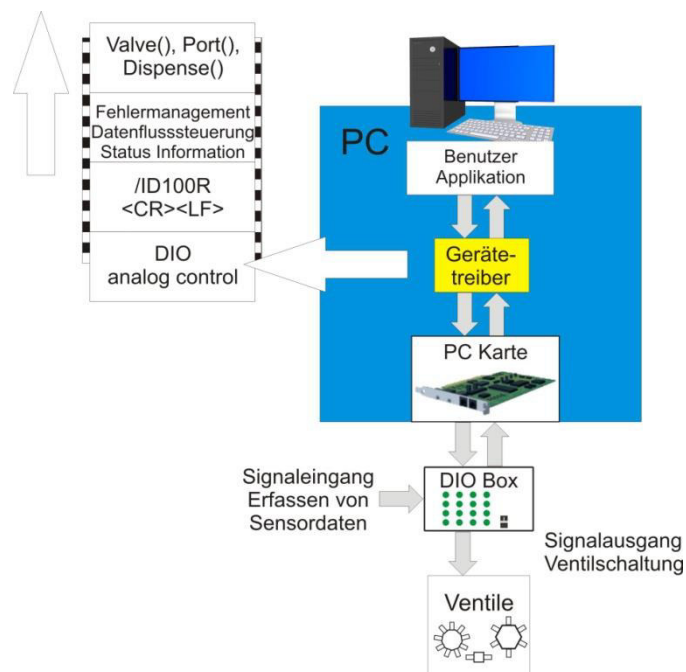


Abbildung 82: In der Gerätetreiberentwicklung für die Spulenventile wird die unterste Abstraktionsebene der Sprachentwicklung für den Synthesizer administriert: binärer, hardwarekonformer Steuercode.

Die Spulenventile werden aufgrund eines digitalen Zugriffs geschaltet und können die Zustände „An“ und „Aus“ annehmen. Diese Funktionalität lässt sich in der Programmierung auf unterschiedliche Weise umsetzen.

Im einfachsten Fall wird mit der Betätigung einer Schaltfläche in LabVIEW der Status der zugeordneten LED geändert. Da die Schaltfläche den Datentyp Boolesche Werte¹⁹ (true, false) ausgibt, kann diese auch direkt mit dem LED Symbol in LabVIEW kontaktiert werden, welches am Eingang ebenfalls Boolesche Werte akzeptiert. In der virtuellen Welt von LabVIEW funktioniert dies durch direktes „kabelorientiertes“ Übergeben der zugeordneten Werte. Dies erfolgt im Normalfall datenstromorientiert, d. h. durch Verbindung mit einem virtuellen Kabel. Sonderfälle hierzu sind lokale oder globale Variablen, sowie funktionale Variablen oder andere dem Datenstromkonzept nicht in erster Instanz entsprechende Datenaustauschprozesse. Ein realer Hardwarezugriff erfolgt damit jedoch nicht.

Der einfachste Programmierweg zum Hardwarezugriff ist die Verwendung des entsprechenden Express-VIs: der DAQ – Assistent. Die Express VIs haben allgemein den Vorteil durch benutzerinteraktiv geführte Dialoge auch sporadischen Nutzern komplexe Parametrierungssituationen sicher zugänglich zu machen. Hierzu werden im geführten Benutzerdialog Portzugriffe (d. h. Zugriff auf das Gesamtgerät, also aller digitalen Ein- und Ausgänge der Datenerfassungskarte) auf die digitale Steuer-Einheit definiert, die eine ereignisbasierte Signalausgabe ermöglichen. Auch die Simulation von Hardware ist möglich. Die Programmierung lässt sich damit vollständig auch an einem Entwicklungsrechner durchführen und testen, der keinen direkten Hardwarezugriff besitzt. Aus den vorgenommenen Einstellungen wird ein Laufzeit-VI erzeugt, welches sich dann nach den Vorgaben den Anschlussterminal anpasst.

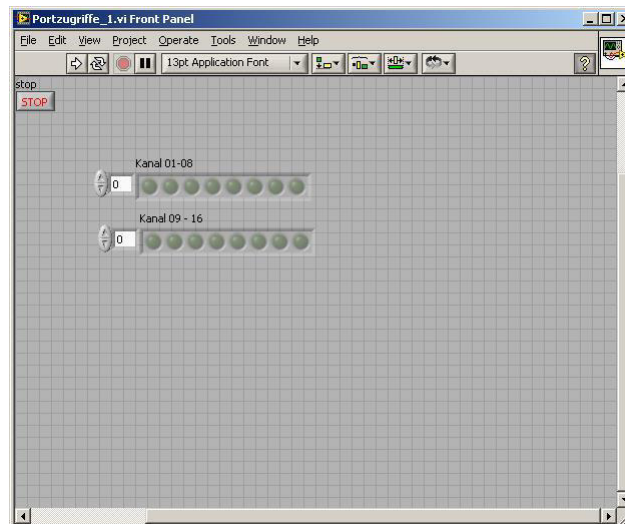


Abbildung 83: Frontpanel mit 16 LEDs, die über ein Array verwaltet werden. Jede LED kann direkt ausgewählt und auf 0 oder 1 bzw. „An“ oder „Aus“ gesetzt werden.

Statt die LEDs für die 16 Ausgänge einzelnen zu steuern, verwaltet ein LED-Array in der Frontpaneldefinition mit einer Stopp-Schaltfläche als Bedienterminal die Schaltzustände aller

¹⁹ Der Datentyp Boolescher Wert kann nur zwei zulässige Werte annehmen: true (wahr) oder false (falsch) und bezeichnet damit, ob ein Sachverhalt wahr ist oder nicht.

Ausgänge. Die LEDs können einzeln ausgewählt und ihr Zustand über die Eingabe-Elemente „An“ oder „Aus“ geschaltet werden (logisch 0 oder 1, false / true) gesetzt werden.

Die Auswertung der Schaltzustände im dazugehörigen Blockdiagramm (Abbildung 84) erfolgt in diesem Beispiel durch Addition aller Binärwerte des Gesamtsystems. Sind also alle 8 Digital-I/Os einer Reihe auf „An“ geschaltet, entspricht dies dem Binärwert: 11111111

$$1 * 2^0 + 1 * 2^1 + 1 * 2^2 + 1 * 2^3 + 1 * 2^4 + 1 * 2^5 + 1 * 2^6 + 1 * 2^7 = 255$$

bzw. HEX- Wert: FF.

Die Auswertung des LED-Arrays erfolgt durch eine For-Schleife, welche die Werte ereignisorientiert aus dem Eingabe – LED-Array herausliest. Dazu wird der Datentyp Boolescher Wert (true, false) in einen Binärwert (0 und 1) und anschließend in einen numerischen Wert (U32²⁰) umgewandelt. Alle Werte des Arrays werden aufaddiert und parallel im Anzeige-Element visualisiert, sowie dem DAQ – Assistenten übergeben. Dort wird der aktuelle LED-Status als Hardwarezustand an die DIO-Box (Verstärker) ausgegeben und die entsprechenden Ventile bestromt bzw. abgeschaltet.

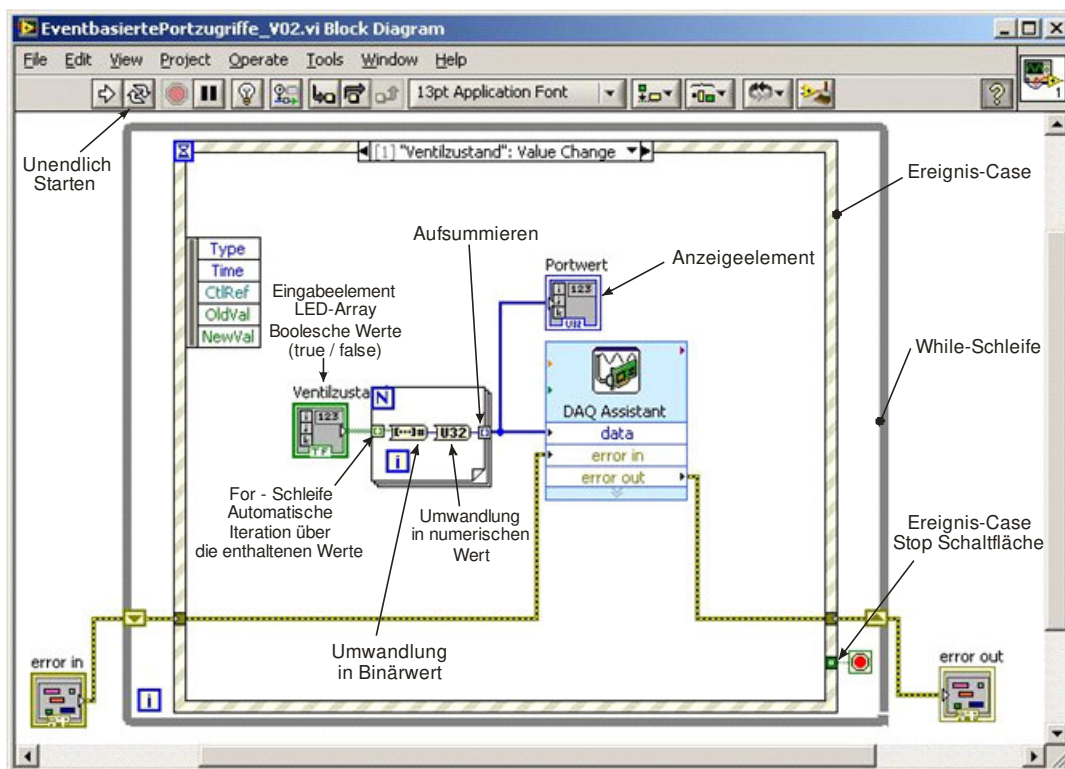


Abbildung 84: Blockdiagramm zur ereignisbasierten Anzeige der Ventilzustände mittels DAQ Assistent. Die Zuordnung der einzelnen LED-Schaltzustände erfolgt durch Addition der Binärwerte des Gesamtsystems. Dieser Wert wird an den DAQ Assistenten übergeben, der die aktuellen Werte des Bedien-LED-Array herausschreibt.

²⁰ Datentyp „UNSIGNED 32“ bezeichnet vorzeichenlose, ganze Zahlen mit einem Wertebereich von 32 Bit.

Am Blockdiagramm zeigt sich der minimal notwendige Programmieraufwand für einen Hardwarezugriff, mit dem sowohl ein einfaches Error-Management als auch die ereignisorientierte Abarbeitung der Änderungen am Eingabeterminal Rechnung getragen wird. Da hierbei das Express-VI DAQ – Assistent zur Anwendung kommt, ist die hardwarenahe Programmierung vereinfacht. Der DAQ – Assistent initialisiert die Hardware, konfiguriert und terminiert, arbeitet jedoch als Einbahnstraße. Um den aktuellen Zustand des LED-Arrays zu erfragen, muss eine unendliche Schleife ausgeführt werden. Dies kann zwar in der Entwicklungsumgebung über die Schaltfläche „unendlich Starten“ geschehen. Damit existiert jedoch kein definiertes Timing. Deshalb ist es sinnvoll, um die Programmierung eine While - Schleife mit einem definierten Abbruchkriterium (z. B. eine Schaltfläche Stopp) zu setzen und dieser Schleife ein eindeutiges Timing zuzuweisen. Dabei nimmt der Benutzer Interaktionen, die im Zeitbereich von 100 ms bearbeitet werden als Echtzeit wahr.

Eine ereignisbasierte Programmierung verzichtet auf das künstlich aufgeprägte Timing, sondern reagiert auf Veränderungen von Seiten des Benutzers mit einem Ereignis-Case²¹. In der Ereignisstruktur des VIs gibt es für die unterschiedlichen Objekte innerhalb des Ereignis-Case Ereignisquellen, auf die reagiert werden kann. So enthält ein Schaltflächen-Objekt z. B. Tasten-, Maus-, Drag & Drop Ereignisse, sowie Kontextmenü- bezogene Ereignisse. Meistens werden Wertänderungen zur Ereignisdefinition benutzt. Es existieren aber auch Objekte in der Ereignisstruktur, die nicht vom Benutzer programmiert wurden, z. B. Ereignisquellen in Applikationen, VIs oder Fensterbereichs-Instanzen. Dabei sind beliebige Ereignis-Strukturen stapelbar. Es wird immer nur der Ereignis-Case abgearbeitet, der von dem Ereignis angesprungen wurde.

Solange die Ereignisstruktur arbeitet, wird der Rest des VIs nicht bearbeitet. Auch für die Stopp-Schaltfläche muss eine Ereignisstruktur vorgesehen werden. Wird als Ereignisstruktur „Standard verwenden, wenn nicht verbunden“, gilt dieses Ereignis (z. B. Stopp Schaltfläche als Abbruchbedingung) für beliebig viele Ereignisstrukturen, die übereinander liegen, ohne dabei explizit definiert zu werden. Die Ereignis-Abarbeitung erfolgt schnellstmöglich. Tritt ein Ereignis ein – also z. B. die Änderung des LED – Zustandes – erhält das Ereignis höchste Priorität und die dem Ereignis zugeordnete Programmierung (auch der Aufruf eines Sub-VIs) wird abgearbeitet.

Express-VIs eignen sich, um z. B. schnell und einfach Benutzerschnittstellen (Bedienterminal– Assistent) zu programmieren oder einen rudimentären Hardwarezugriff (DAQ-Assistent) zu ermöglichen. Hinter jedem Express-VI liegt jedoch immer eine hoch komplexe Zustandsmaschine, die aufgrund ihrer Möglichkeiten die maximale Komplexität in allen, dem Anwendungsbereich des Express – VIs, umschließenden Funktionen vorhält. Damit liegt weit mehr Funktionscode als für die

²¹ In der ereignisorientierten Programmierung enthält ein Ereignis-Case ein oder mehrere Unterprogramme, die nach Eintritt z. B. eines Ereignisses auf dem Frontpanel (dem übergeordneten Programm VI) ausgeführt werden.

aktuelle Aufgabe Verwendung findet hinter dem VI und bindet Ressourcen, wodurch sich die Abarbeitungsgeschwindigkeit in komplexen Applikationen wie der Synthesizer-Software reduziert. Für zeitnahe und ressourcenoptimierte Programmierung sollte der Funktionsumfang deshalb auf die wirklich notwendigen Elemente reduziert werden.

Alternativ dazu kann mit einem nahezu vom Frontend²² vergleichbaren Konstrukt der Zugriff auf eine gesamte Reihe (Line) digitaler Ausgänge realisiert werden. Jede Reihe des Ports besteht aus 8 Ausgängen. Hierbei kommt anstelle des DAQ Assistenten jedoch die individuelle Verwendung der DAQmx Funktionalitäten zur Anwendung.

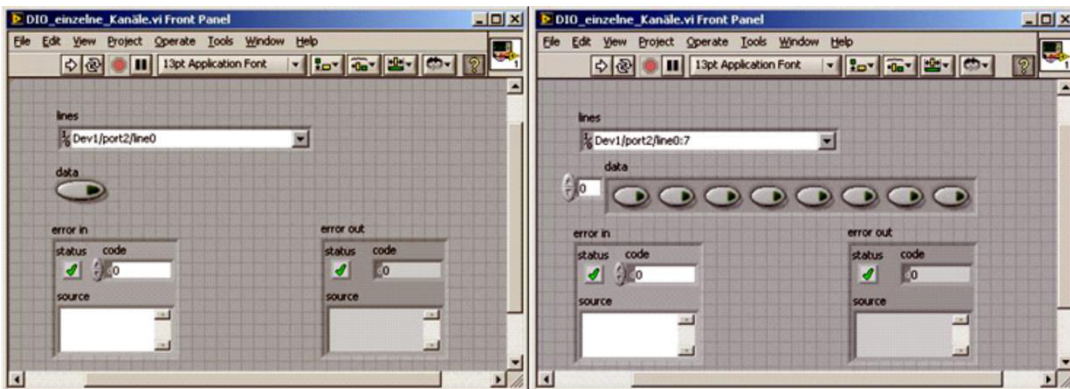


Abbildung 85: Auf digitale Ausgänge wird reihenweise (immer 8 Kanäle gleichzeitig) zugegriffen. Im Frontpanel links lässt sich nur die gesamte Reihe (lines) „An“ und „Aus“ schalten. Am Frontpanel rechts können nach Selektion der Reihe auch die Einzelkanäle geschaltet werden.

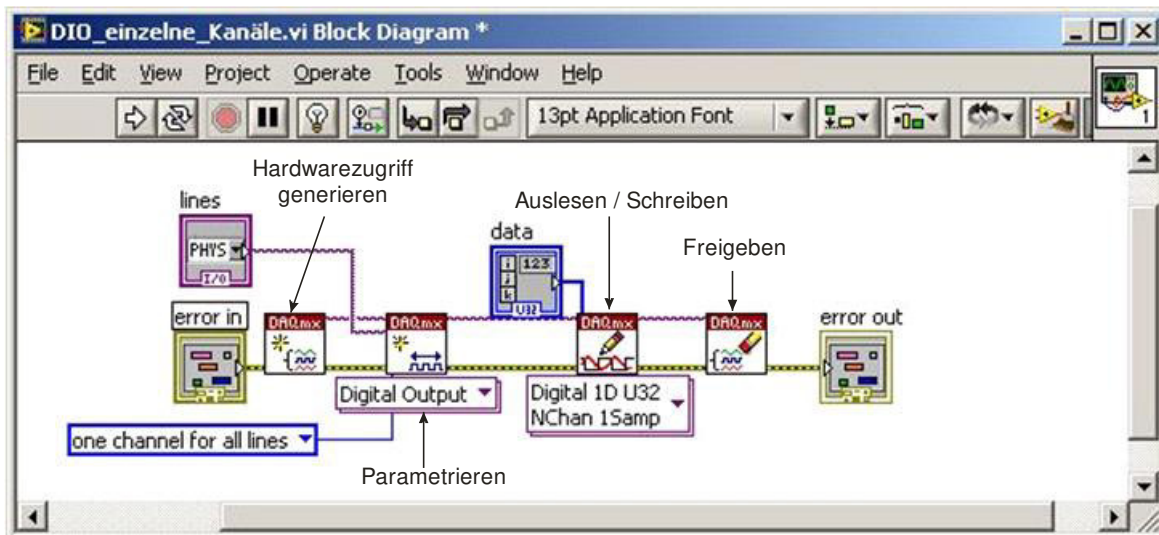


Abbildung 86: Aus dem Blockdiagramm lässt sich die funktionale Schrittkette beim Hardwarezugriff ableiten. Die Programmierung hinter den beiden Benutzeroberflächen aus Abbildung 85 ist dieselbe.

Für den Hardwarezugriff wird zunächst die Hardware für die Aufgabe reserviert. Es folgt die Aufgabendefinition bzw. Parametrierung zum Schalten der Digital Ausgänge. In Abhängigkeit ob mit

²² Basierend auf dem Schichtenmodell der Softwarearchitektur, bezeichnet Frontend in der Informationstechnologie den Bereich, der dem Benutzer zugewandt ist. Es ist eine Software-Anwendung, mit der dessen Benutzer direkt interagieren kann.

Reihen (Lines) oder Einzelkanälen gearbeitet wird, verändert sich die entsprechende Verarbeitungsroutine, die in den Sub-VI'S hinterlegt ist. Die aktuell ausgelesenen Werte werden auf die realen digitalen Ausgänge geschrieben. Nach Terminierung der Aufgabe wird die Hardware freigegeben, bis ein erneutes „Änderungs“- Ereignis von Seiten des Benutzers eintritt.

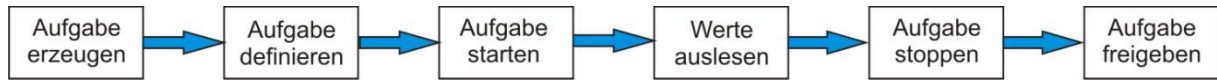


Abbildung 87: Schrittkette bei einem Hardwarezugriff

Diese Schrittkette bleibt immer gleich, auch wenn das Frontpanel ein weiteres Mal überarbeitet und nun z. B. mit Schaltern betrieben wird. Der veränderte Datentyp, im Fall der Schalter der Boolesche Wert, wird einfach dem polymorphen Sub-VI²³ (siehe Kapitel 3.2.1, Seite 64) als Datentyp hinzugefügt und eine entsprechende Verarbeitungsroutine hinterlegt. Das VI erkennt selbstständig, welcher bekannte Datentyp angelegt wurde und wechselt auf die entsprechende Verarbeitungsroutine, die in Form eines Stapels am Objekt abgelegt ist.

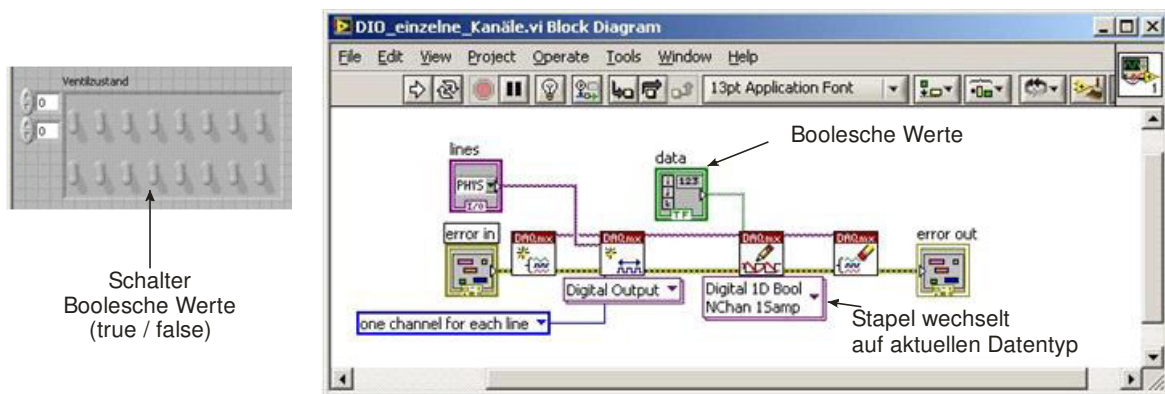


Abbildung 88: In LabVIEW programmierte Benutzeroberflächen lassen sich mit Ein- und Ausgabeelementen ohne Veränderung des funktionalen Programmiercodes beliebig anpassen. Für unterschiedliche Datentypen lassen sich verschiedene Verarbeitungsroutinen am Element hinterlegen (stapeln).

Eine der Anforderungen an die neue Synthesizer-Applikation war der Anwenderwunsch, auch weiterhin mit den bestehenden Ansteuerungsprotokollen des Vorgänger – Synthesizers arbeiten zu können. Hierzu muss jedoch durch Reaktion auf individuelle Kanäle (Channels) eine Stufe detaillierter auf die digitalen Ausgänge Zugriff genommen werden.

²³ Polymorphe VIs passen sich verschiedenen Datentypen an und enthalten mehrere VIs mit gleich aufgebautem Anschlussfeld. Dabei ist jedes VI eine Instanz des polymorphen VIs und jede Instanz enthält mindestens einen Eingangs- oder Ausgangsanschluss, der mit einem anderen Datentyp, als die anderen Instanzen arbeitet.

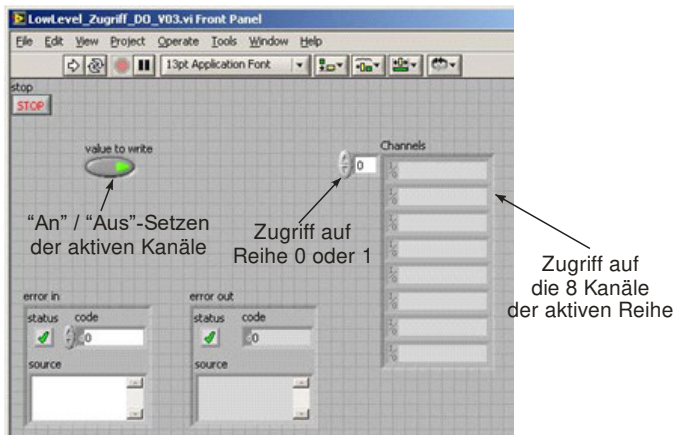


Abbildung 89: Einfacher Zugriff auf die digitalen Kanäle einer Reihe. Jede Reihe besitzt acht Kanäle, die einzeln mit dem Schalter „An“ oder „Aus“ geschaltet werden können

Hierbei wird ereignisorientiert (Blockdiagramm in Abbildung 90) jeweils nur die individuelle Änderung auf den initialisierten Geräte-Tasks²⁴ geschrieben, der innerhalb der Schleife aktiv offengehalten und erst bei der abschließenden Betätigung der Stopp Schaltfläche terminiert wird. Damit sind alle für den Hardwarezugriff des Spulenventil-Gerätetreibers notwendigen Programmierschritte erklärt und die rudimentäre Hardware-Funktion hergestellt.

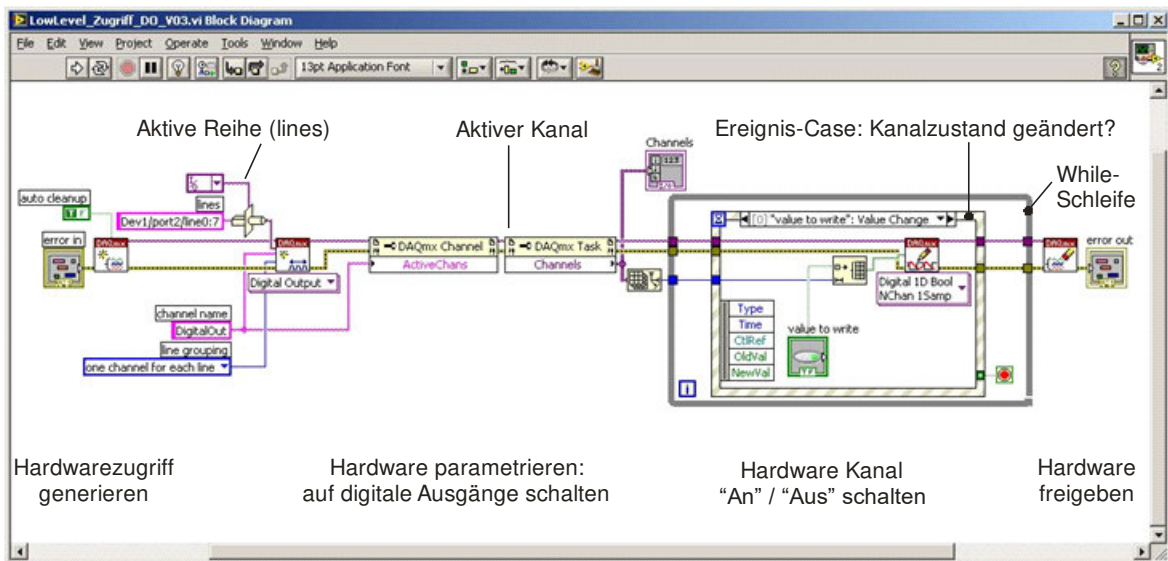


Abbildung 90: Blockdiagramm für den einfachen Hardwarezugriff auf die digitalen Kanäle der Datenerfassungskarte.

²⁴ Task bezeichnet in der Informatik einen Prozess. Dies ist die Ausführung eines Programmes (Instanzierung) auf einem Prozessor und beinhaltet Ressourcenzuteilungen und weitere ergänzende Informationen zur Programmausführung.

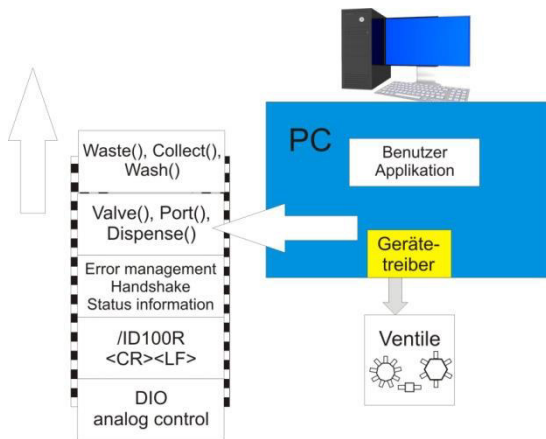


Abbildung 91: Für die aktuellen Gerätetreiber wird in der Software ein Interpretations-Mechanismus etabliert, der die Synthese-Ansteuerung der Vorgänger-Geräte weiter nutzbar macht. Diese Sprachabstraktion findet bereits auf einer höheren Abstraktions- bzw. Interpretationsebene statt.

Damit die Befehle des Altsystems (vgl. Abbildung 28, S. 57) auch im neuen Syntheseautomaten verwendet werden können, müssen die Skripte von der aktuellen Synthesizer-Applikation interpretiert, plausibilisiert und die Funktion an die jeweiligen Hardwaremodule übergeben werden.

Historisch erfolgte der Befehlsaufruf zum Schalten eines Ventils in dieser Form:

„VALVE(1, OFF)“

Der Befehl adressiert die Ventile über einen numerischen Parameter (1 – 16) und dessen Wunschzustand in Form von „ON“ / „OFF“. Dieser Befehl ist auszulesen und die drei Objekte (Befehl, Ventilnummer, Zustand) sind voneinander zu trennen.

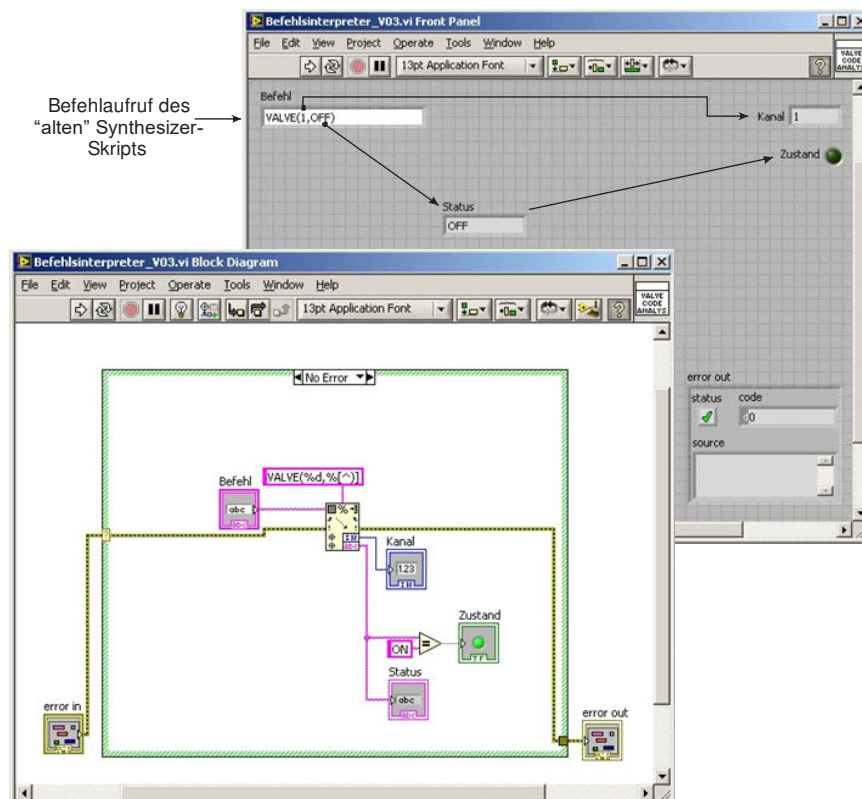


Abbildung 92: Realisierung eines einfachen Befehlsinterpreters. Der vorliegende Befehl wird in Kanalnummer und Status geteilt. Der Textstring „Status“ wird in einen Binärwert umgewandelt und als LED angezeigt.

Am einfachsten lässt sich der übergebene Befehls-Textstring durch Suchen nach entsprechenden Zeichenkombinationen auslesen. Exemplarisch wird dies an einem einfachen regulären Ausdruck²⁵ gezeigt. Der Kanal und dessen Status werden ausgekoppelt und entsprechend der textlichen Information in einen binären Booleschen Wert (true / false) umgewandelt, welches im Frontpanel als LED visualisiert wird. Der Boolesche Wert ist an dieser Stelle zu bevorzugen, da er der zuvor gezeigten hardwarenahen Ansteuerung entspricht. Reine Textinformationen (ON/OFF) hingegen besitzen keine EDV-technische Eindeutigkeit.

Da bei der Ansteuerung eine sequenzielle Abarbeitung von Zuständen benötigt wird, erfolgt im nächsten Schritt die Befehlsvorgabe mittels eines Array²⁶. Die einzelnen Zustände werden über eine Schleifensteuerung zur Ausführung gebracht.

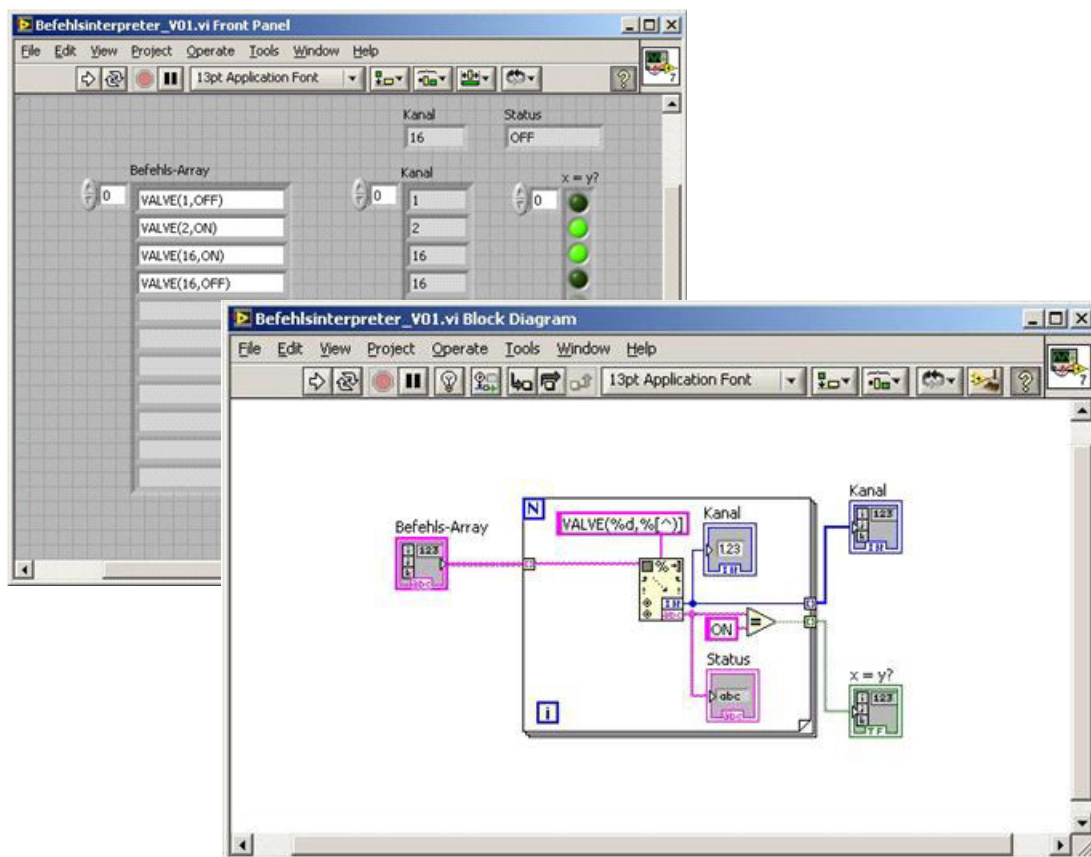


Abbildung 93: Befehlsinterpretierer mit Befehlsvorgabe und Abarbeitung durch ein Array. Die Zustandsänderung für die einzelnen Ventile erfolgt sich über eine Schleifensteuerung.

Um das Problem der Befehlsinterpretation allgemeingültig für alle mit dem Altsystem kompatiblen Gerätekomponten zu lösen, werden für komplexe Anweisungslisten zusätzlich Zeitsteuerungen und Rückkopplungen von Informationen ausgewertet, sowie statistische Informationen der eingefügten Abarbeitungslänge (vgl. *Anhang: 2.1.*) ermittelt. Damit lassen sich Ansteuerungsskripte

²⁵ Regulärer Ausdruck steht in der Programmierung für ein verallgemeinertes Suchmuster mit eigenem Syntax, welches zum Suchen, Prüfen und Bearbeiten von Zeichenketten (Strings) Anwendung findet.

²⁶ Datentyp Array bezeichnet in der Programmierung die Datenstruktur eines Feldes mit mehreren Werten. Der Datenzugriff auf bestimmte Inhalte im Feld erfolgt über Indizes, die dessen Position bezeichnen.

aus der „alten“ Welt durch den Befehlsinterpreter (Abbildung 94) einfach durch Kopieren in die neue Synthesizer-Software überführen.

Über eine Zuordnungstabelle werden Befehle der Anweisungsliste aus den „Alt-Skripten“ den entsprechenden Gerätekomponenten zugewiesen (vgl. *Anhang: 2.2.*). Solche „Übersetzungstabellen“ erweitern die Flexibilität der VIs entscheidend, da durch Pflege und Erweiterung der Zubringerdateien, ohne zusätzliche Anpassung der jeweiligen VIs, die neue Funktionalität verwaltbar bleibt. Diese Zubringerdateien können einfach protokolliert und versioniert werden.

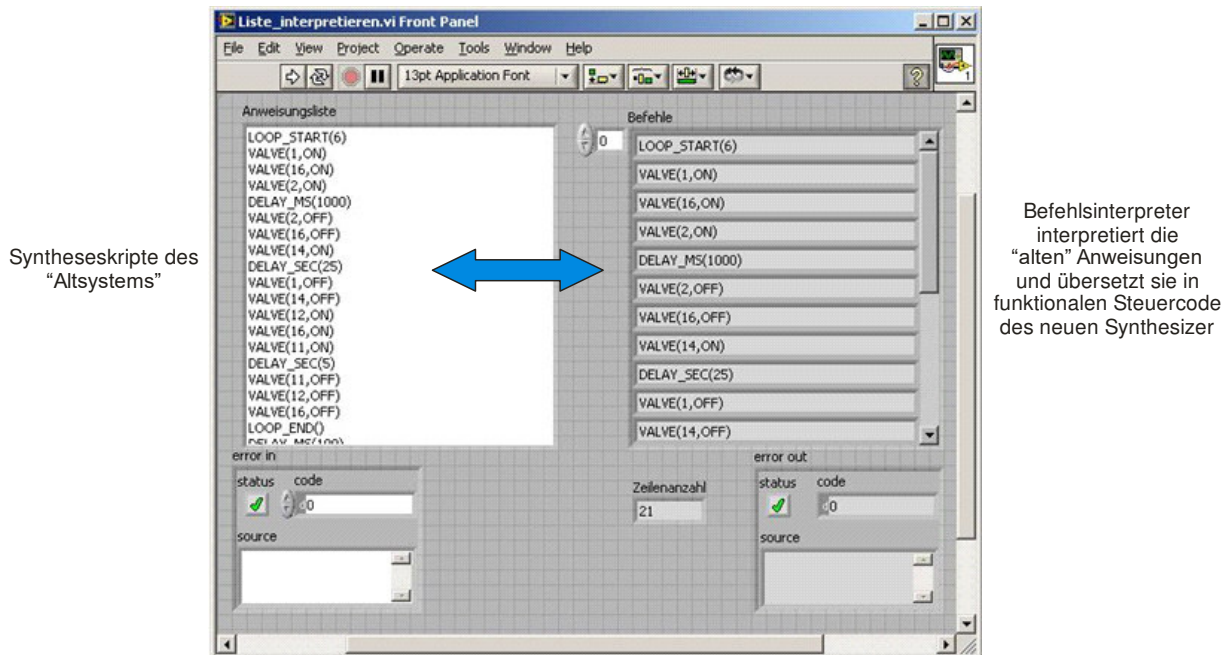


Abbildung 94: Der Befehlsinterpreter ist ein Unterprogramm, das die Kompatibilität mit den Syntheseskripten des Altsystems gewährleistet. Die Skripte werden in die Anweisungsliste kopiert und durch Interpretation der übergebenen Texte in funktionalen Code für das neue System umgewandelt.

LabVIEW Programme sollten sich in maximal einer Bildschirmseite abbilden lassen. Alle weitergehende Funktionalität ist in Unterprogrammen (Sub-VIs) auszulagern. Damit bleibt die Programmierung lesbar und lässt sich auch für Neueinsteiger effizient warten.

Mit jeder höheren Abstraktionsebene der Ansteuerungssoftware wächst jedoch die Notwendigkeit, neben Schleifen, zusätzliche Bedingungsstrukturen einzufügen, die jedoch wieder aus Sub-VIs bestehen können. Damit entwickelt sich ein sinnvoll gestaltetes LabVIEW Programm durch Stapeln der einzelnen Programm-VIs und Unterprogrammstrukturen in die Tiefe.

Der Zugriff auf die jeweiligen Ventile wird nun mit einem Testprogramm realisiert, das die bisher verwendeten Befehlsaufrufe benutzt. In einem Befehlsarray kann hier eine Folge von Ventilstellungen mit den entsprechenden Ventilnummern (1-16) und der gewünschten Ventilstellungen (Off / On) vorgegeben werden. Gerade in der Entwicklungsphase ist es hilfreich, Zustandsinformationen zur Überprüfung der jeweiligen Sub-VIs anzuzeigen. Dies gilt besonders bei

Hardware-Kommunikationen, da es dort durch Auslagerungen von Informationen, sowie Rückgabe seitens der Eingänge, zu unerwartetem Systemverhalten kommen kann. In der Bedienoberfläche des Testprogramms werden deshalb alle Zwischeninformationen angezeigt.

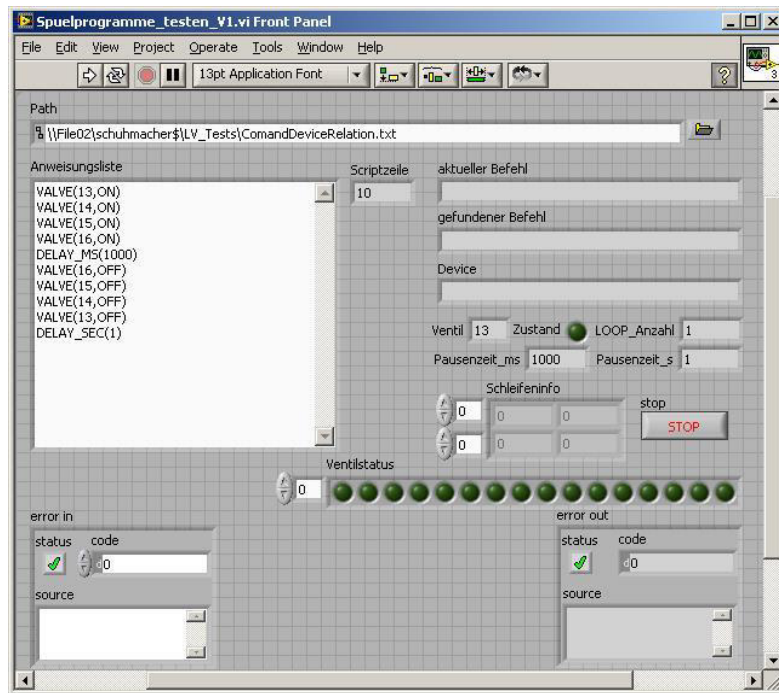


Abbildung 95: Frontpanel für ein Unterprogramm des entsprechenden Gerätetreibers mit dem die Spulventile im Synthesizer auf ihre Funktion getestet werden können.

Das Blockdiagramm für das Testprogramm erweitert sich für den Hardwarezugriff um die Initialisierungsphase, sowie um die Freigabe der reservierten Hardware vor dem endgültigen Beenden der Funktionsausführung, wie in der gezeigten Schrittkette (Abbildung 87, S. 103). Aus dem übergebenen Text ermittelt das Programm sowohl die Ventilnummer als auch den korrespondierenden Status und führt die entsprechende Ventilschaltung nach Plausibilitätsprüfung durch.

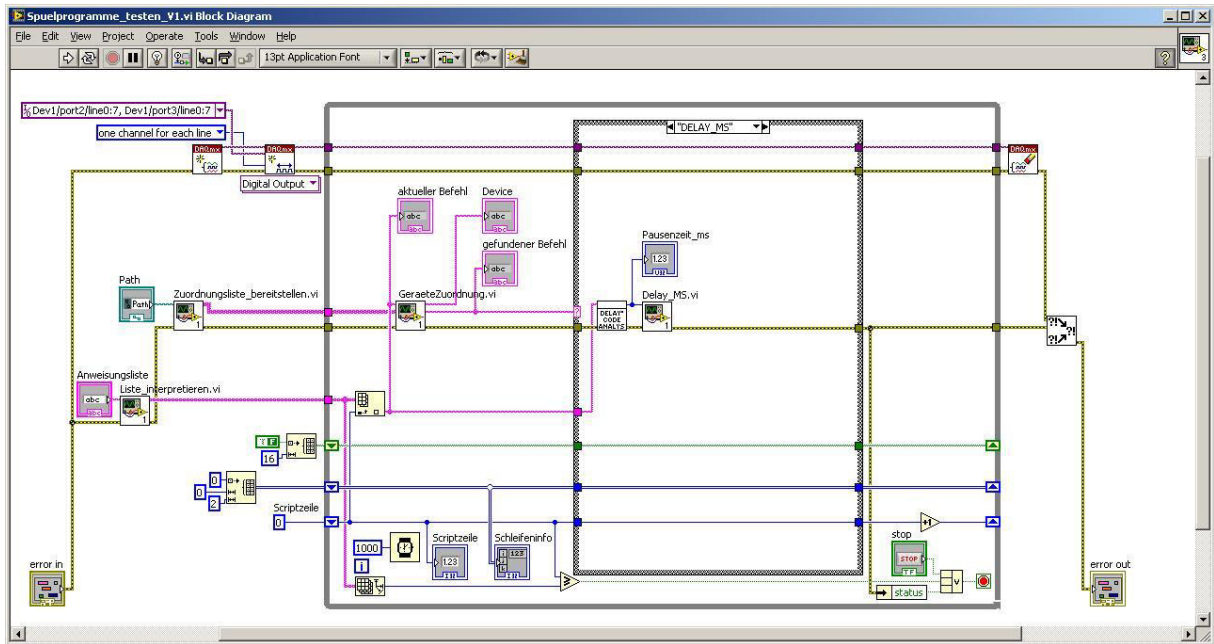


Abbildung 96: Blockdiagramm: Spülprogramm testen (Schleifensteuerung) mit Geräteinitialisierung, Gerätezuordnung, Befehlsinterpretation der Synthese-Skripte des J-KEM-Systems, Schaltung der Ventile gemäß des angegebenen Zustandes und abschließender Hardwarefreigabe nach Funktionsabarbeitung.

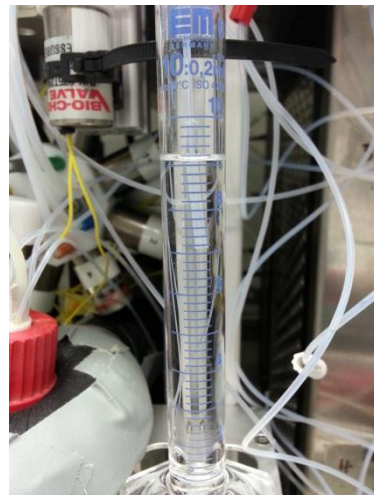
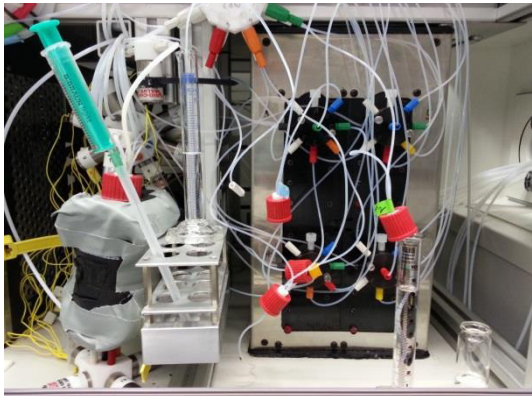


Abbildung 97: Zufuhr von 3 * 3mL DCM zur Überprüfung der geförderten Volumina als Kalibrierung vor der Synthese, um die allgemeine Funktionsbereitschaft zu gewährleisten

In der weitergehenden Programmierung sind die Portzuordnungen über eine Lösungsmittel-/Portzuordnungstabelle flexibilisiert. Aus diesen *.ini- Dateien erfolgt zur Laufzeit des Programmes, die aktuelle Zuordnung der Substanzen zu den gerätetechnischen Anschlüssen. Die Wartung der Zubringerdateien obliegt dem jeweiligen Geräte-Verantwortlichen und ist nur bei grundlegenden Veränderungen im System anzupassen. Für diese Anpassungen stehen dem autorisierten Administrator geräteabhängige Eingabeoberflächen zur Verfügung.

Die gezeigten VIs entstammen der Entwicklungsumgebung. Für die abschließende Anwendung wird

auf den Arbeitssystemen eine *.exe Datei erstellt, die nur die Runtime Version²⁷ von LabVIEW benötigt (vgl. Installation der LabVIEW Runtime Version, *Anhang: 1.2.*).

4.2.2 Steuerung der Spritzenpumpen und Rotationsventile

Die Rotationsventile schalten die entsprechenden Flüssigkeitswege für die Zuckerbausteine, Entschützungs- und Aktivierungsreagenzien. Die genaue Dosierung und den Transport der benötigten Reaktanten in das Reaktionsgefäß übernehmen die Spritzenpumpen. Die Ansteuerung der Spritzenpumpen und Rotationsventile erfolgt über die RS232 oder RS485 – Schnittstelle des Automatisierungs-PCs, womit insgesamt 16 Geräte über einen System-Bus adressiert werden können. Volle Flexibilität im Hinblick auf das modulare Gerätekonzept erhält das System mit dem Einsatz des CAN-Bus Systems, das als Industriestandard eine beliebige Anzahl von Geräten steuern kann.

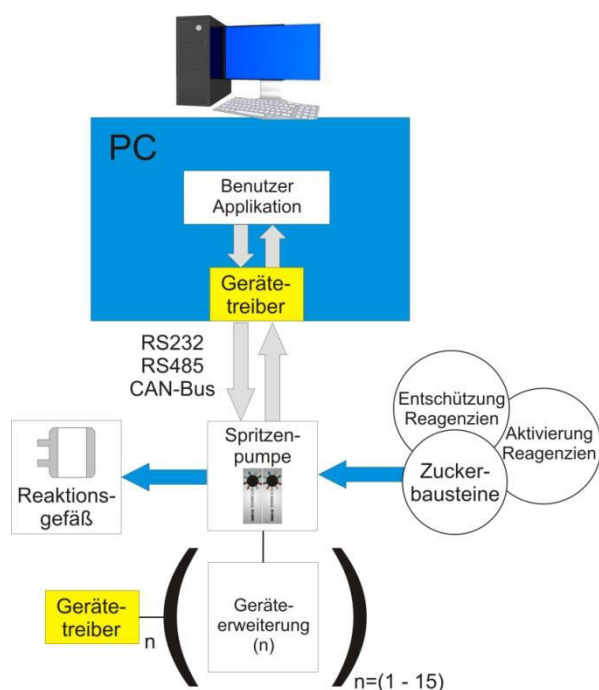


Abbildung 98: Die Ansteuerung der Spritzenpumpe findet über ein serielles Bussystem statt. Damit lassen sich neben der Spritzenpumpe noch bis zu 15 weitere Geräte steuern. Alternativ kann für die Gerätesteuerung auch ein CAN-Bus, das Industriestandard-Bussystem der Automobiltechnik, Verwendung finden. Hier existieren keinerlei reale Einschränkungen hinsichtlich der Geräteanzahl.

Für die Arbeitssynthesizer erfolgt die Gerätekommunikation noch über die RS485-Schnittstelle. Da in den üblichen PC-Varianten die seriellen Ports nicht mehr auf dem Motherboard zur Verfügung stehen, wird die benötigte Anzahl an Bussen mit Hilfe von USB-Seriell-Port Adaptern realisiert. Der

²⁷ Die Runtime Version einer Programmiersprache ermöglicht die Ausführung der erstellten Programmanwendung auch an Rechnern, die keine Vollversions-Lizenz der Programm-Entwicklungsumgebung besitzen.

grundlegende Funktionstest der Spritzenpumpen und Rotationsventile der Firma Kloehn erfolgt über die Gerätesoftware *Kloehn_Control*, sowie mittels VBA Zugriff (vgl. *Anhang: 1.3. ff.*).



Abbildung 99: Spritzenpumpe (li.) und Rotationsventile, Steuerungseinheit der Firma Kloehn (re.)

Die Kloehn – Komponenten sind über 24 V Gleichspannungsnetzteile spannungsversorgt und durch ein seriell Kabel mit dem PC verbunden. Zur Inbetriebnahme ist die Parametrierung der Schnittstelle mit folgenden Standardwerten vorzunehmen: 9600 Baud, 8 data bits, 1 stop bit, no parity, no flow control.

Der Test-Zugriff auf die seriellen Schnittstellen kann in LabVIEW über das Hilfswerkzeug MAX (Measurement & Automation Explorer) erfolgen, welches die Ansicht und Zugriff auf die angeschlossenen Geräte, Bussysteme usw. über ein Hyperterminal (Kommunikations-Programm) ermöglicht. Damit lassen sich die Befehle für Spritzenpumpe und Rotationsventile direkt abschicken. Zusätzlich zur gesamten Textstruktur der Befehlsstrings kommen dabei Steuerzeichen für die Interpretation eines Zeilenumbruchs zur Anwendung. Die angeschlossene Spritzenpumpe erhält auf Befehlsanforderung `/1W4R\r` den Rückgabewert: #6 (Abbildung 100).

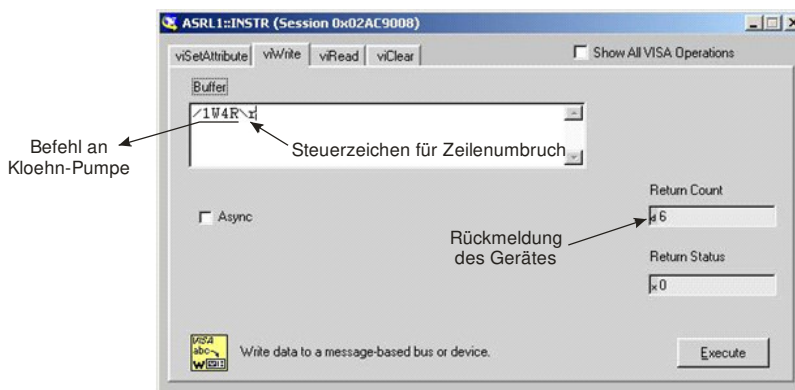


Abbildung 100: Innerhalb von LabVIEW können direkt Befehle an die angeschlossene Spritzenpumpe / Rotationsventil über das Kommunikationsprogramm (MAX) an die serielle Schnittstelle abgeschickt werden. Zusätzlich zum Befehlsstring, müssen hier Steuerzeichen angegeben werden. Die Rückmeldung des Gerätes wird direkt angezeigt.

Mit dem Konsolenprogramm *Kserial* liefert der Hersteller eine eigene Zugriffsmöglichkeit auf die Gerätekomponenten über die serielle Schnittstelle. In der ersten Entwicklungsphase wurde das Programm in LabVIEW eingebunden und als effiziente Testumgebung genutzt, um schnell die benötigte Gerätefunktion für den Syntheseautomaten bereitstellen zu können. Aufgrund

geräteinterner Controllersoftware besteht außerdem die Möglichkeit, Kloehn-Komponenten direkt miteinander zu vernetzen.

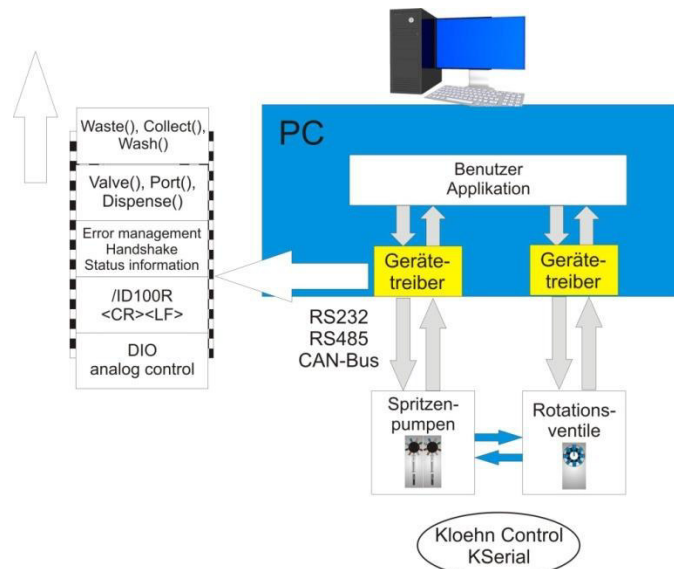


Abbildung 101: Abstraktionsebene der Software bei der Gerätetreiber-Entwicklung der Spritzenpumpen und Rotationsventile. Die geräteinterne Software ermöglicht eine Direktkommunikation der Komponenten untereinander.

Das LabVIEW Testprogramm (Abbildung 102) für die Kloehn-Komponenten enthält alle Elemente zur vollständigen Administration der seriellen Schnittstelle. Die Befehlsübergabe erfolgt per Textstring. Im Falle eines Fehlers wird aufgrund des Booleschen Elemente-Status in den alternativen Abarbeitungszweig des Blockdiagramms gesprungen. Sollte kein Fehler vorliegen, sich jedoch der Fehlercode im positiven Wertebereich > 0 aufhalten, wird dies als Vorhandensein einer lokalen Warnstufe interpretiert. Reale Fehler besitzen negative Werte. Dabei werden in der Regel weiterführende Informationen in die Textvariable übergeben.

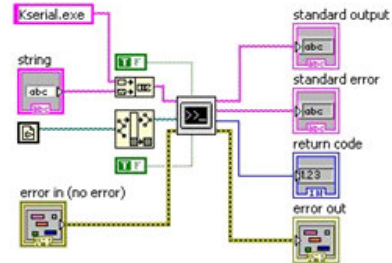
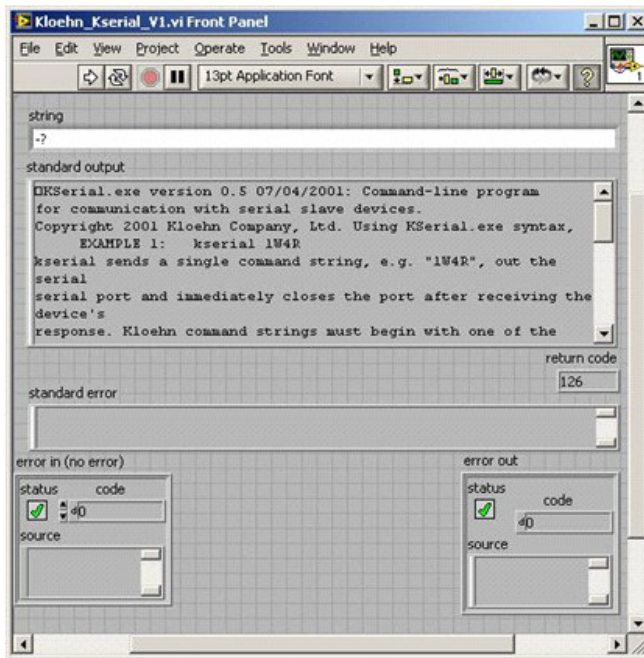


Abbildung 102: Einbinden des Konsolenprogramms Kserial im LabVIEW-Testprogramm enthält alle Elemente zur vollständigen Administration der Programmierschnittstelle.

Der entwickelte Gerätetreiber interpretiert sowohl den Fehlercode als auch die zusätzlichen Textinformationen und nutzt diese zur Realisierung der verschiedenen Fehler-Abarbeitungsroutinen, sowie für Wartungsinformationen. Im einfachsten Fall einer Fehleroutine wird der nachfolgende Programmcode komplett übersprungen und der übergebene Fehler am Ausgang für anknüpfende Elemente bereitgestellt.

Die beiden Systemkomponenten verwenden Befehlsstrings, die sich grundlegend nach den NAMUR-Empfehlungen^[169] aufbauenden Cavro-Protokoll^[173, 174] ergeben. Diese Befehlsstrings sind abstrakt und damit für die direkte Benutzerbedienung ungeeignet. Deshalb erfolgt die „Übersetzung“ zum Altsystem kompatibler Befehle mit Hilfe des Befehlsinterpreters (*Anhang: 2.3. ff*). Die sich daraus ergebenden Befehle beinhalten, wie bei jedem Hardwarezugriff, eine Initialisierung (*init – Befehl*) zur Bereitstellung der Hardware.

Im Auslieferungszustand sind alle Kloehn-Rotationsventile auf sechs Ventilwege voreingestellt. Durch Programmierung der internen Gerätespeicher wird auf die benötigten 8-Ventilwege umgestellt. Die aktuelle Ventileinstellung kann mit ~V unter Angabe der entsprechenden Adresse abgefragt werden. Die Parametrierung zum 8-Wege-Ventil erfolgt mit ~V10 im jeweiligen Gerätespeicher. (vgl. *Anhang 1.9.*).

Der Gerätetreiber liefert auch informative Parameter zur Erhöhung der Bedieneffizienz für den Anwender. Durch Kalkulation der Gesamtvolumina an verwendeten Lösungsmitteln, können sowohl vorhandene Restvolumina als auch Wartungseingriffe zum Auffüllen der Reservoirs sinnvoller im

Laboralltag geplant werden. Auch die Abschätzung der Gesamtsynthesezeit ist hilfreich für den Anwender. Hierbei kommt es parameterbedingt (Raumtemperatur, Füllvolumina) zu Toleranzen.

Damit die Spritzen nicht direkt mit den verwendeten Reaktanzen oder Lösungsmitteln kontaminiert werden, kommen zur Pufferung Trägerlösungsmittel zum Einsatz. Hierbei werden im Augenblick THF und DCE verwendet. Aufgrund des unterschiedlichen physikalischen Verhaltens (Viskosität, Siedepunkt), sind ggfs. individuelle Dosiergeschwindigkeiten notwendig. Gerade beim Befüllen der Spritze kann es - aufgrund der geringen Durchmesser der angeschlossenen PTFE-Schläuche - bei zu schnellen Verfahren des Pumpenstempels zu Unterdruck kommen. Der Siedepunkt einer Substanz erniedrigt sich jedoch bei sinkendem Druck. Das Dampfdruck – Temperaturverhältnis kann entsprechend der Clausius-Clapeyron'schen Gleichung näherungsweise beschrieben werden:

$$\ln p \approx \frac{1}{T \cdot p} \quad p - \text{Dampfdruck der Flüssigkeit, } T - \text{Temperatur}$$

Formel 1: Näherungsformel Dampfdruck entsprechend der Clausius-Clapeyron'schen Gleichung

Durch ungeeignete Spritzendosierung kann es aus diesem Grund infolge des sich einstellenden Unterdrucks zu Siedeverhalten^[175] mit daraus resultierender Gasbildung kommen. Mit Anpassung der Standard-Dosiergeschwindigkeit, die bei 10000 sps (steps per second) für die 5 mL Spritzenpumpen liegt, über den Befehl (RATE(,)) lässt sich dieser Effekt vermeiden. Da die Skalierung sps wenig benutzerfreundlich ist, wurde eine sinnvolle Umrechnung innerhalb des Befehlsinterpreters ermöglicht.

Die sich aus den Grundfunktionalitäten der Kloehn-Komponenten ergebenden Basis VIs wurden in Abbildung 103 strukturellen Funktionalitäten zugeordnet, die sich aus der Schrittkette des Hardwarezugriffs ergeben und ebenso die Konfiguration der spezifischen Schnittstellenparameter enthalten.

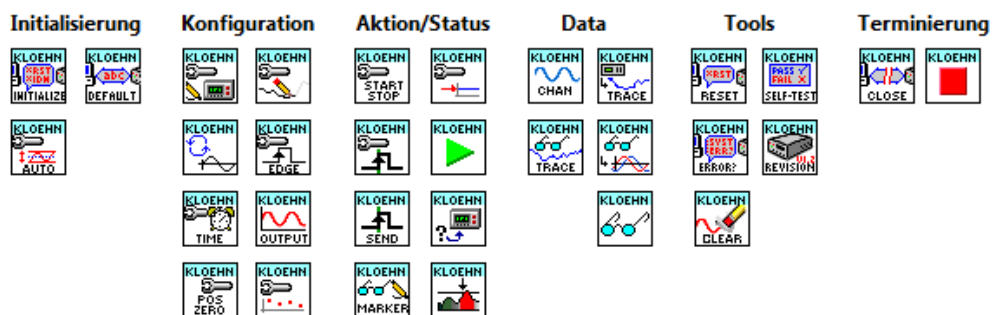


Abbildung 103: Übersicht der einzelnen VIs zur Kommunikation mit den Kloehn - Geräten

Im Testprogramm des Gerätetreibers für die Kloehn-Pumpen können komplette Befehlslisten abgearbeitet werden. Nachfolgend wurden Kalibrierungsdurchläufe für die Spritzenpumpen durchgeführt. Die Überprüfung des Flüssigkeitstransports erfolgt mit verschiedenen

Testprogrammen z. B. 1 mL schrittweise Befüllung und komplette Entleerung, komplette Befüllung und 1 mL schrittweise Entleerung, Pumpentest an Adresse 1 oder 2. (s. *Anhang 1.8.*).



Abbildung 104: Test der Spritzenpumpen mit der Gerätetreibersoftware und Testprogrammen zur Befüllung und Entleerung

4.2.3 Steuerung Druckverteilung

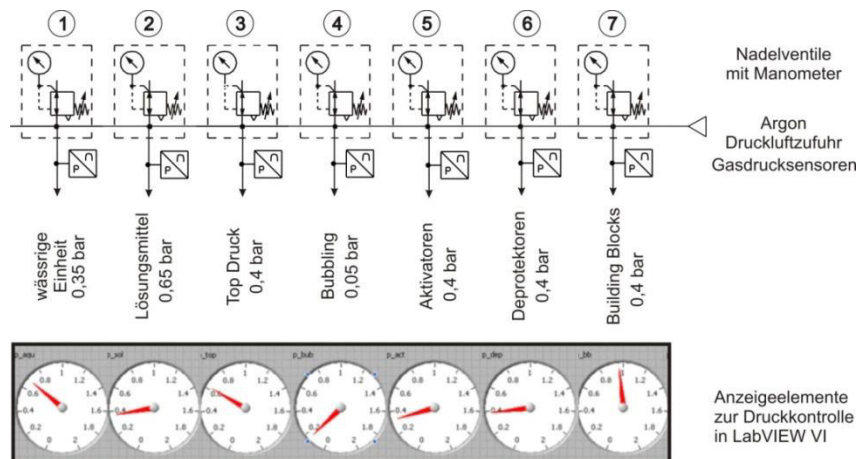


Abbildung 105: Drücke am Eingang des Druckverteilers. Aufgrund unterschiedlicher Viskositäten bei den Lösungsmitteln und Reagenzien sind die benötigten Drücke individuell anzupassen, um gleiche Volumina über die Ventilsteuerung zu transportieren.

Im Synthesizer werden alle Transportleitungen durch einen Druckverteiler (Manifold) auf Überdruck gehalten. Dies führt zum Flüssigkeitstransport innerhalb des Syntheseautomaten, auch gegen prozessdingte Druckveränderungen und verhindert damit eine Kontamination der Anlage durch Rückfluss von Substanzen aus dem Reaktionsgefäß. Da die Zuführung der Lösungsmittel über die Spulventile zeitgesteuert erfolgt, hängt der Volumentransport vom jeweiligen Fließverhalten des Lösungsmittels ab und wird durch dessen Viskosität bestimmt. Um auf allen ventilgesteuerten Lösungsmittelleitungen gleiche Volumina pro Zeiteinheit zu transportieren, wird deren Fließgeschwindigkeit jeweils durch manuelle Druckanpassung auf 1 mL Flüssigkeit pro 1000 ms kalibriert. Dabei werden die Transportleitungen für Aktivatoren, Entschützungsreagenzien und Zuckerbausteine in der Regel mit 0,4 bar beaufschlagt, Lösungsmittel liegen bei ungefähr 0,65 bar, die wässrige Einheit bei 0,35 bar und für das Durchmischen mittels Gasblasen wird 0,05 bar Druck benötigt.

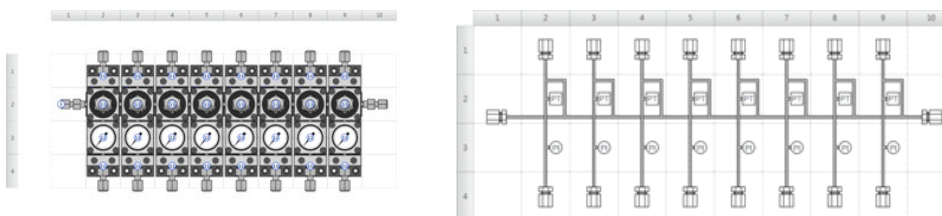


Abbildung 106: Technische Zeichnung des Manifolds der Fa. Swagelok, der an Arbeitssynthesizer II eingebaut wurde.

Bei den Arbeitssynthesizer erfolgt die Druckregulierung manuell. Die Gasversorgung liefert der Hausanschluss, alternativ eine Druckgasflasche. Zum Test der Druckstabilität, wurde eine Druckvergleichsmessung am Honeywell_24PCC, Honeywell 40PC1506 mit dem Gasdrucksensor 24PCCFA2G durchgeführt. Die lokalen Text-Messdateien sind mit über LabVIEW aufgenommen Werten zusammengeführt worden (Abbildung 107).

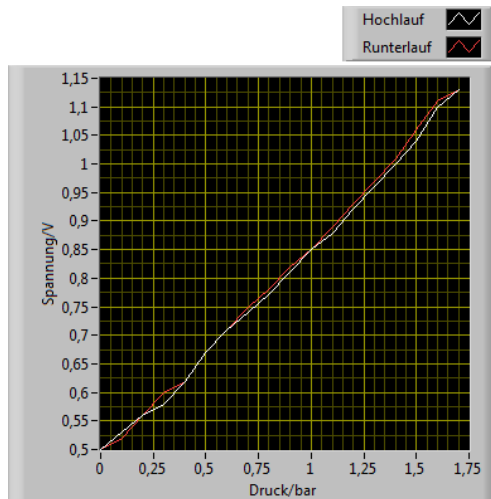


Abbildung 107: Kalibrierung der Drücke und Hysterese-messung

Zur Kontrolle wurde dazu ein Auswertungsprogramm (vgl. *Anhang 2.7.*) geschrieben, das die gemessenen Manometerdrücke visualisiert. In den Versuchen wurde verschiedene Drücke eingestellt und auch über längere Versuchsintervalle zum Test der Langzeitstabilität gemessen. Im folgenden Beispiel (Abbildung 108) war ein Ventil nicht vollständig geschlossen und verursachte das Wegdriften des Signals. Da die Bediener des Synthesizers die zeitlichen Verlaufswerte der Druckversorgung im Normalfall nicht benötigen, wird die Oberfläche der Druckkontrolle im Normalfall vereinfacht, ohne Zeitverlauf, dargestellt.

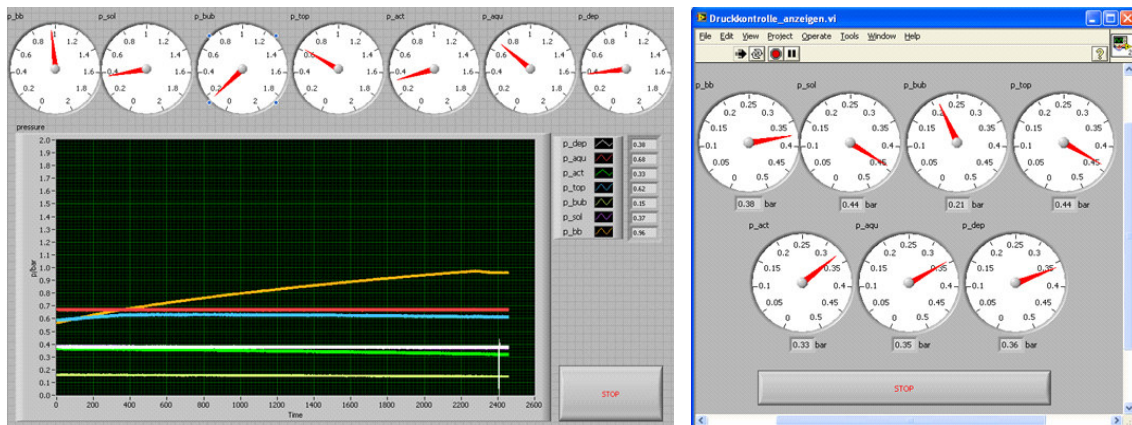


Abbildung 108: Druckmessungen am Druckverteiler für die unterschiedlichen Lösungsmittel und Reagenzien. Das „Wegdriften“ eines Signals zeigt an, dass ein Ventil nicht vollständig geschlossen wurde. Die Druckkontrolle lässt sich auf eine Übersichts-Oberfläche vereinfachen.

4.2.4 Steuerung des Fraktionssammlers

Der Fraktionssammler sammelt Reaktionsauskopplungen während der Synthese, sowie eventuell Synthese-Endprodukte. Dabei erfolgt der Probentransport aus dem Reaktionsgefäß druckbedingt über die Spulenventile. Der Gerätetreiber des Fraktionssammlers steuert die Positionierung des Roboterarms über den jeweiligen Probengefäßen in Abhängigkeit der Probenvolumina an unterschiedlichen Positionen.

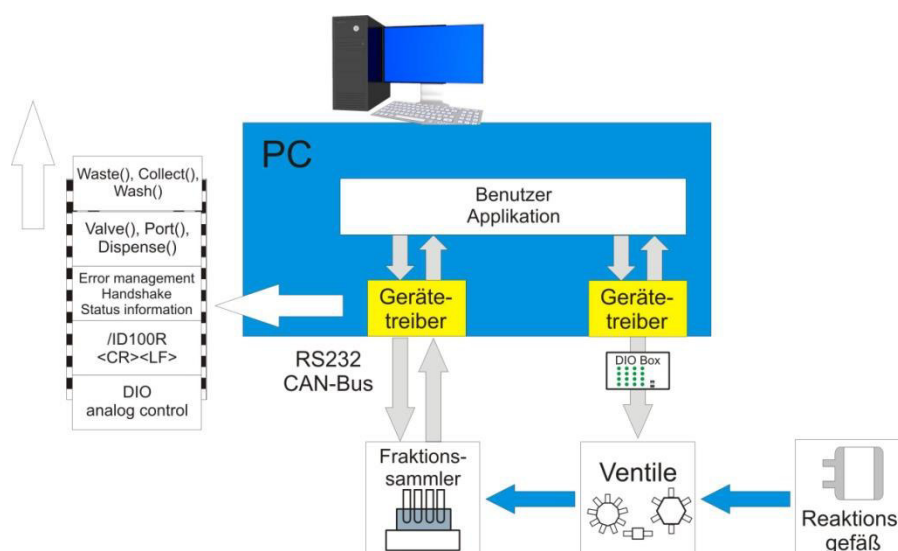


Abbildung 109: Der Fraktionssammler wird über eine rudimentäre RS232 Schnittstelle angesteuert. Die Probenentnahme aus dem Reaktionsgefäß erfolgt über die Spulenventile in den Fraktionssammler. Somit sind für die Bestückung des Fraktionssammlers Anforderungen an die Gerätetreiber der Ventile zum Flüssigkeitstransport, sowie des Fraktionssammlers zur Positionierung der Probe notwendig.

Für den Bau der neuen Arbeitssynthesizer wurden vorhandene Gerätekomponenten aus den Altsystemen genutzt, um möglichst kosteneffizient das neue Konzept umsetzen zu können. Im Gegensatz zu den Kloehn-Komponenten konnte beim Fraktionssammler leider nicht auf vorhandene Benutzerhandbücher, technische Spezifikationen, Schaltpläne oder herstellereigene Bediensoftware zurückgegriffen werden. Aus diesem Grund wurden – wie bereits zum Auslesen des J-KEM-Controllers - Hardwaremodule aufgebaut, mit denen die Kommunikation auf der RS232 Schnittstelle bidirektional mitprotokolliert werden konnte (vgl. *Anhang 1.12.*). Da einzelne Kommandos seitens des PC in mehrere Anweisungen für den Fraktionssammler umgesetzt werden, ist an der lokalen Anzeige des Fraktionssammlers eine Möglichkeit gegeben, die Plausibilität am Zielsystem zu kontrollieren.

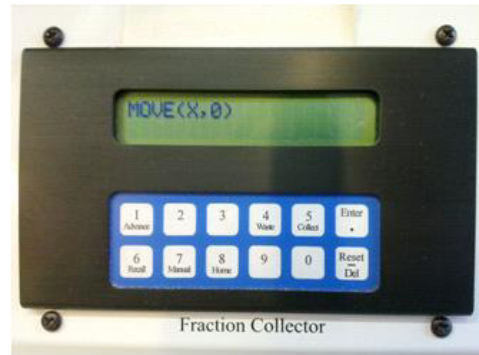
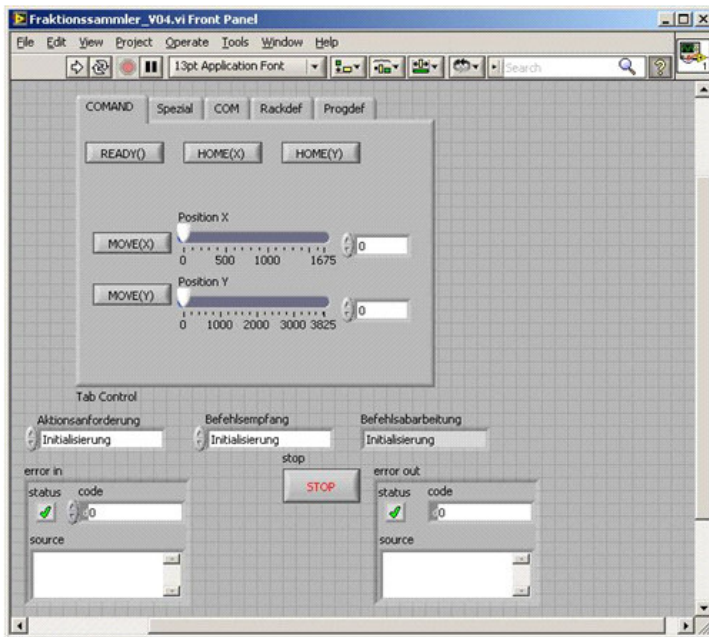


Abbildung 110: Im Fraktionssammler-VI können zu Testzwecken auch Einzelkommandos abgeschickt werden, die in der lokalen Anzeige des Zielsystems kontrollierbar sind. Die Positionierungsaflösung der drei vorhandenen Fraktionssammler für die Arbeitssynthesizer ist jeweils unterschiedlich und wird über gerätebezogene *.ini – Dateien eingelesen.

Mit dem Gerätetreiber konnten die geräteinternen Funktionen vollumfänglich zur programmatischen Ansteuerung erschlossen und zur Verfügung gestellt werden. Damit wurde für die Altgeräte sogar eine Funktionserweiterung ermöglicht. Der Fraktionssammler ist nun in der Lage mit unterschiedlichen Racks, die verschiedene Volumina bereitstellen, betrieben zu werden. Diese Racks sind theoretisch auch zur Laufzeit austauschbar, wenn das Gerät danach eine Reinitialisierung erfährt. Innerhalb der weiterführenden Programmierung des Synthesizers kommt dieses VI als eigenständige funktionale Zustandsmaschine zur Anwendung. Die Positionierungsaflösung und HOME – Position der drei für die Arbeitssynthesizer vorhandenen Fraktionssammler ist jeweils unterschiedlich und wird über gerätebezogene *.ini – Dateien eingelesen.

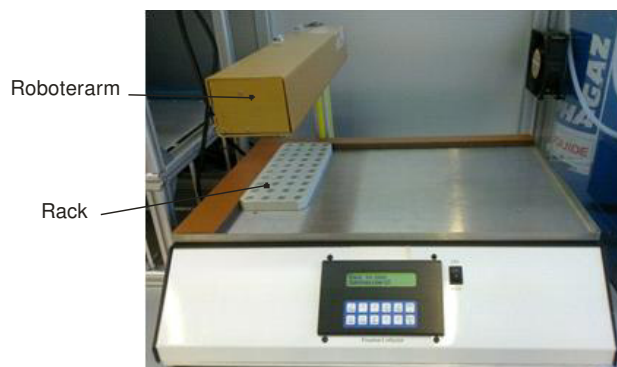


Abbildung 111: Die vorhandenen Fraktionssammler ist vollständig vom neuen Synthesizer-Programm ansteuerbar.

Damit bei unterschiedlichen Rack-Strukturen aufgrund veränderter Volumina in den Auffangbehältern nicht jeweils der absolut anzufahrende Positionswert berechnet werden muss,

lassen sich diese durch Direktdefinition im μ Prozessor des Fraktionssammlers hinterlegen. Dort kann sowohl der Abstand der jeweiligen Auffangbehältermittelpunkte in x- und y-Richtung, sowie die Anzahl der Behältnisse in Zeilen- und Spaltenorientierung abgespeichert werden.

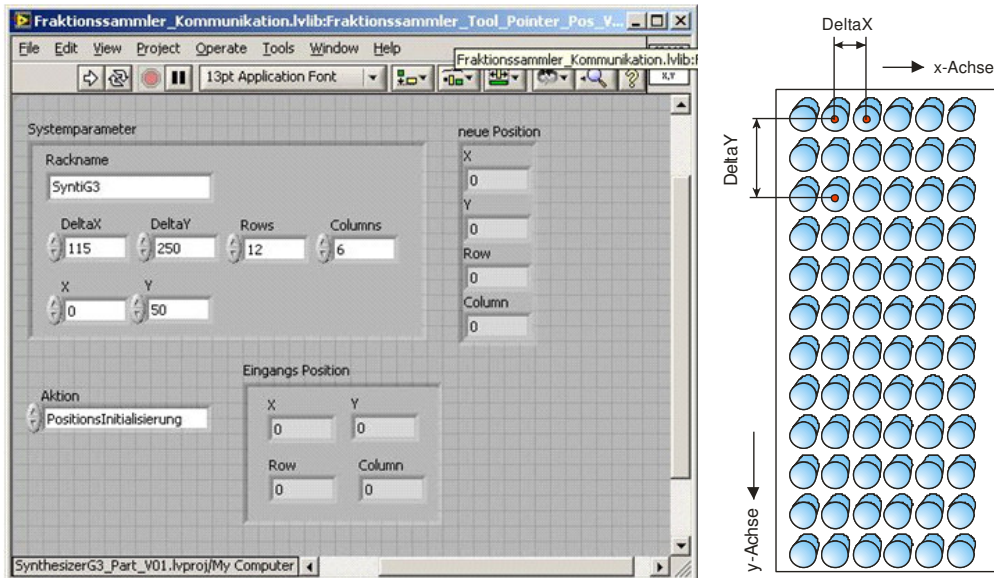


Abbildung 112: Im Fraktionssammler-VI lassen sich unterschiedliche Rack-Strukturen über den Abstand der jeweiligen Auffangbehältermittelpunkte in x- und y-Richtung, sowie reihen- und spaltenbezogene Anzahl der Behältnisse definieren.

Der neu erstellte Gerätetreiber konnte im Testprogramm des Befehlsinterpreters, das bisher für die der Kloehn-Geräte zur Anwendung kam, erfolgreich ausgeführt werden.

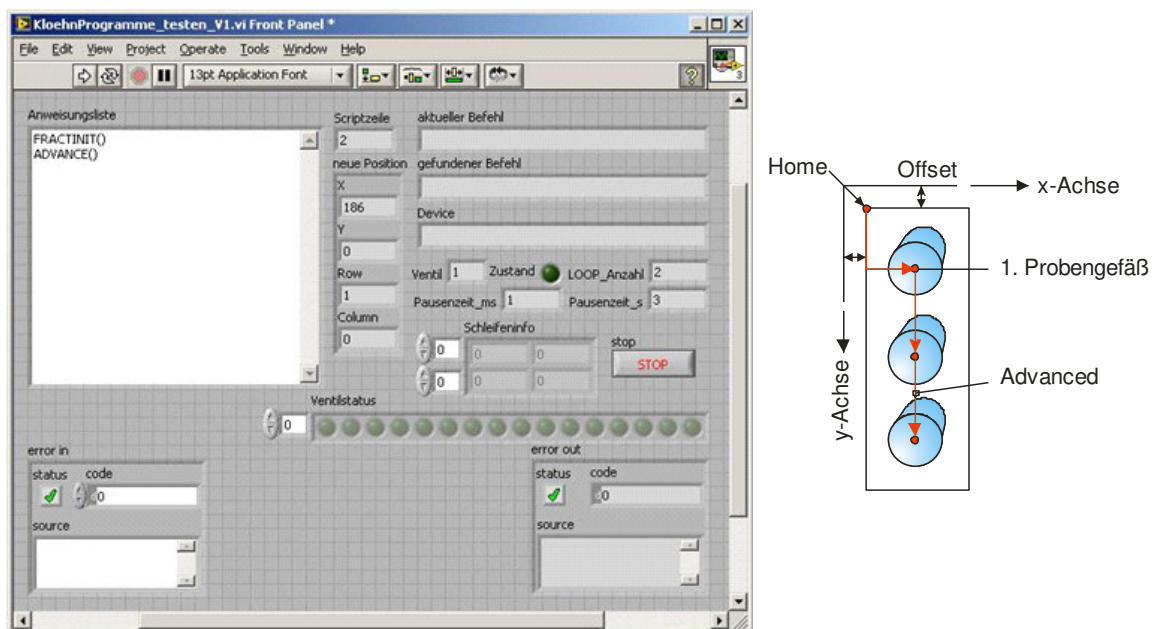


Abbildung 113: Im Testprogramm wurden, die mittels Befehlsinterpreter übersetzten Befehle des Altsystems ausgeführt. Hier erfolgt nach der Geräteinitialisierung mit dem Befehl *Advanced* eine Direktdefinition zur nächsten gültigen Sammelstellenposition.

Da die Racks in ihrer absoluten Position auf der gesamten Plattform frei positioniert werden können, besteht die Möglichkeit einen absoluten Bezugspunkt als Nullkorrektur vorzugeben. Sobald die Direktdefinition erfolgt ist, lässt sich durch die Übergabe der Befehlsaufforderung *Advanced* die

nächste gültige Sammelstellenposition seitens des Gerätetreibers automatisiert ansteuern. Die Reproduzierbarkeit und Stabilität wurden durch die Testprogramme (vgl. *Anhang 1.13.*) mehrfach kontrolliert.

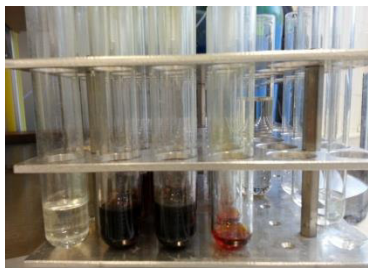


Abbildung 114: Reaktionsauskopplungen während der Synthese

Das Beispiel zeigt die Sammlung von Reaktionsauskopplungen während der Synthese, um über die jeweiligen Zyklen quantitative Aussagen über die chemische Reaktion z. B. den Umsatz / Ausbeute zu ermöglichen. Die Reagenzgläser an der Position 5 sind in diesem Fall gerade für die manuelle Fmoc-Analyse entnommen.

4.2.5 Steuerung der Temperiereinheit

Nach den bisher vorliegenden Erkenntnissen ist die Temperaturführung eine der wichtigsten Einflussgrößen innerhalb der Glykosylierungsreaktion. Hierüber kann in entscheidendem Maße zwischen der Ausbeuterelation (α , β) unterschieden werden. Für alle drei Arbeitssyntheserunden am Ende dieser Arbeit jeweils ein Julabo LH85 Temperiergerät zur Verfügung. Aus diesem Grund wurde auch hier ein umfangreicher Gerätetreiber entwickelt, der die gesamte Funktionalität dieser Systeme abbildet und die Gerätekomponenten über die serielle Schnittstelle (RS232) vom Automatisierungsrechner ansteuert.

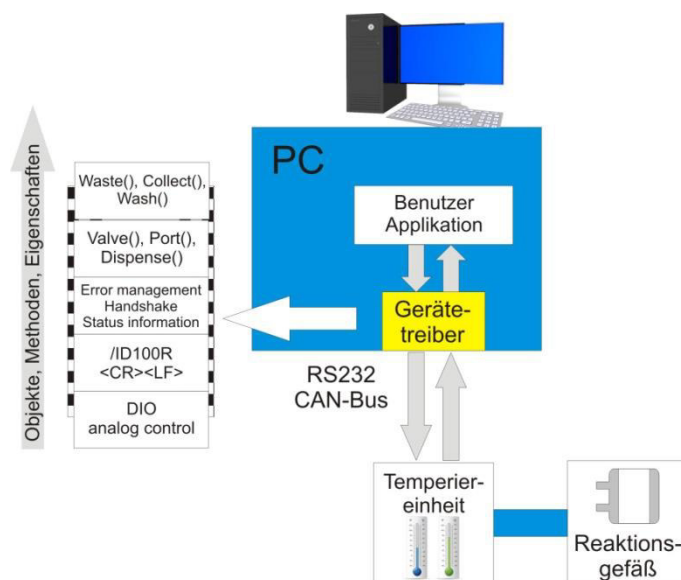


Abbildung 115: Die Temperiereinheit wird über die RS232 – Schnittstelle (alternativ CAN-Bus-System) angesprochen. Für die Glykosylierungsreaktion ist die Temperierung eine der wichtigsten Einflussgrößen.

Der Vorteil, der sich aus dieser im Laborbetrieb weit verbreitenden Temperierungseinheit ableitet, ist das bereits eine umfangreiche Testsoftware (Julabo Easytemp Professional) seitens Julabo zur Verfügung gestellt wird. Da die Geräte sowohl im Handmodus als auch im Analogbetrieb

eigenständig arbeiten können, müssen sie beim Verwenden über die RS232 Schnittstelle lokal vorprogrammiert und der gewünschten Gerätefunktion durch anschließende Tests sichergestellt werden (vgl. *Anhang 1.14. ff.*).

Beim Vorgängersystem zeigte die Temperierung einige Optimierungsmöglichkeiten auf. Gerade bei längeren Synthesen (> Hexamere) kam es gehäuft zu Kommunikationsdefiziten, bis hin zum kompletten Syntheseabbruch. Hier zeigte die konsequente Umsetzung der Entwurfsmethoden zur Gerätetreiberentwicklung mit Fehlermanagementsystem seine sichtbarste Wirkung, denn die Kommunikationsdefizite konnten durch ausgereifte Fehlervermeidungsroutinen vollständig beseitigt werden.



Abbildung 116: Julabo_LH85plus ist die Temperierungskomponente für die drei aufgebauten Arbeitssynthesizer

Dies führte insgesamt zur hohen Stabilität und Fehlertoleranz des Gesamtsystems, mit dem nun beliebig lange und komplexe Synthesen unterbrechungsfrei durchführbar sind.

Die Anforderungsbefehle an das Temperiergerät können im Gerätetreiber über eine Selektionsliste ausgewählt und anschließend parametrisiert werden.

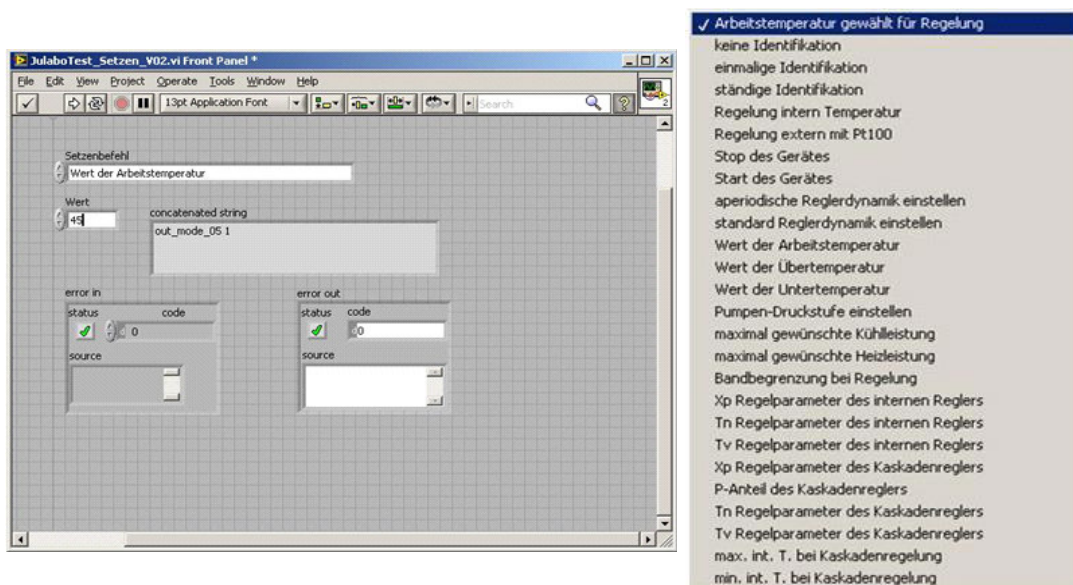


Abbildung 117: Das Gerätetreiber-VI für das Temperiergerät stellt alle Gerätefunktionen in einer Selektionsliste bereit. Die individuellen Parameter können dann direkt eingegeben werden.

Das Temperiergerät bietet dabei jedoch nicht nur die Möglichkeit Anforderungen zu setzen, sondern kann in umgekehrter Richtung über alle gesetzten Geräteparameter abgefragt werden.

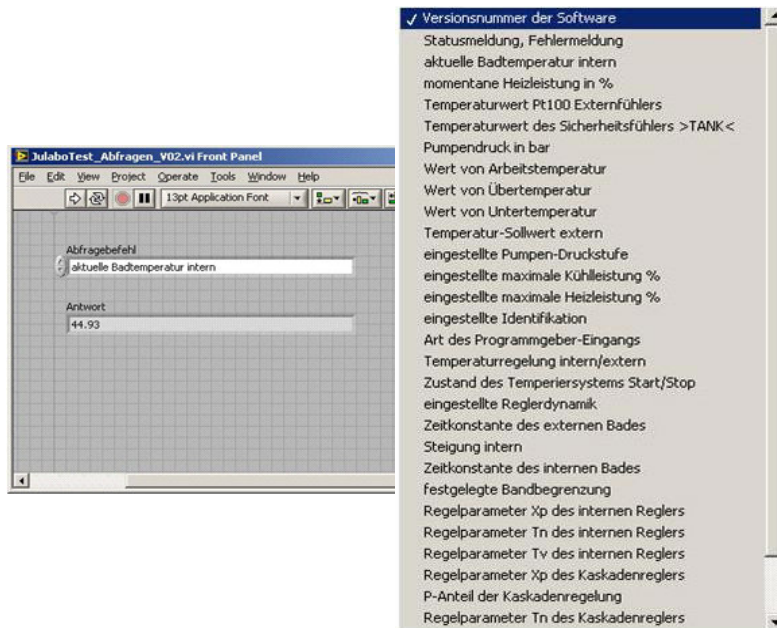


Abbildung 118: Mit den Abfragefunktionen können die aktuellen Geräteparameter aus der Geräteeinstellung ermittelt werden.

Der Anwender erhält mit dem Geräte-VI die Möglichkeit die jeweilig individuelle Gerätefunktionalität in einer 1:1 Kommunikation zur Julabo Schnittstelle zu testen. Auch dieses Geräte-VI wurde mittels Befehlsinterpreter-VI zu den „Altbefehlen“ kompatibel umgesetzt und vollständig getestet.

Neben den reinen Geräteparametrierungen werden PC seitig auch Überprüfungs- und Pegelkommandos implementiert, bei dem nach einem Setzen einer Zieltemperatur die Abarbeitung der Anweisungsliste solange terminiert ist, bis ein Kontrollwert erreicht wurde. Das nachfolgende Beispiel zeigt das Anfahren einer Temperatur von -15°C mit dem Befehl CHILLER (SETPOINT,-15), bei der durch den Befehl PTEMPLOW (-14°C) die nachfolgenden Befehle erst abgearbeitet werden, wenn mindestens -14°C seitens des Julabo als IST-Temperatur des aktuellen Bades zurückgeliefert werden (vgl. Abbildung 119). Auch der umgekehrte Fall, dass eine Temperatur überschritten sein muss, wird entsprechend programmatisch bereitgestellt (vgl. Testprogramme *Anhang 1.18.*)

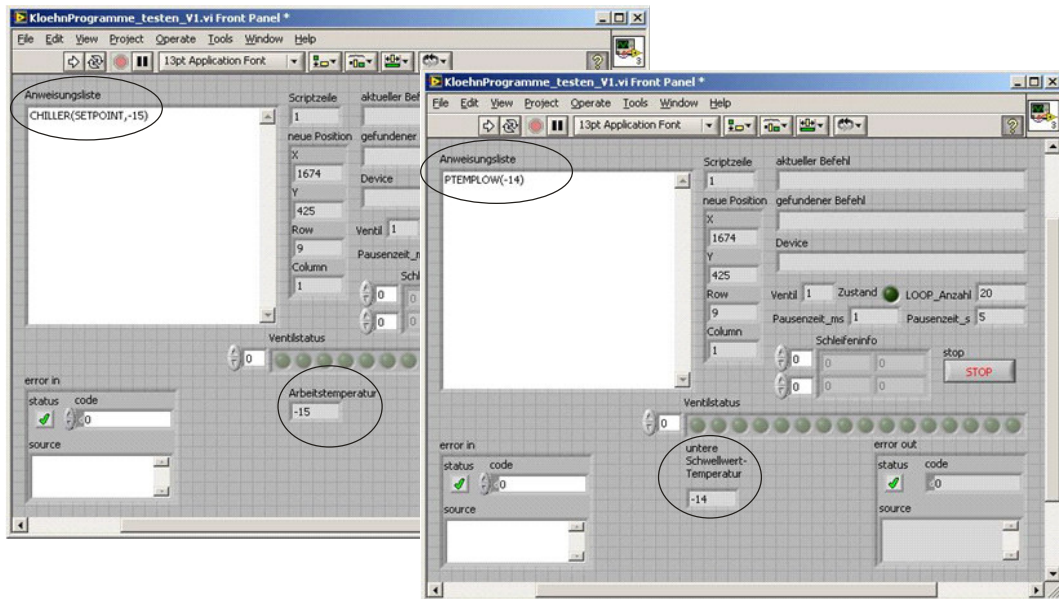


Abbildung 119: Testprogramm im Befehlsinterpreter-VI zum Setzen einer Zieltemperierung und anschließender Schwellwertüberwachung, nachdem die weiteren Befehle erst ausgeführt werden, wenn eine untere Schwellwert-Temperatur von -14 ° C erreicht ist.

Damit für die Qualität der Synthesen im Bereich der Temperaturführung eine Aussage möglich wird, erfolgten verschiedene Tests der Temperaturverläufe am Julabo. Dazu wurden sowohl die lesenden als auch schreibenden Prozesse während der Kommunikation am seriellen Bus unabhängig voneinander mit Hilfe zweier individuell zu konfigurierender Ports, die ereignisorientiert den gesamten Zeichentransfer in eine Datei wegschreiben, protokolliert. Hierbei kam erneut die Sniffer – Hardware zum Einsatz. Da die jeweiligen Schnittstellen für sich autark, aufgrund des Botschaftserkennens, ihren Puffer bereitstellen, wird die Synchronisierung der ausgelesenen Daten durch einen global zur Verfügung gestellten Zeitstempel ermöglicht.

Die Protokolldateien wurden durch ein Auswerteprogramm (vgl. *Anhang 2.11.*) analysiert, in dem die resultierenden String-Arrays individuell interpretiert in einzelne Signalverläufe überführt werden. So ließen sich die Daten für nachfolgende Funktionstests an den Julabo – Temperiereinheiten visualisieren.

Die Funktionstests zeigten überzeugende Ergebnisse. So lassen sich Temperatursprünge von -65°C auf +40°C innerhalb von 17 min erreichen. Der umgekehrte Temperaturverlauf in den Negativtemperaturbereich (+40°C auf -40°C) liegt mit 43 min um mehr als die Hälfte unter der Temperierungszeit gegenüber vor der Systemverbesserung.

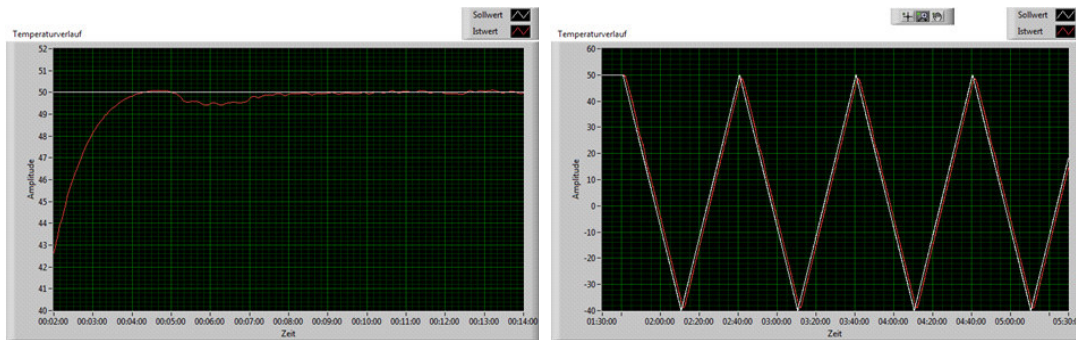


Abbildung 120: Sprungantwort aufgrund der Vorgabe zum Erreichen der Solltemperatur von 50°C und Ergebnisverlauf bei einem periodischen Temperaturwechsel nach Erreichen der jeweiligen Zieltemperaturen von +50°C auf -40°C

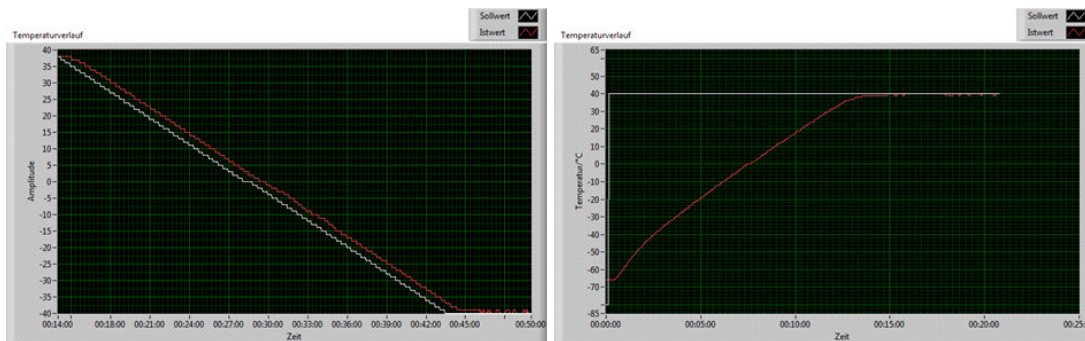


Abbildung 121: Links: Ergebniskurve bei einem Temperaturanforderung von +40°C auf -40 in 43 min
rechts: Sprungantwort bei einem Temperaturverlauf von -65°C auf +40 in 17 min

Die Stabilität und Reproduzierbarkeit der Temperaturführung wurde im nachfolgenden Versuch getestet, bei dem ein Temperaturprofil von -40°C auf +50°C elfmalig durchfahren wurde. Die zeitlich relativ aufgetragenen Verlaufskurven wurden im Nachgang segmentiert dargestellt. Dabei werden die jeweiligen Startpunkte um die Periodendauer des Temperierungsintervalls verschoben und die jeweiligen Durchläufe in einer Wasserfalldarstellung übereinandergelegt, um die Verläufe miteinander optisch vergleichbar zu machen. Die erhaltene Darstellung zeigt eine Reproduzierbarkeit von unter 0,8°C.

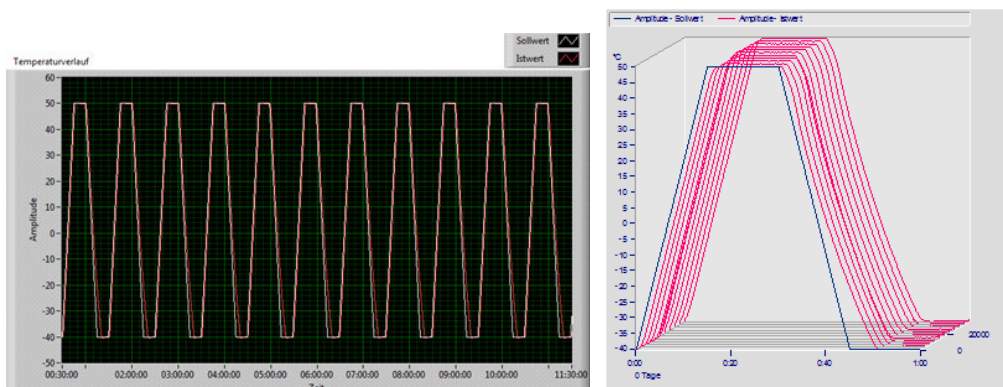


Abbildung 122: Test zur Reproduzierbarkeit der Temperaturführung. Die zeitlich aufgetragenen Verlaufskurven wurden segmentiert und in der weiteren Analyse als Wasserfalldarstellung zur besseren optischen Vergleichbarkeit hintereinander aufgetragen.

4.2.6 Update der Arbeitssynthesizer

Die drei in Berlin –Dahlem realisierten Arbeitssynthesizer wurden aufgrund des Umzugs in die neu fertig gestellten Institutsräumlichkeiten in Potsdam-Golm entsprechend den dort vorliegenden Gegebenheiten angepasst.



Abbildung 123: Aufbauort der neuen Systeme Synthesizer I, II und III in Golm

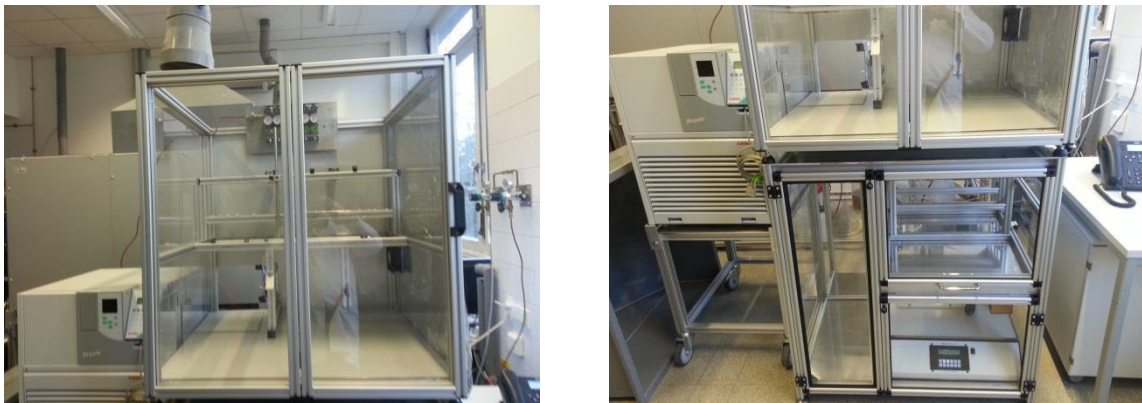


Abbildung 124: Neuaufbau des Rahmensystems für Synthesizer I. Umsetzung des modularen Konzeptes

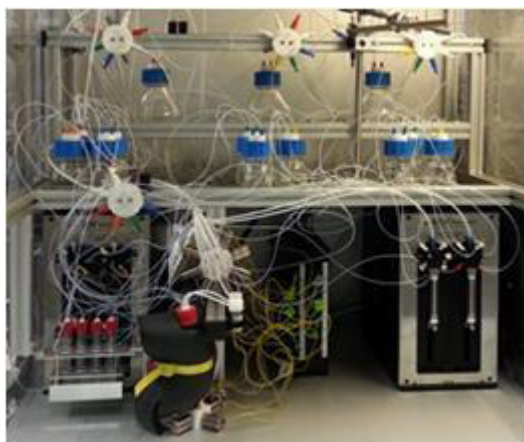


Abbildung 125: umgebauter Synthesizer

Da aufgrund der hier installierten Abzugsvarianten; die bisher zum Einsatz gekommenen Profilrahmenträgersysteme für die Abzugshauben räumlich nicht weiterverwendet werden konnten, mussten die Abmaße der Grundgeräte den räumlichen Gegebenheiten angepasst werden.



Abbildung 126: Bezeichnung der Lösungsmittel in den jeweilig zugeordneten Lösungsmittelsicherheitschrank

Die Lösungsmittelbevorratung konnte in die neben den Abzügen existierenden Lösungsmittelsicherheitschrank positioniert werden. Die Zufuhr der Lösungsmittel erfolgt durch die in der Mitte gezeigten PTFE-Schläuche, die aktuell durch Edelstahlrohre auch gegen Brandfälle geschützt wurden. Hierdurch hat sich das Handling mit den Lösungsmitteln und dem Abfall-Lösungen erheblich vereinfacht und entspricht den aktuellen Sicherheitsrichtlinien.

Die hieraus resultierenden Umbaumaßnahmen in den jeweiligen Versorgungsschläuchen wurden durch Kalibrierungsläufe in ihrer Funktionalität getestet.



Abbildung 127: 19\"/>

Parallel hierzu wurde die Modularisierung der Hardwareseite vorangetrieben. Hierzu entstanden auf Basis von 19-Zoll-Einschubvarianten die jeweiligen Trägermodule, die die einzelnen Synthesizer Komponenten als Funktionseinheit einfach administrierbar machen. Arbeitssynthesizer III wurde bereits mit allen Komponenten auf den genormten 19-Zoll-Standard umgesetzt und in ein entsprechendes 19-Zoll-Gehäuse eingebaut.

Zudem wurden für die Konzeptstudie weitere modulare 19-Zoll-Komponenten als Prototypen gefertigt und getestet. Mit dem Einbau in ein mobiles halbhohes 19-Zoll-Rack wäre damit ein mobiler Synthesizer mit erweiterter Funktionalität (z. B. für Parallelsynthesen) realisierbar (vgl. Kapitel 5).

Wie hier gezeigt, stehen die jeweiligen Gerätekomponenten mit regelmäßigem Wartungsbedarf (Rückschlagventil, Inlinefilter) direkt über der Geräterückseite im einfachen mechanischen Zugriff. Sowohl Versorgungsdruckzubringer als auch die Kontaktierung der elektrischen Ventile wurde mit

Standard-Komponenten ausgeführt. Die gesamte Steuerung wurde über die bisher verwendeten NI PCI Karten realisiert, jedoch sind zusätzliche messtechnische Erweiterungskarten mit in diesem System vorgehalten, um weiterführend elektrische Systemgrößen online auskoppeln zu können.

Da sich während der qualitativen Betrachtung der Synthesen herausstellte, dass das alte Manifold keine ausreichende Konstanz in der Druckversorgung aufrechterhalten konnte, musste hier eine Alternative bereitgestellt werden. Die bisher verwendeten neueren Druckverteiler von Swagelok sind sehr kostenintensiv, weshalb auf eine alternative Standardmodulvariante zurückgegriffen wurde, die in der automobilen Druckversorgungstechnik zum Einsatz kommt. Die hier erworbenen Kenntnisse über Druckregelgüte und Handhabung bei diesem preiswerten Druckverteiler können vor allem bei einer zukünftigen Open-Source-Systemvariante zur Anwendung gebracht werden. Nach den Anpassungen und Umbauten der Arbeitssysteme ergibt sich folgendes Bild:



Abbildung 128: Umbau auf neue Maße am Standort Potsdam-Golm für Synthesizer I, Synthesizer II, Synthesizer III(von links)

4.3 Software der Benutzerschnittstelle

4.3.1 Herstellen der Kompatibilität zu Syntheseskripten des Altsystems

Auf Steuerungsebene des Synthesizers wurden alle „Altbefehle“ 1:1 mit Hilfe des erstellten Befehlsinterpreters auf die Funktionsebene der neuen Geräteapplikation übersetzt, um die Vergleichbarkeit zu bereits gemachten Synthesen zu gewährleisten. Zur Durchführung der ersten Synthese am neuen Arbeitssynthesizer erfolgte zunächst eine funktionale Aufbereitung der Programmlisten des Altsystems.

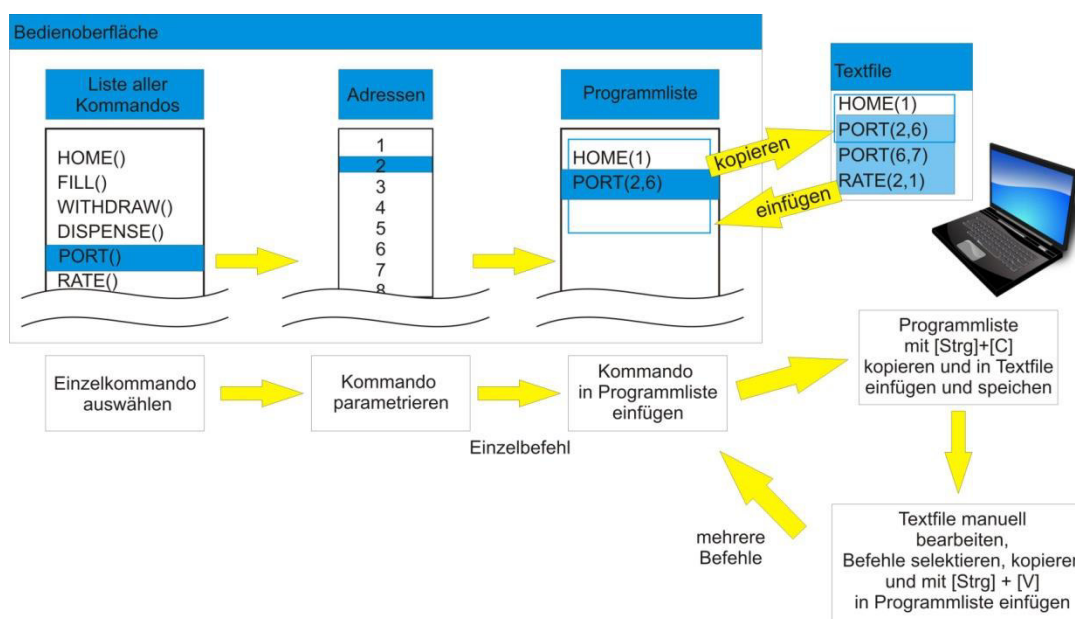


Abbildung 129: Bedienkonzept der alten Synthesizer-Software bis 2012. In der Steuersoftware wurden die Kommandos einzeln ausgewählt, parametrieren und so in die Programmliste eingefügt. Aus der Programmliste konnten Befehlslisten herauskopiert und in einem Textfile editiert werden. Aus dem Textfile wurden funktionale Befehlslisten zur Abarbeitung in die Programmliste der Steuersoftware zurück kopiert und zur Ausführung gebracht.

Das Bedienkonzept der alten Synthesizer bestand aus einer zeilenorientierten Eingabe. Jedes Kommando wurde einzeln aus einer Liste ausgewählt, parametrieren und in die auszuführende Programmliste eingefügt. Die Programmliste erlaubte kein Editieren von Befehlen. Aufgrund der umfangreichen Steueranweisungen, die schon für die Durchführung einer einfachen Oligosaccharidsynthese notwendig sind, etablierte sich eine alternative Eingabe der Befehlslisten. Innerhalb der Programmliste bestand die Möglichkeit des Kopierens und Einfügens über die Zwischenablage des Eingabe-Rechners. Die Kommandozeilen konnten so in ein Textfile kopiert und dort editiert werden. Die editierten Befehlslisten, meist funktionale Einheiten, die einem bestimmten chemischen Prozess-Schritt abbildeten, wurden dann zurück in die Programmliste kopiert und zur Ausführung gebracht. Die Textfiles der Synthese-Skripts konnten – aufgrund der Zeilenbegrenzung der Textfiles - bis zu maximal 65000 Zeilen enthalten.

Diese Programmlisten galt es während der Übergangsphase funktional verfügbar zu machen, damit auf allen Systemen - dem neuen Arbeitssynthesizer und den beiden noch in Betrieb befindlichen Altsystemen, auf Basis gleicher Steueranweisungen gearbeitet werden konnte. Alle bis dahin realisierten Synthesen wurden in ein Excel Sheet überführt und zur Vermeidung von Redundanzen gegeneinander abgeglichen. Veränderungen in der Building Block Anlieferung, der Temperaturen oder Zeiten, konnten mit VBA²⁸ Makros²⁹ flexibilisiert werden.

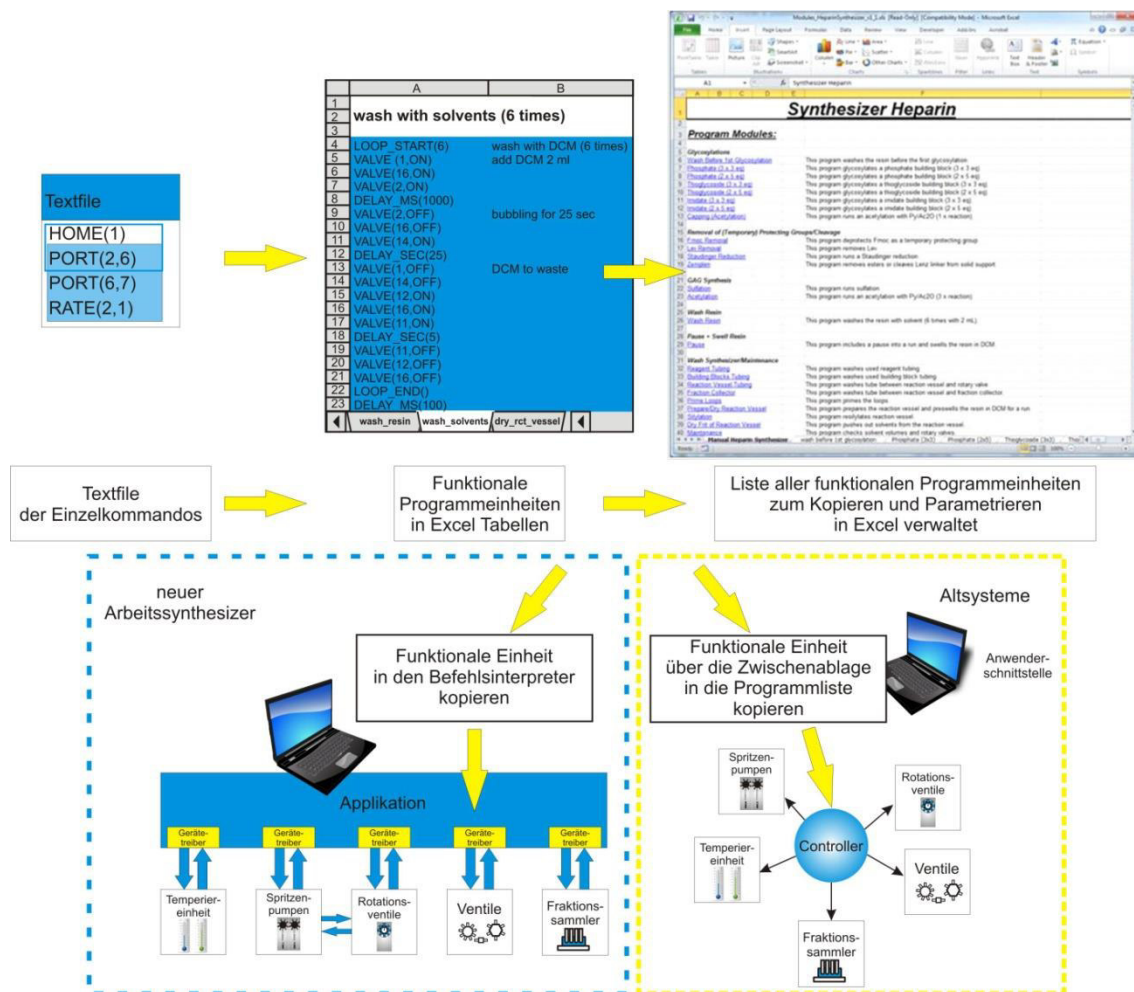


Abbildung 130: In der Übergangsphase 2012-2013 erfolgte die Verwaltung der Synthese-Programme des Altsystems als funktionale Einheiten in Form von Excel-Tabellen, die mit Hilfe von dialoggeführten Makros einfacher parametrierbar wurden. Die Programmcodes steuerten gleichermaßen die beiden controllerbasierten Altsysteme, als auch den neuen dezentralen Arbeitssynthesizers mittels LabVIEW.

Aufgrund der zeilenorientierten Programmierung bzw. des Kopierens von mehreren Befehlen in die Programmliste, resultierten Fehlermöglichkeiten aus der Nichteinhaltung notwendiger Schrittketten in der Synthesesteuerung. Im ersten Schritt wurden deshalb funktional zusammengehörige

²⁸ Abkürzung für engl.: „Visual Basic for Applications“; Skript- / Programmiersprache mit Ergänzungen für verschiedene Anwendungen in Microsoft Office.

²⁹ Makros sind Folgen von Anweisungen und Deklarationen für immer wiederkehrende Aufgaben innerhalb Microsoft Office, die in der Skriptsprache VBA programmiert oder vereinfacht innerhalb einer Office-Anwendung durch „Aufzeichnung“ als VBA-Code erstellt werden kann.

Ansteuerungszeilen in einzelnen Tabellenblättern arrangiert. Aus diesen einzelnen Tabellen gestaltete sich ein umfangreiches Worksheet, in dem die wichtigsten Funktionalitäten zum Steuern des Synthesizers jeweils zusammengefasst sind. In der effizienten Listenverwaltung, die somit jeweils als Kopiervorlage zur Anwendung kam, wurden zusammengehörige Befehle als ein Befehls-Block in die Zwischenablage überführt. Die Einhaltung der empfohlenen Bedienreihenfolgen reduzierte die Fehlerwahrscheinlichkeit erheblich. Da Excel die Copy & Paste Funktionalitäten vollständig unterstützt, ließen sich auch für längere Sequenzen die Ansteuerungslisten effizient und fehlerfrei zusammenstellen.

Auf dieser Realisierungsebene existierte kein individuelles Benutzerinterface, welches einfachere Parameterzugriffe ermöglicht. Da weder Assistentenführung noch ein Hilffsystem implementiert ist, muss der Benutzer Kenntnisse über Hardwaresteuerung und deren Schrittkette besitzen, um die funktionalen Einheiten in der richtigen Reihenfolge zu administrieren/kombinieren.

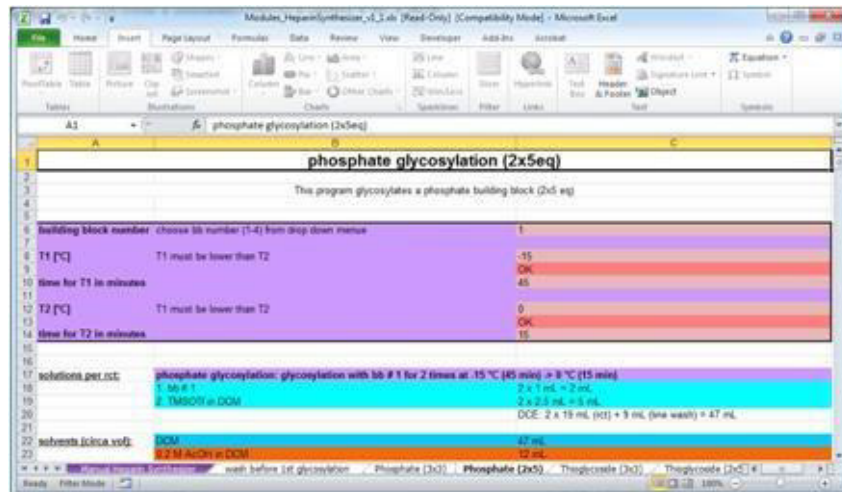


Abbildung 131: Die funktionalen Programmeinheiten der alten Syntheseskripte wurden in Excel-Tabellen überführt. Zur Vereinfachung der Bedienbarkeit und zur Fehlervermeidung wurden mit VBA dialogbasierte Makros realisiert, z. B. zur Parametrierung der Befehle und zur automatischen Codebereitstellung in die Zwischenablage

Die eigentliche Hardwareansteuerung wurde zu diesem Zeitpunkt über das J-KEM-Bedienterminal vorgenommen, dessen einzige Schnittstelle die Zwischenablage war, durch die der Programmtext in die Anweisungsliste kopiert wurde. Aufgrund dieser funktionalen Abkopplung der Altsysteme, existieren keine Debugging-Möglichkeiten, Fehlerverfolgung oder Teach-In Modus, welche die systematischen Analysen bzw. vereinfachte Geräteinteraktion ermöglichen. Somit wurde in einem nächsten Schritt unter Zuhilfenahme der Excel-internen Programmiersprache VBA Programme realisiert, die in Interaktionen mit dem Benutzer, aufgrund von dialogbasierten Rückmeldungen und Parameterselektions-Möglichkeiten, die Bedienung erheblich vereinfachten. Durch diesen Zwischenschritt erhält der Benutzer trotz individueller Oberflächen die gleiche Bedienphilosophie für

alle angewendeten Prozeduren. Da auch die automatisierte Bereitstellung der Codesequenzen in der Zwischenablage realisiert wurde, ist auch hier die Möglichkeit der Fehlbedienung erheblich reduziert.

Für den ersten im Rahmen dieser Arbeit gebauten Arbeitssynthesizer erfolgt der Funktionalitätstest der jeweiligen Codesequenzen parallel zur Weiterentwicklung in VBA auch innerhalb der bis dahin realisierten LabVIEW Benutzerschnittstelle (vgl. *Anhang 2.12.*).

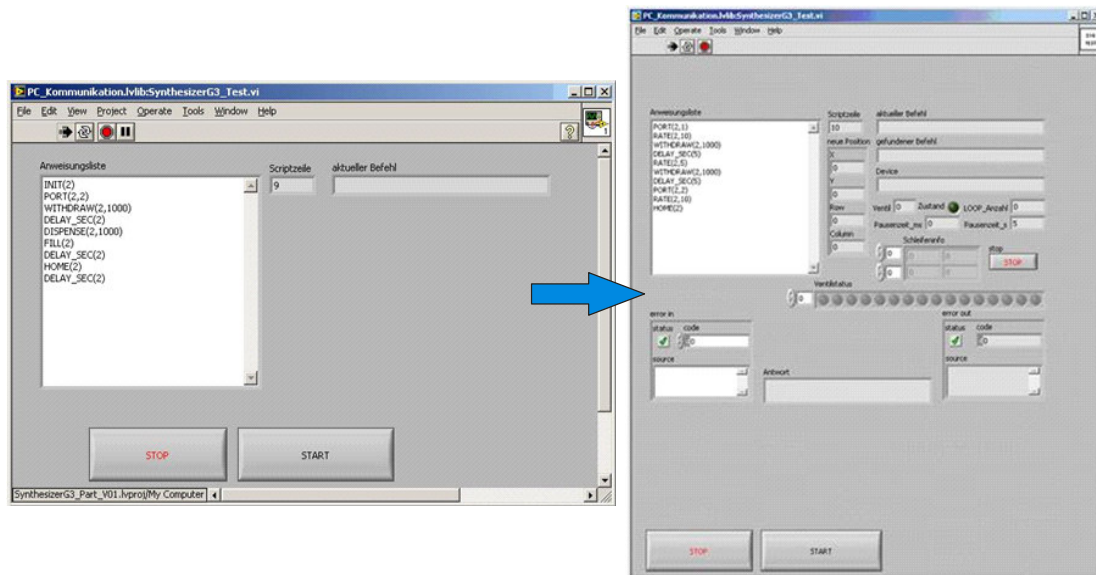


Abbildung 132: Erste Version (links) der Benutzerschnittstelle für den neuen Arbeitssynthesizer zum Test der chemischen Funktionseinheiten.

Das zunächst recht einfach gehaltene Testprogramm wurde während der Testphase mit hilfreichen Oberflächenelementen ausgestattet, aus denen gerade bei längeren Wartezeiten (Spülprozeduren, Temperierung) informative Systemzustände hervorgehen. In der endgültigen Benutzeroberfläche erfolgte die Reduktion der Anzeigen auf die notwendigen Elemente. Dafür erhielten die Objekte weitergehende Funktionalitäten in Hinblick auf Code-Visualisierung und der Möglichkeit Befehlszeilen zu kommentieren. Da zu diesem Zeitpunkt die Zusammenstellung der Anweisungslisten aus Excel heraus erfolgte, wurde dort die parallel liegende Spalte zur Kommentierung genutzt. Das LabVIEW Programm interpretiert dazu das Trennzeichen der beiden Spalten in Excel, den Tabulator Chr(9), als Kommentar. Allgemeingültig erfolgt die Implementierung des Semikolons als alternatives Zeichen für die gleiche Funktionalität.

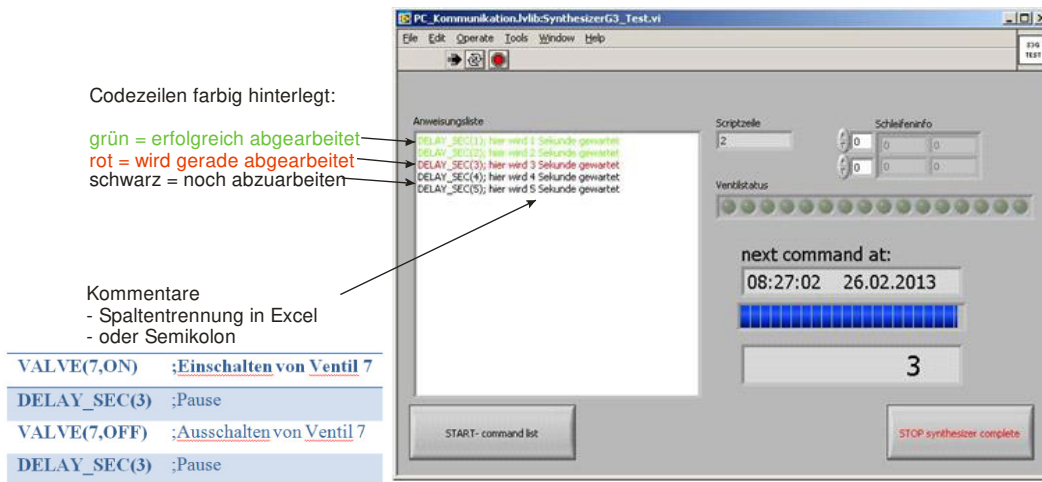


Abbildung 133: In der Benutzerschnittstelle Version 1.6 wird der Abarbeitungszustand der Codeelemente farblich hervorgehoben. Es ist möglich in einer zweiten Excel-Spalte neben dem Code auch Kommentare zu hinterlegen und diese mit in die Anweisungsliste zu kopieren. Alternativ können Kommentare durch Semikolon definiert werden.

Damit konnte die erste Synthesizer-Programmversion als lauffähige *.exe – Datei in eine Runtime Version kompiliert werden, die sich – unabhängig von der Entwicklungsumgebung – wesentlich leichter auf die verschiedenen Automatisierungs-PCs exportieren lässt. Unterschiede im Pfadaufbau zu den Applikationen wurden dabei programmatisch abgefangen (vgl. *Anhang 2.13.*).

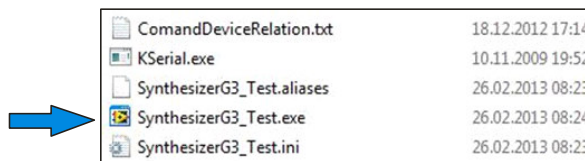


Abbildung 134: Die erste vollständig lauffähige *.exe Programmversion für die neue Synthesizer-Generation.

Bereits in dieser Version wurde die Möglichkeit implementiert, die Software im Demo-Modus (Abbildung 135) zu betreiben. Damit lässt sich ohne Hardwarezugriff der Funktionsablauf der chemischen Synthese durchspielen. Insbesondere bei neuen Syntheseprogrammen können so bereits im Vorfeld eventuelle Schrittkettenfehler oder Syntax-Probleme erkannt werden.

4.3.2 Befehle, Module und Skripte

In der Weiterentwicklung der Benutzer-Applikation strukturiert sich die Software seitens der Bedienung in Befehle, den grundlegenden Bedienungskommandos zur Hardwaresteuerung, aus Modulen, die sich als Funktionseinheit aus verschiedenen Befehlen aufbauen, und den statischen Skripten, die letztlich die gesamte Synthesizer-Steuerung beinhaltet.

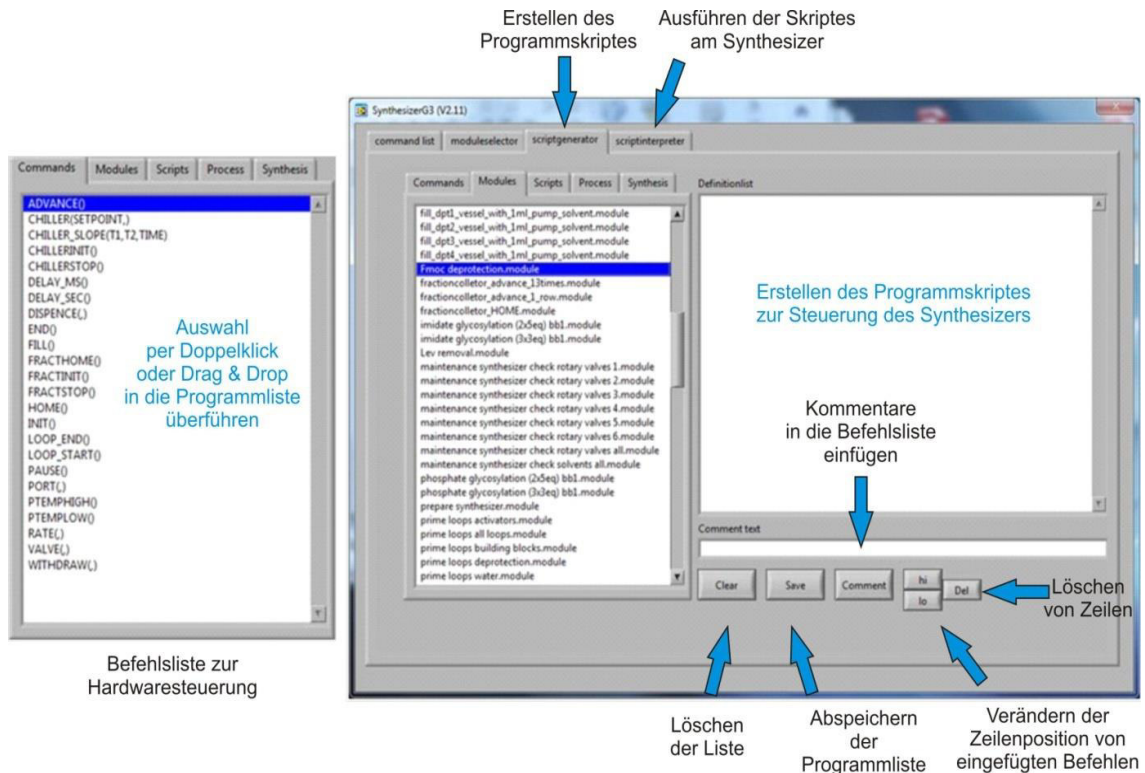


Abbildung 136: Bedienoberfläche der Arbeitssynthesizer. Aus der Kommandoliste lassen sich im Skriptgenerator funktionale Programmeinheiten als Module zusammenstellen und abspeichern. Skripte können sich aus Einzelbefehlen und Modulen zusammensetzen. Die Abarbeitung der Skripte erfolgt im Skriptinterpreter.

Befehle stellen die unterste Ebene dar, an der ein Anwender in die Synthesesteuerung eingreifen kann. Ein Überblick über die Befehle und deren Funktionsbeschreibung liefert Tabelle 4.

Jeder Benutzer kann sich aus den Kommandos listenorientiert individuelle Module zusammenstellen, die mit der Endung **.module* im Synthesizer-Verzeichnis abgespeichert werden. Alle dort abgespeicherten Module stehen auf der Registerkarte „Modules“ im Skriptgenerator zur Verfügung.

Befehl	Beschreibung
VALVE(1,ON)	Öffnen des Nadelventils 1
VALVE(1,OFF)	Schließen des Nadelventils 1
VALVE(0,OFF)	Schließen aller Nadelventile
RATE(1,10)	Für die Spritze an Adresse 1 wird die Flussrate auf 10 mL/min gesetzt
PORT(1,1)	Das Ventil an Adresse 1 wird auf den Ausgang 1 gestellt
FILL(1)	Die Spritze an Adresse 1 wird vollständig befüllt => 5 mL Volumen
DISPENSE(1,1000)	Durch die Spritze an Adresse 1 werden 1000 µL Flüssigkeit geliefert
WITHDRAW(1,1000)	Die Spritze an Adresse 1 wird um 1000 µL aufgezogen
HOME(1)	Die Spritze an Adresse 1 wird vollständig entleert unabhängig ihrer aktuellen Position
DELAY_SEC(10)	Seitens der Abarbeitung der Anweisungsliste wird 10 Sekunden gewartet
DELAY_MS(100)	Die Abarbeitung wird für 100 ms angehalten
PAUSE()	Die Abarbeitung der Anweisungsliste wird angehalten und es wird auf eine Benutzerinteraktion gewartet
LOOP_START(10)	Der nachfolgende Schleifen-Anweisungsblock wird 10 Mal abgearbeitet
END_LOOP()	Definiert die letzte Anweisung in einem Schleifen-Anweisungsblock
CHILLER(SETPOINT,-30)	Vorgabe der Zieltemperatur für das Thermostat
PTEMPLOW(-19)	Wartet mit der Abarbeitung der Anweisungsliste bis die aktuelle Temperatur niedriger ist als -19 °C
PTEMPHIGHT(40)	Wartet mit der Abarbeitung der Anweisungsliste bis die aktuelle Temperatur höher ist als 40 °C
ADVANCE()	Fraktionssammler in die nächste Rack - Position weiterbewegen

Tabelle 4: Befehlsliste - Funktionsbeschreibung

Eine aussagekräftige Benennung der Module mit Parametern zur Prozessführung ist empfehlenswert, um die Lesbarkeit umfangreicher Skripte zu erhöhen. Bewährt hat sich folgende Struktur bei der Namensvergabe:

Modul

Funktion (Wash, Delivery, Prime, Empty), Lösungsmittel (DMF, THF, ...), Menge (n * x mL), Temperatur, Transportweg (RV (Reaktionsgefäß), Waste (Abfallbehälter), Fraction (Fraktionsammler))

z. B.: *Modul*: Wash, DMF, 3 x 3 mL, 25°C, Waste

Sinnvollerweise beinhalten Module komplette Funktionsblöcke ohne komplexe Schleifenbildung und bilden somit die kleinste, aufeinander aufbauende Schrittkette ab. Ein Beispiel wäre das Hinzufügen einer Gasblase zwischen dem Pufferlösungsmittel und dem Zuckerbaustein, das Rückführen der Building Blocks in den Vorratsbehälter, aber auch komplette Spülvorgänge des Reaktionsgefäßes mit einem Lösungsmittel. Aus diesen lassen sich größere Einheiten, z. B. komplette Waschzyklen oder Spülprozeduren der Transportwege zusammenfassen. Diese bestehen im Normalfall aus Modulen, sind jedoch auch in Kombination mit Einzelbefehlen möglich.

```
1  +; //<Info: Header
13 +; //<Modul: Wash, DMF, 3x 3 ml, 25°C, Waste
36 +; //<Modul: Delivery, DMF, 3 ml, -, RV
47 +; //<Modul: Wash, deprotection line
85 +; //<Modul: Prime, line deprotection
102 +; //<Modul: Empty, , 25°C, Waste
114 +; //<Modul: Delivery, Dep1, -, Fraction
162 +; //<Modul: Delivery, Dep1, -, Fraction
210 +; //<Modul: Delivery, Dep1, -, Fraction +
264 +; //<Modul: Wash, DMF, 2ml, -, Waste
302 +; //<Modul: Wash, deprotection line
337 +; //<Modul: Wash, DCM, 1 x 3 ml, 25°C, Fraction +
355 +; //<Modul: Wash, DMF, 3 x 3 ml, 25°C, Waste
377 +; //<Modul: Wash, THF, 3 x 3 ml, 25°C, Waste
399 +; //<Modul: Wash, DCM, 3 x 3 ml, 25°C, Waste
```

Abbildung 137: Das Skriptbeispiel einer Fmoc – Entschützung besteht aus einzelnen funktionalen Modulen, die entsprechende Prozess-Schritte abbilden.

Skripts bilden dabei einen chemischen Prozess, einen Reaktionsmechanismus, z. B. eine Fmoc Entschützung-Reaktion, oder auch die Gesamtsynthese ab. Das Skriptbeispiel zu Fmoc - Entschützung besteht aus 14 Prozess-Schritten, die jeweils in einem eigenen Modul funktional abgebildet sind.

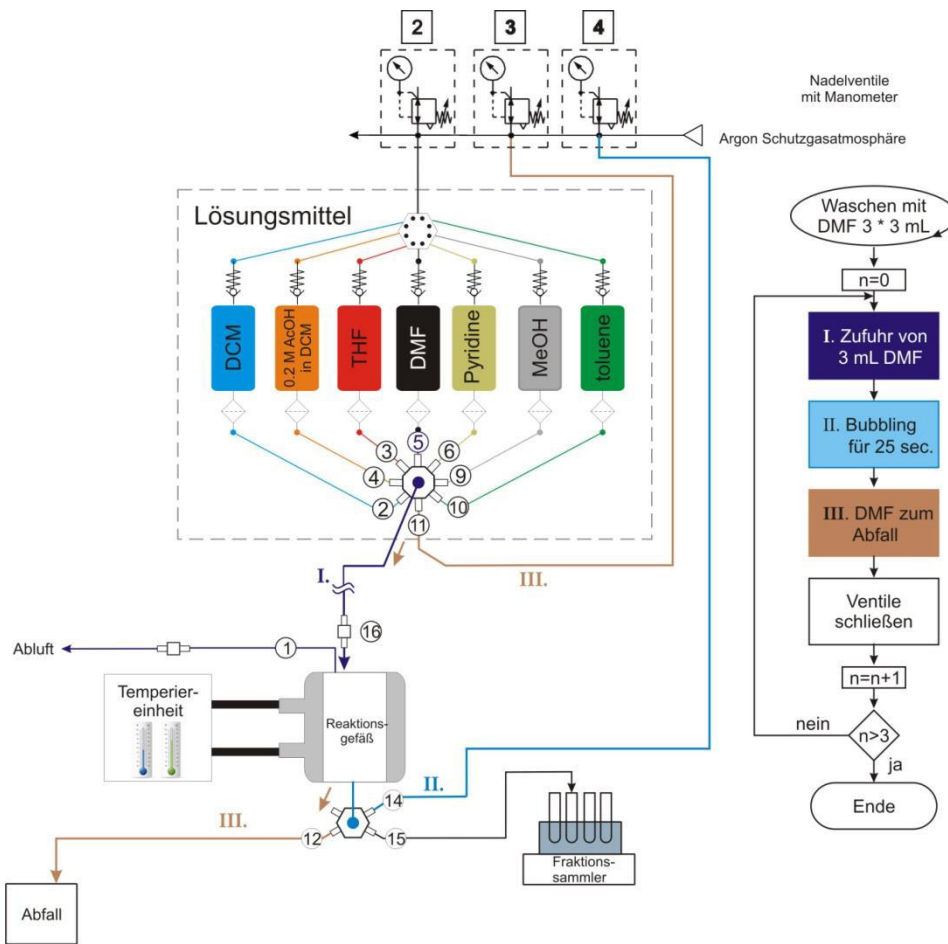


Abbildung 138: Der Transport des Lösungsmittels DMF in das Reaktionsgefäß erfolgt durch Schaltung der entsprechenden Spulenventile und infolge der Überdruckatmosphäre. Die Mengenzufuhr wird über die Schaltungsdauer der entsprechenden Ventile gesteuert.

```

Fmoc_DeProtection_V1
1  //<Info: Header
13 //<Modul: Wash, DMF, 3x 3 ml, 25°C,Waste
14 LOOP_START(3) wash with DMF (3 times)
15 VALVE(1,ON) add DMF 2 ml
16 VALVE(16,ON)
17 VALVE(5,ON)
18 DELAY_MS(3000)
19 VALVE(5,OFF) bubbling for 25 sec
20 VALVE(16,OFF)
21 VALVE(14,ON)
22 DELAY_SEC(15)
23 VALVE(1,OFF) DMF to waste
24 VALVE(14,OFF)
25 VALVE(12,ON)
26 VALVE(16,ON)
27 VALVE(11,ON)
28 DELAY_SEC(5)
29 VALVE(11,OFF)
30 VALVE(12,OFF)
31 VALVE(16,OFF)
32 LOOP_END()
33 DELAY_MS(100)
34 CHILLER(SETPOINT,25)
35 //Modul>
    
```

Abbildung 139: Funktionsmodul zum Lösungsmitteltransport in das Reaktionsgefäß bei Raumtemperatur (25°C)

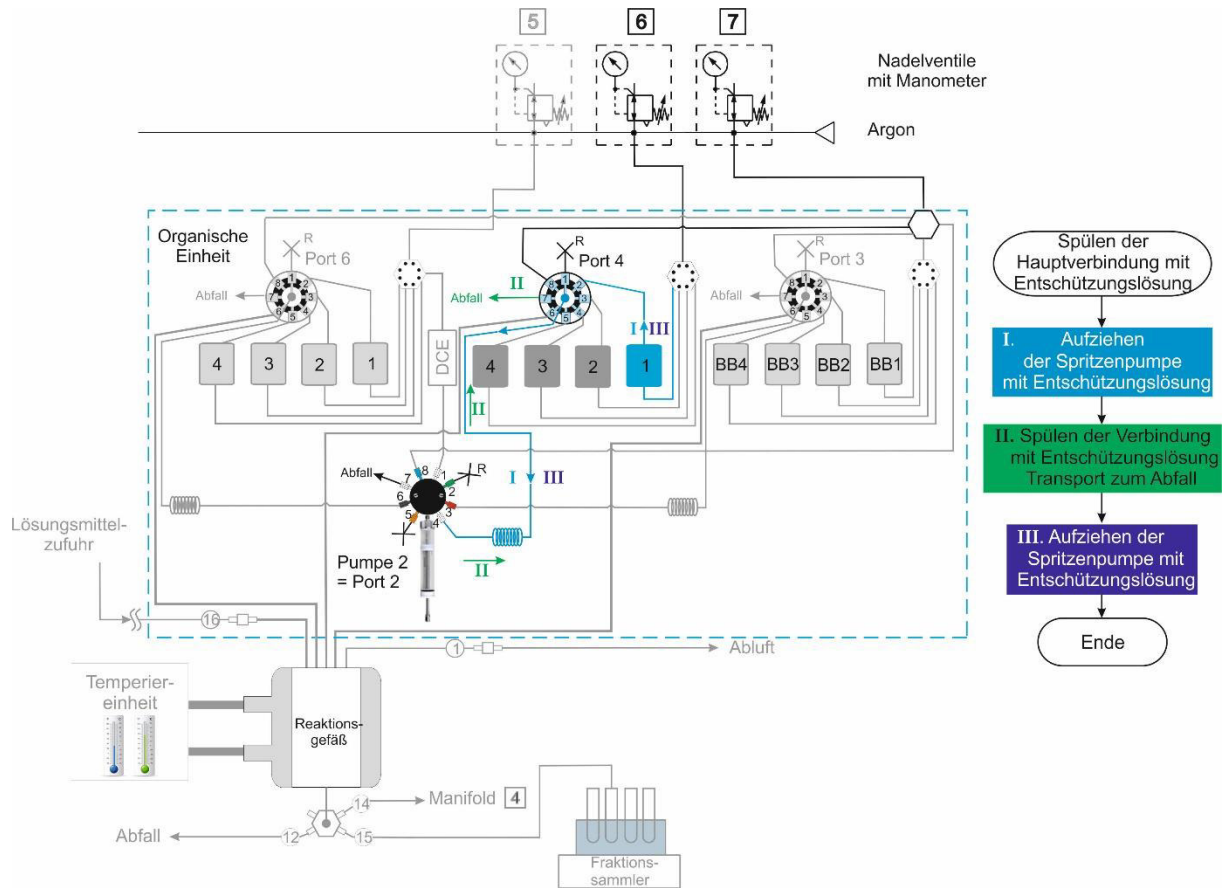


Abbildung 140: Funktionsweise des Moduls zum Spülen der Hauptverbindung mit Entschützungslösung. Zunächst wird Entschützungslösung mit der Spritzenpumpe aufgezogen, die über die Hauptverbindung zwischen dem Zulieferungs-Port und der Transporteinheit zur Unterbindung von Kontaminationen in den Abfallbehälter transportiert wird. Anschließend erfolgt das Aufziehen der Spritzenpumpe mit 2 mL der Entschützungslösung.

```

Fmoc_Deprotection_V1
1  //<Info: Header
13 //<Modul: Wash, DMF, 3x 3 ml, 25°C,Waste
36 //<Modul: Delivery, DMF, 3 ml, -, RV
47 //<Modul: Wash, deprotection line
85 //<Modul: Prime, line deprotection
86 PORT (2,4)   prime line deprotection 1
87 PORT (4,2)
88 RATE (2,1)
89 WITHDRAW (2,1000)
90 DELAY_SEC (2)
91 PORT (2,4)   deprotection 1 to waste
92 PORT (4,7)
93 RATE (2,10)
94 DISPENSE (2,1500)
95 DELAY_SEC (2)
96 PORT (2,4)   deprotection 1 into loop
97 PORT (4,2)
98 RATE (2,1)
99 WITHDRAW (2,2300)
100 DELAY_SEC (2)
101 //</Modul>
    
```

Abbildung 141: Funktionsmodul zum Spülen der Verbindung zwischen Spritzenpumpe 2 und Zuführung der Entschützungslösungen.

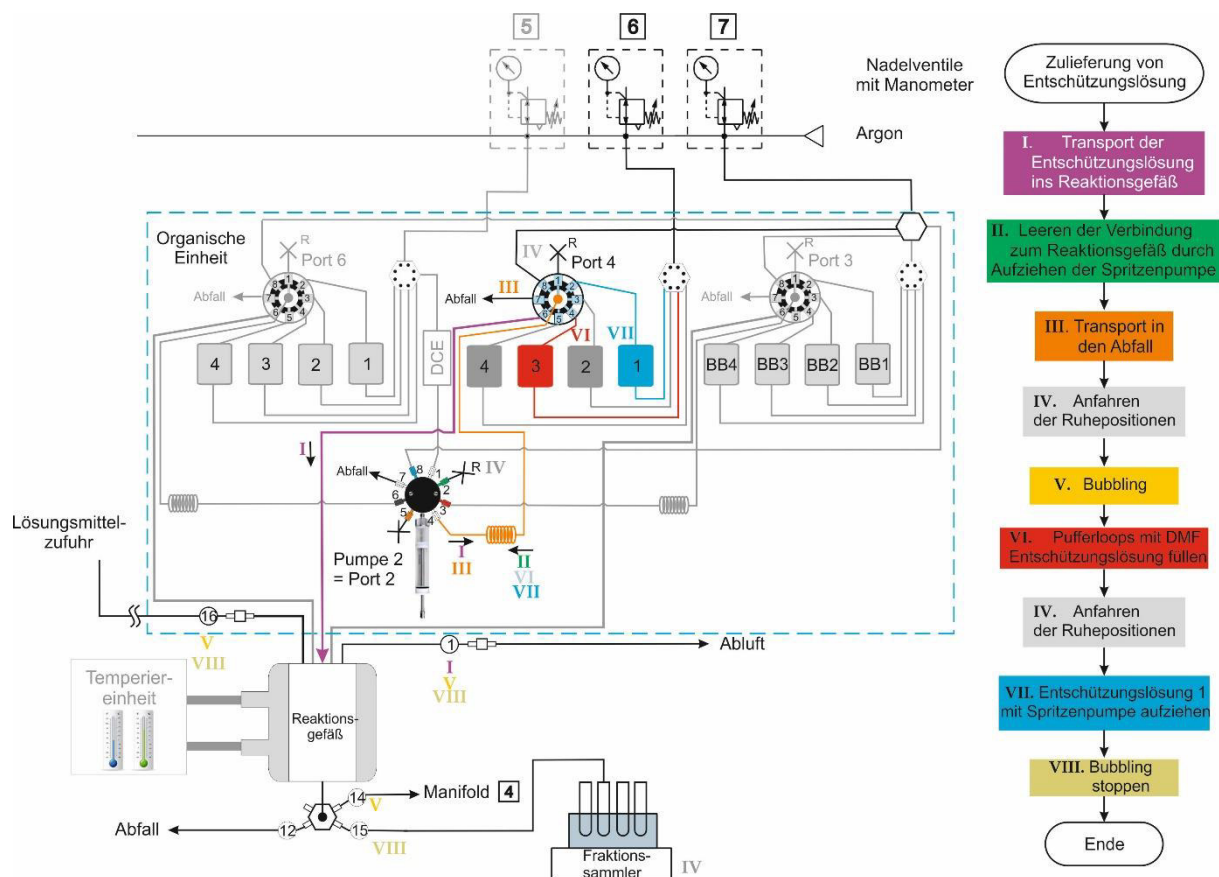


Abbildung 142: Funktionsmodul zum Transport einer definierten Menge Entschützungslösung ins Reaktionsgefäß, das bereits von der Spritzenpumpe aufgezogen war. Anschließend werden die Schläuche geleert und Restmengen in den Abfallbehälter befördert. Im Reaktionsgefäß werden die Reagenzien durch Argonzufuhr (Bubbling) durchmischt. Der Pufferloop wird mit DMF als Entschützungslösung gespült. Anschließend wird eine definierte Menge Entschützungslösung 1 von der Spritzenpumpe aufgezogen, die Argonzufuhr gestoppt und ein Teil der Lösung aus dem Reaktionsgefäß in den Fraktionssammler transportiert.

```

85  +; //<Modul: Prime, line deprotection
102 +; //<Modul: Empty, , 25°C, Waste
114 +; //<Modul: Delivery, Depl, ,-, Fraction
115 VALVE(1,ON) open outlet for delivery
116 PORT(2,4) 2 mL of deprotection 1 into rv
117 PORT(4,6)
118 RATE(2,1)
119 DISPENSE(2,2150)
120 DELAY_SEC(2)
121 PORT(2,4) empty rv line
122 PORT(4,6)
123 RATE(2,1)
124 WITHDRAW(2,500)
125 DELAY_SEC(2)
126 PORT(2,4) flush loop with DMF
127 PORT(4,7)
128 RATE(2,10)
129 HOME(2)
130 DELAY_SEC(2)
131 PORT(2,2) Resting positions
132 PORT(4,1)
133 DELAY_SEC(2)
134 VALVE(1,ON) bubbling rct for 5 min
135 VALVE(16,OFF)
136 VALVE(14,ON)
137 DELAY_SEC(300)
138 PORT(2,4) fill loop deprotection with DMF
139 PORT(4,4)
140 RATE(2,1)
141 WITHDRAW(2,1000)
142 DELAY_SEC(2)
143 PORT(2,2) Resting positions
144 PORT(4,1)
145 DELAY_SEC(2)
146 PORT(2,4) deprotection 1 into loop
147 PORT(4,2)
148 RATE(2,1)
149 WITHDRAW(2,2300)
150 DELAY_SEC(2)
151 VALVE(14,OFF) stop bubbling and solution to fraction collector
152 VALVE(1,OFF)
153 VALVE(15,ON)
154 VALVE(16,ON)
155 VALVE(11,ON)
156 DELAY_SEC(10)
157 VALVE(11,OFF)
158 VALVE(15,OFF)
159 VALVE(16,OFF)
160 DELAY_MS(100)
161 ;Modul>
    
```

Abbildung 143: Anweisungsliste des Funktionsmoduls: Zulieferung der Entschützungslösung 1 und Transport der Reaktionslösung in den Fraktionssammler

Für den ordnungsgemäßen Betrieb des Synthesizers sind Funktionstests vor Durchführung einer Synthese, aber auch regelmäßige Wartungsarbeiten, z. B. Durchspülen der Transportleitungen, notwendig. Diese häufig genutzten Funktionsmodule sind unter der Registerkarte „Moduleselector“ auf Schaltflächen gelegt, die nach Betätigung direkt ausgeführt werden. In diesem Bereich sind alle Inbetriebnahme- und Kalibrierungs-Funktionen eingepflegt, die für die Wartung, das Testen und die Vorprüfungen hilfreich sind.

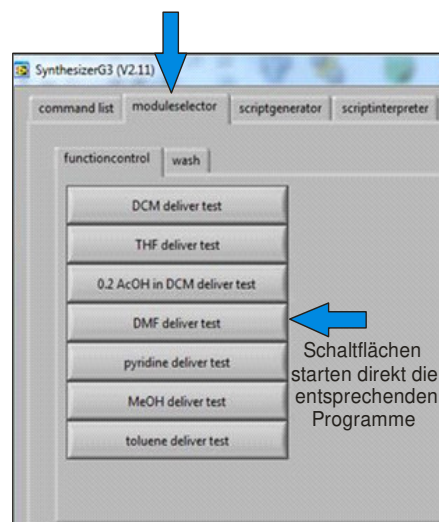


Abbildung 144: Schnellzugriff auf Standardfunktionen zur Funktionskontrolle oder für Wartungsarbeiten, die direkt ausgeführt werden.

Zur individuellen, flexiblen Anpassung der Benutzeroberfläche nach Anwenderbedürfnissen, können unter dem „Moduleselector“ eigene Favoriten-Funktionen auf Schaltflächen gelegt werden. Dazu wurde ein Mechanismus implementiert, der es ermöglicht, die Favoriten-Schaltflächen, ohne programmatischen Eingriff zu erzeugen. Diese direkt ausführbaren Schaltflächen ergeben sich durch Dateien innerhalb der Ordnerstruktur des „Buttons“- Verzeichnisses, welche automatisch in den zugehörigen Registerkarten des „Moduleselectors“ generiert werden und dadurch sehr einfach zu pflegen sind (Abbildung 145).

Nach Implementierung der Benutzerverwaltung ist es hier möglich, in Abhängigkeit des angemeldeten Benutzers und dessen Anwendungsbereich komplett unterschiedliche individuelle Funktionalitäten bereitzustellen.

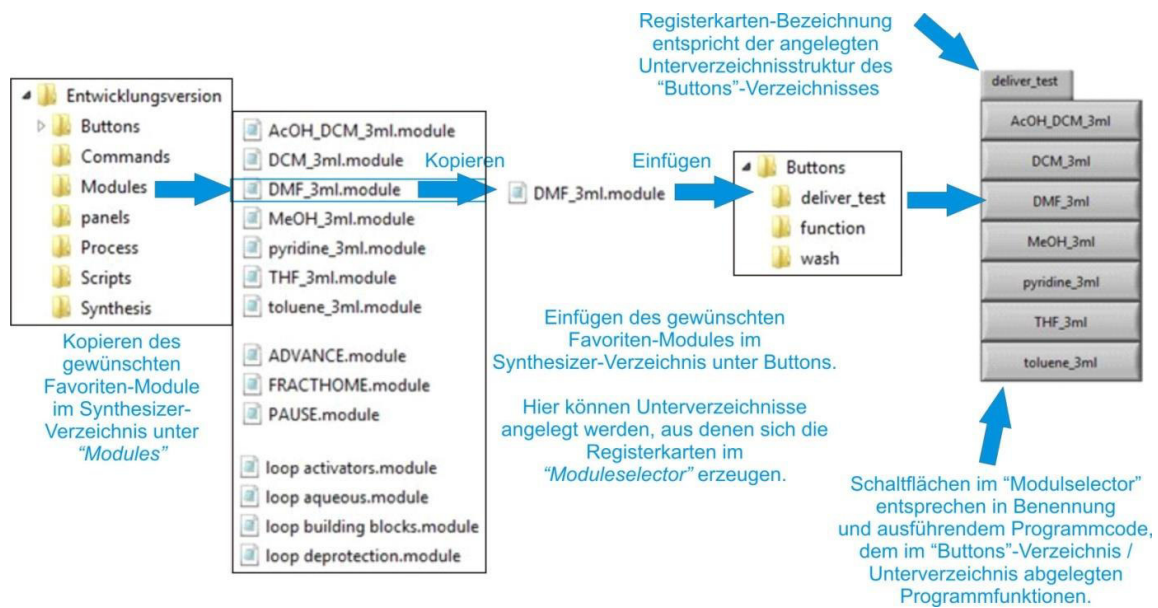


Abbildung 145: Anlegen von direkt ausführbaren benutzerindividuellen Favoriten-Funktionen auf Schaltflächen innerhalb der Synthesizer-Benutzeroberfläche durch Kopieren des gewünschten Programm-Moduls und Einfügen in das Buttons – Verzeichnis. So lässt sich ohne Programmieraufwand Einfluss auf die individuelle Funktionalität der Bedienoberfläche der Synthesizer-Applikation nehmen.

4.3.3 Prozesse

Bislang bestanden die Steuerskripte des Synthesizers aus statischen Befehlsanweisungen. Viele daraus resultierende Programm-Module unterscheiden sich in ihrer Funktionalität jedoch oftmals nur durch einige unterschiedliche Parameter. In der Synthesizer-Software wurden die Möglichkeiten der Prozessoptimierung mit Hilfe flexibler Parametereingaben weiterentwickelt. Am Beispiel der Lösungsmittelzufuhr zum Reaktionsgefäß erfolgt nun die Umsetzung eines statischen Skripts in einen dynamischen Prozess.

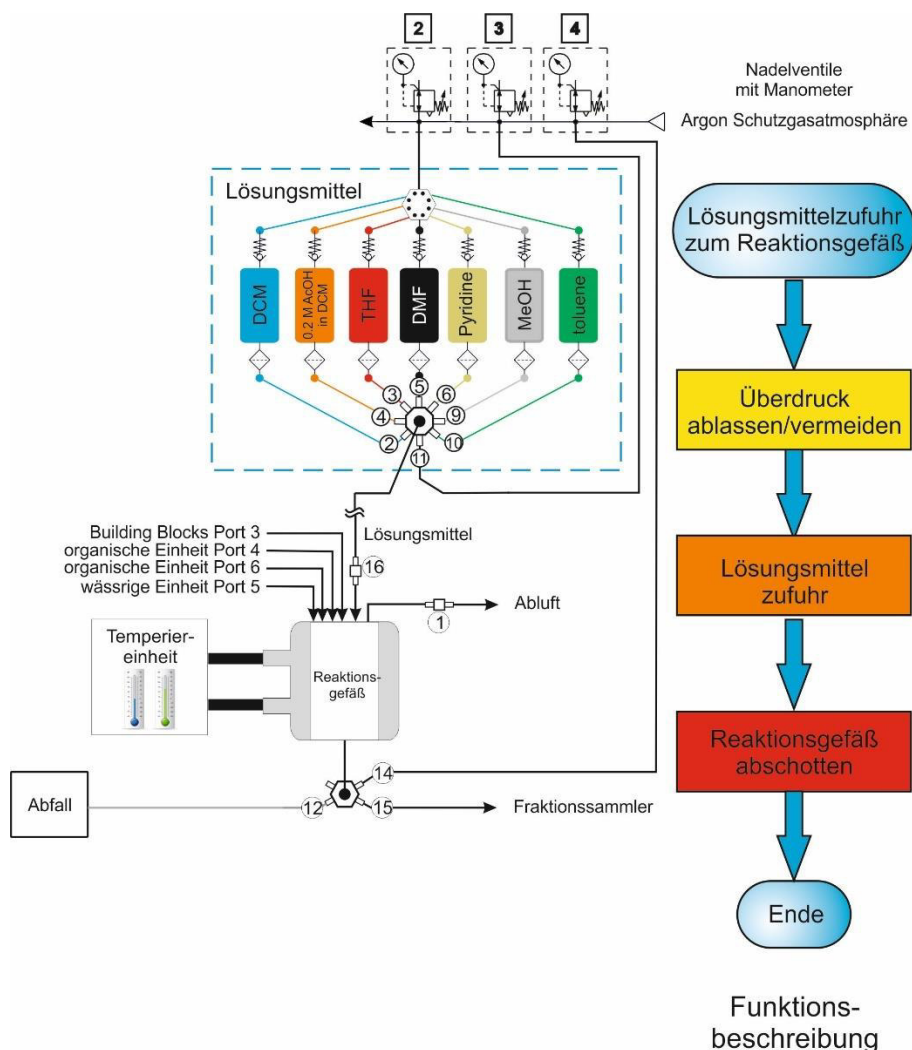


Abbildung 146: Schematischer Ablauf für eine Lösungsmittelzufuhr zum Reaktionsgefäß. Für jedes der am Synthesizer verwendeten Lösungsmittel ist ein eigenes Ventil zuständig. So leitet Ventil 2 das Lösungsmittel DCM in die Transportleitung zum Reaktionsgefäß und Ventil 5 transportiert DMF. Ventil 16 öffnet die Zufuhr zum Reaktionsgefäß.

Aktuell stehen sieben verschiedene Lösungsmittel für die Reaktionen bereit. Für eine Lösungsmittelzufuhr von 3 mL DCM ins Reaktionsgefäß besteht das Programm-Modul aus folgenden Vorgängen: Schaltung von Ventil 1 zur Vermeidung bzw. Ablassen des Überdrucks im Reaktionsgefäß. Öffnen von Ventil 16 zur Freigabe der Lösungsmittelzufuhr ins Reaktionsgefäß. Schaltung von Ventil 2 zum Öffnen der Verbindung zum Lösungsmittel DCM. Die Wartezeit von 3000 ms entspricht dabei

der Mengenzufuhr von 3 mL des Lösungsmittels, das sich aus Schlauchdurchmesser, Schlauchlänge und Transportgeschwindigkeit ergibt. Mit dem Schließen der Ventile endet der Funktionsschritt.

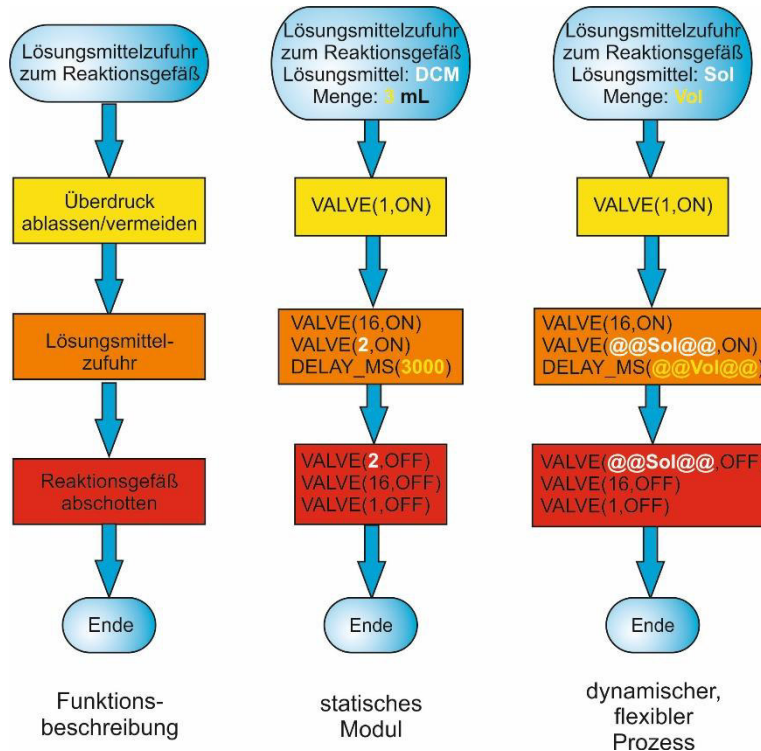


Abbildung 147: Die allgemeine Funktionsbeschreibung führt zur spezifizierten Umsetzung als statisches Modul, das individuell für jedes Lösungsmittel und für jede Mengenänderung angepasst werden muss. Daraus abstrahiert sich der dynamische Prozess, bei der Parameter für die jeweiligen Lösungsmittel und Mengenangaben Verwendung finden. Die Parameter dienen als Platzhalter-Variablen für die spezifischen Angaben, die später in den ausführbaren Programmcode ersetzt werden. Dadurch verringern sich die benötigte Module der Lösungsmittelzufuhr für jedes der sieben Lösungsmittel * x mL Mengenangaben auf einen einzigen Prozess.

Derzeit werden für jedes der Lösungsmittel eigene statische Module verwendet. Ändert sich die Menge des zugeführten Lösungsmittels, müssen die Änderungen im Programm-Modul durchgeführt werden, was wieder zu weiteren zusätzlichen statischen Modulen führt.

Abstrahiert man diese Funktionalität und ersetzt die veränderlichen Bestandteile des Moduls durch flexible Parameter, die später im Programmcode als „Schlüsselwort“ in Form **@@Parameterbezeichnung@@** durch die gewünschten spezifischen Daten ersetzt werden, reduziert sich die Vielzahl an ähnlichen Programm-Bestandteilen auf einen einzelnen flexiblen Prozess.

Um die veränderlichen Bestandteile des Prozesses mit den gewünschten Werten zu ersetzen, benötigt der Prozess eine zusätzliche Eingabe- bzw. Auswahlmöglichkeit der vorhandenen Prozessdaten. Am komfortabelsten geschieht dies durch einen zusätzlichen Abfragedialog. In einer weiteren Abstrahierung der Synthesprache, erfolgt die Verwaltung von systemrelevanten Daten, wie die Ventizuordnung der Lösungsmittel oder Skalierungsfaktoren (z. B. für die transportierte Lösungsmittelmenge) nun durch die Applikation. Der Benutzer kann damit direkt und ohne

Systemkenntnis die erforderlichen Syntheseparameter eingeben. Der Austausch der Schlüsselparameter mit den gewählten Angaben, sowie die Erzeugung des ausführbaren Programmcodes erfolgt automatisch mit der Prozessbearbeitung.

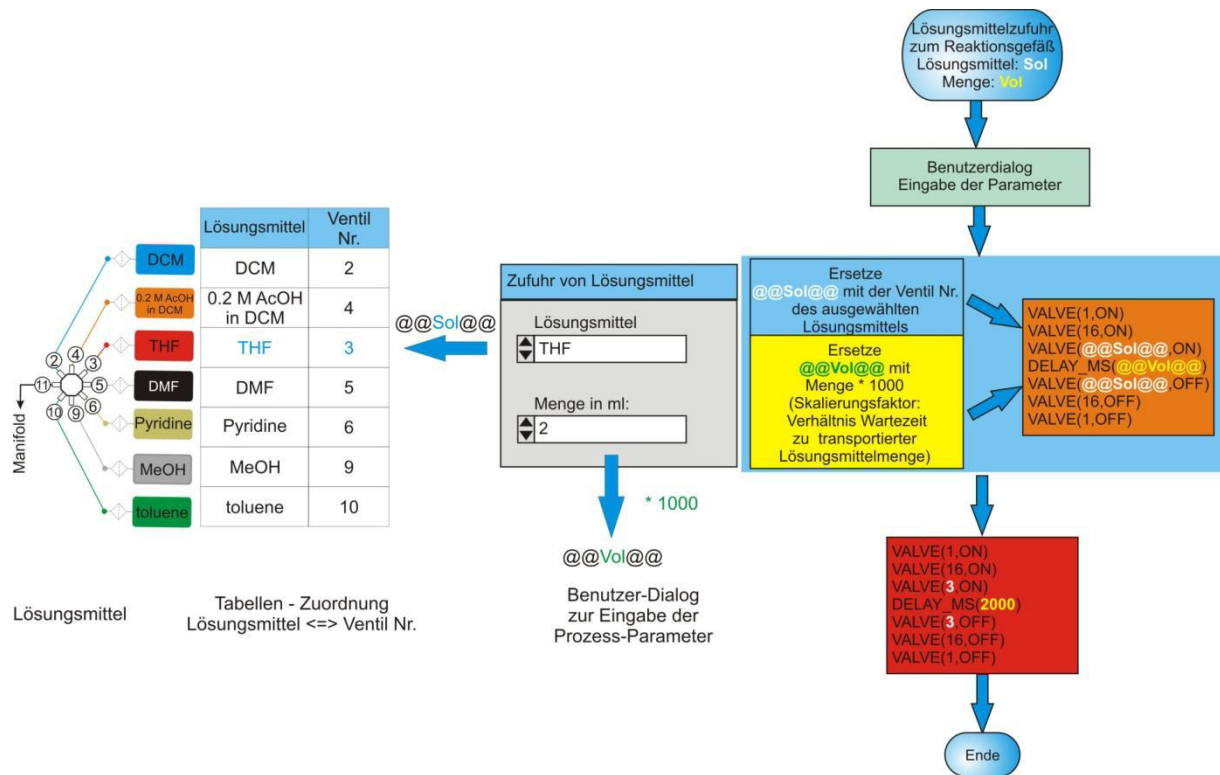


Abbildung 148: Schematischer Programmablauf für den dynamischen Prozess der Lösungsmittelzufuhr. Lösungsmittelauswahl und Zufuhrmenge erfolgt über einen Benutzerdialog. Eine hinterlegte Zuordnungstabelle verwaltet das dem Lösungsmittel zugeordnete Ventil. Der Benutzer kann direkt die Zufuhrmenge angeben, ein Skalierungsfaktor berechnet die Transport- bzw. Wartezeit. In der Programmabarbeitung werden die „Schlüsselwörter“ durch die interpretierten Parameter aufgefüllt und in ausführbaren Programmcode überführt.

Jedes statische Modul oder Skript mit Parametern (z. B. Schleifenzahl, Zeiten, Temperaturen, Lösungsmittel, Building Blocks) lässt sich zu einem dynamischen Prozess generalisieren. Da der ausführbare Programmcode als reine ASCII-Datei in den Synthesizer-Verzeichnissen vorliegt, können nachfolgend beschriebenen Änderungen in jedem üblichen Texteditor mit der Funktion „Suchen und Ersetzen“ vergleichbar durchgeführt werden. Im Beispiel wird die bislang statisch eingesetzte Temperatur von -15°C durch Ersetzen mit dem String @@T1@@ zu einem Parameter. Weitere Prozessparameter werden analog ersetzt. Für die Aufrechterhaltung der Programmintegrität ist jedoch auf Eindeutigkeit bei der Namensvergabe für die Schlüsselwörter der variablen Parameter zu achten.

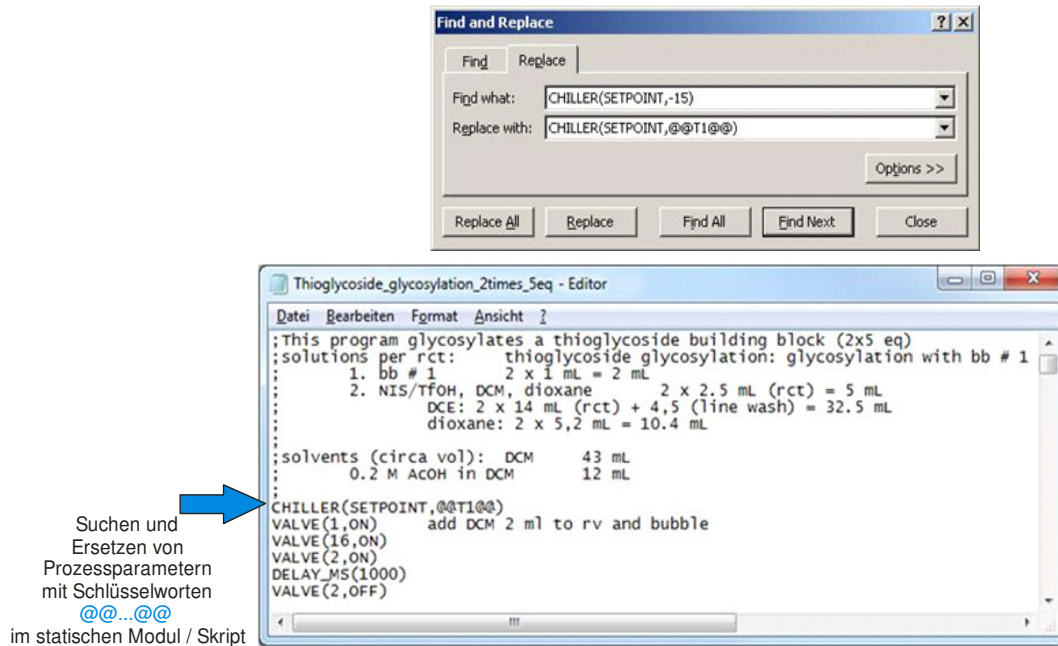


Abbildung 149: Konvertieren eines statischen Skripts zu einem dynamischen Prozess durch Austauschen der Prozessparameter mit @@Parameterbezeichnung@@ durch die Funktion „Suchen und Ersetzen“. Der Programmcode für Module / Skripts liegt als ASCII – Datei vor und kann somit in jedem beliebigen Texteditor geöffnet/bearbeitet werden.

Damit der dynamische Programmteil in der Synthesizer-Benutzeroberfläche als Prozess erkannt wird, erfolgt das Abspeichern des konvertierten Skripts/Modules unter der Registerkarte *Process* im Synthesizer-Verzeichnis mit der neuen Dateieindung **.process*.

Der prozessindividuelle Benutzerdialog zur Parameterabfrage lässt sich mit Hilfe des Express-VIs für den Anwenderdialog in LabVIEW gestalten (siehe Kapitel 3.2.1, Seite 64). Diese können auch vom Anwender ohne umfangreiche LabVIEW Erfahrung individuell konfiguriert und parametrisiert werden. Im Prozessdialog in Abbildung 150 werden die Parameter, z.B. T1 für das erste Temperaturniveau und Parameter t1 für die dazugehörige Haltezeit, mit den gewünschten Eingabe-Elementen verknüpft. Das Aufstarten des Parameter-Dialogs erfolgt automatisch mit dem Starten des dynamischen Skriptes. Der hier zugrunde liegende Mechanismus verlangt seitens des Anwenders die Bereitstellung eines Prozess-VIs mit dem gleichen Namen wie die dazugehörige Prozess-Datei im Unterverzeichnis „panels“.

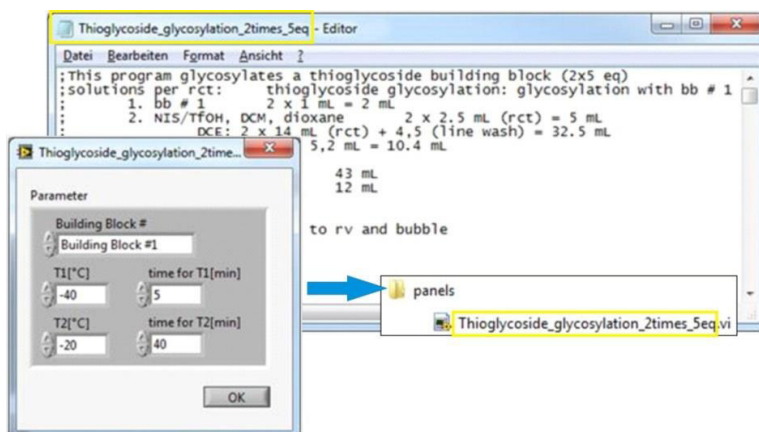


Abbildung 150: Das dynamische Prozess-Skript wird mit dem korrespondierenden Benutzerdialog zur Parametereingabe über den gleichlautenden Namen verknüpft. Das Prozess-VI muss dafür im Unterverzeichnis „panels“ des Synthesizer-Verzeichnisses abgespeichert werden. Für diesen Beispielprozess sind sowohl die zugeführten Building Blocks als auch zwei Reaktionstemperaturen und deren dazugehörige Haltezeit variabel.

4.3.4 Synthesesteuerung mit der neuen Synthesizergeneration

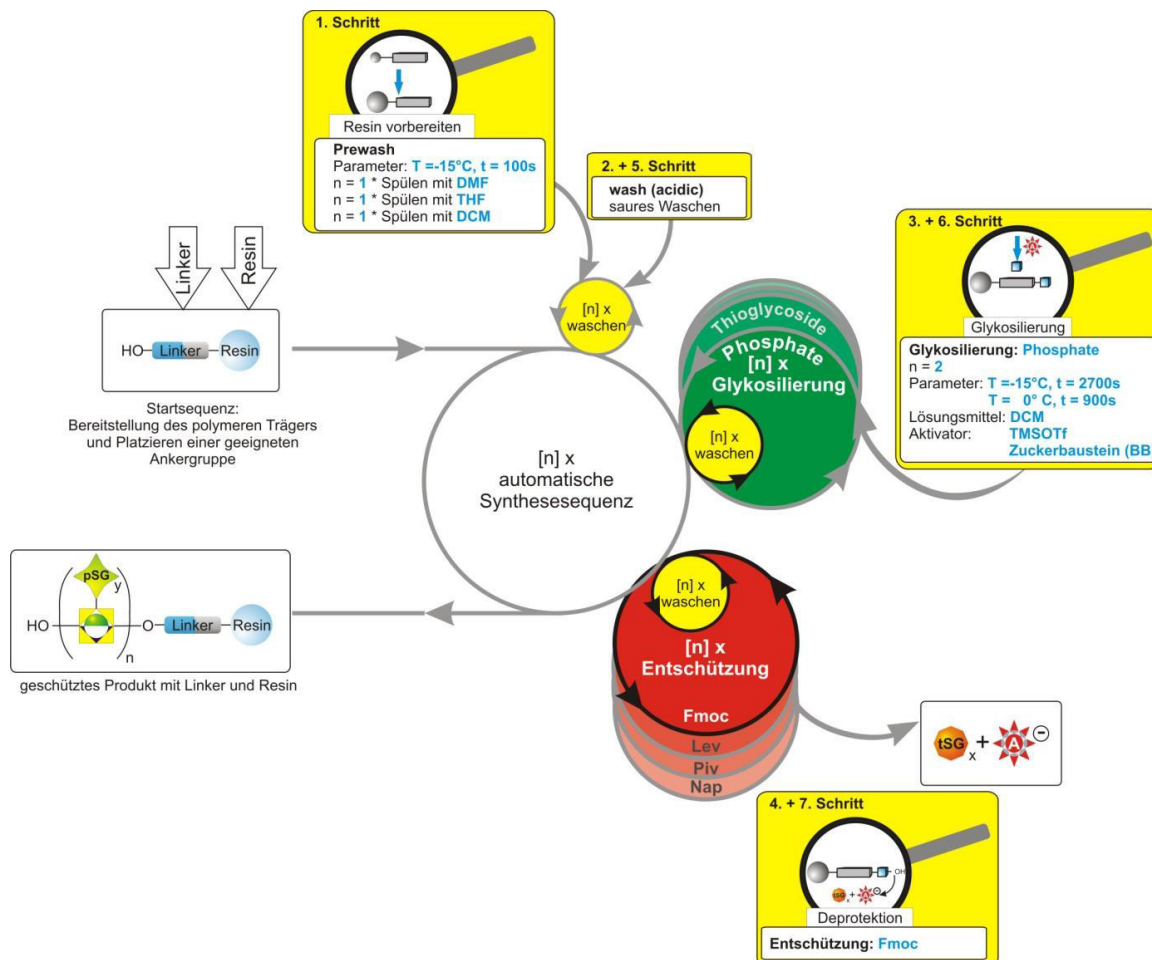


Abbildung 151: Schematisch programmatischer Ablauf einer Disaccharid-Synthese mit Prozessparametern. Die einzelnen Syntheseschritte bilden sich in der Benutzeroberfläche mit jeweils eigenen Prozess-Skripten ab.

Aus den vorgenannten Elementen (Befehle, Module, Skripts, Prozesse) lassen sich die gesamten Synthesen zusammenstellen. Nachfolgend gezeigt ist das Beispiel einer Disaccharid-Synthese. Beim

Aufstarten der Software werden die einzelnen Hardwarekomponenten automatisch initialisiert und die Kommunikationskanäle getestet. Der Synthesizer ist nach dieser internen Prüfung betriebsbereit.

Da auch diese Synthesen vom Benutzer bearbeitet werden können, sind diese jeweils zur Protokollierung unter sinnvollem Namen abzuspeichern. Daraus resultierende Textdateien erhalten standardmäßig die Endung *.syn zugewiesen und stehen, wenn sie im entsprechenden Sequenzverzeichnis gelistet sind, den zukünftigen Bedienern direkt über die Registerkarte Synthesis zur Auswahl zur Verfügung. Damit wird das Reproduzieren von bereits durchgeführten Synthesen sehr einfach möglich.

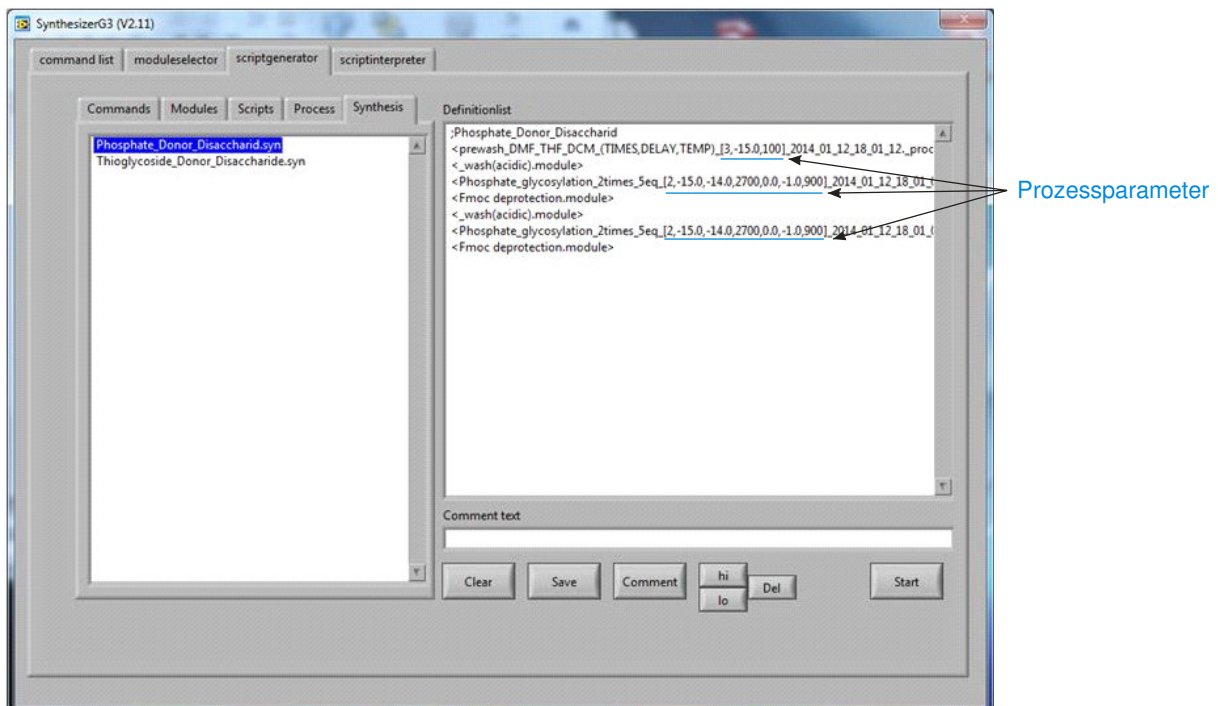


Abbildung 152: Das Syntheseskript zur automatischen Synthese eines Disaccharids besteht aus den einzelnen dynamischen Prozess-Schritten. Die jeweiligen Prozessparameter erschließen sich direkt aus der Befehlszeile.

Im gezeigten Beispielskript für die Disaccharid-Synthese wird zuerst das Resin mit verschiedenen Lösungsmitteln gespült (Abbildung 152). Die Lösungsmittelauswahl erfolgt in dynamischen Prozess-Skripten. Hier übernehmen die zuvor gezeigten Zuordnungstabellen die hardwaretechnischen Parameterzuordnungen (z. B. Lösungsmittel zu Ventil-Nr.). Der nachfolgende saure Waschprozess schafft einen definierten Ausgangszustand für die Reaktion. Bei den Synthesen ist neben den Building Block-Informationen, die Temperierung und auch die entsprechenden Wartezeiten durch den Prozessaufruf eindeutig protokolliert. Die Prozessparametereingabe erfolgt auch hier bequem über Parameter-Dialoge. Da die Fmoc Abspaltung standardisiert ist, sind keine weiterführenden Parameter aufgelistet. Gleiches gilt für den erneuten sauren Waschvorgang. Der zweite Kopplungsvorgang enthält die gleichen Parameterzuweisungen wie zuvor. Im letzten Schritt erfolgt wiederum die Fmoc Abspaltung.

Ein bereits erfolgreich abgelaufenes Syntheseskript kann auf der Registerkarte Skriptinterpreter über die Dateiauswahl selektiert werden. Der zugrundeliegende reduzierte Syntheseablauf wird daraufhin visualisiert. Die erneute Abarbeitung der Synthese kann auf der Oberfläche der Anweisungsliste mit der „Start“- Schaltfläche direkt gestartet werden. Hierbei wird der reduzierte Synthesetext in die zugrundeliegenden Hardwarekommandos transferiert und ermöglicht die individuelle Kontrolle der aktuell ausgeführten Funktionszugriffe.

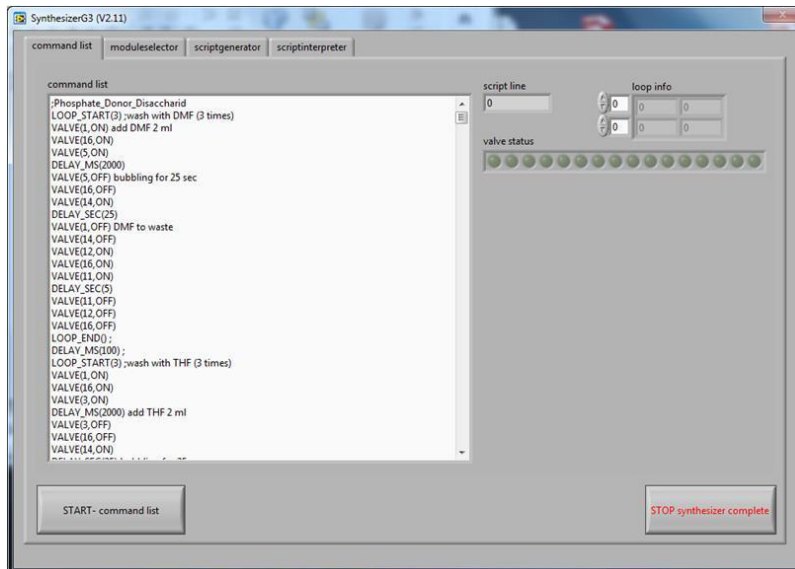


Abbildung 153: Anweisungsliste des selektierten Syntheseskripts kann direkt mit START ausgeführt werden

Alle der angesprochenen funktionalen Einheiten (Synthesen, Prozesse, Skripte, Module) können seitens des Benutzers auch offline zusammengestellt werden. Die daraus resultierenden reinen ASCII Textdateien müssen nur in den dazugehörigen Verzeichnisstrukturen der jeweiligen Systeme durch Kopieren bereitgestellt werden und stehen dann in den Auswahllisten zur Verfügung. Dies geschieht durch eine sofortige Aktualisierung beim Wechsel der jeweiligen Registerkarte (vgl. *Anhang 2.15.*).

Aufgrund der positiven Erfahrung mit dem ersten vollständig neu aufgebauten Synthesizer wurden auch die beiden älteren Systeme auf den gleichen technischen Stand gehoben. Da die einzelnen Systeme mit unterschiedlicher Hardware aus dem vorhandenem Laborgerätepool zusammengestellt wurden, unterscheidet sich die jeweilige Gerätezusammenstellung in Funktion und Parametrierung. So besitzen die Automatisierungsrechner teilweise unterschiedliche Schnittstellenbelegung und auch die Betriebssysteme sind verschieden. Dennoch ist die komplette Austauschbarkeit der Synthesizer-Software, sowie weiterer Dateien, z. B. individuelle Benutzeroberflächen, gewährleistet. Die individuelle Anpassung an die jeweilige Hardware-Umgebung wird in lokal vorhandenen Konfigurationsdateien (*.ini-Datei) verwaltet.

4.4 Synthesen an der neuen Gerätegeneration

Obwohl sich die Hardwarekomponenten der drei Synthesizer neuester Generation voneinander unterscheiden, lassen sich die Syntheseprogramme durch hardware-sensitive Initialisierung geräteübergreifend ausführen. Der implementierte Befehlsinterpreter gewährleistet die Kompatibilität mit den Syntheseprogrammen des Synthesizers der Vorläufergeneration. Damit konnten sich die neuen Arbeitssynthesizer bereits während der Projektzeit als vollwertige Laborsysteme etablieren. Alle drei Arbeitssysteme bewiesen in parallel arbeitenden Synthesen ihre chemische Vergleichbarkeit und Reproduzierbarkeit. Seit Bereitstellung des ersten Arbeitssynthesizers ist die Anzahl an durchgeführten Synthesen drastisch gestiegen. Nachdem die Protokollierung der einzelnen Runs ab Ende 2012 eingeführt wurde, summierten sich bis Mitte 2016 bereits 7993 Ausführungen/Nutzung des Applikationsprogramms an allen Arbeitssynthesizern.

Synthese von Arabinofuranosid-Oligosacchariden

Nach intensiven Geräte- und Softwaretests, bewies der erste Arbeitssynthesizer mit der von J. Kandasamy et al. erfolgreich durchgeführten Synthese von Arabinofuranosid-Oligosacchariden^[172] seine Leistungsfähigkeit.

Oligoarabinofuranoside eignen sich als Werkzeug zur Untersuchung biosynthetischer Enzyme, die an der mykobakteriellen Zellwandsynthese beteiligt sind.^[176] Die Herstellung von Arabinofuranose Fragmenten und Analoga ist jedoch anspruchsvoll, da sie viele diskrete Operationen und mehrere Reinigungsschritte erfordert. Die Synthese des Arabinandodecasaccharids der Gruppen von Lowary und Ito wurden über einen konvergenten Fragmentansatz erreicht.^[177]

Mit der Synthese von einfachen linearen (**41**, **42**) und verzweigten (**43**, **44**) Strukturen von Oligoarabinosid mit Arabinofuranose (Araf)-Resten, wurde nun erstmals die Möglichkeit der automatisierten Herstellung dieser Strukturen mit dem neuen Arbeitssynthesizer getestet. Das lineare α -D- (1-5) -Araf **41** und **42** wurden über iterative Glykosylierungen unter Verwendung eines Thioglykosid -Bausteins **45** mit 9-Fluorenylmethoxycarbonyl an C-5 als temporäre, und Benzoaten an C-2 und C-3 als teilnehmende, Schutzgruppen hergestellt (Abbildung 154). Die Glykosylierung erfolgte bei Temperaturen zwischen $-40\text{ }^{\circ}\text{C}$ und $-20\text{ }^{\circ}\text{C}$ mit einer Mischung aus N-Iodsuccinimid und Trifluormethansulfonsäure als Aktivator.

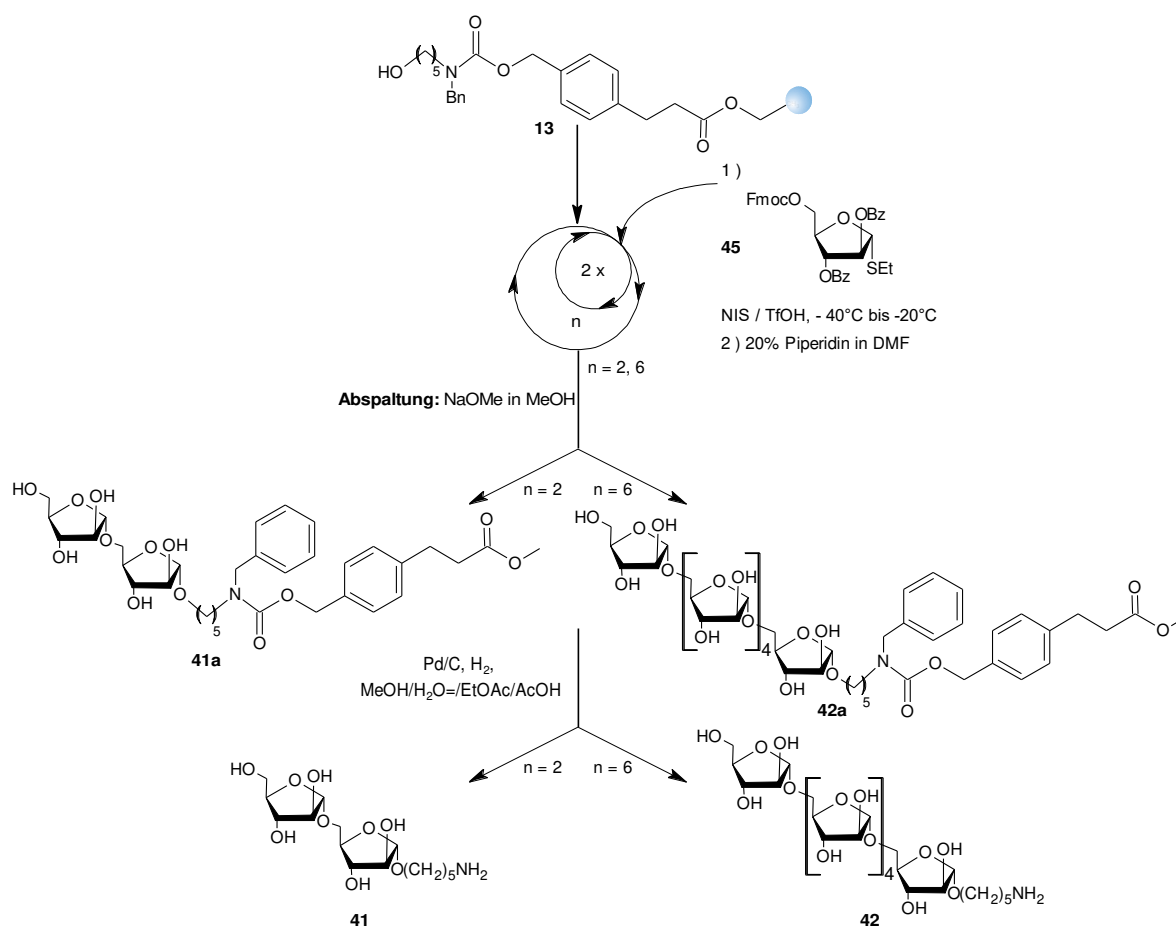


Abbildung 154: Automatische Synthese linearer Di- und Hexa-Arabinofuranoside unter Verwendung des neuen vollautomatischen Arbeitssynthesizers.

Jeder Zyklus wurde als Doppelkupplung unter Verwendung von fünf Äquivalenten des Bausteins **45** durchgeführt. Die Fmoc-Gruppe wurde mit Piperidin entfernt und legte die zur Verlängerung bereitstehende Hydroxylgruppe frei. Der Glykosylierungs - Entschützungszyklus wurde zweimal für die Synthese von Disaccharid **41** und sechs Mal für die Herstellung von Hexasaccharid **42** wiederholt. Nach Abspaltung vom Harz und Reinigung durch präparative Umkehrphasen-HPLC, Hydrogenolyse, gefolgt von einer einfach Sep-Pak-Reinigung ergab sich das Disaccharid **41** und Hexasaccharid **42** mit einer Gesamtausbeute 85% bzw. 65%, bezogen auf die Harzbeladung.

Die Synthese der α - (1-3) -verzweigten Oligo-Arabinofuranoside **43** und **44** erfolgte unter Verwendung des Bausteins **46**, dessen zwei Fmoc-Gruppen die Entfernung von temporärem Schutz und Glykosylierung sowohl der C-3- als auch C-5-Hydroxylen in einem einzigen Entschützungs-Glykosylierungs-Zyklus ermöglichen (Abbildung 155). Nach dem Beladen des Harzes mit dem Baustein **46** unter Verwendung des Glykosylierungs-Entschützungs-Zyklus, wurde die erste Verlängerung der Synthese von Oligo-Arabinofuranosid **43** mit dem Thioglykosid **45** unter den Synthesebedingungen von **41** und **42** durchgeführt. Unter Verwendung des Thioglykosids **45** wurde die folgende Glykosylierung zur Erzeugung der α - (1-5) -Bindung und dem Einbau der α - (1-3) -Verzweigung durchgeführt. Nach Entfernen der Fmoc-Gruppen, Abspaltung vom Harz, Aufreinigung

durch Adsorption auf einem Kieselgelkissen, gefolgt von Waschen mit Hexan und CH₂Cl₂, sowie Elution mit einem Gemisch von CH₂Cl₂: MeOH, und anschließender Hydrogenolyse, ergab sich das entschützte Trisaccharid **43** in 78% Gesamtausbeute bezogen auf die Harzbeladung.

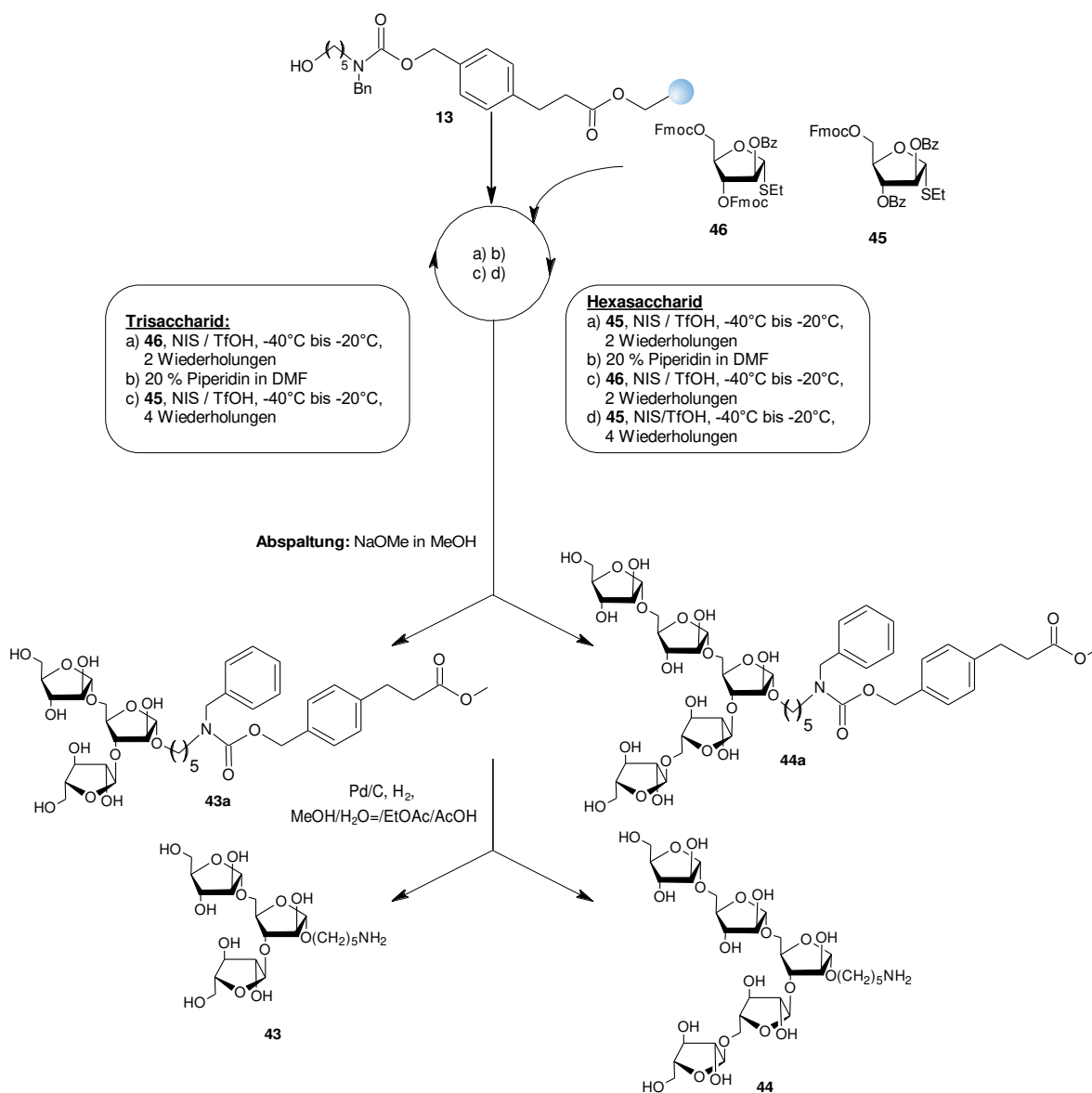


Abbildung 155: Automatische Synthese von verzweigten Tri- und Hexasaccharide unter Verwendung des neuen vollautomatischen Arbeitssynthesizers.

Die automatisierte Synthese des verzweigten Hexasaccharids **44** wurde über vier Glykosylierungszyklen unter Verwendung der Bausteine **45** und **46** durchgeführt. Nach Abspaltung vom Harz, dem Reinigen mit Hilfe einer Silikat-Filterplatte, und anschließender Hydrogenolyse, konnte das Amin-funktionalisierte Hexasaccharid **44** in einer Gesamtausbeute von 63%, bezogen auf die Harzbeladung, hergestellt werden. Die automatische Synthese des teilweise geschützten Hexasaccharids **42** benötigte 16, des Hexasaccharids **44** aufgrund des vereinfachten Reinigungsprotokolls 14 Stunden.^[172] Die Synthesen konnten ohne Unterbrechungen kontinuierlich

innerhalb der technischen Vorgaben durchgeführt werden und bewiesen somit erstmals im praktischen Einsatz die Leistungsfähigkeit des neuen Synthesizer-Konzepts.

Synthese von Dermatansulfat-Oligosacchariden

Das automatisierte Protokolle einen schnelleren Zugang zu strukturell definierten Oligosacchariden erlauben, wurde auch mit Synthesen von Glykosaminoglykanen (GAG) wie Hyaluronan^[104] und Chondroitinsulfat^[178] am Synthesizer der alten Generation gezeigt. Mit der Herstellung konjugationsfähiger Dermatansulfat-Oligosaccharide durch J. Kandasamy et al., konnten nun auch am neuen Arbeitssynthesizer Erfahrungen in der Herstellung von Glykosaminoglykan-Polysacchariden gesammelt werden.^[179]

Dermatansulfat, das vorwiegend in der Haut vorkommt und in vielen Säugetiergeweben vorhanden ist, erfüllt als Teil der extrazellulären Matrix eine Vielzahl biologischer Funktionen.^[180] Die alternierenden Disaccharideinheiten aus N-Acetyl-D-galactosamin (GalNAc) und L-Iduronsäure (IdoA) können variabel an den C-4- und C-6-Positionen in GalNAc und der C-2-Position in IdoA O-sulfatiert sein. Abhängig von der Quelle sind 63-97% des Dermatansulfatpolymers aufgebaut von IdoA - GalNAc4SO₃-Disaccharid-Wiederholungseinheiten, die als charakteristisch für Dermatansulfat angesehen werden.^[181]

Bisherige manuelle Syntheseverfahren wurden in Lösung durchgeführt.^[182] Bei der automatisierten Festphasensynthese wurden die Dermatansulfat-Oligosaccharide mit den orthogonal geschützten D-Galactosamin (GalN) **48**- und L-Iduronsäure **49** – Bausteinen und dem fotospaltbaren Linker **47** am neuen Arbeitssynthesizer hergestellt.

Der GalN – Baustein **48** wurde mit der temporären 9-Fluorenylmethoxycarbonyl-Aminoschutzgruppe (Fmoc) an C-3 und mit Levulinylester (Lev) an der C-4-Position maskiert. Levulinylester (Lev) kann unter milden Bedingungen selektiv abgespalten werden und ermöglicht so eine Sulfatierung der freien Hydroxylgruppe. Die GalN-Aminogruppe wurde als Trichloracetamid geschützt, um gute β -Selektivitäten in Glykosylierungsreaktionen sicherzustellen und die Installation des gewünschten Acetamids durch direkte Reduktion im späten Stadium der Synthese zu ermöglichen.

Die C-4-Position des IdoA - Bausteins **49** wurde mit der temporären Fmoc -Schutzgruppe versehen und die C-2-Hydroxylgruppe als Benzoatester geschützt, um eine α -Selektivität bei Glycosylierungsreaktionen sicherzustellen. Die restlichen Hydroxylgruppen, die während des Zusammenbaus nicht modifiziert wurden, wurden als Benzylether dauerhaft geschützt.

Die automatisierte Synthese von Disaccharid **50** erfolgt im ersten Glykosylierungs-Entschützungszyklus unter Verwendung von Baustein **48**, gefolgt vom zweiten Glykosylierungs-Entschützungszyklus unter Verwendung von Baustein **49** (Abbildung 156). Die resultierende freie

Hydroxylgruppe am terminalen Zucker wurde auf dem Synthesizer unter Verwendung von Essigsäureanhydrid in Pyridin acetyliert. Die Abspaltung vom Harz erfolgte mittels UV-Licht durch Anwendung eines Foto-Flow-Reaktors.^[178] Nach Reinigung des Rohproduktes unter Verwendung der HPLC, konnte das Disaccharid **50** mit 66% Gesamtausbeute bezogen auf die Harzbeladung in sechs Schritten hergestellt werden. Das Tetrasaccharid **51** wurde durch abwechselnde Verwendung der Bausteine **48** und **49** über vier Glykosylierungs- und Entschützungszyklen, gefolgt von einer Acetylierung der freien Hydroxylgruppe am terminalen Zucker, hergestellt. Nach der Abspaltung vom Harz und Aufreinigung des Rohproduktes mittels HPLC, konnte das Produkt **51** mit 28% Gesamtausbeute bezogen auf die Harzbeladung in zehn Schritten, gewonnen werden. Deletionssequenzen wurden bei der automatisierten Synthese von **50** und **51** nach Beurteilung durch HPLC nicht festgestellt.

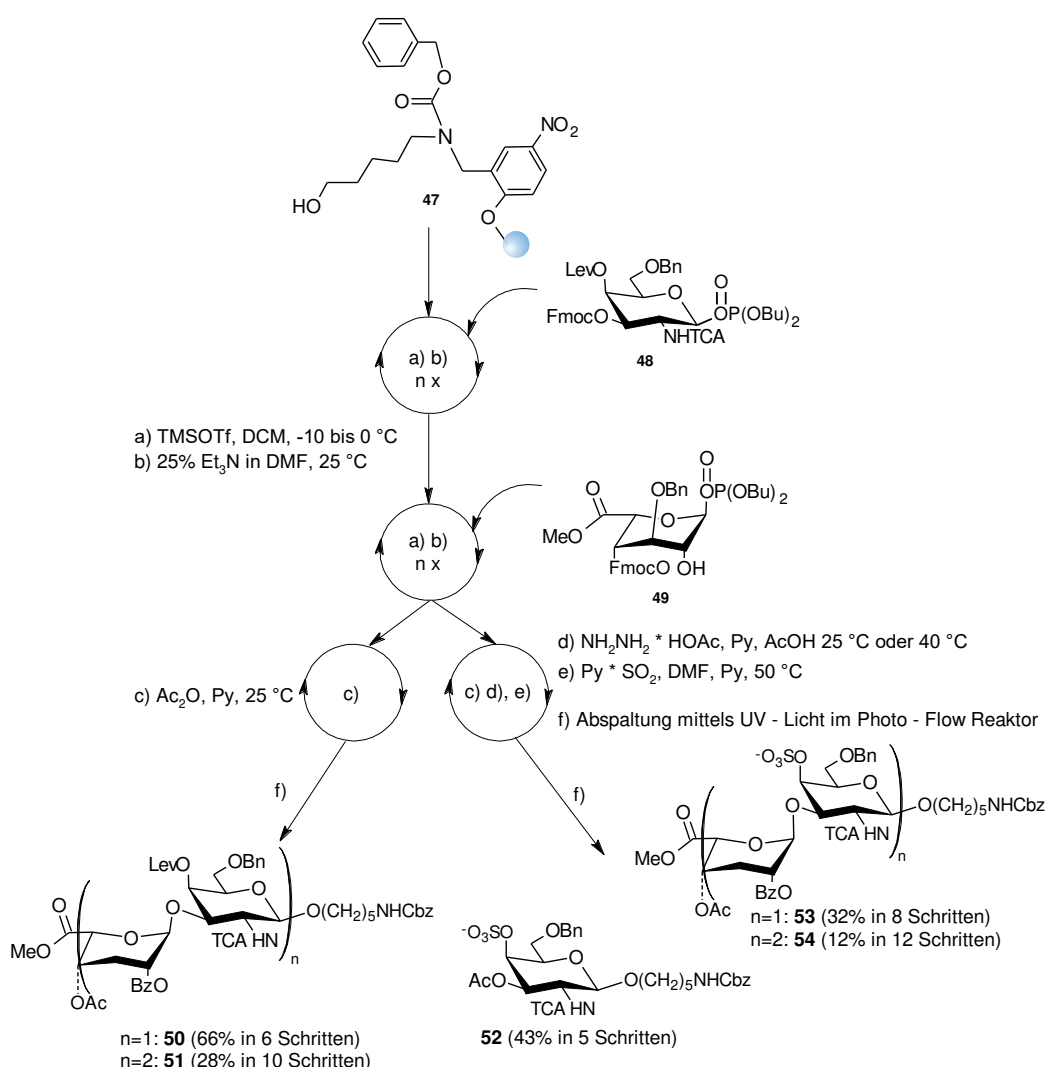


Abbildung 156: Automatische Synthese von Dermatan – Oligosacchariden unter Verwendung des Arbeitssynthesizers.

Mit der selektiven Entfernung der Lev-Schutzgruppen und Sulfatierung der resultierenden Hydroxylgruppen, sind bei der automatisierten Synthese von sulfatierten Dermatanoligosacchariden

zwei weitere Syntheseschritte (Abbildung 156 d)e)) erforderlich. Die Entfernung der Lev-Gruppen erfolgt mittels einer Lösung von Hydrazinacetat in Pyridinessigsäure bei 25 °C. Der Pyridin-Schwefeltrioxid-Komplex in Dimethylformamid-Pyridin wurde für die Sulfatierung freier Hydroxylgruppen bei 50 °C am Synthesizer verwendet.^[178]

Die Synthese des sulfatierten Monosaccharids **52** und des Disaccharids **53** erfolgte auf einem festen Träger in fünf bzw. sieben automatisierten Schritten unter den gleichen Glykosylierungs-Entschützungsbedingungen wie bei der Herstellung der Produkte **50** und **51**, gefolgt von automatisierter Acetylierung, Lev-Schutzgruppenentfernung und Sulfatierung. Nach Abspaltung des fotoempfindlichen Linkers, wurden die Rohprodukte mittels Umkehrphasen (RP) -HPLC analysiert und durch Größenausschlusschromatographie (Sephadex LH-20) gereinigt, um das Monosaccharid **52** mit 43% und das Disaccharid **53** mit 32% Gesamtausbeute bezogen auf die Harzbeladung zu erhalten.

Der automatisierte Zusammenbau von sulfatiertem Tetrasaccharid **54**, unter den für die Synthese von **52** und **53** festgelegten Bedingungen, ergab eine Mischung von Verbindungen, die aus der unvollständigen Entfernung von Lev-Estern mit Hydrazinacetat bei 25 °C resultierte. Die Erhöhung der Temperatur für die Lev-Schutzgruppenentfernung auf 40 °C im automatisierten Lauf verbesserte die Reinheit und Ausbeute des gewünschten Tetrasaccharids **54**, das in 12% Ausbeute über 12 Stufen isoliert wurde.

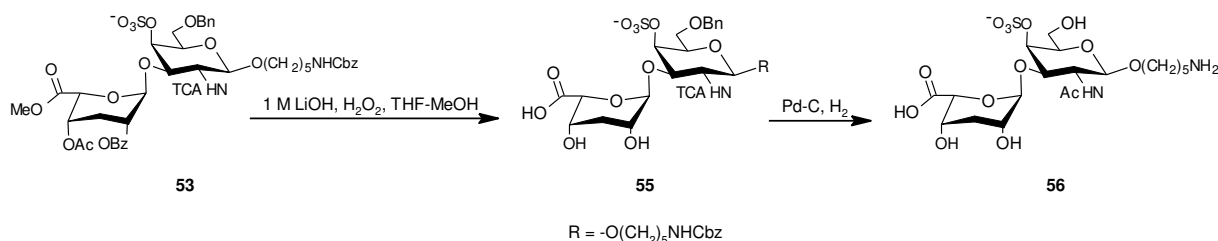


Abbildung 157: Entschützung des sulfatierten Disaccharids in der Lösungsphase

Nach dem Einbau der geschützten Oligosaccharide auf dem festen Träger wurde die Schutzgruppenabspaltung des sulfatierten Disaccharids **53** in Lösungsphase getestet (Abbildung 157). Die Hydrolyse von Acetat, Benzoat und Methylestern wurde unter Verwendung von Lithiumhydroxid / Wasserstoff Peroxid in THF-MeOH durchgeführt, um das Disaccharid **55** mit 79% Ausbeute herzustellen. Die Hydrogenolyse von **55** lieferte ein vollständig entschütztes, konjugationsfertiges Disaccharid **56** mit 80% Ausbeute. Konjugationsbereite Dermatansulfat-Oligosaccharide können direkt auf Mikroarrays immobilisiert oder an Proteine konjugiert werden.^[179]

Synthese von Oligosaccharid-Fragmenten pflanzlicher Arabinoxylane, Xyloglucane und Typ-II-Arabinogalaktane

Da strukturell verwandte Oligosaccharide aus üblichen Bausteinen hergestellt werden können, ermöglicht die automatisierte Oligosaccharidsynthese einen einfachen und schnellen Aufbau von Kohlenhydratbibliotheken. Mit der Herstellung verschiedener Oligosaccharid-Fragmente pflanzlicher Arabinoxylane aus Monosacchariden^[183] durch D. Schmidt et al., Xyloglucan Oligosaccharid-Fragmenten aus Mono – und Disacchariden^[184] durch P. Dallabernardina et al. und verschiedene Typ-II-Arabinogalaktane unterschiedlicher Größe und Komplexität auf der Basis von vier Monosaccharid-Bausteinen^[185] durch Bartetzko et al., konnten weitere umfangreiche Erfahrungen für die systematische Synthese von Kohlenhydratstrukturen am Arbeitssynthesizer gewonnen werden.

Sowohl Arabinoxylan, als auch Xyloglucan, Xylan, Glucomannan und Glucuronoxylan gehören einer heterogenen Gruppe von Polysacchariden an, die in Hemicellulose (Abbildung 158), dem Hauptteil der amorphen Matrix der primären Zellwand aller höheren Pflanzen, vorkommen^[186] und je nach Zelltyp und Pflanzengruppe in der Zusammensetzung stark variieren kann. Xylane sind eine der Hauptkomponenten der Pflanzenzellwand und damit die am häufigsten vorkommenden nichtcellulosehaltigen Polysaccharide in der Natur.^[187,188] Xyloglucan (XG) ist das am besten untersuchte Hemicellulose-Polysaccharid und mit einem Anteil von 20% bis 25% in der Zellwand-Trockenmasse,^[189] Hauptbestandteil der Hemicellulosen in der Primärwand bei Eudikotyledonen³⁰, sowie bei etwa der Hälfte der Monokotyledonen³¹.^[187, 190, 191]

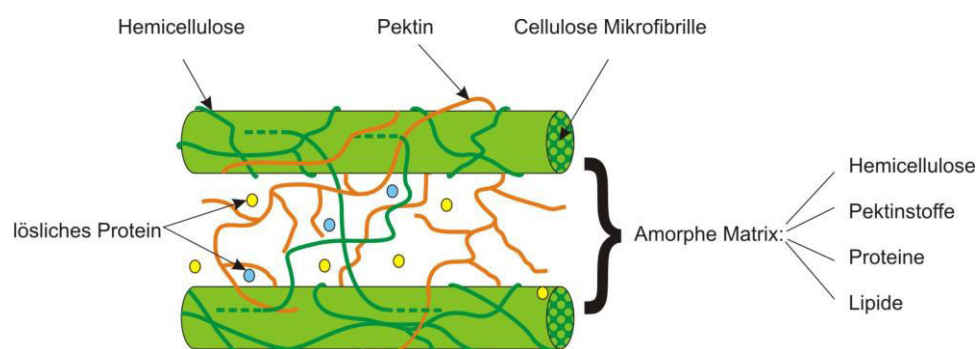


Abbildung 158: Hemicellulosen sind ein Bestandteil pflanzlicher Zellwände, deren Matrix aus fibrillärer, teilweise kristalliner Cellulose besteht.

In den Pflanzenzellwänden sind zudem Glykoproteine³² und Proteoglykane³³ eingebettet. Speziell Arabinogalaktan - Proteine (AGP) sind stark glykosylierte Proteoglykane aus Typ-II-Arabinogalaktan - Polysacchariden^[192], die an vielen biologischen Prozessen beteiligt sind, wie Zellteilung^[193],

³⁰ In der Botanik die Gruppe von „bedecktsamigen“ Pflanzen, die einen Großteil der „Zweikeimblättrigen“ (= Dikotyledonen) umfasst.

³¹ In der Botanik „einkeimblättrige“ Pflanzen, die der Gruppe der „bedecktsamigen“ Pflanzen angehören.

³² Makromolekül, bestehend aus einem Protein und einer oder mehrerer kovalent gebundenen Kohlenhydratgruppen. Vorkommen, z.B. als struktureller Bestandteil von Zellmembranen.

³³ Klasse besonders stark glykosylierter Glykoproteine. Wichtiger Bestandteil der tierischen Interzellularmatrix.

Zellverlängerung^[194], Zelltod^[195], Musterbildung, Erhaltung des postembryonalen Musters^[196] und Wurzel- Mikrogen-Interaktionen.^[197]

Automatisierten Synthese von Arabinoxylan-Fragmente

Bei der von S. Eller durchgeführten automatisierten Synthese der Arabinoxylan-Fragmente wurden für den Aufbau des β -1,4-verknüpften Xylan-Rückgrats zwei verschiedene D-Xylopyranose-Bausteine verwendet. Die C-4-Hydroxyle wurden jeweils mit einer Fluorenylmethoxycarbonyl (Fmoc)-Gruppe geschützt, die sich während der automatisierten Synthese unter Verwendung einer Aminbase selektiv entfernen lässt. Mit den Benzoatestern an den C-2-Hydroxylgruppen wurde die erforderliche β -Selektivität in den Glykosylierungsschritten sichergestellt. Das C-3-Hydroxyl wurde entweder mit einem Benzylether als permanente Schutzgruppe oder mit einer (2-Naphthyl)-methyl (Nap) -Gruppe als temporäre Schutzgruppe für die Substitution des Grundgerüsts mit Arabinoseresten ausgestattet. Die in der Literatur beschriebenen oxidativen Bedingungen zur selektiven Spaltung von Nap-Ethern in Gegenwart von Benzylethern unter Verwendung von 2,3-Dichlor-5,6-dicyano-1,4-benzochinon (DDQ),^[198] fanden auch erfolgreich in der automatisierten Synthese Anwendung.

Nap-Ether weisen ähnliche elektronische Eigenschaften wie Benzylether auf und erlaubt die relativ einfache Synthese des jeweiligen Xylose-Bausteins. Dibutylphosphat als anomere Abgangsgruppe zeigte bislang die besten Ergebnisse beim Einbau großer oligomerer Glykane.^[149,199]

Der Einbau der Arabinose-Substituenten am Xylan-Gerüst beruhte entweder auf einem perbenzoylierten oder einem 2-Fmoc geschützten L-Arabinofuranosyl-Baustein in Abhängigkeit der weiteren Seitenkettenverlängerung. Mit dem neuen Arbeitssynthesizer und den benötigten Bausteinen wurden elf Arabinoxylan-Fragmente, spezifische Substrukturen natürlich vorkommender Arabinoxylane, hergestellt.

Durch iterative Addition der Bausteine **57a, b** und **58a, b** des am Linker-funktionalisierten Harzes wurden lineare Oligoxylane (**59 - 62**) und die Oligoarabinoxylane unterschiedlicher Komplexität (**63 - 69**) hergestellt (Abbildung 159). Die Abspaltung vom fotospaltbaren Linker **47** erfolgte mittels UV-Licht an einem mikrofluidischen Fotoreaktor.^[200] Die Glykosylierungsschritte wurden doppelt unter Verwendung von fünf Äquivalenten Glykosyldonor und entweder äquimolarem TMSOTf oder N-Iodsuccinimid zusammen mit katalytischen Mengen an TfOH zur Aktivierung durchgeführt. Jeder Glykosylierung unter Verwendung eines Xylose-Bausteins folgte entweder die Entfernung der temporären Fmoc-Schutzgruppe an C-4, was die Verlängerung des Xylan-Grundgerüsts bewirkte, oder der Nap-Schutzgruppe an C-3, welches die Installation des Arabinose-Substituenten ermöglichte. Nach Optimierung der Reaktionsbedingungen, konnte ein zuverlässiges Protokoll für die selektive Spaltung von Nap-Gruppen auf festem Träger etabliert werden, welches die Auswahl an temporären, orthogonalen Schutzgruppen für die automatisierte Oligosaccharidsynthese erweitert.

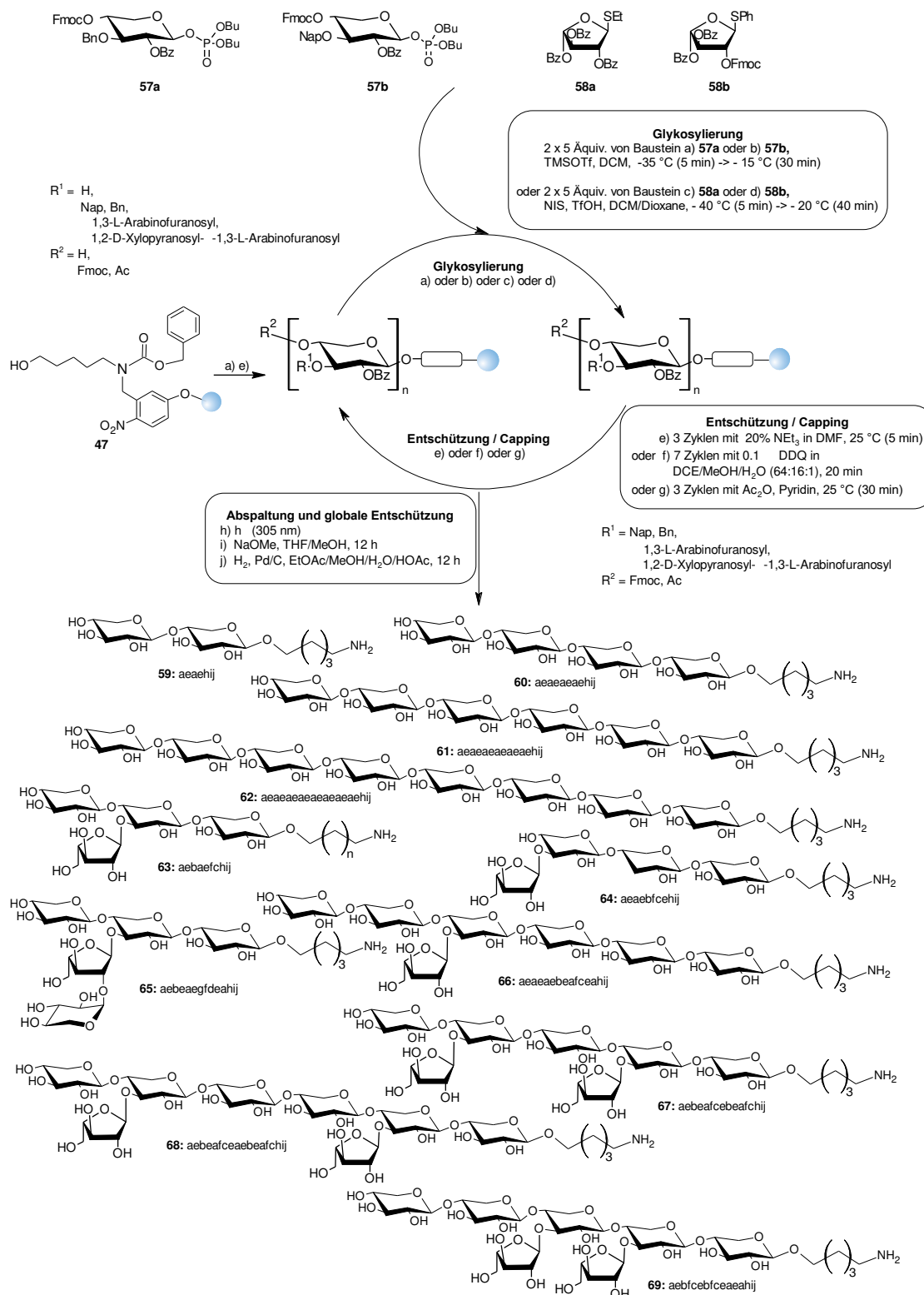


Abbildung 159: Automatisierte Festphasensynthese von Arinoxylan – Fragmenten. Der Buchstabencode unter den Strukturen zeigt die jeweilige Synthesesequenz am Arbeitssynthesizer.

Nap-Ether unterscheidet sich von den routinemäßig verwendeten Fmoc- und Levulinoyl-Schutzgruppen dadurch, dass er auf den jeweiligen Baustein keinen „disarmed“-Effekt ausübt, was die Feineinstellung der Glykosyldonorreaktivität in Festphasensynthesen erleichtert. ^[121,201] Für die Synthese des Pentasaccharids **65**, das einen β-1,2-D-Xylopyranosyla-1,3-L-Arabinofuranosyl-

Substituenten enthält, wurde das lineare Xylan-Grundgerüst durch Reaktion mit Essigsäureanhydrid nach der Montage verkappt. Nach Entfernen der Nap-Gruppe an der zentralen Xylose-Einheit und Einbau des Arabinoserestes unter Verwendung von Baustein **58b** wurde die Seitenkette durch Entschützen des Fmoc-substituierten Arabinose-Restes und abschließende Glykosylierung mit Xylose-Donor **57a** selektiv verlängert.

Nach dem Oligosaccharidaufbau wurde das Harz aus dem Synthesizer entfernt und die rohen Oligoarabinoxylsäuren vom festen Träger abgespalten.^[110] Die anschließende Entfernung der Benzoate mit Natriummethoxid und die hydrogenolytische Spaltung der Benzylether und der Cbz-Schutzgruppe lieferte die vollständig entschützten Oligoarabinoxylane **59 - 69** in einer Gesamtausbeute von 7 - 43%, bezogen auf die berechnete Beladung des Linker-funktionalisierten Harzes **47**. Die Produkte wurden durch präparative Reversed-Phase-HPLC entweder auf der Stufe der halbgeschützten Oligosaccharide vor der Hydrierung oder in der Endstufe gereinigt, wobei die reinen Produkte erhalten wurden. Alle Produkte wurden durch analytische HPLC, NMR-Spektroskopie und hochauflösende Massenspektrometrie charakterisiert.^[183]

Synthese von Xyloglucan-Strukturen

Bei der von P. Dallabernardina durchgeführten Herstellung von Xyloglucan (XG) -Oligosacchariden aus Monosaccharid-Bausteinen, erwies sich die Installation der α -Xyloside als Schwierigkeit. Die Reaktion führte zu einer Mischung von α / β -Isomeren mit einem relativen Verhältnis von 1: 1 und enthielt weniger des gewünschten α -Isomers als in ähnlichen Reaktionen in Lösung.^[202] Da mehrere α -Xyloside in die Synthese größerer XG-Oligosaccharide eingebaut werden müssen, wurden die XG-Oligosaccharide mit einem Mono- und einem Disaccharid-Baustein mit vorinstallierter α -xylosidischen Bindung hergestellt. Der Glukose-Baustein **70** wurde in der C-2 - Position mit Benzylestern (Bz) als beteiligten Schutzgruppen ausgestattet, um die Bildung von β -Bindungen zu begünstigen. Fluorenylmethoxycarbonyl (Fmoc) und Levulinoyl (Lev) dienen als temporäre Schutzgruppen, um entweder die Verlängerung des β -1,4-Glucan-Rückgrats oder die Substitution der 6-Position durch Xylose zu ermöglichen. Alle übrigen Positionen wurden als Benzylether dauerhaft geschützt. Im Disaccharid **71** wurde, da die Installation einer Fmoc-Schutzgruppe aufgrund sterischer Hinderung scheiterte, das C-4-Hydroxyl der Glukose als Lev-Ester geschützt.

Die Xyloglucanfragmente wurden durch iterative Addition der Bausteine **70** und **71** mit am Linker-funktionalisiertem Harz **47** unter Verwendung des Arbeitssynthesizers hergestellt.

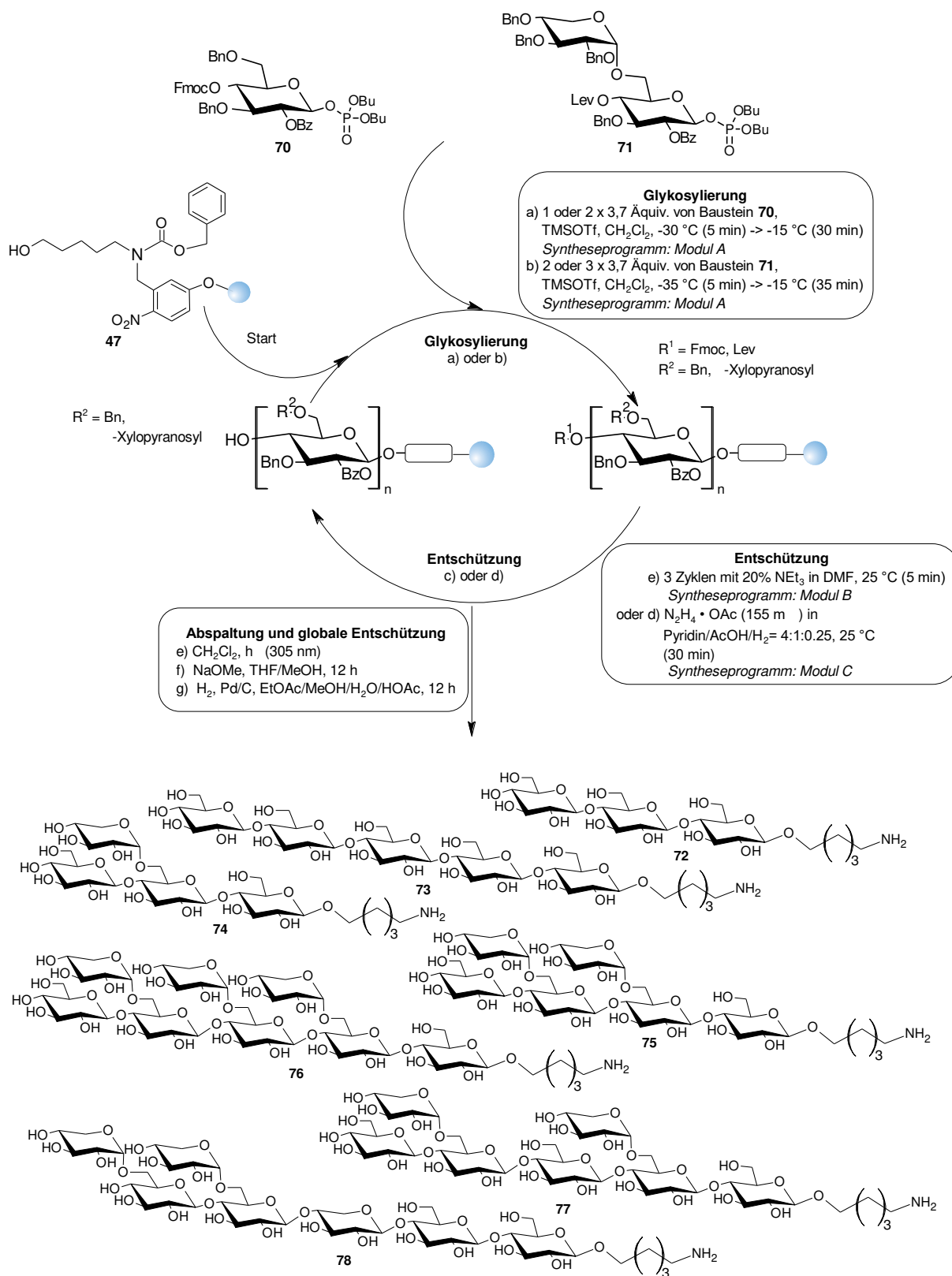


Abbildung 160: Automatische Synthese von Xyloglucan – Oligosacchariden am Arbeitssynthesizer.

Nach Abspaltung der geschützten Form vom festen Träger konnte das Xyloglucanfragment **74** vollständig erhalten werden. Oligosaccharide mit zwei α -Xylosid-Substituenten wurden jedoch mit verringerter Effizienz gebildet, und die gewünschten Produkte waren von Deletionssequenzen

begleitet, denen eine der Disaccharideinheiten fehlte. Dennoch wurden die vollständig entschützten Verbindungen **75**, **77** und **78** in Gesamtausbeuten von 9-16% nach Abspaltung vom festen Träger, Behandlung mit Natriummethoxid und einer abschließenden Hydrierungsreaktion isoliert. Das Octasaccharid **76**, das drei Disaccharideinheiten enthält, wurde in 2% Gesamtausbeute hergestellt, wenn drei statt zwei Mal 3,7 Äquivalente des Bausteins **71** für die beiden Schlüsselglykosylierungen verwendet wurden. Aufgrund der sterischen Hinderung ist der Zusammenbau von Ketten mit mehr als drei Disaccharideinheiten nicht praktikabel. Um den Satz molekularer Werkzeuge für die Xyloglucan-Biologie zu vervollständigen, wurden zwei Zellulosefragmente (**72**, **73**) hergestellt. Dabei waren die Kupplungen mit einmal 3,7 Äquivalenten des Bausteins **70** ausreichend, um vollständige Glykosylierungen sicherzustellen.^[184]

Synthese von Typ-II-Arabinogalaktan - Strukturen

Basierend auf vier Monosaccharid-Bausteinen, konnten über die automatisierte Glykansynthese am Arbeitssynthesizer 14 Arabinogalaktan-Glykane vom Typ II unterschiedlicher Größe und Komplexität von M. P. Bartetzko hergestellt werden. An den drei Galaktose-Bausteinen **79**, **80** und **81** wurde, abhängig von der zu verlängernden Position, die temporäre Fluorenylmethoxycarbonyl (Fmoc) - Schutzgruppe an der C-3- (**79**, **80**) - oder C-6-Position (**81**) installiert. Die Hydroxylgruppen, die während der Festphasensynthese nicht manipuliert werden mussten, wurden permanent geschützt. In der C-2-Position stellten Benzylester (Bz) eine Beteiligung benachbarter Gruppen an den Glykosylierungsreaktionen bereit. Alle verbleibenden Positionen wurden mit Benzylethern (Bn) geschützt. Um eine Verzweigung zu ermöglichen, wurde Baustein **80** zusätzlich mit einer Levulinoyl (Lev) -Schutzgruppe versehen, die vollständig orthogonal zur Fmoc-Gruppe ist und vielfach erfolgreich in der automatisierten Glykansynthese Verwendung findet.^[136,146,203,204] Die Galaktose Bausteine wurden mit Dibutylphosphat als anomere Abgangsgruppe versehen, da die Herstellung von Gal- β 1,3-Gal-Bindungen mit anderen anomeren Abgangsgruppen in der Literatur häufig schlechte Stereoselektivitäten und niedrigen Ausbeuten aufwies.^[205] Die anspruchsvolle Gal- β 1,3-Gal-Bindung konnte mit drei Glykosylierungen am Linker-funktionalisierten funktionalisierten Harz **47** mit dem Baustein **79** und vollständiger Umwandlung nach Spaltung des fotolabilen Linkers in einem kommerziellen Fotoreaktor,^[200] an der festen Phase hergestellt werden. Ebenso gut verlief die Synthese eines β - (1, 6) -verbundenen Trigalactosids unter Verwendung von Baustein **81**. Durch iterative Glykosylierungs- und Entschützungsreaktionen ließen sich nun die linearen und verzweigten Galaktan-Oligosaccharide **83** – **89** am Arbeitssynthesizer herstellen (Abbildung 161). Die Glykosylierungen wurden zweimal unter Verwendung von 3,75 Äquivalenten der Galaktose-Bausteine **79** - **81** und äquimolaren Mengen TMSOTf durchgeführt. Obwohl für die automatisierte Glykan-Assemblierung die zweifache Verwendung von fünf Äquivalenten der Bausteine Standard

ist,^[136,146,204] ließ sich bei der Verwendung der weniger überschüssigen Bausteine keine reduzierte Umwandlung beobachten.

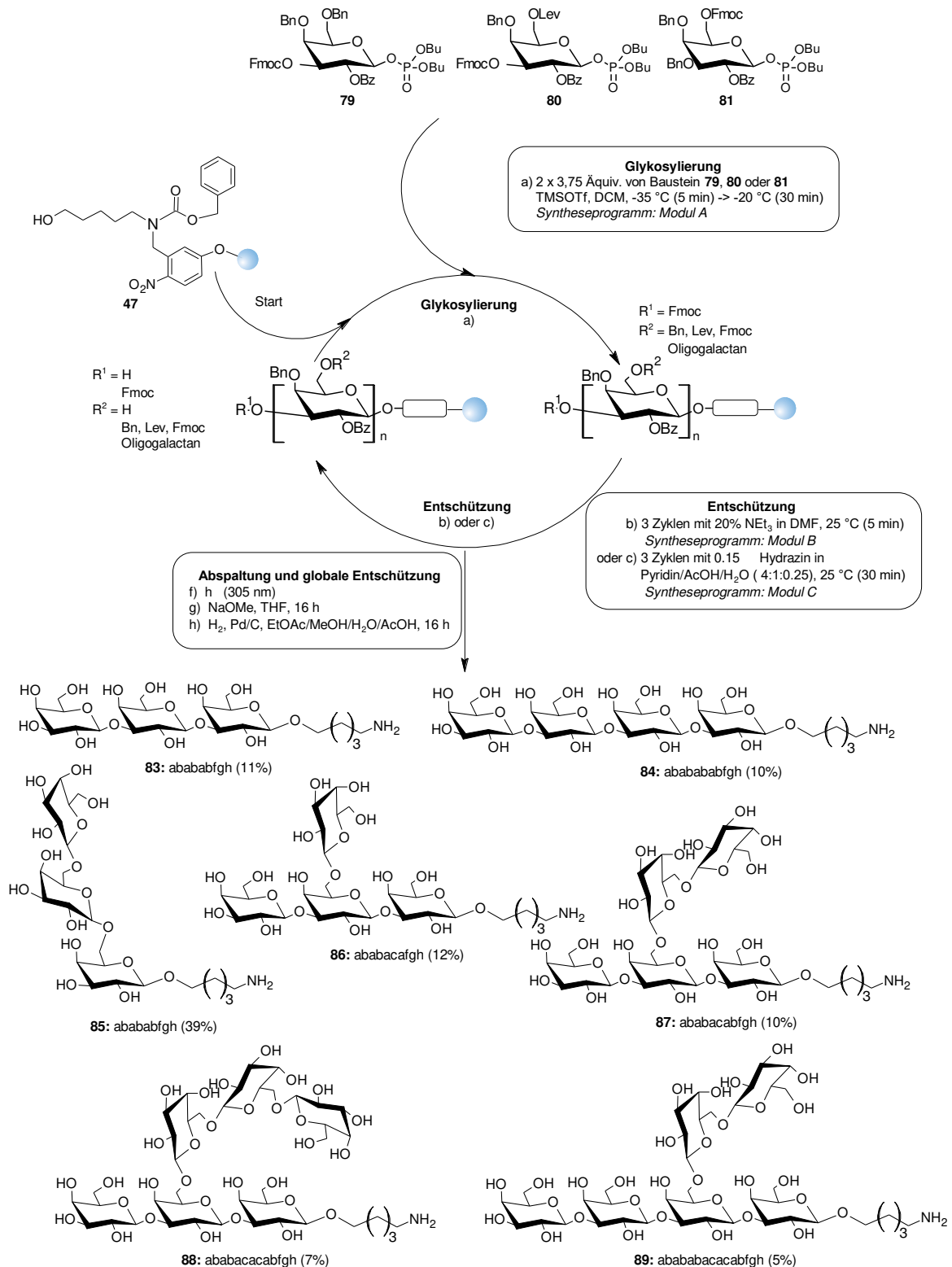


Abbildung 161: Automatische Synthese von Type-II Galaktan Fragmenten 83 – 89 am Arbeitssynthesizer.^[185] Der Buchstabencode unter den Produkten bezeichnet die jeweiligen Syntheseschritte.

Nach jeder Glykosylierung wurde entweder Fmoc oder Lev entfernt, um die Position auf Galaktose für die nachfolgende Verlängerung zu aktivieren. Alle verzweigten Strukturen wurden synthetisiert, indem zunächst das lineare β - (1,3) -gebundene Rückgrat zusammengebaut, und anschließend die β - (1,6) -gebundene Seitenkette angebracht wurde.

Nach Abspaltung der geschützten Oligosaccharide vom festen Träger und der Behandlung mit Natriummethoxid in Methanol, wurden die resultierenden halbgeschützten Produkte einer Hydrogenolyse unterzogen. Die vollständig entschützten Oligogalaktane **83 - 89** konnten damit in einer Gesamtausbeute von 5 bis 39% bezogen auf die berechnete Beladung des Linker-funktionalisierten Harzes **47** hergestellt werden. Die Reinigung der Produkte erfolgte über präparative Umkehrphasen-HPLC, sowohl im vollständig als auch im halbgeschützten Zustand. Die Reinheit aller Produkte wurde durch Charakterisierung mit analytischer HPLC, NMR-Spektroskopie und hochauflösender Massenspektrometrie sichergestellt.

Mit einem zusätzlichen L-Arabinose-Baustein **82** und den Galaktose Bausteinen **79 – 81** ließen sich sieben weitere Arabinogalaktan-Substrukturen vom Typ II herstellen (Abbildung 162). Bei den Produkten ist die Arabinose-Einheit entweder mit dem β - (1,3) -gebundene Galaktan - Rückgrat (**90, 91, 93**) oder mit der β - (1,6) -gebundenen Galaktan - Seitenkette (**92, 94 – 96**) verknüpft. Die Herstellung der komplexeren Oligosaccharide **94 – 96** erforderte zusätzliche Verkappungsschritte während der automatisierten Synthese, um die selektive Bindung der Arabinose an die gewünschte Galaktoseeinheit sicherzustellen. Dazu wurde nach dem Zusammenbau des (1,3) -Galaktan-Rückgrats die endständige Fmoc-Gruppe entfernt, und die resultierende freie Hydroxylgruppe als Ester maskiert. Zum Verkappen wird das harzgebundenen Oligogalaktan mit einer Lösung von Benzoessäureanhydrid und DMAP in DCE für 30 min bei 40 °C inkubiert. Dabei entsteht Benzoyl ester als Verkappingsrest, das stabil ist gegenüber Hydrazin, welches zur Entfernung der Lev-Schutzgruppe verwendet wird.^[206] Nach Verkappen des β - (1,3) -gebundenen Galaktan - Rückgrats wurde die Seitenkette durch iterative Lev Entschützung und Glykosylierungsreaktionen eingeführt. Nach der Fmoc-Entschützung wurde der Arabinose - Baustein **82** selektiv an die Seitenkette gebunden. Der Thioglykosid – Baustein **82** wurde dabei mit N-Iodsuccinimid und katalytischen Mengen an TfOH in DCM aktiviert. Nach Abspaltung der Produkte von dem festen Träger, der globalen Entschützung und Reinigung, konnten die Typ-II-Arabinogalaktan-Oligosaccharide **90 - 96** in Ausbeuten zwischen 9 und 17% bezogen auf die berechnete Harzbeladung hergestellt werden.^[185]

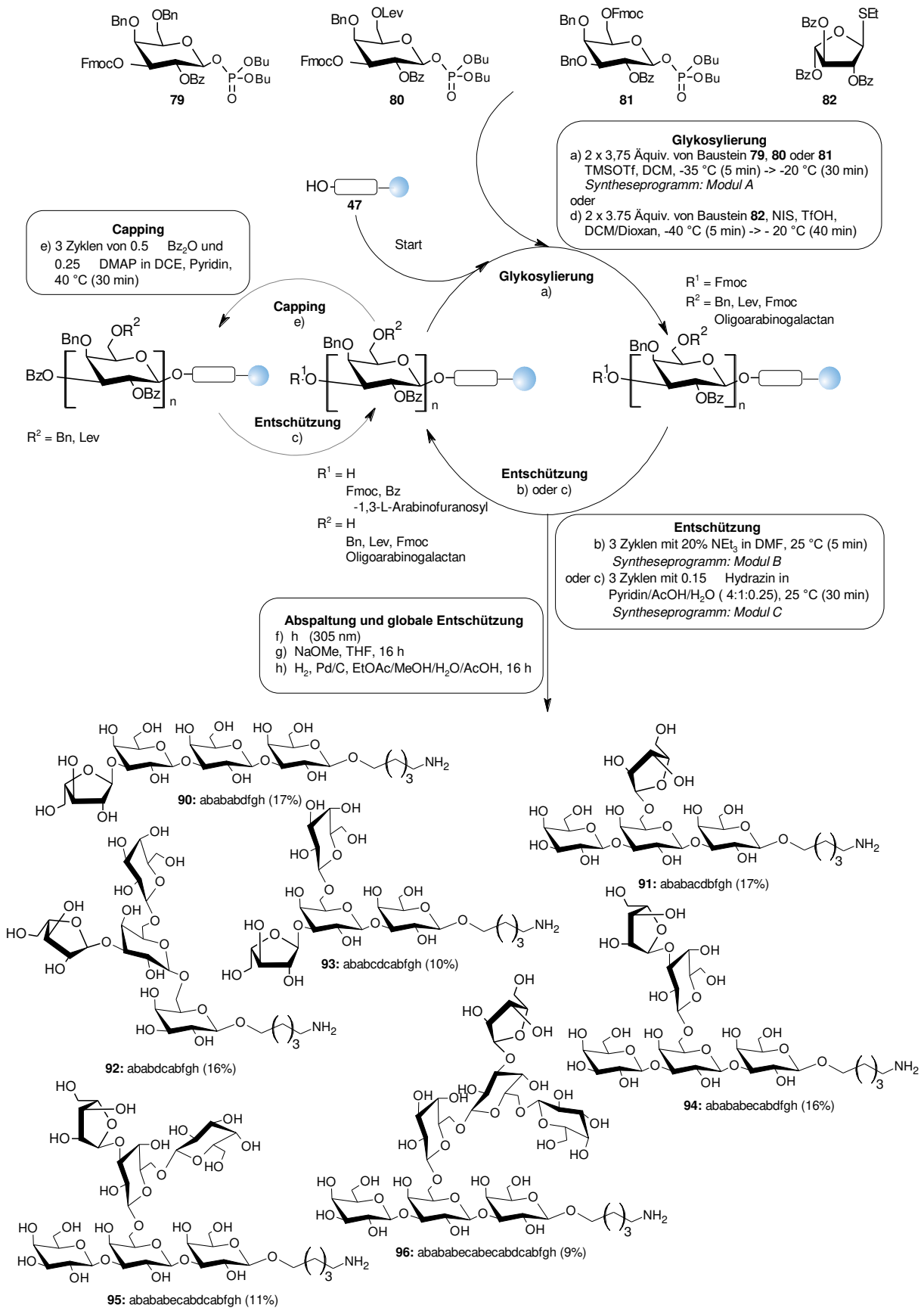


Abbildung 162: Automatische Synthese von Type-II Galactan Fragmenten 90 – 96 am Arbeitssynthesizer.^[185] Der Buchstabencode unter den Produkten bezeichnet die jeweiligen Syntheseschritte.

Automatische Synthese eines 50-mer Polymannosids

Der bisherige Rekord eines mit der automatischen Synthese hergestellten 30-mer Oligomannosids, basierte bereits auf dem neuen Softwarekonzept.^[178] Hierzu wurde das Altgerät in der Entwicklungsphase der Arbeitssynthesizer mit der neuen Steuerung versehen und softwareseitig im Kompatibilitätsmodus betrieben (siehe Kapitel: Herstellen der Kompatibilität zu Syntheseskripten des Altsystems, Seite 129). Der Erfolg der neuen Umsetzung ermutigte zum nächsten Schritt, der Herstellung eines 50-mer Polysaccharides durch Naresh et al., welcher für die neu gebauten Arbeitssynthesizer den ultimativen Test zur Verifizierung der Gesamtsystemstabilität darstellte.

Bei der automatisierten Festphasensynthese hängt die Gesamtlänge von Polymeren von sehr hohen Ausbeuten bei jedem Schritt ab. Neben der chemischen Herausforderung der Stereo- und Regio-Kontrolle während der Glykosylierung, aufgrund inhärenter struktureller Komplexitäten bei der chemischen Synthese der Polysaccharide, spielt die Systemstabilität bei den einzelnen Prozessschritten im automatischen Arbeitssynthesizer und die strikte Einhaltung der Prozessparameter wie Temperaturführung über einen langen Zeitraum eine große Rolle. Die Synthese langer Polymere bildet somit die Effizienz und Zuverlässigkeit synthetischer Prozesse am Arbeitssynthesizer ab. Bisherige Syntheseversuche an den Altsystemen gelangten entweder an ihre Stabilitätsgrenzen oder waren nicht in der Lage über die gesamte Synthesezeit die Prozessparameter stabil zu regeln, was im schlimmsten Fall zum Abbruch des Syntheseprozesses mit Verlust der Zuckerbausteine führte.

Am neuen Arbeitssynthesizer gelang über eine 102-stufige Synthese die Herstellung eines 50-mer Polysaccharides aus Monosacchariden. Damit wurde mit der automatischen Festphasensynthese das damals längste synthetische Kohlenhydrat aufgebaut. Die wenigen bislang in der Literatur beschriebenen Synthesen langkettiger Polysaccharide, wie das mykobakterielle Arabinogalaktan 92-mer, das Heparinbezogene 40-mer und das mykobakterielle Mannose-blockierte Arabinomannan 21-mer wurden unter Verwendung von Blockkopplungen durchgeführt.^[207]

Für die Studie wurden drei unterschiedlich geschützte Thiomannosid-Monomere (**97** - **99**, Abbildung 163) entwickelt, aus denen der optimale Baustein zur Synthese langer linearer Polysaccharide mit Man- (1 → 6) - α -Man-Bindungen selektiert wurde. Die Monomere **97** - **99** tragen eine permanente Benzoylgruppe an der C-2 - Position, um α -Selektivität (dh. 1,2-trans-glykosidische Bindungen) durch Nachbargruppenbeteiligung sicherzustellen, und eine temporäre Fmoc-Carbonatschutzgruppe an der C-6 - Position zur Kettenverlängerung. Die C3 - und C4 - Hydroxylgruppen sind mit „bewaffneten“^[121,208] - Benzylethern (**99**), „Entschärfungs“ - Benzylestern (**97**),^[209] oder einer Kombination der beiden (**98**)^[210] geschützt.

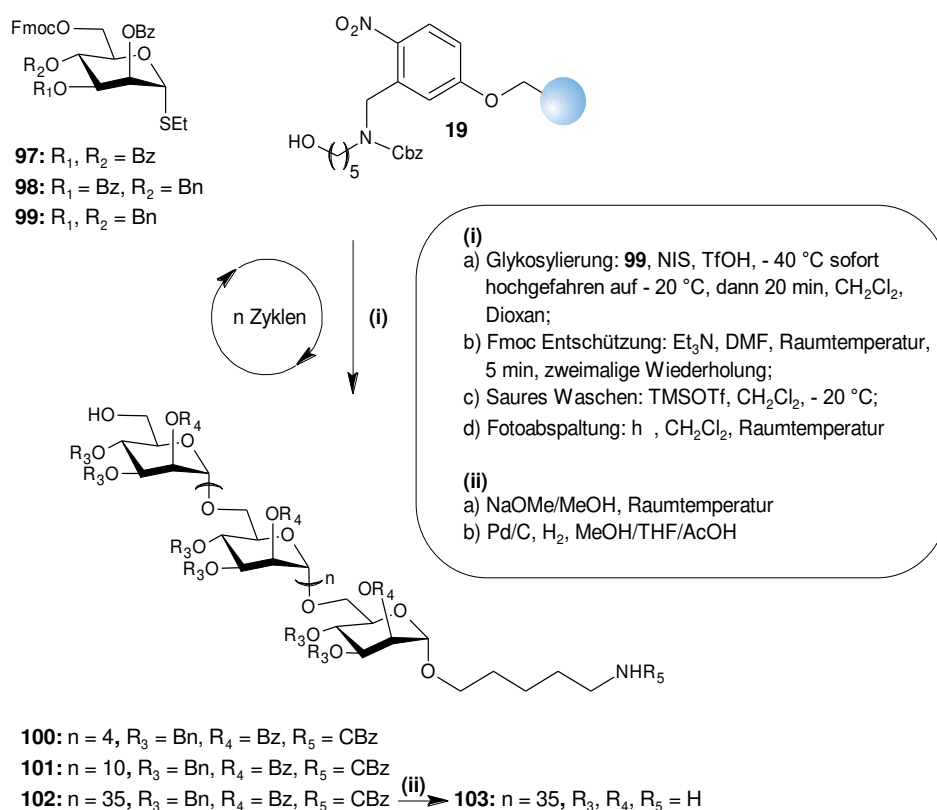


Abbildung 163: Automatische Festphasensynthese von Mannose Oligomeren am neuen Arbeitssynthesizer

In getrennten Chargen wurden Trisaccharide auf einem Merrifield-Harz synthetisiert, das mit einem fotolabilen Linker (**19**) ausgestattet war, um die Wirkung der Schutzgruppen auf Reaktivität und Produktreinigung (ESI⁺) zu bestimmen.^[200] Dabei erwies sich Baustein **99** mit seinen zwei schützenden Benzylethergruppen bei der Analyse der Syntheseprodukte aufgrund seiner größeren Reaktivität am besten für die Synthese von Man- (1 → 6) - α -Man-Oligosacchariden geeignet. Das Oligosaccharidprodukt ließ sich gut durch HPLC von den Deletionssequenzen trennen.

Mit den für die Trisaccharidsynthese entwickelten Bedingungen, wurde jeder Glykosylierungsschritt unter Verwendung von 6,4 Moläquivalenten, basierend auf der Fmoc - Quantifizierung der anfänglichen Zuckerbeladung^[211] des Bausteins **99** durchgeführt und damit lineare Mannoseketten mit einer Länge von sechs (**100**) bis 50 (**105**) Monomereinheiten hergestellt (Abbildung 163 und Abbildung 164).

Durch Behandlung mit Triethylamin wurde die temporäre Fmoc-Schutzgruppe an der C-6 – Position entfernt und vor der nachfolgenden Glykosylierung eine Säurebehandlung mit TMSOTf-Lösung zur Entfernung von Wasser und Base durchgeführt.

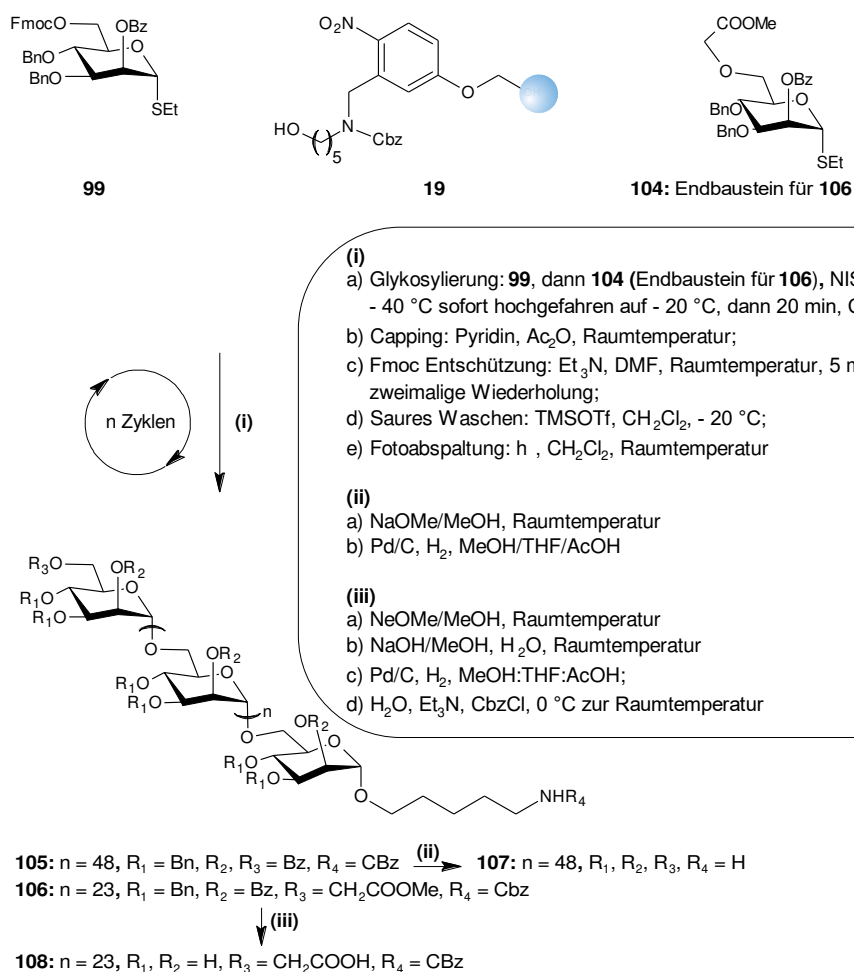


Abbildung 164: Automatische Festphasensynthese von Mannose Oligomeren am neuen Arbeitssynthesizer

Zur Abspaltung vom Harz wurden die geschützten Oligosaccharidprodukte nach der automatischen Synthese am Arbeitssynthesizer einer UV-Bestrahlung mit Hilfe einer kontinuierlichen Durchflussvorrichtung unterzogen.^[212] Hexasaccharid **100** (13 Stufen) und Dodecasaccharid **101** (25 Stufen) ließen sich leicht durch ein Normalphasen-HPLC von allen Deletionssequenzen reinigen und konnten mit 50 % bzw. 38 % Ausbeute, bezogen auf die Harzbeladung des ersten Zuckers, hergestellt werden.

Mit zunehmender Kettenlänge wird jedoch die Abtrennung des gewünschten Produktes von der unmittelbaren Deletionssequenz (d. H. n vs. n-1) zunehmend schwieriger, so eluiert z. B. das 37-mer **102** gemeinsam mit dem 36-mer, was eine Reinigung durch HPLC ausschließt. Auch steigt der Anteil der n - 1 Deletionssequenz mit zunehmender Kettenlänge. Deshalb wurde während der Herstellung des 50-mers **105** ein zusätzlicher Acetylierungs- oder Verkappungs-Schritt verwendet, um Trennprobleme und unvollständige Umwandlung zu adressieren. Dieser in der Oligonukleotid- und auch in Peptidsynthese üblichen Ansatz, wurde zusätzlich durch die Einführung eines zweiten Glykosylierungsschritts in jedem der letzten fünf Elongationszyklen (n = 46 - 50) ergänzt (6,4 Äquivalente des Bausteins in jedem Kopplungsschritt; zwei Kopplungsschritte). Das 50-mer **105**

konnte nach 102 automatischen Synthese Schritten in 5% Gesamtausbeute hergestellt werden. Die durchschnittliche stufenweise Ausbeute betrug 97,1% (ESI †). Das Produkt war durch die HPLC in voller Länge gut auflösbar und konnte bei Verwendung des Verkappungs- und zusätzlichen Kupplungsschrittes leicht aus den verkürzten Sequenzen gereinigt werden. Obwohl der Verkappungsschritt in jedem Zyklus die Reinigung von 50-mer **105** erleichtert, erwies es sich als zeitaufwändig (2 h für jeden Verkappungsschritt), was für die Hälfte der Syntheselaufzeit von insgesamt 250 h steht.

Das Mannose 37-mer **102** und das 50-mer **105** wurden zur Entfernung der Estergruppen mit Methanolyse (NaOMe / MeOH) nach Zemplén entschützt. Mit Hilfe von Hydrogenolyse erfolgte die Abspaltung der Benzylgruppen in Gegenwart von Pd / C. Aufgrund der sehr großen Anzahl an Benzylgruppen wurde die Reaktion über einen längeren Zeitraum (72 bis 96 h) und mehrere Runden durchgeführt, um den Abschluss der Reaktion sicherzustellen.^[213]

5 Zukunftsausblick und Konzeptstudie: „Das cyber-physische Chemie-System“

5.1 Einleitung

Bereits das zu dieser Arbeit erstellte Exposé von 2010^[159], welches die Entwicklungsgrundlage der neuen Gerätegeneration von Synthesizern darstellte, beinhaltet die Zukunftsvision eines umfassend standardisierten, modularen und flexiblen Gerätekonzepts und der daraus abgeleiteten Möglichkeiten (Abbildung 66): strikte Trennung von Gerätesteuerung (Gerätetreiber für Hardware-Funktionseinheiten) und Applikationsebene, Abstrahierung der Parametriersprache für die Synthesizer-Software bis zur Ebene, die eine Einarbeitung des Anwenders obsolet werden lässt, Simulation des Syntheseverlaufs zur Evaluierung, physikalische Prozessüberwachung zur Regelung im Gegensatz zur einfachen Steuerung des Syntheseverfahrens, Funktionserweiterung durch Vernetzung mit weiteren Anwendungen, wie z. B. Datenbanken, Visualisierungssoftware, usw., sowie Etablierung autonomer Steuerungsmöglichkeiten und Vernetzung mit anderen Gerätekomponenten. In dieser Zukunftsvision stellt die Umsetzung der neue Gerätegeneration an Oligosaccharid-Synthesizern nur ein erster entscheidender Schritt zu einer weit umfassenderen und funktionaleren Gerätegeneration dar: dem Molekülsynthesizer als vernetztes Laborgerät.

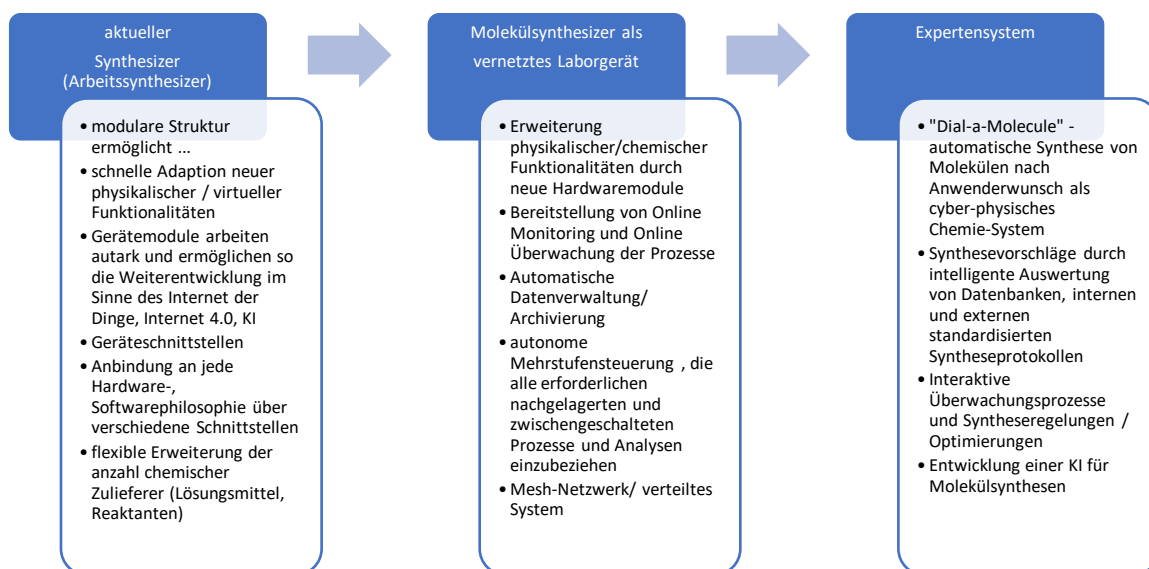


Abbildung 165: Zukunftskonzept - Funktionalitäten von Arbeitssynthesizer, konzeptionierten Molekülsynthesizer und Expertensystem

Bei der bisherigen Entwicklung lag der Fokus auf der Steuerungssoftware durch Gerätetreiber, die als eigenständige Funktionsmaschinen mit selbständigen Fehlermanagement agieren. Dieser

wesentliche Erfolg des Entwicklungsschrittes führte zu einer belastbaren, hohen Stabilität in der Gerätesteuerung. Bereits die drei soft- und hardwareseitig konstruierten Oligosaccharid-Synthesizer beinhalten die Möglichkeit flexibler Erweiterung und effizienter Anpassung an Anwenderwünsche. Hardwareseitig geschieht dies durch weitere Modularisierung von Funktionskomponenten. Man stelle sich das Endprodukt vereinfacht als Baukasten-System vor, bei dem Basis- und Zusatzmodule hardwareseitig leicht miteinander verbunden werden und dann sofort funktional miteinander interagieren können. Was hardwareseitig in dieser Gerätegeneration noch nicht umgesetzt werden konnte, weil dazu eine Neukonstruktion der Module für ein vereinfachtes und standardisiertes Montagesystem notwendig wäre, softwareseitig gilt dies bereits jetzt für die Steuerungssoftware. Die erreichte Komplexitätsstufe der umgesetzten Steuerungsarchitektur geht weit über das hinaus, was zur reinen Ansteuerung einer Oligosaccharidsynthese erforderlich ist und umfasst auch die Möglichkeit mit derselben Applikation einen, im Verhältnis wesentlich einfacher zu steuernden, Peptid- und DNA-Synthesizer zu realisieren. So erfordert der Betrieb des Synthesizers als Peptid-Synthesizer lediglich den Einbau weiterer Pumpen-Module, die softwareseitig in die Applikationsebene der Gerätesteuerung eingelinkt werden, während das Modul der Temperierung in der Regel dafür keine Verwendung finden wird. Der Betrieb des Synthesizers zur Erzeugung von DNA / RNA Segmenten benötigt nur wenig weitere Standard-Hardwaremodule. Hier wäre z. B. aufgrund der verwendeten Chemikalien hoch materialbeständige Pumpensysteme zu bevorzugen. Aber auch diese Möglichkeiten sowohl Oligosaccharide, Peptide und DNA am gleichen Gerät synthetisieren zu können, wäre nur ein weiterer Schritt auf dem Weg zum umfassenden Molekülsynthesizer. Mit der Entwicklung neuer Hardwaremodule können weitere Syntheseverfahren umgesetzt werden, u. a. automatische Parallelsynthesen an mehreren Reaktionsgefäßen, Modulsysteme für Durchflusssynthesen zum Parallel- oder Alternativbetrieb oder vernetzte Synthesen mit der gleichzeitigen Steuerung mehrerer Syntheseapparate. Unter Verwendung zusätzlicher Überwachungsmodule wie Sensoren und Aktoren könnte zudem ein bislang rein steuerungs-basiertes System in ein regelungs-basiertes System überführt werden.

Stand im ersten Schritt der Entwicklung von Zustandsmaschinen für die Steuerungssoftware das Back-End³⁴ der Applikation im Mittelpunkt, so liegt die Herausforderung der nächsten Entwicklungsstufen besonders auch in der Flexibilisierung der Benutzerschnittstelle (Front-End³⁵). Mit der Abstrahierung der Parametriersprache zur Syntheseführung ließe sich das Konzept einer prozess-gesteuerten Automations-Architektur in eine service-basierte Automations-Architektur überführen. Insbesondere die Vernetzung mit funktionaler Standard-Software könnte weitreichende

³⁴ Back-End bezeichnet in der Informationstechnologie nach dem Schichtenmodell die Software, näher am System verwendet und der Datenverarbeitung dient, wie hier die Gerätesteuerung.

³⁵ Front-End bezeichnet in der Informationstechnologie die Software, die sich näher am Anwender befindet, z.B. die Eingabeoberfläche.

Vereinfachungen und Vorteile für den Anwender mit sich bringen. In einem fortschrittlichen Workflow entwirft nun als Beispiel der Chemiker seine Synthese in ChemDraw, während der Biologe oder Mediziner mit seinem gewohnten Moleküldesign-Werkzeug die gewünschte Struktur erzeugt. Alle diese Anwendungen sind über definierte Schnittstellen in die Synthesizer-Software integriert. Das gewünschte Synthesergebnis aus ChemDraw oder anderen Anwendungen wird als Syntheseanforderung von der Synthesesteuerung interpretiert. Der Benutzer benötigt selbst keine weiteren Gerätekenntnisse mehr. Die autarken Zustandsmaschinen der einzelnen Synthesizer-Komponenten überprüfen die Syntheseanforderung, geben eine Rückmeldung an die Steuerung, welches die Informationen mit der vorhandenen Hardware-Funktionalität abgleicht. Nach erfolgter Simulation der gewünschten Synthese durch Evaluierung der benötigten Syntheseschritte, kann durch Datenbankabgleiche mit verwandten Synthesen die Syntheseführung weiter optimiert werden, auch werden Informationen zur Gefahrenbewertung der gewünschten Endproduktes und der Zwischenstufen gesammelt oder Vorschläge für alternative Syntheserouten dem Benutzer angezeigt. Ist die Funktionalität der gewünschten Syntheseanforderung hardwaretechnisch gewährleistet, erstellt sich das Steuerungsprogramm automatisch. Der Benutzer erhält die notwendige Schritt-für-Schritt Anleitung zur Inbetriebnahme des Gerätes für die geplante Synthese. Intelligente Sensoren überwachen Synthesefortschritte, optimieren oder regeln Syntheseverläufe und protokollieren auch funktionale Zwischenschritte in der Syntheseverfahrensdatenbank.

Die Übersetzung von chemischen Code in Hardware-Steuerungscode übernehmen dabei die für den Benutzer unsichtbaren Funktionsmodule der Software analog zu den autonom arbeitenden Gerätetreibern der Hardware-Komponenten.

Dabei ist auch der dezentrale Betrieb des Syntheseapparates möglich, um z. B. gefährliche chemische Synthesen in einem separaten Syntheseraum mit entsprechenden Sicherheitsvorkehrungen durchführen zu können. Die Überwachung der Synthesefortschritte kann über sichere und damit verschlüsselte Verbindungen via WLAN, über das Internet, über eine App auf dem Handy oder einem Tablet erfolgen. Hierbei überwacht eine Kamera und verschiedene Sensoren sowohl den sicheren Betrieb des Syntheseapparates als auch die Syntheseschritte, wobei entsprechende neue funktionale Hardwaremodule Analysen von Zwischenproben vornehmen, z. B. Detektion von Farbumschlägen, PH-Wert-Messungen, Online-NMR Analysen usw. Der Anwender kann dabei jederzeit den Fortschritt seiner Synthese beobachten, erhält im Fehlerfall Warn- oder Alarmmeldungen, wird informiert, wenn weitere Aktionen durch den Benutzer notwendig sind (z. B. Nachfüllen von Chemikalien) oder die Synthese vollendet wurde.

Für die nächsten Schritte zur Weiterentwicklung des Systems in einen service-basierten Syntheseautomaten (Abbildung 166), konnten bereits eine Vielzahl von Vorversuchen und

Machbarkeitsanalysen durchgeführt werden, die in den nachfolgenden Kapiteln weiter vertieft werden.

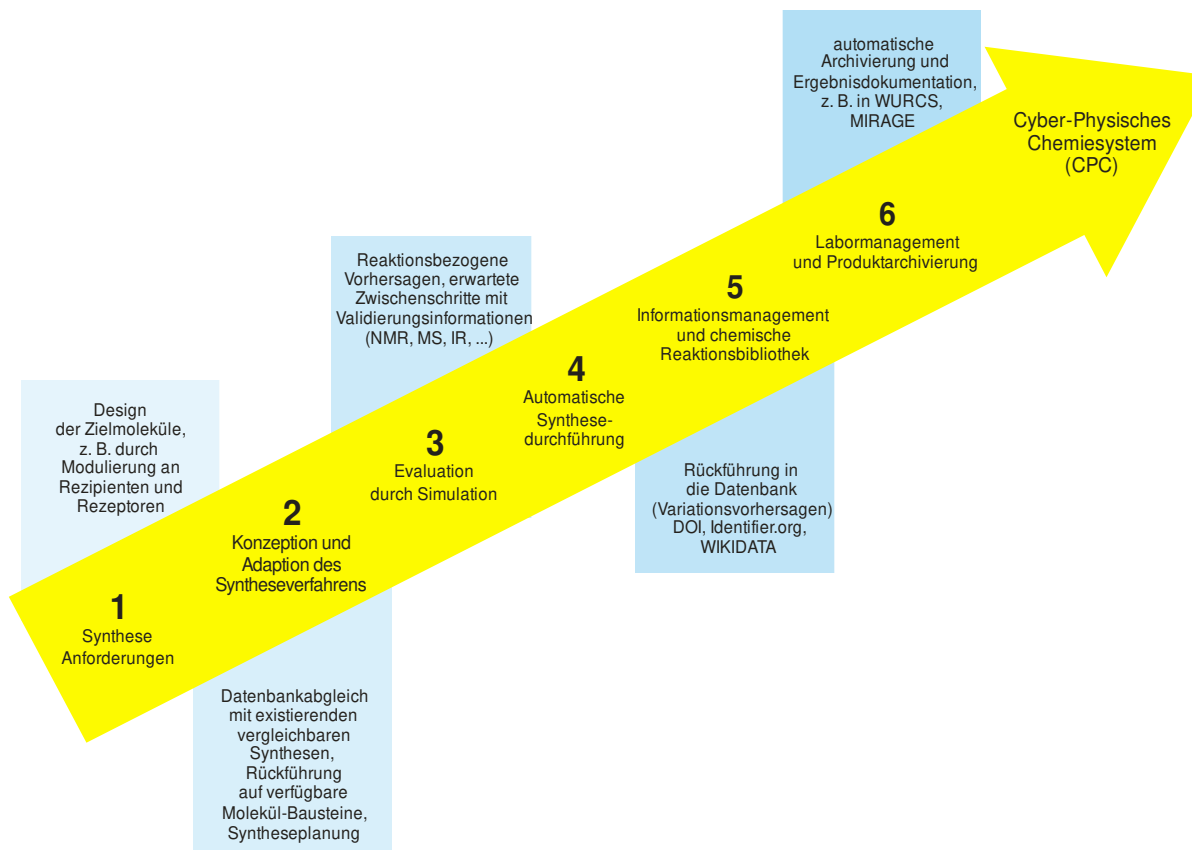


Abbildung 166: Arbeitsfluss zur Entwicklung eines service-basierten Labormanagementsystems. Mit den Begriffen aus dem Konzept Industrie 4.0 kann damit die Vorgehensweise beim Entwurf eines cyber-physischen „Chemie – Automatisierungs“ - Systems beschrieben werden.

Grundsätzlich erweckte das strikt umgesetzte Konzept der Flexibilität, Standardisierung und Modularisierung in der Applikationsebene autonom agierender Geräteeinheiten, in Anbetracht der Komplexität und Größenordnung der Entwicklungsarbeit, durchaus Zweifel an der Umsetzbarkeit bzw. Zukunftssicherheit einer solchen Konzepts. Es entstand die Frage, ob sich die Technik zukünftig nicht vielmehr in Richtung hoch spezialisierter „Stand-Alone“-Anwendungen entwickelt, die für sich genommen effizient einen definierten Funktionsumfang abdecken.

Bestätigung für die zukunftsweisende Richtung, welches die Idee, die 2010 mit dem Exposé begann und ab 2011 mit der Realisierung der drei Synthesizer umgesetzt wurde, erhielt ich 2013 mit der Veröffentlichung der Forschungsagenda: Plattform Industrie 4.0.^[214] Es entstammt einem Hightech-Strategie-Projekt der Bundesregierung und mehrerer deutscher Wirtschaftsverbände mit dem Ziel der Modernisierung der industriellen Produktion durch innovative Informations- und Kommunikationstechnik. Dabei zielt es auf die Vereinbarkeit mehrerer aktueller Zukunftstrends wie Automatisierung, Konnektivität, Sicherheit, Mobilität und Globalisierung.

Technologisch steht hinter dem Begriff „Industrie 4.0“ auch das Konzept „Internet der Dinge“³⁶. Darunter wird die Vernetzung von physischen und virtuellen Gegenständen mit dem Internet verstanden. Durch selbstständige Kommunikation von Gegenständen über das Internet kann so ein Material- oder Produktionsfluss festgelegt werden.^[215]

Das in der neuen Synthesizergeneration umgesetzte Konzept lässt sich durch zusätzliche Kommunikation über ein Netzwerk, z. B. einer HTML5-Benutzeroberfläche^[216], mit den Begrifflichkeiten des Industrie 4.0 -Projekts zu einem cyber-physischen System³⁷ weiterentwickeln, in denen informatische, softwaretechnische Komponenten mit mechanischen und elektronischen Teilen interagieren können. Das cyber-physische System soll smarte Fabriken ermöglichen, die selbstständig handeln, sich organisieren und optimieren können, wobei während des Prozesses Informationen online zur Verfügung gestellt werden.^[217] Gleiches kann mit dem Molekülsynthesizer als smartes Laborsystem umgesetzt werden.

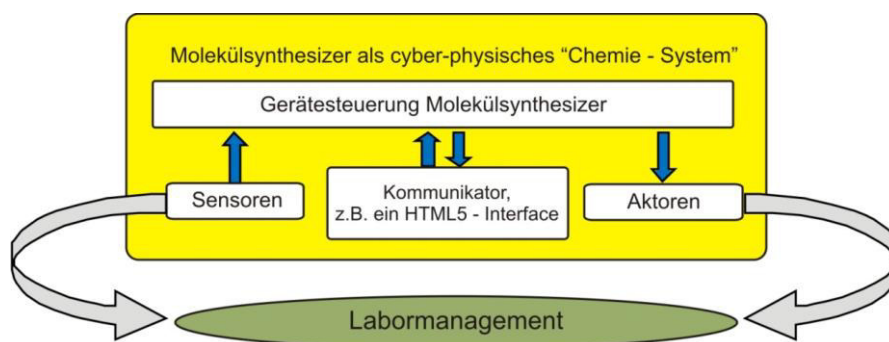


Abbildung 167: Schematische Darstellung des Molekülsynthesizers als cyber - physisches System im Kontext des Zukunftsprojektes Industrie 4.0

Aufbauend auf der beschriebenen aktuellen Synthesizer-Architektur, kann sich der um verschiedene funktionale Hardwaremodule zu ergänzende Molekülsynthesizer langfristig zu einem Expertensystem weiterentwickeln, das in der Lage sein kann durch digitalen Zugriff auf die gesammelten biologischen und chemischen Daten und deren effiziente Nutzung, eine Vielzahl von funktionellen Molekülen bedarfsgerecht und in großem Umfang herzustellen und zu reinigen, sowie das Ergebnis von Reaktionen vorherzusagen oder zu optimieren.

³⁶ Engl. „Internet of Things“, Kurzform: „IoT“

³⁷ Systeme, die reale (physische) Objekte und Prozesse mit informationsverarbeitenden (virtuellen) Objekten und Prozessen über offene, teilweise globale und jederzeit miteinander verbundene Informationsnetze verknüpfen. Optional nutzen CPS lokal oder entfernt verfügbare IT-Dienste, verfügen über Mensch-Maschine-Schnittstellen und bieten die Möglichkeit zur dynamischen Anpassung des Systems zur Laufzeit.

Internationale, alternative Bezeichnungen für Industrie 4.0 sind neben „cyber-physical systems“, auch „advanced/smart/digital manufacturing“, „smart factory“ oder „advanced/smart/digital production/industry“.

In diese Richtung zielen auch die derzeit aktiven Initiativen wie das „Dial-a-Molecule Grand Challenge Network“³⁸ in Großbritannien und das von DARPA gesponserte Projekt „Make it“³⁹ in den USA, die Reihe von Entwicklungen angeschoben haben, um die dazu erforderlichen Protokolle zur Erkennung und den Zugriff durch automatische Datenverarbeitung zu entwickeln.^[218]

5.2 Weiterentwicklung der Hardware

5.2.1 Zusatz-Module und hardwaretechnische Verbesserungen

Modularisierung und Standardisierung von Funktionskomponenten auf ein 19“ Rack Standard-System erlaubt eine flexible Erweiterung mit weiteren Modulen, welche den Funktionsumfang des Syntheseautomaten erheblich ausweiten können. Dies wurde bereits in einigen Prototyp-Modulen für die Basisfunktionalität des Synthesizers umgesetzt, die im nachfolgenden Kapitel vorgestellt werden. Dafür ist eine robuste, flexible und zukunftssichere Kommunikations-Infrastruktur zwischen den einzelnen Gerätekomponenten notwendig. In den aktuellen Systemen leisten dies noch die geräteeigenen Schnittstellen. Für ein Konzept mit hohem hardwaretechnischen Freiheitsgrad ist jedoch die Umsetzung über ein modernes, zukunftssicheres Bussystem notwendig. Aufgrund der bisher absehbaren Komplexität und der Kosten, bietet sich dafür CAN⁴⁰/CANOpen⁴¹ oder Ethernet⁴² an.

Mit der Unterstützung einer SPS⁴³ als Sicherheitselement, um erkennbares Gefahrenpotential bereits frühzeitig zu analysieren und Abhilfemaßnahmen (Notabschaltung, Druckreduktion, Überlauf) einzuleiten, kann der Sicherheitslevel des Automatisierungssystems erheblich erweitert werden. Bei automatisierten Systemen besteht die Notwendigkeit den Sicherheitsstandard dem Grad der Automation anzupassen, da ein solches System aufgrund der leichten Bedienbarkeit, auch von fachfremden Personal benutzt werden kann.

Durch ein elektronisches Manifold zur individuellen Druckbereitstellung im Automatisierungssystem aus industriellen Standardkomponenten der Steuerungs- und Automatisierungstechnik, können den unterschiedlichen Anforderungen, die sich aus den verschiedenen Schutzgasvarianten (Temperatur, Druck) ergeben Rechnung getragen werden. Der Prototyp wird in einem nachfolgenden Kapitel vorgestellt.

³⁸ Zusammenarbeit des Engineering and Physical Sciences Research Council mit der Royal Society of Chemistry und der Institution of Chemical Engineers, Vereinigtes Königreich, zur Ermittlung der Chancen zu den großen Herausforderungen in der chemischen Wissenschaft und des Ingenieurwesens.

Online unter: <http://generic.wordpress.soton.ac.uk/dial-a-molecule/> [letzter Abruf: 10.01.2019]

³⁹ Online unter: <https://www.darpa.mil/program/make-it> [letzter Abruf: 17.01.2019]

⁴⁰ Akronym für Controller Area Network, serielles Bussystem.

⁴¹ Auf CAN basierendes Kommunikationsprotokoll, Verwendung in der Automatisierungstechnik und zur Vernetzung innerhalb komplexer Geräte

⁴² Spezifiziert Software und Hardware für kabelgebundene Datennetze; häufigste verwendeter LAN Typ in der Netzwerktechnik

⁴³ Abkürzung für speicherprogrammierbare Steuerung.

Die Temperaturführung innerhalb der jeweiligen Module, kann vor allem bei der Kühlung durch die Verwendung von Peltier - Elementen, Kompressor oder, für besonders große Gradienten, mit Flüssigstickstoff erreicht werden. Ein Heizmodul ist sowohl durch Mikrowelle als auch IR-Strahlung realisierbar.

Um die Syntheseführung schrittweise zu einem Optimum in den unterschiedlichsten Syntheseparametern zu fahren, ist die Einbindung eines Online Monitoring (NMR, UV/VIS, Raman, MS usw.) unerlässlich. Jüngst wurde die Verwendung von Dünnschichtchromatographie in Verbindung eines fotolabilen Linkers auf einem Harz vorgestellt, welches eine bequeme und einfache Methode zur Echtzeit-Überwachung von Festphasenreaktionen darstellt^[219] und sich ebenfalls in die automatischen Protokoll-Mechanismen des Molekülsynthesizers integrieren ließe.

Nur über die kontinuierliche Auskopplung von definierten und reproduzierbaren Informationen in sinnvoll strukturierte Datenbanken, können selbstlernende Verfahren (KI und neuronale Netzwerke), sowie kombinatorische Chemie zum Einsatz kommen.

Aufgrund der zentralen Bedeutung des Online-Monitoring für den Molekülsynthesizer und das nachfolgende Expertensystem, wird das Konzept im nachfolgenden Kapitel „Integration von Online-NMR, NMR Simulation, sowie Validierung & Konsolidierungsprozesse“ weiter konkretisiert.

Damit für die Kombinatorik die Datenbanken mit den notwendigen Informationen möglichst zeitnah befüllt werden können, ist ein Parallelverteiler innerhalb der Automatisierungssysteme zu implementieren, mit dem die gleiche Synthese in verschiedenen Reaktionsgefäßen zeitgleich durchgeführt werden kann. Hierbei sollte nur ein definierter Freiheitsgrad in der Reaktion variiert werden, um eine bisher unerreichte Reproduzierbarkeit zu gewährleisten. Neben den einzelnen oder mehreren Reaktionsgefäßen, lässt sich auch eine Funktionsmodul-Gruppe für Durchfluss-Synthesen aufnehmen, um alternativ oder auch parallel Festphasen-Eintopf-Synthesen und Durchfluss-Synthesen an einem Gerät durchzuführen. Bei Durchflussverfahren kann durch die Inline-Integration von Analysen in Echtzeit eine große Menge an Informationen über den Reaktionsfortschritt, die katalytische Aktivität und Stabilität usw. zum Aufbau der Systemdatenbank für den Molekülsynthesizer erhalten werden. Darüber hinaus ist es möglich, den Reaktionsfortschritt und die Selektivität durch manuelle oder automatisierte Feedback-Optimierung zu beeinflussen.^[220] Die Kombination von Durchflussreaktoren mit Überwachungstechnologien und einem Rückkopplungsalgorithmus ermöglicht es einzelne Syntheseschritte systematisch zu optimieren. Damit lässt sich ein einfaches selbstoptimierendes System-Modul aufbauen. Aktuelles Beispiel ist ein autonomer, sich selbst optimierender Strömungsreaktor zur Synthese des Naturprodukts Carpanon.^[221]

Zur Durchführung fotoselektiver Synthesen wurden Vorversuche an einem selbstentwickelten

Fotoreaktor durchgeführt. Auch für die Abspaltungsreaktion hat die Verwendung frequenzselektiver Substituenten eine Vielzahl von Anwendungsmöglichkeiten. Der Fotoreaktor-Prototyp wird in einem eigenen Kapitel nachfolgend vorgestellt.

Gerade für die Kombinatorik oder für die Ausführung geplanter flexibler und paralleler Syntheserouten, lässt sich so ein hoher Zuwachs an intelligenter Funktionalität erzielen. Aufgrund der verwendeten Reaktanten und Lösungsmittel sollten die Oberflächen der jeweiligen Komponenten, die den Chemiebereich kontaktieren, PTFE-Folien beschichtet oder aus Edelstahl gefertigt sein. Die Verwendung von Quickkupplungen ermöglicht den komfortablen Austausch der Reaktionsgefäße, um mit unterschiedlichen Mengen und Methoden (Katalysatoren, elektrochemischer Stimulation, Microwelle) arbeiten zu können. Hierfür wurden bereits einige Vortests zur Dichtigkeitsprüfung der Quickkupplungen durchgeführt. Bei den bisher getesteten Varianten ist mit Swagelok (SS-QTM2A-B) als einzige der Temperaturbereich von +60°C bis -40°C ohne Flüssigkeitsaustritt mehrfach mit Förderungsdruck durchfahrbar gewesen. Eine Füllstandsmessung, sowohl der Substanzgefäße wie auch vom Abfallbehälter, kann den Anwender aktiv unterstützen und vermeidet im Vorfeld bereits mögliche Systemfehler.

Durch eine zusätzliche Evakuierungsmöglichkeit des Systems, ließe sich der Restfeuchteanteil absenken. Sowohl ein Ausheizen des Reaktors als auch das abschnittsweise Spülen mit Schutzgasatmosphäre und Einkopplung von Molsieben ist eine zielführende Ergänzung. Zur Vorbeugung von Alterung und Verunreinigungen durch Lagerung des Lösungsmittels Dimethylformamid, welches das am häufigsten verwendete Lösungsmittel für die Fmoc – Abspaltung ist, kann eine automatische Regeneration vor Verwendung im Syntheseverlauf durch Durchblasen mit Inertgas (N₂) oder alternativ Vakuumbeschallung erreicht werden, wie in einer aktuellen Veröffentlichung gezeigt wurde.^[222] Für andere Lösungsmittel müssten ebenfalls automatische qualitätssichernde Maßnahmen Standard werden, um eine gleichbleibende Prozessqualität zu gewährleisten, die sich an der Ausbeute des Syntheseproduktes zeigt.

Im Anschluss an eine Synthese sollte sowohl eine automatisierte Proben-Aufreinigung (HPLC) wie auch eine temperierte Einlagerung standardisiert erfolgen. Hierdurch wären Wiederaufbereitungen bei nachfolgenden Synthesen möglich, eine wesentliche Grundlage für die Durchführung von kombinatorischen Synthesen. Die technische Realisierung wurde bereits mit der Firma Festo komplett analysiert und konzipiert. In dieser Variante wurden die Buildingblockverwaltung mit Schutzgasabfüllung, Temperierung und neuer mechanischer Zuführung veranschlagt.

Da eine moderne Liftsteuerung für die Probenverwaltung zum Einsatz kommt, würde die Grundfläche des Gesamtsystems trotz erheblicher Funktionserweiterung nicht zunehmen. Jedoch scheitert eine direkte Umsetzung aufgrund der Kosten von aktuell ca. 500.000,- €. Ziel ist die

Bereitstellung der neuen Funktionalitäten sowohl von der Software wie auch der Hardware über API's und Webserver (Swagger).

Hierdurch lassen sich komplexe verteilte Systeme vollwertig mit allen angesprochen Funktionen realisieren. Um jedoch auch eine kostengünstigere visuelle Überwachungsmöglichkeit bereitzustellen wurden mit dem Raspberry Pi in Kombination mit dem Kameramodul verschiedene Vorversuche durchgeführt. Im einfachsten Fall stehen die Bilder durch:

```
image = DeviceRead["RaspiCam"]  
Export["/home/pi/image.jpg", image]
```

als eine Datei zur Verfügung, auf die LabVIEW zugreifen kann. Falls kein Vision Paket von LabVIEW vorliegt, können auch die Bildbearbeitungen bereits mit Mathematica auf dem Raspberry vorgenommen werden. Die Implementierung von Makerboards ist ebenfalls ein zentrales Konzept im Molekülsynthesizer und wird deshalb in einem eigenen Kapitel ausführlich behandelt.

5.2.2 Hardware-Modularisierung und Standardisierung

Der Hardware-Modularisierung wurde bei der Konzeptionierung höchste Priorität zugemessen. Es erleichtert den Zugriff auf die Gerätekomponenten und die Wiederverwendbarkeit einzelner Module in anderen Systemen und ermöglicht insgesamt eine größere Flexibilisierung bei der Komponentenordnungen. Dies ist wichtig, wenn weitere Module den Basisfunktionalität des zukünftigen Molekülsynthesizers ergänzen. Für einzelne Grundmodule des Oligosaccharidsynthesizers wurde für die Konzeptstudie bereits ein Re-Design durchgeführt und als Prototyp realisiert.

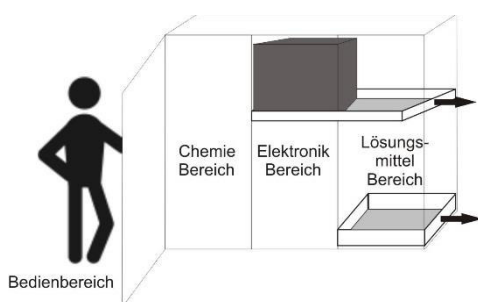


Abbildung 168: Konzeptioneller Aufbau des Synthesizers als mobiles 19“ Rack in der Seitenansicht mit den drei Systembereichen: Chemie, Elektronik, Lösungsmittel. Elektronik und Lösungsmittelbereich sind auf Vollauszugsschienen montiert.

Als Hauptträgerrahmen der modularen Komponenten kommt ein mobiles 19“ Standard-Racksystem zur Anwendung, in dem durch Standardbaugruppensysteme die einzelnen funktionalen Komponenten integriert werden. Die obere Einschubebene strukturiert sich in sechs Modulbaugruppen: zwei Spritzeneinheiten, zwei Rotationsventileinheiten, ein Reaktionsgefäß und eine Lösungsmiteleinheit. Hierdurch sind die wichtigsten Komponenten der aktuellen Arbeitssynthesizer platzsparend zusammengeführt. Die weiteren Funktionseinheiten sind rückseitig in der gleichen Funktionsbaugruppe untergebracht.

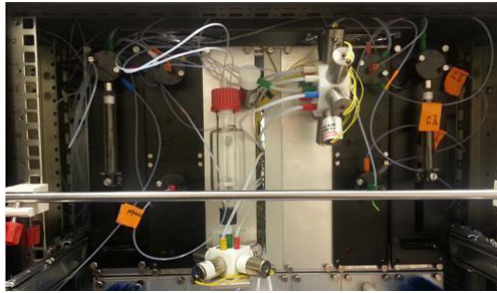


Abbildung 169: Prototyp modularer Systemkomponenten eingebaut in einem 19" Rack. Maximale Flexibilität im Einbau oder Umbau neuer Komponenten, sowie die leichte Wiederverwendbarkeit in weiteren Systemen ist damit gewährleistet



Abbildung 170: Übersicht über rückseitige Hardware-Module in kompakter, für die 19" Gerätevariante optimierter Bauweise.

Die gesamte Elektronik und Steuerungseinheit ist im Mittelteil des Baugruppeneinschubs platziert, wobei sich der Platzbedarf bei gleicher Funktion außer bei den Flüssigkeitsbehältern auf neun Höheneinheiten (HE) reduziert.



Abbildung 171: Zur Reduzierung der Gesamtgeräteabmaße ist eine platzeffiziente Anordnung der Gerätekomponenten, hier der Datenerfassungskomponenten, notwendig.

Die Druckverteilungseinheit (Abbildung 172a) hat eine lokale digitale Anzeige zur Visualisierung des Versorgungsdrucks des verwendeten Schutzgases. Durch die Verteilung auf zehn unabhängige Druckstränge stehen hier flexible Erweiterungsmöglichkeiten zur Verfügung, da aktuell nur sieben unterschiedliche Druckpegel zur Anwendung kommen.

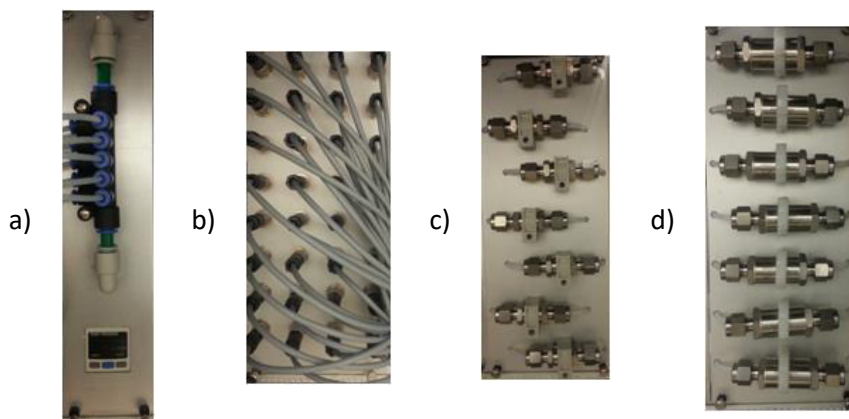


Abbildung 172:
a) Druckversorgung
b) Ventilsteuerung
c) Rückschlagventilblock
d) Filterblock

Die Einbauposition innerhalb des 19“ Rahmens ermöglicht den direkten Zugang zu den Spulenventilen (Abbildung 172b) von der Anwenderseite, während die Verkabelung mit der DIO-Anschlussbox im mittleren Elektronik-Bereich des Synthesizers untergebracht ist. Bei der Anslusstechnik kommen, wie in Abbildung 173 gezeigt, arretierbare Steckerkombinationen zum Einsatz, die auf Seiten des elektronischen Systembereichs fest mit der DIO-Anschlussbox verkabelt sind. Die arretierbaren Zwischenstecker ermöglichen den leichten selektiven Ein- und Ausbau der Spulenventile zu Wartungszwecken, ohne dabei Eingriff in die essenzielle Systemverkabelung nehmen zu müssen.

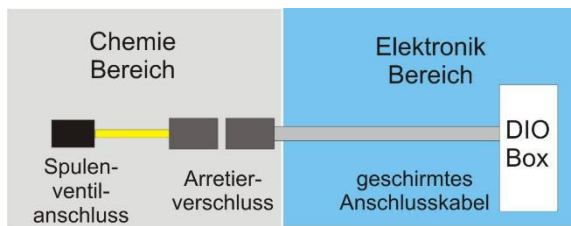


Abbildung 173: Der Zugriff (Wechsel/Wartung) auf den Anschluss der Spulenventile ist durch den Einbau eines Adapters im Mittelteil des Synthesizers vereinfacht.

Für diese Verbindung wurden qualitativ hochwertige Anschlusskabel eingesetzt, wobei der ausreichend große Kabelquerschnitt die elektrischen Verlustleistung durch thermische Erwärmung minimiert und die hochwertige Schirmung Signalstörungen verhindert. Die Verkabelungen sind langlebig, garantieren eine hohe Signaltreue und sind basierend auf den Erfahrungen mit dem Arbeitssynthesizer wartungsfrei.

Die Rückseite des mobilen 19“ Racks ist mit einer verschließbaren Tür versehen, um dem Anwender einfachen Zugang für Wartungs- der Reparatureingriffen zu ermöglichen. Dies gilt vor allem in Hinblick auf die Wartung der Rückschlagventile (Abbildung 172c). Dazu wurden diese in einem eigenständigen Modulrahmensystem platziert. Mit Hilfe versetzter Haltesysteme ist die Ausrichtung räumlich optimiert, wobei der Zugriff auf alle sieben Rückschlagventile auch für unerfahrene Bediener möglich ist.

Auch für die Inline Filter (Abbildung 172d) ist eine wiederverwertbare Befestigungsmöglichkeit installiert, so dass alle sieben Filterelemente einfach gewartet oder ausgetauscht werden können. Durch Beschriftung und symbolische Anzeige der Flussrichtung werden Einbaufehler minimiert. Alle vorgestellten Baugruppenelemente sind auf Trägersystemen untergebracht, die in Vollauszugsschienen eingeklinkt sind. So kann bei Wartungsaufgaben auch auf die Mittelteile schnell und ohne zusätzliches Werkzeug zugegriffen werden.

5.2.3 Prototyp eines druckgesteuerten Manifolds

Die druckbasierte Zulieferung der Lösungsmittel ist abhängig von den angelegten, im Syntheseverlauf konstanten Versorgungsdrücken. Durch die verfahrensbedingten Schwankungen der Füllzustände in den Flüssigkeitsreservoirs und dem sich daraus dynamisch einstellenden Dampfdruckgleichgewicht, garantieren einfache mechanische Druckregler, wie sie im Arbeitssynthesizern verbaut wurden, keine langfristige Druckstabilität. Ein höherwertiges, druckgeregeltes Massen - Durchflusssystem wurde konzeptioniert und als Prototyp realisiert.

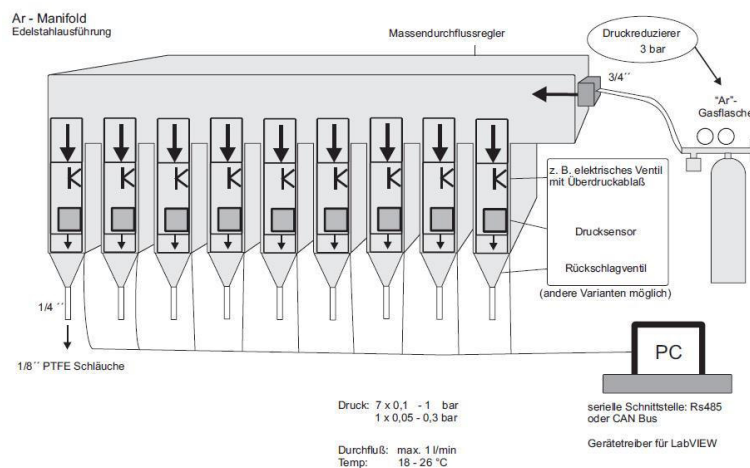


Abbildung 174: Schematische Darstellung des elektronisch druckgeregelten Argon-Manifolds

Die Regler können über ein serielles Bus System mit dem Steuer- PC kommunizieren wodurch sowohl die elektronische Druckaufzeichnung während der Synthese als auch individuelle Anpassungen vorgenommen werden. Die bereitgestellten Drücke speisen eigenständige Druck- Reservoirs, von denen die jeweiligen Flüssigkeitsgefäße mit elektronischen Ventilen beaufschlagt werden. Schwankungen in der Druckversorgung, wie bei der Entleerung der lokalen Einzelflaschenversorgungen, können dadurch weitgehend kompensiert werden.

Parallel zu dieser Regelung erfolgt die Information des Benutzers zum Austausch/Befüllung der Versorgungsflaschen in einem geeigneten Synthesestatus. Auch kann nun direkt die entnommene Gasmenge aufgezeichnet und online als Statusinformation zur Verfügung gestellt werden. Somit ist es bereits zu Synthesebeginn möglich, sich über die aktuellen Pegelwerte einen Überblick zu verschaffen. Bei einer Fehlersuche, im Falle von Problemen mit der Dichtigkeit der Verschraubungen unterstützen die erfassten Werte den Anwender.



Abbildung 175: Prototyp des elektronischen Druckverteilers zur individuellen Einstellung der benötigten Versorgungsdrücke

Die verwendeten Module erlauben eine elektronische Auskopplung der Stellgrößen, sowie der aktuellen Zustandsinformationen über serielle Schnittstellen. Diese Variante ist kleiner, kann Versorgungsfehler melden und zeichnet sich durch ein kontrollierbares Strömungsverhalten aus. Diese speziell gefertigten Druckverteilungseinheiten können bereits bei einer Abnahme von 50 Einheiten pro Jahr vom Komponenten-Hersteller mechanisch fertiggestellt, druckgeprüft und kalibriert bezogen werden.

Die für die Kommunikation mit dem elektronischen Manifold zusätzlich benötigten seriellen Schnittstellen wurden über einen USB-RS232 Adapter in den PC integriert und dessen Funktion mit den Betriebssystembordmitteln getestet. Die einzelnen Regler sind untereinander über Standard D-Sub-Stecker zu einem RS485 Bus verkabelt. Die vom Hersteller definierten Standard Kommunikationsparameter werden als Schnittstellenparameter beibehalten, um die Wartung und den Austausch zu vereinfachen. Die Funktion der Einzelkomponenten kann über die „get red-y“ – Software (vgl. Anhang 1.21.) geprüft werden.

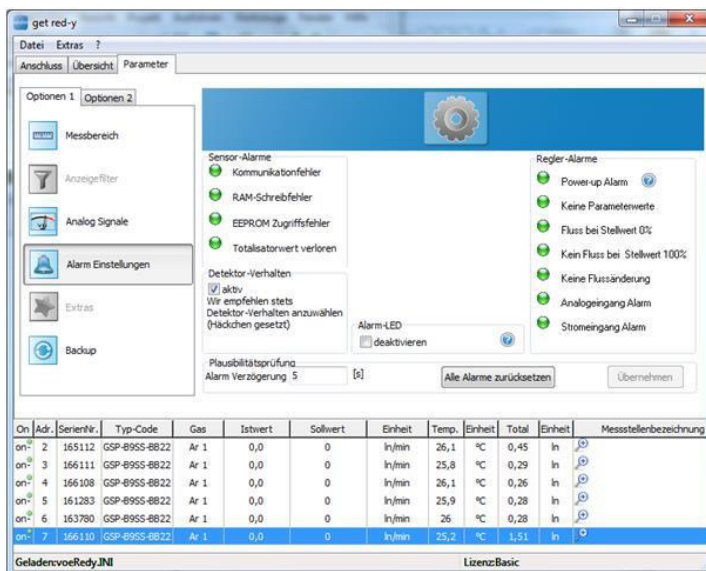


Abbildung 176: Funktionstest mit der herstellereigenen „get red-y“-Software

Durch den Zugriff auf die unterschiedlichen Geräteadressen, können die Statusinformationen abgefragt, sowie die einzelnen Komponenten getestet werden. Die Systemsituation wird jeweils online visualisiert. Neben der Erkennung der Spannungsversorgung und der Schnittstellenparametrierung, können auch die weiterführenden Gerätekonfigurationen ausgelesen

werden. Warnfunktionen und aktuelle Systemzustände lassen sich bei Fehlern über diese Oberfläche effizient für alle Regler kontrollieren.

Der Zugriff auf die einzelnen Regler erfolgt in der LabVIEW-Applikation mit selbst entwickelten LabVIEW-Gerätetreibern nach dem Consumer / Producer Konzept⁴⁴ über Warteschlangen (*Queue*)⁴⁵. Dazu müssen die entsprechenden Register für die Zugriffe aus der Synthesizer-Applikation, lesend und schreibend administriert werden. Durch die Verwendung benannter Warteschlangen sind diese wiederum in das vorhandene Gesamtkonzept ohne grundlegende Anpassungen integrierbar (vgl. *Anhang 2.16.*).

Auch für die Benutzeroberfläche wurde das Consumer / Producer – Konzept vollständig umgesetzt. Steuerungs- und Regelungsaufgaben laufen komplett unabhängig von den Interaktionen am Benutzerterminal, um benutzerinduzierte Laufzeiteffekte in der Programmabarbeitung auszuschließen. Solche Effekte können sonst durch direkte Dateninteraktion entstehen, wie z. B. bei der Visualisierung und umfangreichen Analyse. Etwaige Auswirkungen auf die geregelte Kommunikationsschritt看ette bei der Hardwarefunktionalität wären Prozessverlangsamung, Prozessabbruch oder Programmabsturz, sowie Schritkettenverstöße.

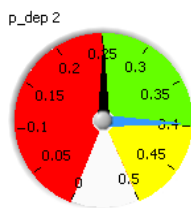


Abbildung 177: Anzeigeelement in der Benutzeroberfläche für die Druckversorgung mit farbiger Hervorhebung der jeweiligen Druckbereiche

Benutzereingaben und Kontrollen auf der Parametrier- und Anzeigeoberfläche des elektrischen Manifolds werden durch farblich codierte Anzeigeelemente unterstützt. Die empfohlenen Druckbereiche sind mit Grün, der Toleranzbereich mit Gelb, und mangelnder Druckpegel mit Rot hervorgehoben. Diese Anzeigeelemente finden sowohl bei der manuellen Einstellung des gewünschten Versorgungsdrucks als auch zur Anzeige der tatsächlich von der Sensorik gemessenen, anliegenden Drücke zur Prozesskontrolle Verwendung.

⁴⁴ Das Consumer / Producer Konzept bezeichnet in der Programmierung Mechanismen zur Prozesssynchronisierung, welches die Zugriffsreihenfolge von schreibenden und lesenden Datenprozessen betrifft. Die meisten Programmierkonzepte beinhalten den wechselseitigen Ausschluss von Prozessen beim Ausführen kritischer Abschnitte, und unterstützen dabei die Kooperation von Prozessen. Daten produzierende Prozesse und Dateninteraktionen werden hierbei als voneinander losgelöste Prozesse realisiert.

⁴⁵ Queue (engl.) bezeichnet in der Informatik eine Datenstruktur zur Zwischenspeicherung von Objekten in einer Reihenfolge vor Weiterverarbeitung.

5.2.4 Fotoreaktor – Prototyp für die fotochemische Synthese

Physikalisch induzierte Prozessschritte stellen eine interessante Optimierungsmöglichkeit bei der automatisierten Synthese von Oligo- und Polysacchariden dar. Neben der exakten Temperatursteuerung, die für den gesamten Syntheseverlauf wichtig ist, wäre Licht ein sehr attraktives Reagenz. Der in der automatischen Zuckersynthese eingesetzte mit Base abspaltbare 3-(4-((Benzyl(5-hydroxypentyl)carbamoyloxy)methyl)phenyl)propionic-Säure Linker^{46, [161]}, wird von einem fotolabilen Benzyl 5-hydroxy-2-nitrobenzyl(5-hydroxypentyl)carbamate Linker^[110] ergänzt. Der „Traceless-Linker“ ist eine Spezialvariante des fotolabilen Linkers, der nach Abspaltung am Zielmolekül ein freies reduzierendes Ende hinterlässt (Abbildung 178). Das Syntheseprodukt eignet sich damit zur direkten Verwendung in Liquid-Chromatographie-Massenspektrometrie (LC/MS) Analysen.^[113]

Die Abspaltung mit Licht unter neutralen Bedingungen stellt eine milde und orthogonale Alternative zur chemischen Variante dar, die mit einer starken Base^{47, [223]} abgespalten wird. Bei der fotoreaktiven Linker-Variante kann das Endprodukt automatisiert vom Resin getrennt werden. Es erübrigen sich nachfolgenden manuellen Aufarbeitungsschritte in homogener Phase, wobei ein sehr reines Produkt entsteht.^[224–226] Aufgrund dieser attraktiven Eigenschaften, steigt die Anzahl der entwickelten fotolabilen Linker für die Festphasensynthese und – Freisetzung kontinuierlich.^[227]

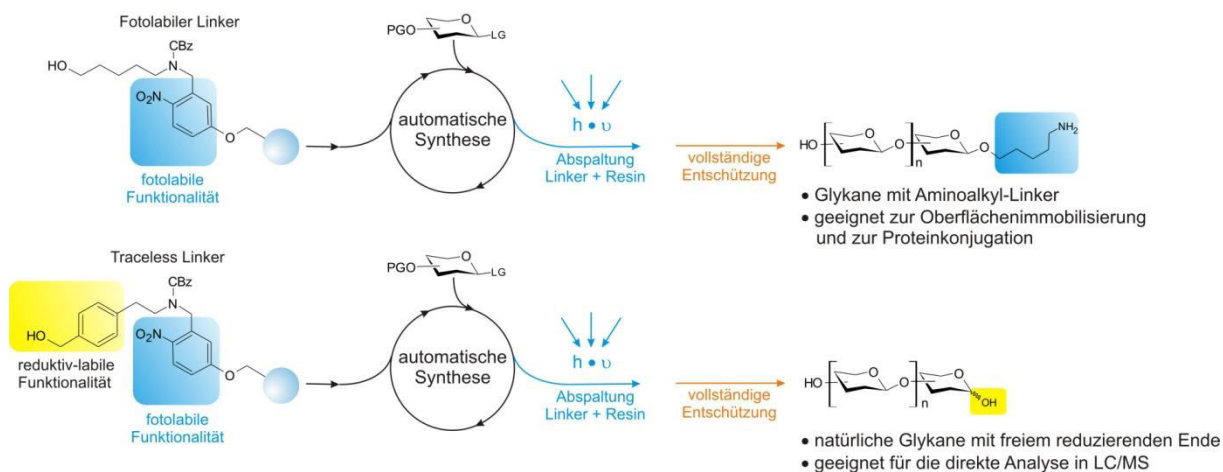


Abbildung 178: Fotolabile Linker in der automatischen Glykansynthese.^[113]

Derzeit erfolgt die Fotoabspaltung manuell in einem kontinuierlichen Fluss-Fotoreaktor (vgl. Abbildung 187) außerhalb des automatischen Syntheseprozesses. Licht als äußerer Stimulus für die Automation ist jederzeit verfügbar und lässt sich fokussiert zuführen. Dadurch ist eine präzise zeitliche und energetische Steuerung des Aktivierungsvorganges möglich, sowohl über die

⁴⁶ „Lenz“-Linker

⁴⁷ NaOMe, MeOH, CH₂Cl₂

Verweildauer des Substrats im Belichtungsbereich, über die Belichtungsstärke der verwendeten Lichtquelle oder über die direkte frequenzselektive Energiebreite.^[228,226]

Die Integration eines Fotoreaktor-Moduls in den Syntheseautomaten beschleunigt und vereinfacht den Arbeitsablauf. Das Fotoreaktor-Modul ermöglicht zudem die fotoselektive Abspaltung fotolabiler, temporärer Schutzgruppen^[109, 225, 228, 229a-e] während der automatisierten Synthese.

Für die orthogonale Schutzgruppenstrategie lassen sich damit im automatischen Syntheseprozess säure-, basen- und fotolabile Schutzgruppen kombinieren, die unabhängig und selektiv in beliebiger Reihenfolge entfernt werden können.

Seit dem Jahr 2000 wird die Kombination unterschiedlich wellenlängensensitiv spaltbarer Fotoschutzgruppen, auch als chromatische Orthogonalität bezeichnet,^[230] untersucht. Die Verwendung fotospaltbarer Schutzgruppen in Biomolekülen verursacht in der Regel eine Veränderung der ursprünglichen molekularen Struktur, Konformation oder der Ladung und bewirkt eine drastische Beeinflussung der biologischen Funktion, deren Bioaktivität durch Licht ausgelöst wird. Bei Verwendung von Kombinationen unterschiedlich wellenlängensensitiv spaltbarer Fotoschutzgruppen kann die Bioaktivität koexistierender Moleküle unabhängig voneinander ausgelöst werden. Diese unabhängige Reaktion beruht auf den verschiedenen Fotosensitivitäten der Chromophore⁴⁸ bei ausgewählter Wellenlänge λ , die durch die Quantenausbeute der Photolyse⁴⁹ und den Extinktionskoeffizienten⁵⁰ der Chromophore bestimmt werden.

Diese Strategie ermöglicht eine individuelle und sehr gezielte Aktivierung verschiedenster Vorgänge an einer einzigen Stelle im Molekül.^[224,225,230]

In den Peptid-, DNA- und RNA-Syntheseverfahren befinden sich bereits einige fotolabile Schutzgruppen, die sogenannten „caging“-Gruppen^[231] (Tabelle 5), in der Anwendung.

Kein Standard-System erfüllt alle notwendigen Anforderungen an einen optimalen Fotoreaktor für die automatische Oligosaccharid-Synthese. Deshalb wurden zwei Fotoreaktor-Prototypen als Erweiterungsmodule zum Einbau in 19“ Racks realisiert, die sowohl die geräte- und steuerungstechnischen als auch die fotochemischen Anforderungen innerhalb des Syntheseautomaten erfüllen können.

⁴⁸ Chromophor (griech. „Farbträger“) wird der Teil eines Farbstoffs bezeichnet, in dem anregbare Elektronen verfügbar sind.

⁴⁹ Fotolyse: Spaltreaktion einer chemischen Bindung in einem Molekül, als unmittelbare Folge der Absorption von Licht.

⁵⁰ Maß für die Absorption elektromagnetischen Strahlung einer Substanz in molarer Konzentration bei einer Durchtrittslänge von 1 cm und einer bestimmten Wellenlänge.

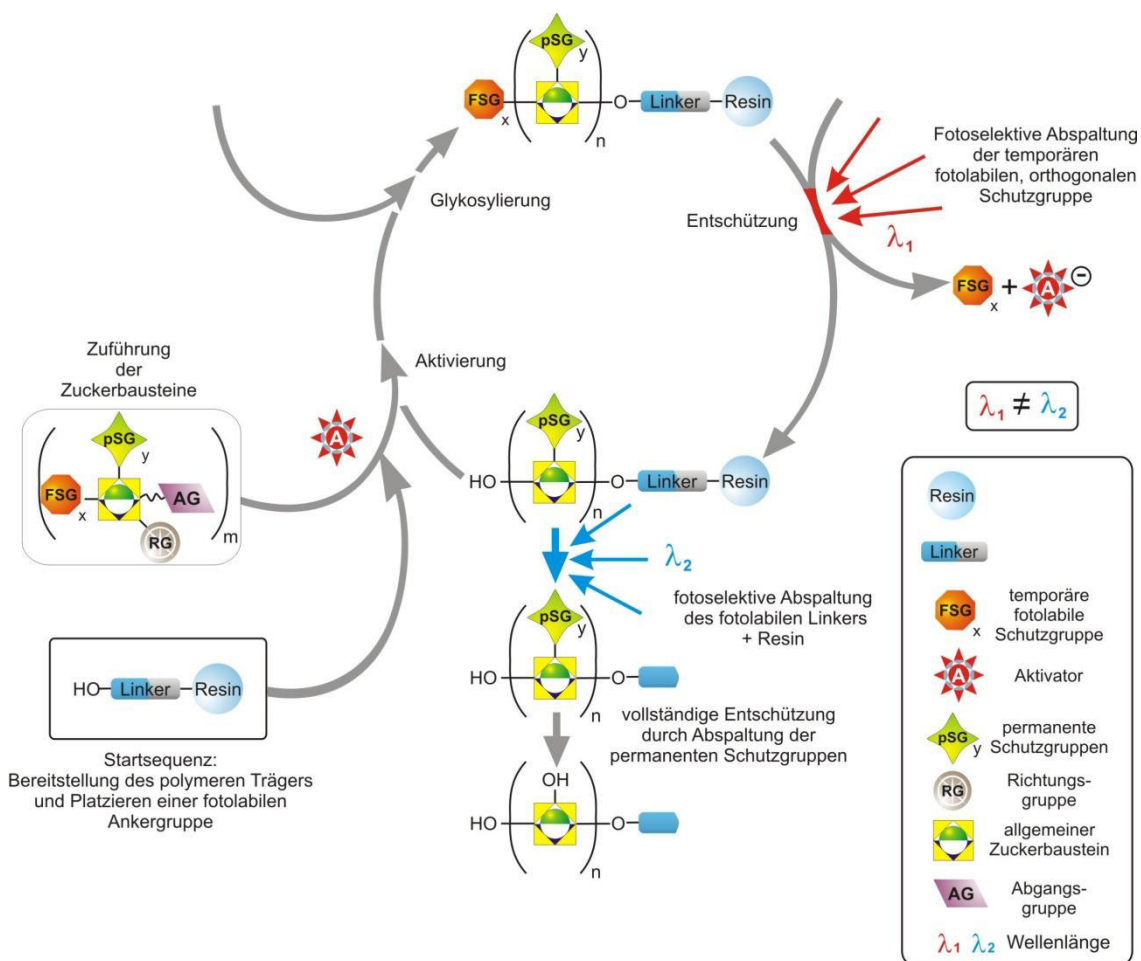


Abbildung 179: Schematischer automatisierter Syntheszyklus mit fotoselektiver Abspaltung einer oder mehrerer temporären fotolabilen Schutzgruppen, sowie integrierter Abspaltung des fotolabilen Linkers.

Schutzgruppe	Abk.	Wellenlänge Λ	Literatur
Nitrobenzyl	ONB	300	[232]
Nitroveratryl	(C)NV	320	[233]
Nitrobenzhydryl	NBH	350	[234]
Nitrobenzylamine	NBA	354	[235]
Nitrovanilin	NA	360	[225]
Nitroveratryloxycarbonyl	NVOC	360	[236]
Nitrobenzylether	NBE	365	[237]
Nitrobenzhydrylamine	NBHA	365	[238]

Tabelle 5: Fotochemisch absplaltbare Schutzgruppen

Fotoselektive Abspaltprozesse erfordern ein möglichst schmal definiertes Frequenzband, bei der idealerweise die Maximal-effizienz der emittierten Wellenlänge der Lampe genau im Absorptionsbereich der reagierenden Komponente liegt.^[239] In den bisher im Labor etablierten Belichtungssystemen sind Quecksilberdampfboegen installiert, die ein breites Spektrum emittieren (Abbildung 180), welches vom Lampentyp, dem Füllgas, dem Druck und der Wärme abhängig ist. Die Einschränkung des Frequenzbereichs erfolgt dabei meist durch Verwendung von Spektralfiltern

(Abbildung 181), die nur einige Wellenlänge übertragen und die anderen absorbieren bzw. reflektieren.

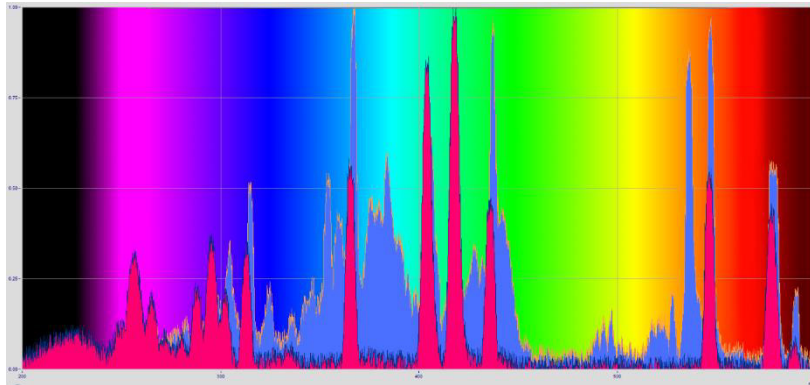


Abbildung 180: Spektralverteilung der Standard-UV-Strahlungsquelle Hg-Z1 (rot) und Hg-Z4 (blau) von Peschel UV. Diese Lampen werden oft im Labor für die photochemische Prozessentwicklung und Produktion eingesetzt.

Bei Quecksilberlampen ist das emittierte Licht zudem meist nicht auf das zu belichtende Ziel fokussiert, was somit zu Streuverlusten führt. Für die automatisierte Synthese spielt auch die Wärmeentwicklung der Lampen eine entscheidende Rolle, die konstruktiv gelöst werden muss. Die angegebene Lampenleistung bezieht sich auf das Gesamtspektrum und ist deshalb ein ungeeignetes Auswahlkriterium da für die Synthese jeweils nur ein kleiner Frequenzbereich nutzbar ist. Die tatsächliche Systemeffizienz mindert sich damit um die Verlustleistung des ungenutzten Spektrums und liegt bei den derzeit verwendeten 750 W Lampen bei nur 8 %. Maßgebliche Dimensionierungsgröße für den Fotoreaktor ist deshalb die Gesamtbetrachtung der effektiven Emissionsreichweite.

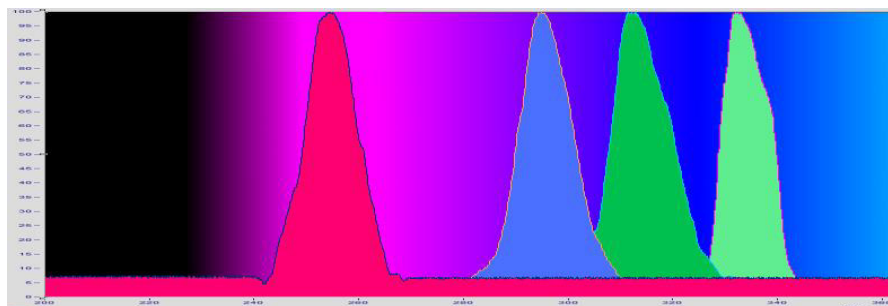


Abbildung 181: Verschiedene Übertragungsspektren für Schmalbandfilter mit typischer Bandbreite von 20 nm NS245(rot), NS297(blau), NS313(dunkelgrün), NS335(grün). Optischer Bandpassfilter blockieren durch Absorption und/oder Reflexion alle Wellenlängen oberhalb und unterhalb des spezifizierten Übertragungsbereichs.

Bei Frequenzen unter 240 nm erzeugen zudem alle UV-Lampen aufgrund ihrer hochenergetischen Strahlung aus dem umgebenden Sauerstoff Ozon. Im Labor muss dieser Fotoreaktor-Typ deshalb unter einem Abzug betrieben werden. Weitere konstruktive Varianten sind der Einsatz von Ozon-Konvertern oder die Verwendung einer Stickstoff-Schutzatmosphäre. Das UV-Schutzgehäuse muss mit elektrischen Trennvorrichtungen betrieben werden, da die Zündspannung bei Hg-Hochdruck-Emittern bis zu 10000 Volt betragen kann. Zur Unfallvermeidung besitzen kommerzielle

Fotoreaktoren eine Notabschaltung bei Gehäuseöffnung, Wiedereinschaltenschutz und sind mehrfach gekapselt. Die typische Lebensdauer dieser Lampen liegt bei ca. 2000 Stunden Betriebsdauer. Neben den Nachteilen in der Effizienzausbeute und der Lebensdauer, wird somit durch diesen Lampentyp in der Gerätekonstruktion die Einhaltung vieler gerätetechnischer Vorschriften zusätzlich erforderlich.

Alternativ hierzu weisen UV-Leuchtdioden eine schmale Frequenzbandbreite bei geringer Wärmeabfuhr auf. Doch ab einem Frequenzbereich kleiner 365 nm ist deren effektive Kosten - / Leistungsbilanz derzeit unwirtschaftlich. Relativ kostengünstig, gerade im Frequenzbereich unter 365 nm Wellenlänge, zeigen UV-Leuchtstoffröhren von den vorgestellten Varianten die geringste Wärmeentwicklung bei einer Systemeffizienz von nahezu 80 % der Lampenleistung. Sie werden im Niedervolt-Bereich betrieben und weisen mit etwa 9000 Betriebsstunden eine hohe Lebensdauer auf. Deshalb wurden sie für den Einsatz im Fotoreaktor-Prototyp I ausgewählt.

Zunächst erfolgte die Optimierung der Prototypen auf die spezifischen Eigenschaften des im Labor gebräuchlichen Standard-Linkers. Zur Vermessung der zukünftigen Spektren wurde eine Modulbaugruppe mit einem Spektrometer ausgestattet. Die hierbei als Lichtquelle zu Einsatz kommende Baugruppe enthält zwei unabhängige Lampen (Lampe1 = Halogen, Lampe2 = Deuterium) Nachfolgend sind die direkt gemessenen Spektren gezeigt, die als Textdatei ausgekoppelt wurde, um für andere Signalanalyseprogramme zur Verfügung gestellt zu werden.

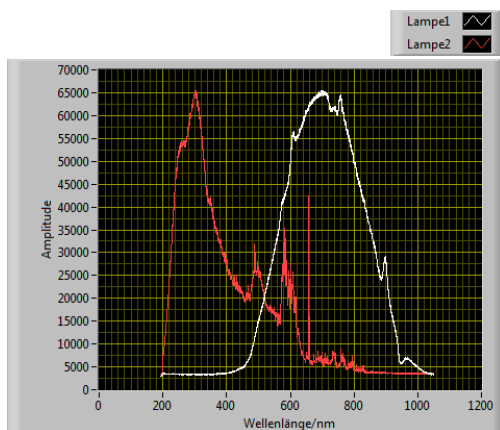


Abbildung 182: Messung der Fotospektren der innerhalb der Belichtungskomponente vorhandenen Lichtquellen, Deuterium (rot), Halogen (weiß)

Anschließend wurden die erfassten Werte in FAMOS, einer Signalanalysesoftware für Messdaten, eingelesen, skaliert, visualisiert und als Kalibrierungsdatensatz archiviert.

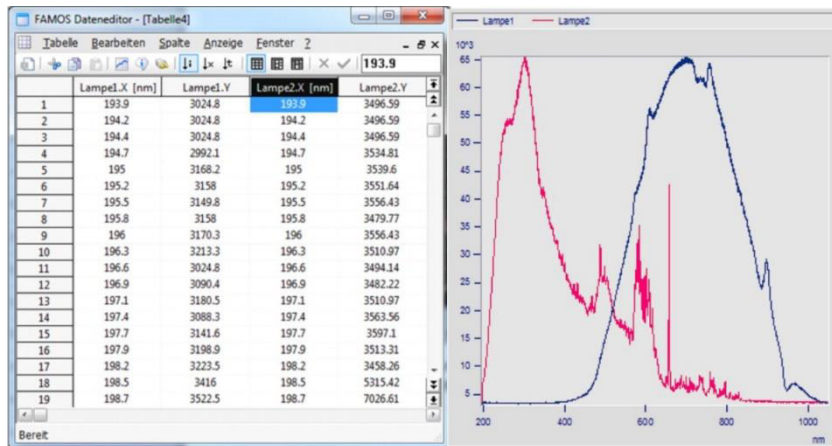


Abbildung 183: Messdaten und Messkurven der Spektren verschiedener Lampen

Die bisher eingesetzten Belichtungslampen haben vergleichbar breite Spektren. Das Maximum des Absorptionsspektrums des Fotolinkers liegt bei 310 nm. Daraus resultierte die entsprechende Lampenauswahl für den Fotoreaktor.

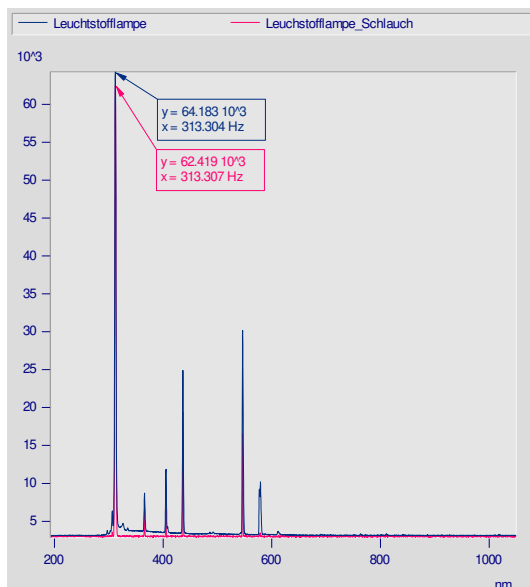


Abbildung 184: Spektrum der bisher verwendeten Leuchtstoff-Lampe direkt (blau) und nach Schlauchmaterial (rot)

Die gewählte Fluoreszenzröhre entwickelt ihre maximale Energieausbeute im Wirkungsbereich des Standard-Linkers bei gleichzeitig minimaler Wärmeabgabe. Damit entfällt die aufwendige Kühlung (Kompressoren, Gas/Trockeneis-Kühlung), was die steuerungstechnische Prozessführung erheblich erleichtert. Das eng gefasste Spektrenband macht entsprechende Schutzmaßnahmen wegen Ozonbildung überflüssig. Der PC übernimmt die Regelung der Belichtungszeit. Mit dem Einbau entsprechender Sensorik ist eine Überwachung des Spektrums und weiterer Online-Effekte möglich.

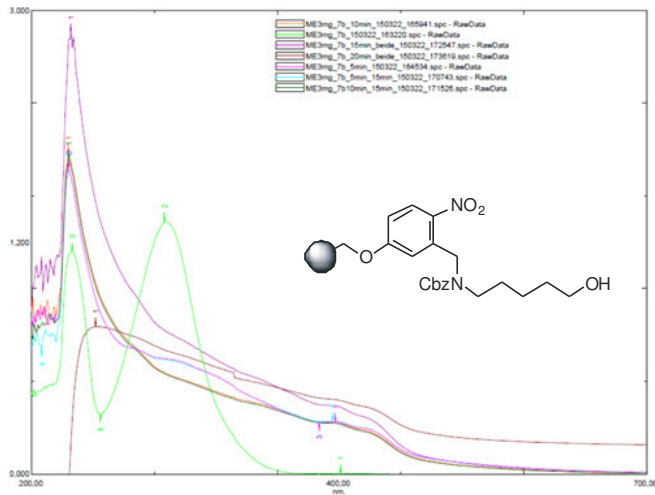


Abbildung 185: Spektrale Veränderung des fotospaltbaren Linkers bei unterschiedlich langen Belichtungszeiten: Referenz keine Belichtung (grün), 5 min (pink), 10 min (türkis), 15 min (lila)

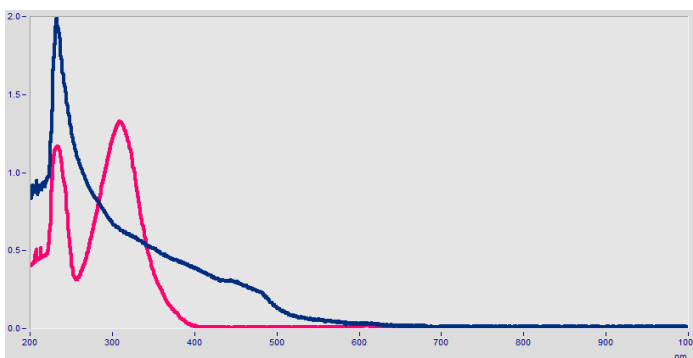


Abbildung 186: Die Absorptionsmessungen erfolgten mit UV / VIS-Spektralphotometer von Ocean Optics (USB4000) gesteuert von LabVIEW. Es zeigt die Absorption des Fotolinkers am Anfang (rot) nach 15 min Belichtung mit 311 nm Lampe (blau) Lösungsmittel: DCM

Beim bisherigen Fotoreaktor wird das Resin im oberen Bereich der Spritze gesammelt, um es als erstes durch den Teflonschlauch in die UV- Belichtungseinheit zu transportieren (Abbildung 187). Die Spritze wird nach manueller Befüllung auf der Transporteinheit fixiert. Die Parametrierung der Lampeneigenschaften, des Kühlprozesses und der Durchflussrate sind in voneinander unabhängigen Geräten vorzunehmen. Da diese Komponenten nicht untereinander kommunizieren, wird selbst bei deutlicher Überhitzung des Belichtungsreaktors ($> 40\text{ °C}$) bei Kühlproblemen weiterhin das thermisch empfindliche Substrat zugeführt.

Im Fotoreaktor - Prototyp I (Abbildung 188) ist die Fluoreszenzröhre als Lichtquelle im Zentrum des Reaktors. Die Bestrahlung wird durch umlaufende optische Reflexion optimiert und Kühlrippen dienen der passiven Wärmeabfuhr. Das Energiemanagement ist im Reaktor enthalten, um die optimalen Parameter für die jeweilig benutzte Lampe einzustellen. Die Belichtungs-Box und die Transportschläuche sind aus Langzeit UV-Licht beständigem Material.^[240]

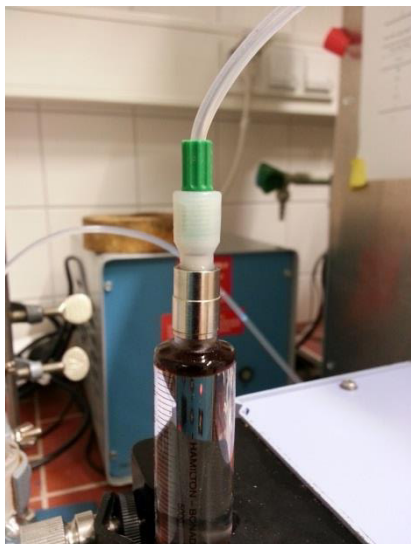


Abbildung 187: Vorbereitung zur Abspaltung des Linkers vom Resin. Verfahrensweise am alten Fotoreaktor bei Verwendung der Arbeitssynthesizer

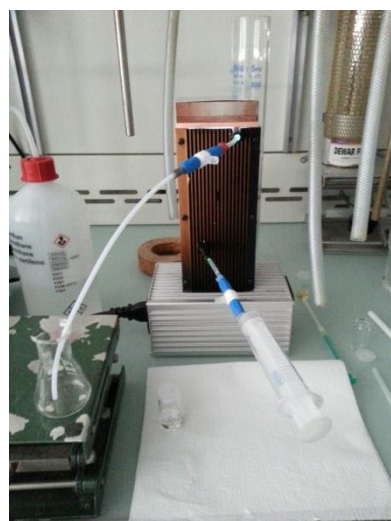


Abbildung 188: Modularer Fotoreaktor – Eigenbau – Typ I – für den integralen Einbau am Entwicklungssynthesizer zur automatisieren Linkerabspaltung

Fotoreaktor Prototyp I wurde für die Abspaltung des Foto-Linkers^[110] konzipiert, der zweite Prototyp für die fotoreaktive Schutzgruppenabspaltung. Das Modulbauteil Typ II (Abbildung 189) ermöglicht durch flexible Drucksteuerung auch die kurzfristige Zuführung von Reaktionskomponenten. Die Kühlkörper dienen der Temperaturkontrolle für die verschiedenen adaptierbaren unterschiedlichen frequenzselektiven LED's zur automatischen fotoselektiven Abspaltung.



Abbildung 189: Weiterführende Systemmodularisierung – Fotoreaktor Prototyp II 19“-Bauvariante.

Da alle UV-Lichtquellen potenziell für Haut und Augen gefährlich sind^[241], muss der Benutzer der Fotoreaktoren in die Sicherheitsregeln für UV-Operationen^[242,200] eingewiesen werden.

In der chemischen Reaktionsführung ist auf die Verwendung eines geeigneten, nicht mit den Photonen wechselwirkenden, Lösungsmittels zu achten. Idealerweise sollten sich auch die entstehenden fotochemischen Nebenprodukte reaktionsneutral verhalten und bei der verwendeten Bestrahlungswellenlänge transparent sein, um eine Absorption der Anregungsenergie zu vermeiden. Die Auswirkungen von begrenzter Reaktanzdurchdringung, strukturellem Fotoschaden, die Handhabung von Fokussierung, Wärmemanagement und mögliche Reaktionsmechanismen für den automatisierten Syntheseprozess werden derzeit von verschiedenen Gruppen untersucht.^[243]

Die beiden Fotoreaktor-Module können hier einen Beitrag zur Entwicklung lichtgesteuerter Synthese-

Anwendungen leisten und zum weiteren Verständnis fotochemischer Prozess-Mechanismen beitragen. Für die Kombinatorik lassen sich im Hinblick auf die Glykosylierungsreihenfolge unter Verwendung fotolabiler Schutzgruppen neue Variationen erreichen. Auch für die Abspaltungsreaktion kann mit frequenzselektiver Substituenten eine Vielzahl von Anwendungsmöglichkeiten erschlossen werden. Es ist eine der zukünftigen Herausforderungen, die Versuchsparameter fortlaufend zu optimieren und die Wirkung dieser Art von elektromagnetischer Stimulation, während der Entstehung neuer Moleküle zu untersuchen.

5.2.5 Integration von Online-NMR, NMR Simulation, Validierung & Konsolidierungsprozesse

Im Gesamtkonzept stellt das Expertensystem die finale Entwicklungsstufe des Molekülsynthesizers dar. Das Expertensystem ist ein cyber-physisches „Chemie-System“, das mit selbstoptimierenden Prozessen die Steuerung und Regelung chemischer Synthese-Prozesse übernimmt und damit eine chemische KI⁵¹ darstellt. Wesentliche Grundvoraussetzung für die Automatisierung chemischer Prozesse ist die Möglichkeit einer Echtzeit-Überwachung des Reaktionsverlaufs, eine begleitende Online Datenanalyse und darauf basierend die Möglichkeit der Rückkopplung in die Synthesesteuerung zur Optimierung des Syntheseverlaufs (Feedback-Optimierung).

Zur Echtzeit-Überwachung werden bislang hauptsächlich rein optische Online-Methoden am Synthesizer eingesetzt, die jedoch einen hohen Kalibrierungsaufwand (Farbabgleich) erfordern. NMR-Spektroskopie, eine unverzichtbare chemische Analysemethode, erwies sich bislang aufgrund der hohen Raumanforderungen, hohen Anschaffungs- und Wartungskosten und der betrieblichen Komplexität für die kontinuierliche Automation als ungeeignet. Dies hat sich während der Projektphase mit der Entwicklung kompakter kernmagnetischer Resonanz (NMR) – Instrumente, z.B. Spinsolve Benchtop NMR (MAGRIT) oder picoSpin NMR (Thermofischer), geändert. Diese Tisch-NMR-Spektrometer sind für automatische Anwendungen prädestiniert und ermöglichen fortgeschrittene chemische Prozessüberwachung sowie Reaktionskontrolle, auch in industriellen und rauen bzw. explosionsgefährdeten Umgebungen, wie die zahlreichen geförderten NMR-Studien der Horizon 2020-Projekte⁵² der Europäischen Union CONSENS zeigen.

Die Spektrometer enthalten Permanentmagnete, die zu Protonen-NMR-Frequenzen zwischen 40 und 80 MHz führen. Aufgrund der Entwicklung in Bezug auf Feldhomogenität und Empfindlichkeit, können hiermit kleine Moleküle, die bei Konzentrationen in der Größenordnung von 1 mmol L⁻¹ gelöst werden, in Single-Scan-Messungen mit 1 Hz Auflösung charakterisiert werden. Ihre begrenzte

⁵¹ KI: Abkürzung für „Künstliche Intelligenz“, als Teilgebiet der Informatik, befasst es sich mit der Automatisierung intelligenten Verhaltens und dem maschinellen Lernen.

⁵² Liste der geförderten NMR-Projekte: <https://www.fabiodisconzi.com/open-h2020/per-topic/nmr/list/index.html> [letzter Abruf 01.03.2019]

Auflösung, die zu Peaküberlappungen führt, kann den allgemeinen Anwendungsbereich jedoch einschränken. Derzeit werden verschiedene Ansätze diskutiert, um die Auswirkungen von chemischen Verschiebungen, Relaxation, Linienformunvollkommenheiten, Phasenlage und Grundlinienverzerrungen des NMR-Signals für die automatische Datenanalyse zugänglich zu machen.^[244] Die zunehmende Anzahl von Publikationen zur Einbindung der NMR-Spektroskopie - Technik, Bildgebung und Relaxometrie im Labor^[245], sowie Verfahrenskonzepte wie Eintopfreaktionen („One-Klick“-Verfahren)^[246], Durchfluss- und Mikroreakorteknologie (FlowChem)^[247] zeigt das große Potential dieser neuen Geräte für die Verfahrensautomation.

Für den Synthesizer ist zur Entwicklung eines regulativen Verfahrens ein Konzept der NMR-Simulation zur Vorhersage des erwarteten Produktspektrums bei gleichzeitiger NMR-Online-Überwachung des Syntheseproduktes vorgesehen. Für dieses Konzept konnten bereits einige technische Vorversuche für die Implementierung der Hardware-Steuerung eines Tisch-NMR's als zusätzliches Funktionsmodul für den Synthesizer gewonnen werden. Die neue Technik eignet sich hervorragend zur Integration in das aktuelle modulare Synthesizersystem.

Im ersten Entwicklungsschritt wurde ein LabVIEW-Beispielprogramm erstellt, das die wesentlichen Funktionen zum Einlesen von NMR-Dateien in LabVIEW und vergleichende Analysen von NMR-Simulation und NMR-Messdaten ermöglicht. Gemessene NMR-Daten können in diesem LabVIEW-Programm auch aus externen Quellen importiert werden (Abbildung 190).

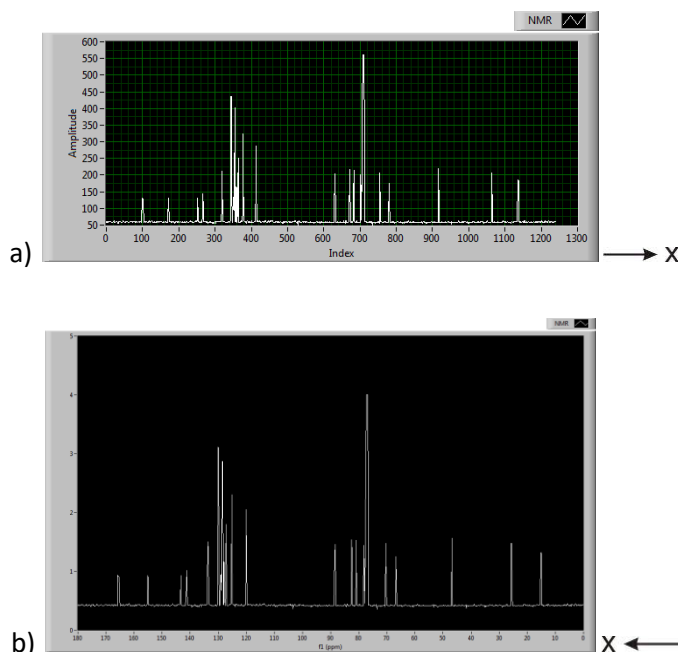


Abbildung 190:

a) Einlesen von NMR-Messdaten in LabVIEW konformer (x-Achse linear steigend) Darstellungsweise in das Beispielprogramm zur Konzeptstudie für die Weiterentwicklung des Synthesizers

b) Übliche Darstellungsweise von NMR's mit Skalierung des chemischen Shifts in negative Richtung steigend

Die eigentliche Herausforderung steckt in der Entwicklung der automatischen Online-Analyse, der anschließenden Rückkopplung der Daten in den Syntheseverlauf zur selbstoptimierenden Steuerung und dem zukünftigen Vorschlagswesen für Synthesepfade auf der Basis bereits erfolgreich durchgeführter Synthesen. Auf diesem Entwicklungsweg stellt der qualitative Abgleich zwischen

Simulationsprozess und Offline- oder Online gewonnenen NMR-Realdaten einen wichtigen Schritt zur Steuerung und Regelung chemischer Reaktionen im Synthesizer dar. Dazu wurden Daten mit ChemNMR, der NMR-Vorhersagefunktion von ChemDraw, die Spektren von verschiedenen Zuckerbausteinen, sowohl als Protonen-, als auch Carbonspektren, simuliert und innerhalb des erstellten LabVIEW-Beispielprogramms den real gemessenen NMR-Daten gegenübergestellt, um zunächst eine rein visuelle Ähnlichkeitsbetrachtung durchzuführen (Abbildung 191 u. ff.).

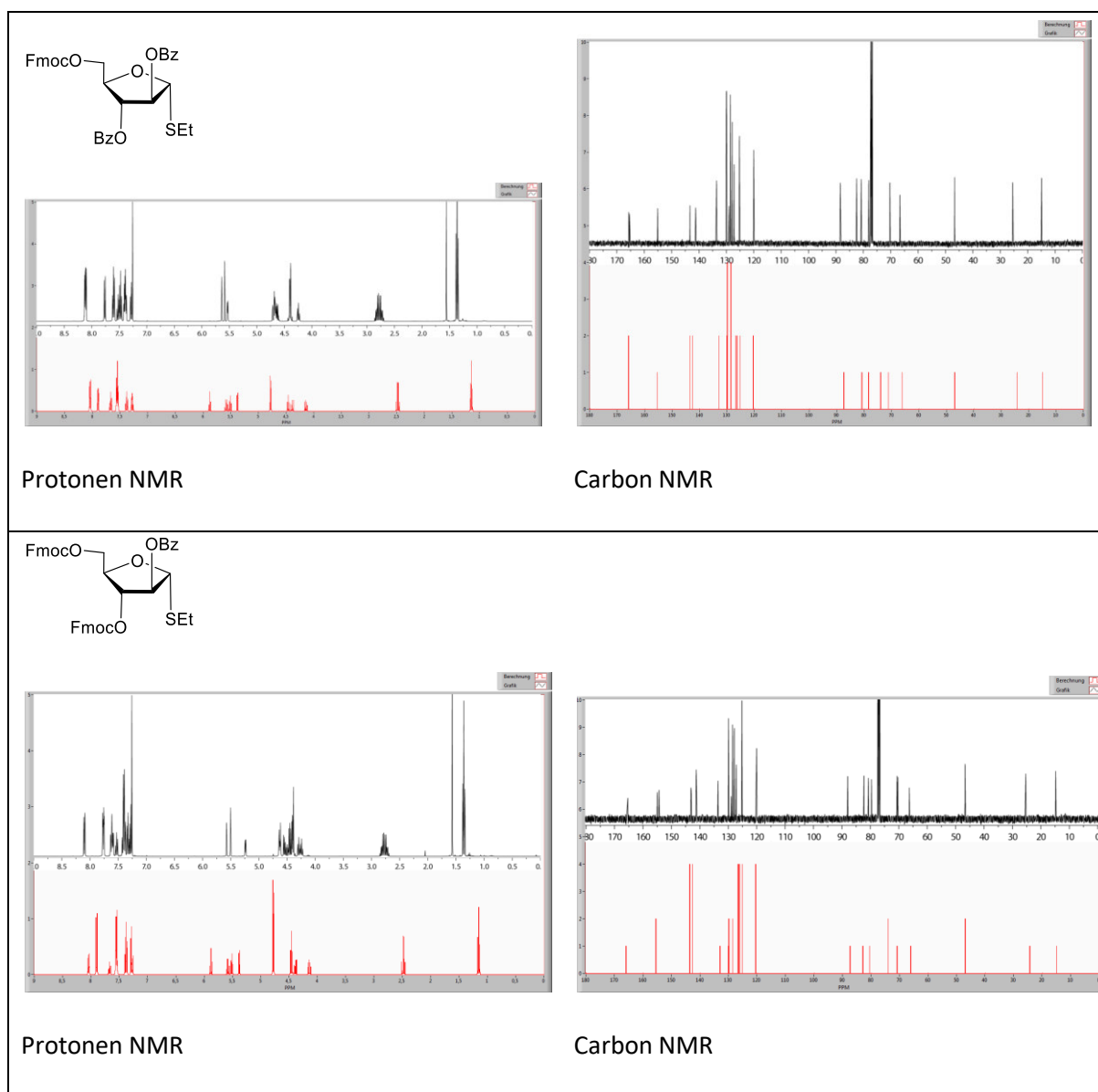


Abbildung 191: Vergleich gemessener NMR-Daten (jeweils obere Kurve) mit simulierten NMR-Daten des abgebildeten Zuckerbausteins (jeweils untere Kurve)

SPINUS⁵³ ermöglicht Vorhersagen zu NMR Spektren unter Verwendung neuronaler Netzwerke. Aufgrund der Eingabemöglichkeiten über eine Java Editor können diese Funktionalitäten in die Anwendungssoftware des automatisierten Systems implementiert werden. Zum vereinfachten

⁵³ Abkürzung für **Structure based Predictions In Nuclear magnetic resonance Spectroscopy**;
Online unter: <http://www2.chemie.uni-erlangen.de/services/spinus/index.html> [letzter Abruf: 01.03.2019]

Zugriff und als Austauschplattform kann die als reine HTML5 Applikation realisierte NMR Predict⁵⁴[248] mit in die Oberfläche eingebunden werden.

Originaldaten

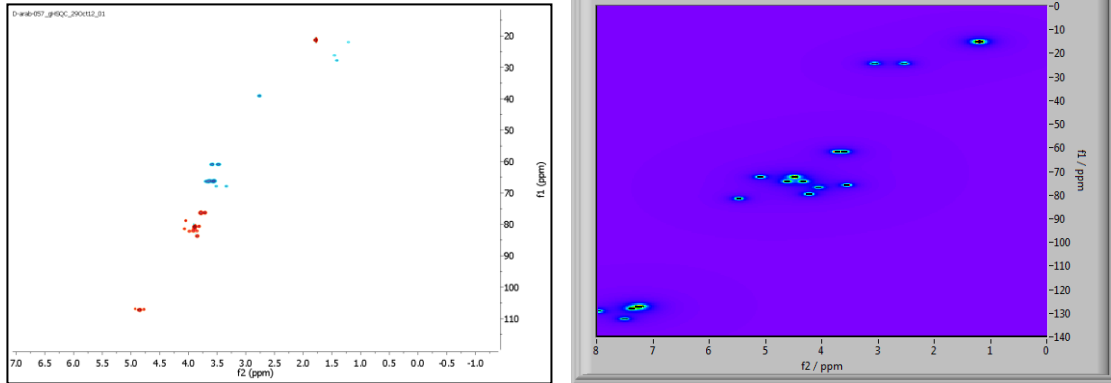


Abbildung 192: Aufbau eines HSQC – Spektrums in LabVIEW aus den Daten von Jey Kandasamy's Arabinofuranosid Synthese. Überlagerung als 2D NMR im LabVIEW Programm als Visualisierungsmöglichkeit der eingelesenen Originaldaten (re.).

Zur automatisierten Charakterisierung von Oligosacchariden stellt GlycoNMRSearch^[249] die grundlegenden Such-Funktionen auf Basis vorhandener NMR's bereit. Über die Eingabe der chemischen Verschiebungen wird durch den Suchalgorithmus innerhalb des Datenbestandes von Glycoscience-Datenbank⁵⁵ ein Abgleich vorgenommen und die erhaltenen Information mit einem Scoring visualisiert. Durch die Bereitstellung solcher Funktionalitäten direkt aus der Applikationsoberfläche des Automatisierungssystems hat der Anwender sofort eine informative Rückkopplung, ob die Synthese in die geplante Richtung läuft. Da hierüber auch eine Bewertung der erhaltenen Informationen abgefragt werden kann, ist die weiterführende Implementierung ein automatisierter Prozess, der den zukünftigen Anwender weitergehend entlastet.

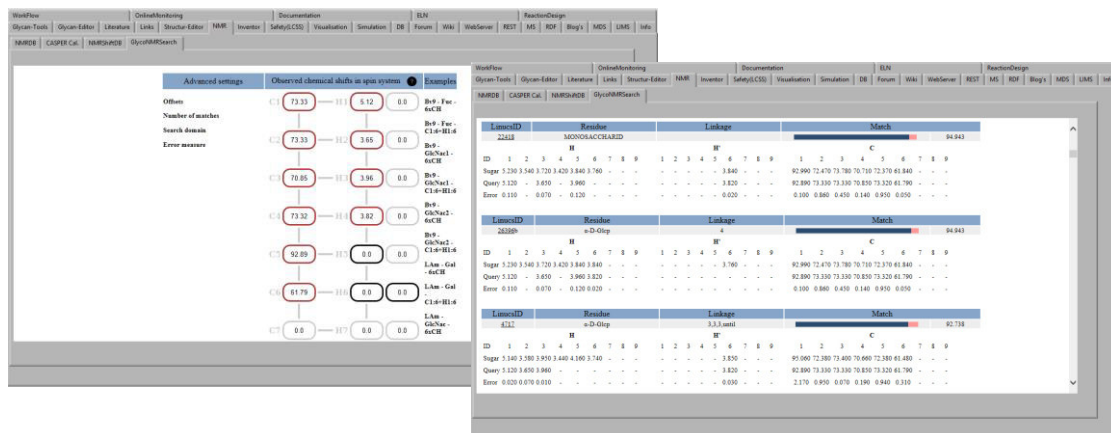


Abbildung 193: Einbindung GlycoNMRSearch in das LabVIEW-Beispielprogramm.

⁵⁴ Online unter: <http://www.nmrdb.org> [letzter Zugriff: 01.03.2019]

⁵⁵ Online unter: <http://www.glycosciences.de/database/> [letzter Abruf: 09.01.2019]

Zur Glykan-Charakterisierung kann auch GlycReSoft⁵⁶[250], eine Anwendung zur Strukturerkennung von Glykanen aus LC / MS-Daten oder GRASS⁵⁷[251], eine Anwendung der Carbohydrate Structure Database (CSDb)^[252] zur NMR-basierten Strukturaufklärung von Sacchariden, in die modulare Synthesizer-Plattform integriert werden.

Durch die auf diesem Weg sukzessiv entstehende Datenbank lassen sich die Lernverfahren kontinuierlich anpassen und optimieren. Die während der Synthese gewonnenen Informationen können automatisiert in die Datenbank von NMRShiftDB⁵⁸, [253,254] eingepflegt werden. Diese offene Datenbank ermöglicht die interoperable Nutzung der gewonnenen Daten, wie z. B. bei der computerunterstützten Strukturaufklärung biologischer Metaboliten^[255] oder der systematischen Untersuchung von NMR-Strukturen von Glykan-Determinanten^[256] und erleichtert die Qualitätskontrolle bei der Zuordnung von Spektren zu organischen Molekülen.^[254] Als weitere offene Ressource zur Verwaltung und Archivierung von NMR-Spektren steht seit 2015 die OSDB⁵⁹, [257] Datenbank zur Verfügung.

Mit dem plattformübergreifenden Spektrum-Visualizer jsNMR^[258] können NMR Spektren sogar auf mobilen Geräten angezeigt werden, welches z. B. bei der Fernüberwachung bzw. Kontrolle automatischer Synthesen von Interesse ist.

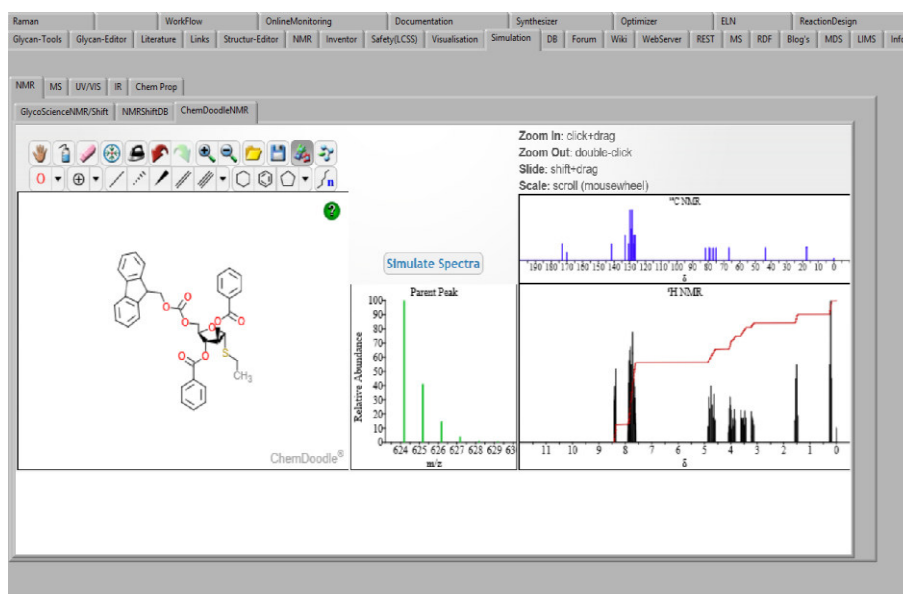


Abbildung 194:
Implementierung von
ChemDoodleNMR in
die LabVIEW-
Programmumgebung

Das Web-Tool CASPER⁶⁰, [259] erleichtert die Vorhersage der chemischen ¹H- und ¹³C-NMR-Verschiebungen von Oligo- oder Polysaccharidstrukturen und wird für die strukturelle Bestimmung unbekannter Glykane unter Verwendung fragmentierter NMR-Spektren eingesetzt. Die ebenfalls für

⁵⁶ Online unter: <http://www.bumc.bu.edu/msr/glycresoft/> [letzter Abruf: 01.03.2019]

⁵⁷ Online unter: http://csdb.glycoscience.ru/biopse/grass_interface.php [letzter Abruf: 01.03.2019]

⁵⁸ Online unter: <https://nmrshiftdb.nmr.uni-koeln.de/> [letzter Abruf: 01.03.2019]

⁵⁹ Online unter: <http://osdb.info/> [letzter Abruf: 01.03.2019]

⁶⁰ Online unter: <http://www.casper.organ.su.se/casper/calculate.php> [letzter Abruf: 01.03.2019]

Glykan-Spektren optimierte Modellbildung GODESS^[260a,261,260b] erreicht sowohl für ^1H ^[262] als auch ^{13}C ^[263] aussagekräftige Genauigkeit. Dies ermöglicht sogar die Vorhersagen von 2D-Spektren (^1H - ^1H TOCSY, COSY/COSY-DQF/COSY-RCT, HSQC-COSY, HSQC-TOCSY sowie HMBC).

Auch bei den Desktopanwendungen (ChemDraw^[264], MNOVA^[265], ChemDoodle⁶¹) werden die NMR Vorhersagemöglichkeiten durch den programmatischen Zugriff für den weiterentwickelten Molekülsynthesizer nutzbar. Die Abhängigkeit der simulierten von den gemessenen Spektren kann mittels gewichteter mathematischer Kreuz-Korrelationsanalyse ermittelt werden. Aufgrund des automatisierten Daten-Abgleichs mit vorhandenen Open-Source Ressourcen, ist durch Anwendung selbstlernender Verfahren eine immer besser werdende Vorhersage auch bei komplexeren Szenarien möglich (vgl. Kapitel Expertensystem).

5.2.6 Makerboards, Internet der Dinge und Internet 4.0

Mit der modularen, offenen Architektur des in dieser Arbeit realisierten, automatischen Oligosaccharidsynthesizers, wurde seit 2012 die Basis für eine Syntheseautomation gelegt, welches eine nahtlose Weiterentwicklung in die gegenwärtige technologische Zukunft erlaubt. Cloud-basierte Systeme, sowohl im Intranet betrieben als auch webbasiert, werden mittelfristig reine Stand-Alone-Lösungen ersetzen. Zudem können sie es dem Forscher ermöglichen, chemische Reaktionen über einen Webbrowser mit jedem beliebigen Gerät von jedem Ort der Welt aus zu überwachen und zu steuern. Schon heute erlebt die Kommunikation in der Chemie eine Revolution aufgrund webbasierter Technologien, die dazu führen, dass wissenschaftliche Meinungen, Daten und Wissen in immer stärkerem Maße Verbreitung finden.^[266] Die Einführung moderner, vernetzter Technologien und Automationsverfahren in der Laborumgebung steht dagegen noch am Anfang. Doch chemische Synthesemethoden könnten von der technologischen Unterstützung durch komplexe Überwachungs- und Kontrollmethoden und der Anwendung maschinengestützter Verfahren erheblich profitieren.^[267]

Parallel zu der vorliegenden Arbeit, entwickelten sich die ersten Versionen an Makerboards, wie Raspberry Pi⁶² oder die Mikrokontroller: Arduino⁶³ und NodeMCU⁶⁴. Ursprünglich für Schulexperimente und kleinere technische Projekte gedacht, entstand in kürzester Zeit eine riesige Open-Source Anwendergemeinde, welche zur rasanten Weiterentwicklung der Mini-Computer beitrug. Noch bis 2016 wurden Makerboards lediglich für „Proof-of-Concepts“-Projekte empfohlen, da ihre Leistungsfähigkeit im Produktiveinsatz je nach Typ und Anwendung begrenzt war. Seit 2018 stehen jedoch auch Makerboards zur Verfügung, die sich für den professionellen industriekonformen Serieneinsatz eignen.

⁶¹ Online Demo: <https://web.chemdoodle.com/demos/simulate-nmr-and-ms/> [letzter Abruf: 01.03.2019]

⁶² Online unter: <https://www.raspberrypi.org/> [letzter Abruf: 25.01.2019]

⁶³ Online unter: <https://www.arduino.cc/> [letzter Abruf: 25.01.2019]

⁶⁴ Online unter: https://www.nodemcu.com/index_en.html [letzter Abruf: 25.01.2019]

Die zunehmende Verfügbarkeit vielfältiger Sensoren und hilfreicher Zusatzmodule, macht den Einsatz dieser Technologie für die Übernahme kleiner Funktionalitäten am Synthesizer und in verteilten Systemen (Mesh-Netzwerke⁶⁵, IoT-Systeme^{66[268]}) hoch interessant. Architekturen mit Arduino-Boards und drahtloser Sensor-Netzwerktechnologie sind als Bausteine zur Implementierung autonomer Fernmess- und Steuerungssysteme eine praktikable, kostengünstige Lösung vieler innovativer Anwendungen mit unterschiedlichen Komplexitätsniveaus.^[269]

Normale Laborgeräte lassen sich mit den Makerboards schnell und effizient um benötigte Zusatzfunktionen, sogenannte Microservices, erweitern. Der Synthesizer kann damit kostengünstig vielfältige Aufgaben der Online-Prozessüberwachung übernehmen, angefangen von einfachen lokalen Temperaturmessungen bis hin zu Interaktionen mit der systeminternen Zustandsmaschine. In vielen Anwendungsfällen werden Geräte miteinander kombiniert, die aufgrund der historischen Entwicklung keine gemeinsam EDV-technisch verwendbare Schnittstelle besitzen und damit in einem IoT-System einander zugänglich gemacht werden. Makerboards finden hier als Schnittstellenadapter Verwendung und vermitteln zwischen den unterschiedlichen Hard- bzw. Software-Philosophien. Da oftmals die für die Analyse verwendeten Geräte nicht im gleichen Netzwerk, geschweige denn in den gleichen Räumlichkeiten zusammengefasst sind, können diese Grenzen durch die Möglichkeiten der Makerboards in cloud-basierten Anwendungen aufgehoben werden. So lassen sich als Beispiel Zugriffe des Synthesizers auf die verschiedenen autonomen Laboranalyseeinheiten wie z. B. NMR oder MS-Geräte realisieren, die weder lokal noch direkt mit dem Synthesizer verbunden sein müssen, und die Analysedaten automatisch prozessbezogen in die Wissensdatenbank nach FAIR-Empfehlung über das ELN/LIMS – System überführen (vgl. Kapitel: Informationsverarbeitung und Wissensmanagement mit ELN und LIMS). Diese standardisierte Herangehensweise an Prozessschritte stellt die zuverlässige Datenbasis bereit, aus dem nachgelagerte Analyseprozesse für zukünftige Synthesevorschläge vom Expertensystem abgeleitet werden können.

Zur Qualitätssicherung bei der chemischen Prozessführung wird die Erfassung von Umgebungsparameter, wie Luftfeuchte oder Temperatur, sowie die Online-Überwachung der Synthese, z. B. unter Verwendung einer Webcam, immer größere Bedeutung gewinnen. Bei automatisierten Verfahren müssen gleiche Prozessparameter immer zu gleichen Ergebnissen in gleichbleibender Qualität führen. Physikalische Parameter, die bislang nicht oder nur unzureichend erfasst wurden, welche aber relevant für Reproduzierbarkeit der Synthese sein könnten, sind automatisch während des gesamten Prozessverlaufs zu überwachen und in der Synthesizer -

⁶⁵ Vermaschtes drahtloses lokales Netzwerk, bei dem jeder Netzwerkknoten mit einem oder mehreren anderen verbunden ist. Die Informationen werden von Knoten zu Knoten weitergereicht, bis sie das Ziel erreichen.

⁶⁶ Abkürzung für Internet of Things (Internet der Dinge): System miteinander vernetzter physischer und virtueller Gegenstände.

Prozessdatenbank zu dokumentieren. Eine Vielzahl der benötigten lokalen Informationen sind bereits in den rudimentären Modelvarianten der Makerboards implementiert. Zur Messung der Luftfeuchte, die relevant für die Qualität und Reaktivität vieler eingesetzter Lösungsmittel oder chemischer Produkte ist, stellt der Sensor DHT22⁶⁷ mit einer Genauigkeit von 2% im Bereich von 0-100% Luftfeuchte eine kostengünstige Ergänzung von nahezu jedem Laborgerät dar. Da sich innerhalb der verteilten Geräte unterschiedliche Temperaturbereiche erfassen lassen müssen, kann durch die Verwendung des Temperatursensors DS18B20⁶⁸ mit wenig Kabelaufwand ein durch Eindraht - Technik verteiltes Messnetz aufgebaut werden und somit ein lückenloses Temperaturprofil thermisch reaktiver Synthesen aufgezeichnet werden.

Der Raspberry Pi kann als Cloud-Server verwendet werden, das als Speichergerät für Echtzeitanwendungen dient.^[270] Echtzeit-Überwachungssysteme können vorhandene Sicherheitssysteme im Synthesizer und im Laborumfeld ergänzen, z. B. als ein zusätzliches Labor-Brandüberwachungssystem^[271] zur Vermeidung von Fehlalarmen. Im zukünftigen Synthesizer ist vor allem eine definierte Notfallabschaltung essenziell, welches die Synthese selbstständig in einen chemisch und physikalisch „sicheren“ Modus parkt. Lokale SPS⁶⁹ - Aufgaben können mit eigener unterbrechungsfreier Stromversorgung (USV) realisiert werden. Das Arduino Uno wird als speicherprogrammierbare Steuerung verwendet, welches auf die Funktionalitäten des Raspberry PI aufsockelt.

Unter Verwendung von MQ-X Sensoren⁷⁰ können direkt verschiedene Gase detektiert werden. Die Kontrolle ob Gerätschaften (Laborgläser, Reagenzgläser) an der geforderten Position sind, lässt sich mit den Abstandssensoren HC-SR04⁷¹ oder GP2Y0A02YK⁷² erfassen. Die Kodierung welcher Halter sich wie ausgerichtet im System befinden, lassen sich über Reed-Relais erreichen.

Füllstandskontrollen, z.B. zur Detektion des Füllstandes von Reaktanten und Reagenzien oder des Abfallbehälters am Synthesizer, sind oftmals auf reine Gewichtskontrollen zurückzuführen und damit durch Wiegedrucksensoren wie HX711⁷³ abfragbar.

Zur Erhöhung des Bedienkomforts am Synthesizer kann ein LCD - Display zum Einsatz kommen, das als kostengünstige zusätzliche Anzeigemöglichkeit lokaler Information dient. Eine lokale Bedienmöglichkeit ergibt sich unter Verwendung der Möglichkeit der Bluetooth Kommunikation, da

⁶⁷ Online Datenblatt: <https://www.sparkfun.com/datasheets/Sensors/Temperature/DHT22.pdf> [letzter Abruf: 24.01.2019]

⁶⁸ Online Datenblatt: <https://datasheets.maximintegrated.com/en/ds/DS18B20.pdf> [letzter Abruf: 24.01.2019]

⁶⁹ Abkürzung für speicherprogrammierbare Steuerung.

⁷⁰ Online Beschreibung unter: <https://playground.arduino.cc/Main/MQGasSensors> [letzter Abruf: 24.01.2019]

⁷¹ Online Datenblatt: https://www.mpja.com/download/hc-sr04_ultrasonic_module_user_guidejohn.pdf [letzter Abruf: 24.01.2019]

⁷² Online Datenblatt: https://www.sparkfun.com/datasheets/Sensors/Infrared/gp2y0a02yk_e.pdf [letzter Abruf: 24.01.2019]

⁷³ Online Datenblatt: https://cdn.sparkfun.com/datasheets/Sensors/ForceFlex/hx711_english.pdf [letzter Abruf: 24.01.2019]

diese in nahezu jedem modernen Handy enthalten ist. Auch umfangreiche Zugriffsmechanismen lassen sich damit administrieren. Alle modernen Komponenten sind sehr einfach in die Steuerung des Synthesizers implementierbar. Auch Touchscreen Panels können direkt in die modulare Erweiterungstechnik integriert werden. Selbst Geräte, bei denen keinerlei zusätzliche Möglichkeit der elektronischen Bedienung vorgesehen ist, sind durch Verwendung von Funksteckdosen in das Messnetz einbindbar.

Die visuellen Überwachungsaufgaben der Versuche kann durch die Verwendung von Kameras mit Livestream auf jeden PC- weltweit gemappt werden. Diese einfachste Methode der Prozessüberwachung mittels Webcam ist Bestandteil zahlreicher Veröffentlichungen, die Echtzeitüberwachung mit Analyse-Technologien basierend auf Open-Source-Anwendungen mit dem Ziel weitergehender Automatisierung in den Forschungslabors thematisiert.^[267,272]

Im zukünftigen Synthesizer werden durch die Implementierung der OpenCV⁷⁴- Module die Grundlagen für ein erweitertes „maschinelles Lernen“⁷⁵ im Expertensystem gelegt. Eine Vielzahl manueller Kontrollen lassen sich damit programmatisch abbilden, sowie Kontrollaufgaben laborübergreifend zusammenfassen und durch unterschiedliche Warnstufen sinnvoll administrieren.

Die weitergehende Automation der Synthese erfordert zunehmend auch die Einbeziehung notwendiger vorbereitender, nachgelagerter und zwischengeschalteter Prozesse und Analysen. Bei der Anforderung mechanischer Aktionen kommen dabei sowohl Servo- wie auch Schrittmotoren zum Einsatz. Durch die Kaskadierung von Treiberboards (PCA9685⁷⁶) lassen sich auch komplexere Positionierungsaufgaben abbilden. Im Extremfall kann auch der Roboterarm (6DOF⁷⁷) Verwendung finden.

Über das RFID⁷⁸-Modul RC522⁷⁹ können vom System benötigte Daten per NFC⁸⁰ bereitgestellt werden, z. B. für die Synthese am Synthesizer verwendete Reagenzien, Reaktanten und Lösungsmittel. Im Gegensatz zu einem QR⁸¹-Code, der nur statische Informationen an das System übermitteln kann, lassen sich mit NFC Informationen dynamisch verändern. So kann nicht nur der Name der eingekoppelten Substanz ausgelesen werden, sondern auch weitere Informationen, wie

⁷⁴ Freie Programmibliothek mit Algorithmen für die Bildverarbeitung und maschinelles Sehen.

Online unter: <https://opencv.org/> [letzter Abruf: 25.01.2019]

⁷⁵ Teilbereich der Künstlichen Intelligenz, beschreibt dynamische Algorithmen, die in der Lage sind Algorithmen Muster und Gesetzmäßigkeiten zu erkennen und Lösungen zu entwickeln.

⁷⁶ Online unter: <https://www.nxp.com/products/analog/interfaces/ic-bus/ic-led-controllers/16-channel-12-bit-pwm-fm-plus-ic-bus-led-controller:PCA9685> [letzter Abruf: 25.01.2019]

⁷⁷ Abkürzung in englisch für: "six degrees of freedom", bezeichnet die sechs Freiheitsgrade eines starren Körpers im dreidimensionalen Raum.

⁷⁸ RFID ist die Abkürzung von radio-frequency identification, welches ein Sender-Empfänger-System zum automatischen und berührungslosen Identifizieren und Lokalisieren von Objekten und Lebewesen mit Radiowellen bezeichnet.

⁷⁹ Online Datenblatt: <http://anleitung.joy-it.net/?goods=rfid-modul-rc522> [letzter Abruf: 25.01.2019]

⁸⁰ Abkürzung von Near Field Communication, bezeichnet einen auf RFID-Technik basierenden internationalen Übertragungsstandard zum kontaktlosen Austausch von Daten.

⁸¹ Abkürzung für **Q**uick **R**esponse, zweidimensionaler grafischer Code mit automatischer Fehlerkorrektur.

der Füllstand bzw. die aktuelle Inhaltsmenge des Behälters. Statusinformationen, wie eine kritische Bestandsmenge, können so über Messenger-Funktionen übermittelt werden, ohne dass der eigentliche Bediener Vorort sein muss.

Auch Parametrierdatenbanken können per NFC hinterlegt werden. Das „intelligente“ Probegefäß kann so z. B. Informationen enthalten, mit welchem Temperaturprofil die Substanz behandelt werden soll. Diese Information übermittelt es berührungslos an das entsprechende Temperierungsgerät. Damit ist das eigenständige Abarbeiten von Temperaturprofilen auf einer Vielzahl verschiedener Heizmodule möglich, die zwar eine RS232-Schnittstelle besitzen, jedoch nicht mit einem zusätzlichen PC ausgestattet sind und somit einer direkten, flexiblen Programmierung entzogen waren. In einem Konzeptvorschlag für ein webbasiertes Labor-Anwesenheitssystem wird damit sogar der Mensch durch Integration von RFID-ARDUINO-Technologie zum vernetzten IoT-„Objekt“.^[273]

Zur Ansteuerung und zum Testen dieser verteilten Systeme steht mit OpenHAB⁸² bereits eine umfangreiche Software zur hersteller- und protokollneutralen Gebäudeautomatisierung zur Verfügung. Gerade bei Funknetzen ist die übersichtliche Parametrieroberfläche zur schnellen Einarbeitung von Vorteil. Die Reichweiten können hierbei selbst im Labor mehrere 100 Meter betragen und reichen somit in vielen Fällen zur Vernetzung der Arbeitsmittel einer Arbeitsgruppe aus. Durch die sich aus diesem Konzept ergebenden Varianten, ist die ereignisorientierte Benachrichtigung mit modernen Kommunikationswegen GSM⁸³ (SMS, email, WhatsApp...) direkt möglich.

Seit 2013 wurde von Wolfram Research Mathematica für die Raspberry Community gebündelt und steht somit über NOOBS⁸⁴ zur Verfügung. Damit ist ein vollwertiger Zugriff auf mathematisch komplexe Analysen, Vorhersagen bis hin zu selbstlernenden Verfahren auf aktuellen wissenschaftlichen Level gewährleistet. Mit den aktuellen Befehlserweiterungen ist Mathematica direkt in der Lage die Hardwareeigenschaften des Raspberrys zu adressieren. Dynamische Regelungsprozesse werden somit beschleunigt. Dies ermöglicht die direkte Interaktion von chemischer Reaktionskinetik, mathematischer Modellbildung bis hin zu Echtzeit-Hardwareansteuerung.

Die Industrietauglichkeit kann über die Verwendung der Baureihe Revolution PI von Kunbus⁸⁵,

⁸² Abkürzung von **Open Home Automation Bus**. Online unter: <https://www.openhab.org/> [letzter Abruf: 25.01.2019]

⁸³ Abkürzung für **Global System for Mobile Communications**.

⁸⁴ Abkürzung für **New Out Of Box Software**; online verfügbar unter: <https://www.raspberrypi.org/downloads/noobs/> [letzter Zugriff: 25.01.2019]

⁸⁵ Online unter: <https://revolution.kunbus.de/> [letzter Zugriff: 25.01.2019]

netPivon Hilscher⁸⁶, emPC-A/RI von Janz Tec⁸⁷, PiXtend von Qube Solutions⁸⁸, Rpi 07 von Mass⁸⁹ erreicht werden. Die Standardisierung des Aktionsanforderung kann mit MQTT⁹⁰ synchronisiert werden. Auf dem MQTT Protokoll basiert auch das Konzeptbeispiel einer webbasierten IoT-Lösung, zur Überwachung, Verfolgung und Analyse von Daten im landwirtschaftlichen Bereich.^[274] Für die Verwaltung der Teilnehmer eignet sich u. a. Node-Red⁹¹ und Mosquitto⁹².

Die beachtlichen Fortschritte der Open-Source-Technologien und deren Möglichkeiten, finden derzeit zunehmend in den chemischen Forschungslaboren Beachtung und Anwendung, wo sie bei weitem nicht mehr nur zu Schulungszwecken eingesetzt werden wie z. B. bei der Konstruktion und Optimierung eines mit einem Raspberry Pi kontrollierten Absorptionsspektrometers,^[275] sondern auch für fortgeschrittene Automatisierungs- und Analysetätigkeiten. Kürzlich wurde eine cloudbasierte Reaktorplattform vorgestellt, die es ermöglicht, Batch- und Fließreaktionen entweder einzeln oder gemeinsam durchführen zu können.^[276] Auch die Reaktionsautomatisierung zur Maximierung des Ausstoßes eines festen Katalysatorvolumens^[277] basiert auf einem cloudbasierten modularen Softwaresystem unter Anwendung von Open-Source-Technologien. Weitere Entwicklungen zielen auf die Bereitstellung von Hard- und Software zur Ermöglichung einer autonomen Mehrstufensteuerung.^[267] Eine Reihe von Veröffentlichungen beschäftigen sich mit der Durchflusschemie. So kommt ein Raspberry Pi-Computer bei der Steuerung mehrerer Durchflusschemiegeräte für eine automatisierte Präparation von Piperazin-2-carboxamid zum Einsatz.^[278] Webcams in Verbindung mit einem „Computer-Vision“⁹³-Ansatz, finden in einem kontinuierlichen Flüssig-Flüssig Extraktionssystem zur effizienten Inline-Extraktion überschüssiger Reagenzien Anwendung.^[279] Bei der fließchemischen Entschützung von Silylethern mit einem selbstgebauten Autosampling-System kamen Open-Source-Softwaretechnologien für die automatisierte Flüssig-Flüssig-Extraktion und ein Raspberry Pi-Einplatinencomputer zur Steuerung der Motoren eines kostengünstigen Autosamplers und der Fluidventile zum Einsatz.^[280] Bei der Synthese von 2-Amino-adamantan-2-carbonsäure wurde ein intelligentes, selbstregulierendes,

⁸⁶ Online unter: <https://www.netiot.com/netpi/industrial-raspberry-pi-3/> [letzter Zugriff: 25.01.2019]

⁸⁷ Online unter: https://www.janztec.com/fileadmin/user_upload/Produkte/embedded/emPC-AiMX6/manual_empc-aimx6_hardware.pdf [letzter Zugriff: 25.01.2019]

⁸⁸ Online unter: <https://www.pixtend.de/> [letzter Zugriff: 25.01.2019]

⁸⁹ Online unter: <https://www.mass.de/de/produkte/industrial-all-in-one-pcs/produkt-details/rpi-07.html> [letzter Zugriff: 25.01.2019]

⁹⁰ Abkürzung für Message Queuing Telemetry Transport, bezeichnet ein offenes Nachrichtenprotokoll für die Gerätekommunikation in IoT (Internet der Dinge) – Anwendungen.

⁹¹ Grafisches Entwicklungswerkzeug von IBM zur Erstellung von IoT-Prototypen. Online unter: <https://nodered.org/> [letzter Abruf: 25.01.2019]

⁹² Open Source Software für die Machine-to-Machine-Kommunikation. Online unter: <https://mosquitto.org/> [letzter Zugriff: 25.01.2019]

⁹³ Computergestützte Lösung von Aufgabenstellungen, die sich an den Fähigkeiten des menschlichen visuellen Systems orientieren, wird auch als maschinelles Sehen oder Bildverstehen bezeichnet.

integriertes System für fortschrittliche mehrstufige Flussesequenzen unter Verwendung von Webcams und modulare Komponenten in Verbindung mit modernen Steuerungstechnologien eingesetzt.^[281]

5.3 Weiterentwicklung der Synthesizer-Software

5.3.1 Übersicht der Bedienebenen der Benutzerschnittstellen

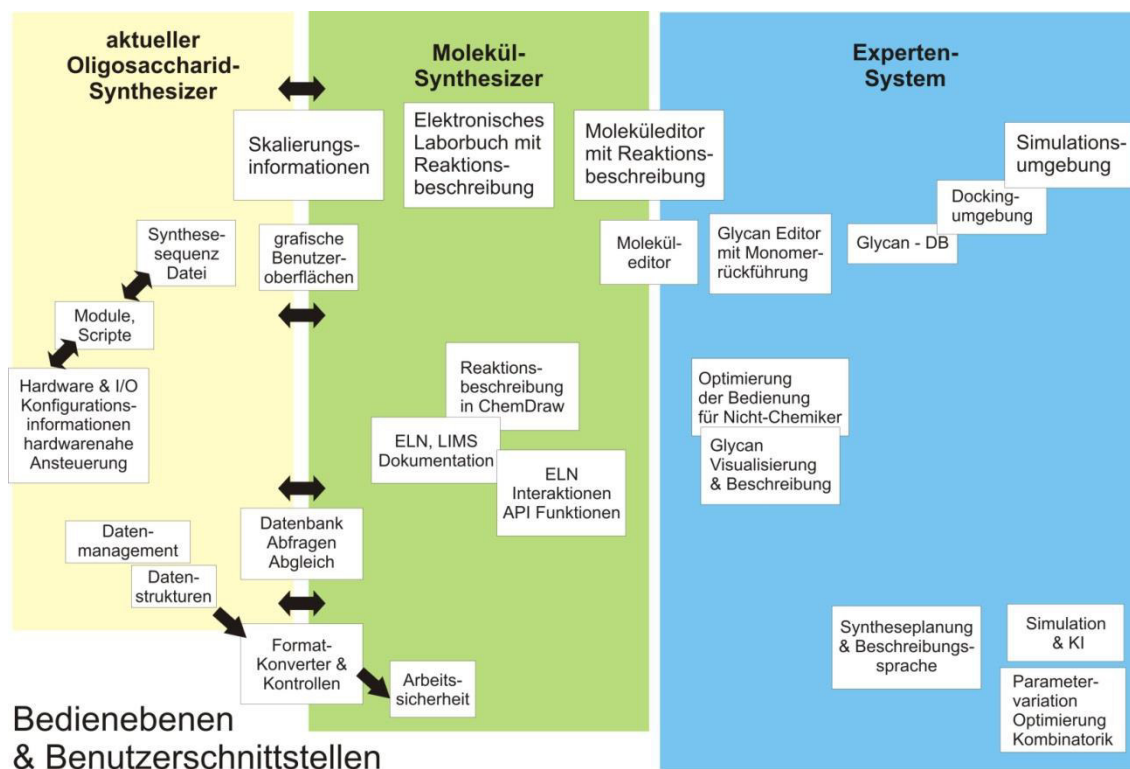


Abbildung 195: Konzept der notwendigen Entwicklungsstufen für die Bedienebenen und Benutzerschnittstellen von Molekülsynthesizer und Expertensystem, ausgehend vom aktuellen technologischen Entwicklungsstand des Oligosaccharid-Synthesizers.

Die Benutzeroberfläche soll die intuitive Bedienung des Syntheseautomaten trotz vielfältiger und sehr unterschiedlicher Anwenderbedürfnisse ermöglichen. Daher wird sich die Bedienung des Synthesizers zukünftig von einer rein gerätetechnischen Programm-Parametrierung im aktuellen Oligosaccharid-Synthesizer hin zu einer anwenderbasierten Synthesesteuerung im Molekülsynthesizer und weiterführend zu einem umfassenden Expertensystem entwickeln, welches nicht nur die anforderungsbezogene Synthesesteuerung übernimmt, sondern auch eine eigene „Synthese-Intelligenz“ für Syntheseverfahren, Prozesse und zur Optimierung von Synthesewegen bereitstellen bzw. empfehlen kann. Abbildung 195 stellt die wichtigsten Schwerpunkte und Themengebiete dar, die in der Synthesizer-Applikation realisiert und implementiert werden müssen, um langfristig das Expertensystem zu ermöglichen. Für die Entwicklung der intuitiven Benutzeroberfläche sind im Hintergrund der Applikation viele weitere Funktionalitäten, wie Datenbankoperationen, Formatkonvertierung, elektronisch auswertbare Laborbücher usw., parallel zu entwickeln, einzubinden und erweiterte Schnittstellenwerkzeuge bereitzustellen. Die Implementierung frei verfügbarer Open-Source-Software wie den **GlycanBuilder**^[282,283] zur Generierung von Glykancode über die Schnittstellen in die Synthesizer-Oberfläche, wird das

Anwendungsspektrum und Bedienfreundlichkeit der Synthesizer-Oberfläche erheblich erweitern. Innerhalb dieser glyko-spezifischen Softwarepakete steht die Behandlung und Verwaltung von Glykanen im Vordergrund, wodurch Funktionalität aus aktuell wissenschaftlich relevanten Themenkomplexen zeitnah für neue Synthesekonzepte am Synthesizer bereitstehen. Die daraus auskoppelbaren Informationen lassen sich zur Ansteuerung des Syntheseautomaten nutzen und ergänzen zudem kontinuierlich die Datenbank des zukünftigen Expertensystems.

Der für den allgemeinen Anwender effizienteste Weg eine Synthese am Synthesizer zu parametrieren, wird die Möglichkeit sein, die Synthese an einer funktionalen Standard-Software (z. B. ChemOffice, ChemAxxon-Tools, ACDLabs, u.a.) seiner Wahl zu planen und die Steuerungsabläufe zu simulieren. So könnte in Zukunft eine komplette Synthese innerhalb ChemOffice geplant und als Steuercode für das Synthesegerät genutzt werden. Eine eigene Einarbeitung in die hardwarenahe Parametrieroberfläche der Synthesizer-Software wird somit nicht mehr notwendig. Als Test für die Umsetzbarkeit dieser Variante wurden bereits einige für die Parametrierung der Synthesizer-Software notwendige Funktionalitäten von ChemOffice über die Software-Schnittstelle des Steuerprogrammes erschlossen. Mit den aus ChemDraw abrufbaren Daten lässt sich die Schrittkette für die Steuerung innerhalb des Syntheseprogrammes durchführen. Wie in einem nachfolgenden Kapitel gezeigt, ist damit eine automatische Syntheseführung aus der ChemDraw-Oberfläche (oder vergleichbaren Programmen) prinzipiell möglich.

Unterstützend bei der Bedienung des Synthesizers wurde der Prototyp einer Geräteassistentz für die Synthesizer-Anwendung entworfen (Abbildung 196), die zunächst in Form einer elektronischen Checkliste Hilfestellung zur Inbetriebnahme des Syntheseautomaten leistet. Hierzu werden die wichtigsten Arbeitsschritte in programmatische Checklisten zusammengefasst, die eine sinnvolle und effiziente Reihenfolge (Zeitersparnis, Fehlervermeidung) bei der Abarbeitung darstellen. Gerade für die effiziente Einarbeitung können in diesem Umfeld interaktive Hilfen und unterstützende Informationsmöglichkeiten (Youtube, Moodle, Anki) Einzug halten. Auch zusätzliche Kontrollen wie Kameraauswertung, Strichcode oder NFC⁹⁴-Tag helfen Fehlbedienungen zu vermeiden.

Die größte, effizienteste Erweiterungsmöglichkeit des Funktionsumfangs der neuen Synthesizergeneration besteht jedoch in deren Flexibilität, was die Implementierbarkeit von Schnittstellen zu kommerziellen Chemieapplikationen, vielversprechender frei verfügbarer Open-Source-Software und allgemein zugänglicher Datenbanken betrifft. Gerade in den letzten Jahren hat

⁹⁴ Nahfeldkommunikation (engl. Near Field Communication), internationaler Übertragungsstandard zum kontaktlosen Austausch von Daten per elektromagnetischer Induktion mittels loser gekoppelter Spulen über kurze Strecken von wenigen Zentimetern.

sich der Arbeitsfluss bei Automationsprojekten unter dem Industrie 4.0 Konzept auf den Weg zu service-basierten Systemstrukturen und damit gezielt in Richtung Systemvernetzung entwickelt.

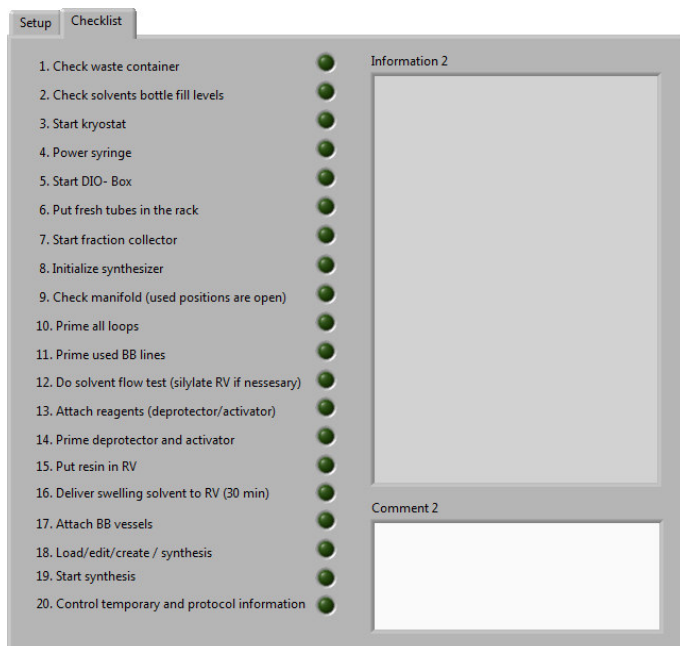


Abbildung 196: In der Weiterentwicklung des Synthesizer-Konzepts ist eine Benutzer-Assistenz zur einfachen Gerätebedienung vorgesehen.

Eine der größten hieraus resultierende Chance/Möglichkeit besteht darin, nicht nur auf eigene Daten der lokalen Arbeitsgruppe, sondern auch auf allgemein verfügbare aktuelle Daten / Datenbanken zuzugreifen, diese automatisch abzugleichen, zu validieren, zu variieren und diese Datenbasis für zukünftige Synthesen (Synthesevorhersagen) bereitzuhalten. Die Vernetzung unterschiedlichster Ressourcen ermöglicht dem Anwender eine zeiteffiziente Verwaltung komplexer Syntheseprojekte, z.B. durch Automatisierung von Protokollierungspflichten, dynamischer Abruf von Analysedaten vernetzter Laborgeräte, usw. ... Die Übergabe von Syntheseinformationen in komplexe Datenbanksysteme wie lokale ChemFinder-Instanzen wurde bereits konzeptionell ermöglicht. Die Möglichkeiten die sich aus einer programmatischen Ankopplung an SciFinder⁹⁵ ergeben sind nahezu unerschöpflich. Testweise wurde für weitere Datenbanken rudimentäre programmatische Anbindungen durchgeführt, die in den nachfolgenden Kapiteln vorgestellt werden.

Mit dem Informationsgewinn und darauf basierende Synthesevarianten, die sich aus dem systematischen, funktionellen Abgleich eigener geschützter, noch nicht publizierter Daten, mit Erkenntnissen verwandter Syntheseprojekte aus allgemeinen Forschungsdatenbanken ergibt, könnte durch intelligente Analysen, Optimierungssynthesen, ein Expertensystem für den Synthesizer entwickelt werden, mit dem Vorhersagen der Syntheseschritte, kombinatorische Synthesen u. a. möglich wären.

Dazu bedarf es jedoch noch vieler weiterführenden automatisierten Interaktionsmöglichkeiten wie

⁹⁵ Online unter: <https://scifinder.cas.org/scifinder/login> [Letzter Abruf: 10.02.2019]

den Zugriff auf UV/VIS – Spektrometer, HPLC, NMR, usw., kurz gesagt, alle im Laboralltag genutzten elektronischen Systeme. Auch für diese Hardware – Schnittstelle konnte bereits erfolgreich eine rudimentäre Anbindung an die Synthesizer-Steuerungssoftware getestet werden.

Ein Offline Simulationsprozess ergänzt das Gesamtkonzept. Um zeit- und kostenintensive Verfahren auf Konzeptmängel, Fehler in der Synthesesteuerung oder unerwünschten Nebeneffekte zu überprüfen, ist ein Evaluierungsprozess für die automatische Syntheseführung vorteilhaft. Dazu wurden bereits einige rudimentäre Programme zur Simulation der technischen Seite erstellt, die ebenfalls nachfolgend vorgestellt werden. Die Evaluierung durch Simulation ist ein wichtiges Element bei cyber-physischen Systemen und dient sowohl der Qualitätskontrolle als auch technischen Sicherheitsaspekten.

Aufgrund der Komplexität der angesprochenen Bereiche, werden die essenziellen Funktionalitäten im Zukunftskonzept des Synthesizers in jeweils eigenen Kapiteln behandelt. Nachfolgend wird in verschiedenen Varianten und Kombinationen die schrittweise Umsetzung der Benutzeroberfläche des Synthesizers von der hardwarenahen Bedienung hin zu einer anwenderbezogenen Applikationsoberfläche vollzogen. Die Varianten bauen größtenteils aufeinander auf und liefern mit den Erfahrungswerten, die durch intelligentes Datenmanagement in einer geeigneten Datenbankstruktur verwaltet und analysiert werden, die notwendigen Verfahrensinformationen zur Umsetzung nachfolgender Komplexitätsstufen für die Bedienung eines universellen Molekülsynthesizers.

5.3.2 Grafische Bedienoberfläche des Synthesizers

5.3.2.1 Schnellparametrier – Benutzeroberflächen

Eine unmittelbar umsetzbare Verbesserungsvariante des aktuellen Oligosaccharidsynthesizers ist die programmatische Bereitstellung eines möglichst selbsterklärenden grafischen Werkzeugs zur Erstellung individueller Benutzeroberflächen, die z. B. als Schnellparametrierdialoge eine erhebliche Bedienerleichterung im Sinne des Bedienkomforts bei der Parametrierung von Synthesen bedeutet. Anhand typischer Synthesen, die an den drei Synthesizern neuer Generation durchgeführt wurden, stellen die Konzeptbeispiele funktionale Umsetzungen individueller Schnellparametrierdialoge dar. Diese eignen sich vor allem zur Rationalisierung wiederholt durchgeführter Struktur-Synthesen, wie Reproduzierbarkeitsstudien oder die Produktbereitstellung für kaskadierte Synthesen.

Unter anderem wurden Oligosaccharid-Fragmente pflanzlicher Arabinoxylane aus Monosacchariden^[183], Xyloglucan Oligosaccharid-Fragmenten aus Mono – und Disacchariden^[184] und verschiedene Typ-II-Arabinogalaktane unterschiedlicher Größe und Komplexität auf der Basis von vier Monosaccharid-Bausteinen hergestellt.^[185]

Die spezifischen Anforderungen der Arabinoxylan-Oligosaccharid Synthese sind in einem

Schnellparametrier - Dialog zusammengefasst, der zu erheblichen Vereinfachungen bei der Eingabe der notwendigen Synthese-Parameter führt, und damit zur Fehlervermeidung in der Syntheseprogrammierung beiträgt (Abbildung 197). In der Oberfläche können die voreingestellten Standard-Parameter der Prozessführung für die Struktursynthese überprüft, bestätigt oder individuell editiert werden.

Mit Hilfe von individuell konfigurierbaren Schaltflächen lassen sich Synthesesyklen einfach und fehlerfrei zusammenstellen und die Struktur von Zielmolekülen definieren. Im einem weiteren Konzeptbeispiel dienen die Definitions-Schaltflächen in der funktionalen Schnellparametrier-Umgebung als Zeichenwerkzeug, mit denen das angestrebte Zielmolekül dargestellt wird. Der Steuercode setzt sich im Hintergrund aus bestehenden Synthese-Skripten zusammen. Das vorgestellte Konzept „Erstellung des Synthesizer-Steuercodes durch Zeichnen der angestrebten Strukturvariante“ wird für die Synthese von XG-Oligosacchariden^[184] (Abbildung 198a) und für die Herstellung von Typ II Galaktan Fragmenten^[185] (Abbildung 198b) mit den Symbolen des Glykancodes⁹⁶ demonstriert. Die, für den verwendeten Zuckerbaustein, typischen Entschützungsmodule können dabei automatisch dem Steuercode zugewiesen werden. Die komplette Syntheseanweisung wird dabei automatisch aus den Standard-Syntheseparametern der hinterlegten Skripte generiert und dem Synthesizer zur Abarbeitung übergeben.

Dem Gesamtkonzept des Syntheseapparates entsprechend, besteht - neben der vereinfachten Benutzer- und Steuerungsoberfläche – immer noch die Möglichkeit, auf die jeweiligen Parametrieroberflächen der Prozessmodule zuzugreifen und in einem „Administratormodus“ individuelle Veränderungen vorzunehmen.

⁹⁶ Glykan-Symbolik basierend auf den Veröffentlichungen vom Consortium for Functional Glycomics (CFG) unter <http://www.functionalglycomics.org> [Abruf: 09.11.2018]

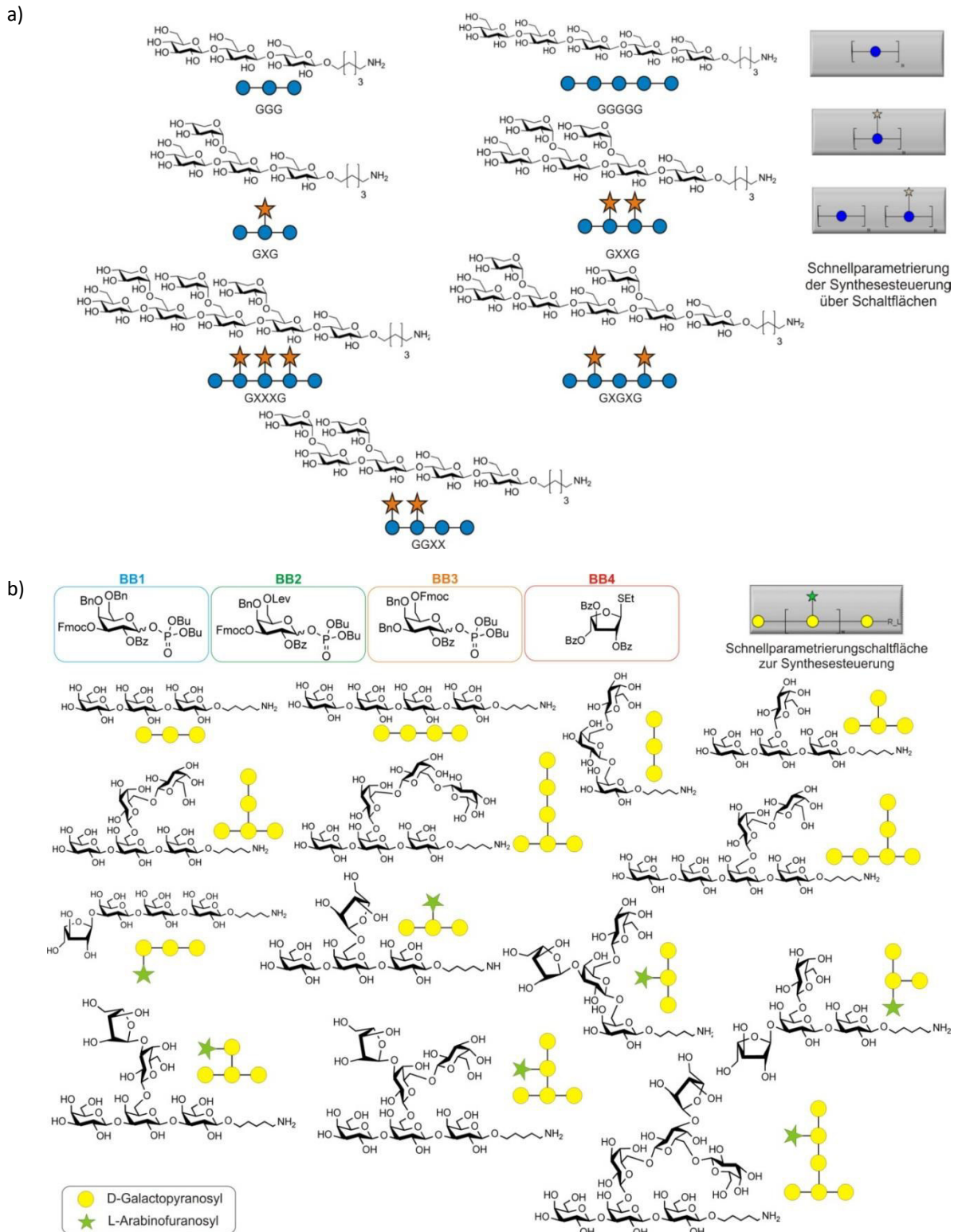


Abbildung 198: Schnellparametrierung von Strukturvariationen über anwenderindividuelle Schaltflächen. Der Zugriff über den „Glykan-Code“ des gewünschten Endproduktes definiert so durch minimale Anpassungen die automatische Synthesesteuerung. a) Strukturvariationen von Cellulose- und XG Oligosaccharid – Fragmenten ^[184] b) Herstellung von Typ II Galaktan Fragmenten aus vier Grundbausteinen. ^[185]

Im Prinzip zwei unterschiedliche Entwicklungen demonstriert die folgende Variante des Konzepts „Erstellung des Synthesizer-Steuercodes durch Zeichnen der angestrebten Strukturvariante“ in Abbildung 199. Hierbei wurde die Anwenderoberfläche zum einen um eine Benutzerverwaltung ergänzt. Der berechtigte Anwender erhält damit den einfachen Zugriff auf seine bereits durchgeführten Synthesen und kann sich seine eigene, benutzerindividuelle Synthesebibliothek anlegen und verwalten, die im Beispiel als namentliche Registerkarten aufgeführt werden. Diese Registerkarten sind im gezeigten Entwurf der Benutzeroberfläche aufgrund des Administrator Logins alle sichtbar. In der programmatischen Umsetzung der Software-Erweiterung wird aus Datenschutzgründen jedoch ein Benutzer- und Freigabemanagement etabliert, in dem der Anwender differenziert den Zugriff auf seine Synthesebibliotheken gänzlich oder teilweise gegenüber anderen Benutzern verbieten, beschränken oder erlauben kann.

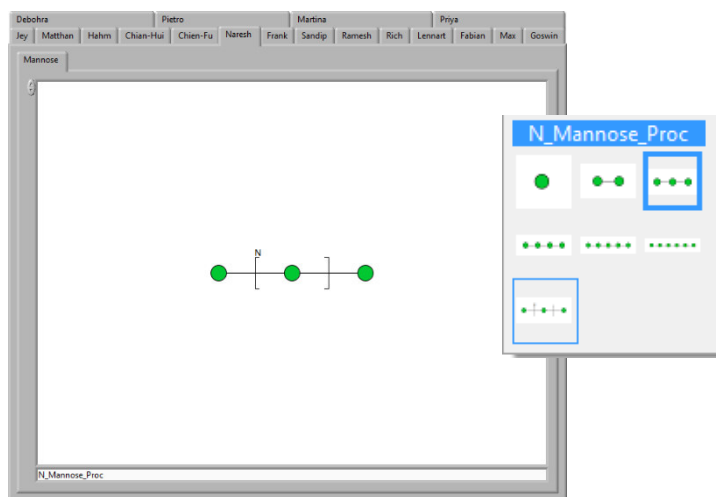


Abbildung 199: Ausschnitt aus der Benutzerverwaltung und vereinfachte Anwenderoberfläche zur Selektion der Synthesesteuerung, z. B. von Mannosen- Synthesen. Das Kontextmenü listet die benutzerindividuelle Synthesebibliothek des selektierten Benutzers auf.

Am neuen Synthesizer wurden anhand von Man- α - (1 \rightarrow 6) –Man –Synthesedurchläufen systematisch, mit ansteigender Zahl von Kopplungen, langkettige Polysaccharide bis hin zum 50-mer erzeugt.^[213] Systematische Struktursynthesen sind ein prädestiniertes Anwendungsgebiet der Synthesizer-Automation. Als zweite Weiterentwicklung der Bedienoberfläche wird in einem zusätzlichen Benutzermenü im Beispiel eine Bibliothek von Mannose – Produkten angezeigt. Durch Selektion des gewünschten Endproduktes lässt sich die Synthesesteuerung direkt aufrufen. Das Syntheseprogramm kann mit den hinterlegten Standardparametern sofort gestartet oder zusätzlich durch Aufrufen eines Kontext-Dialogs individuell parametrisiert werden. Der Kontext-Dialog kann zuvor vom Benutzer als Schnellparametrier - Dialog erzeugt worden sein und wird dann über eine Prozessverwaltung gesteuert.

Das Beispiel in Abbildung 200 zeigt eine projektbezogene Syntheseparametrierung für die Synthese einer n-Mannose mit vereinfachter Eingabe von Kopplungsanzahl, Temperaturwerten und Zeitprofile. Der Bedienungsaufwand für den Anwender reduziert sich auf das absolute Minimum.

Zur Parametrierung der Synthesesteuerung, insbesondere für die Synthese homogener Polysaccharide, spielt in der neuen Synthesizer-Applikation ein besonders leistungsstarkes neu ermöglichtes Programmkonstrukt: die kaskadierbaren Schleifen, eine entscheidende Rolle. In der entwickelten Software stellt diese Umsetzung eine Besonderheit dar, denn üblicherweise administrieren die meisten skriptbasierten Entwicklungsumgebungen maximal 7 Schleifenebenen. Die Kaskadierbarkeit der Schleifen ist in der aktuellen Synthesizer Software – lediglich abhängig vom Speicherplatz – theoretisch unbegrenzt möglich. Die Anpassung auf neue Kopplungsanzahlen stellt dabei nur eine benutzergeführte Parametrierung dar.

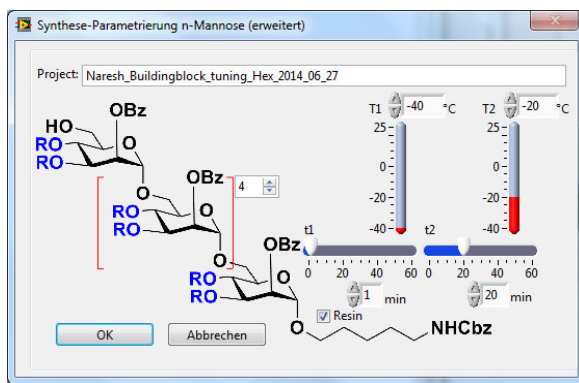


Abbildung 200: Projektbezogene, individuell parametrierbare Benutzeroberfläche zur Synthesesteuerung einer n-Mannose.

Aufgrund des bereits umgesetzten neuen modularen Geräte/Softwarekonzepts, können funktional alle gezeigten Benutzeroberflächen bereits jetzt vollumfänglich die Synthesizer-Steuerung parametrieren und die Synthese damit real hardwaretechnisch durchlaufen.

Effizient und schnell realisierbar ist die Bereitstellung einer Auswahl bereits vorgefertigter Standard-Dialoge innerhalb der Entwicklungsumgebung von LabVIEW für typische Synthesen, wie in den Beispielen gezeigt.

Für die Erstellung benutzerindividueller, ausführbarer Schnell - Parametrieroberflächen zur Steuerung der Synthese ist eine Software-Erweiterung erforderlich, die es dem Anwender ohne Programmierkenntnisse ermöglicht, diese funktionalen, prozessbezogenen Dialoge einfach per Drag- & Drop zu erstellen, und diese automatisch mit der vorhandenen Synthese-Steuerung zu verbinden. Unplausibilitäten, Logikfehler und Schrittketten-Probleme, die sich durch die Erstellung benutzerindividueller Schnellparametrier-Dialoge ergeben könnten, müssen durch ein qualifiziertes, intelligentes Fehlermanagement automatisch abgefangen werden.

Für die individuelle Benutzer- & Datenverwaltung ist ein entsprechendes Berechtigungsmanagement in die vorhandene Software zu integrieren.

5.3.2.2 Der Building Block Selektor zur vereinfachten Dateneingabe

Der **Building Block Selektor** ist ein Konzept zur Umsetzung einer anwenderorientierten Eingabemaske mit struktureller Synthese-Steuerung vor allem für Prototyp-Synthesen. Während des Entwurfs dieser Funktionsweiterung für die aktuelle Synthesizer-Software, konnten die zentralen Hauptpunkte der Programmierung unter LabVIEW bereits gelöst und funktionale Tests zur weiteren Realisierbarkeit des Konzepts durchführt werden.

Die wesentliche Konzeptidee besteht in einer „chemischen Synthesesteuerung“ durch Eingabe der verwendeten Zuckerbausteine (Building Blocks) und dem resultierenden Wunschmolekül. Der **Building Block Selektor** erlaubt eine schnelle, strukturierte und chemisch eindeutige/vorselektierte Dateneingabe. Die zugeordneten Building Blocks werden automatisch mit der Syntheseparametrierung abgespeichert und stehen intern zur Verarbeitung zur Verfügung z. B. für die steuerungstechnische oder chemische Datenanalyse, wenn weitergehende automatisiert erfasste Prozessdaten vorliegen. Die Verwaltung der Building Blocks als aktive Strukturelemente ermöglichen die direkte Verwendung als interpretierbaren chemischen Code, sowohl mit Strukturformel als auch mit funktionalen Referenzen.

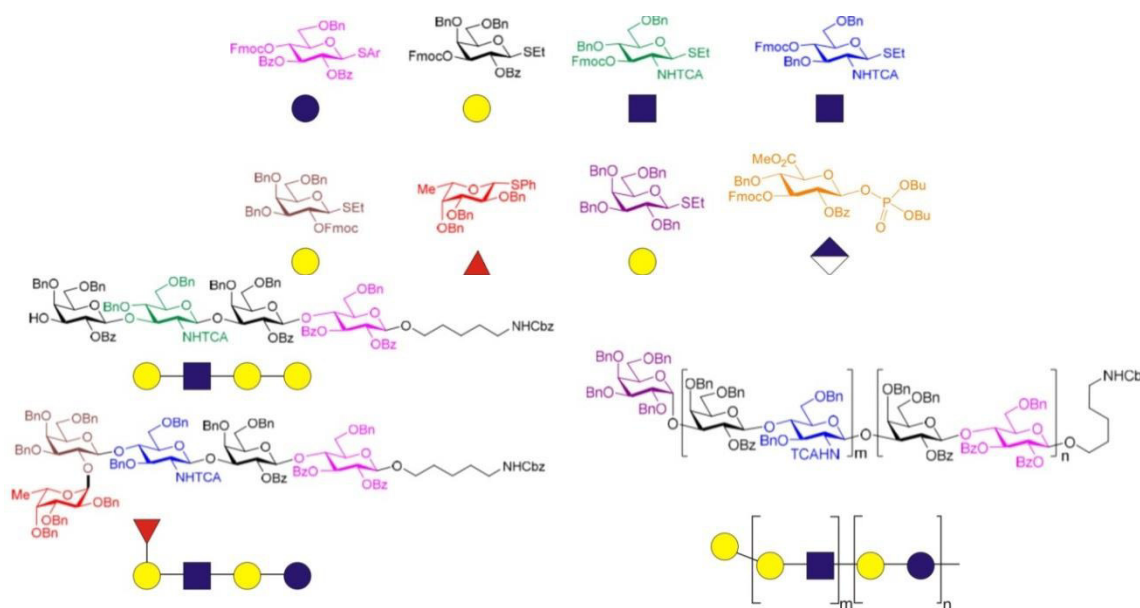


Abbildung 201: Exemplarisch stehen die Monosaccharid – Building Blocks zur automatischen Synthese von Lactotetraosyl (Lc4) und Neoleactotetraosyl (nLc4) – Kernstrukturen für das Projekt „Automated Glycan Assembly of Complex Oligosaccharides Related to Blood Group Determinants“^[284] für die Vielfalt möglicher Zuckerbausteine. Dies erfordert eine effektive Datenverwaltung von Prozessinformationen in Relation mit den verwendeten Building Blocks zur Durchführung von System- / Prozessanalysen.

Als Beispiel ist nachfolgend die Umsetzung einer α -D-Glukoseeinheit innerhalb des **Building Block Selektors** vorgeführt. Aus der bekannten Strukturinformation seitens ChemDraw lässt sich die Parallelität in der Verwendung direkt nachvollziehen. Der **Building Block Selektor** unterstützt den Anwender aktiv bei der Eingabe und ermöglicht die einfache Auswahl aller bisher zur Anwendung gekommenen chemischen Grundeinheiten (Building Blocks).

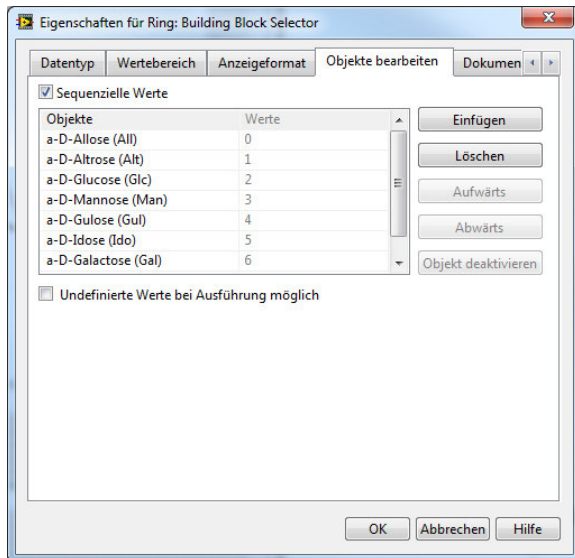


Abbildung 202: Definitions-Dialog für den Building Block Selektor

Bei den hierbei verwendeten Building-Block Einheiten unterscheiden sich nur die verschiedenen Substituenten, weshalb hier Selektionsobjekte innerhalb eines Selektionsrings zur Anwendung gekommen sind.

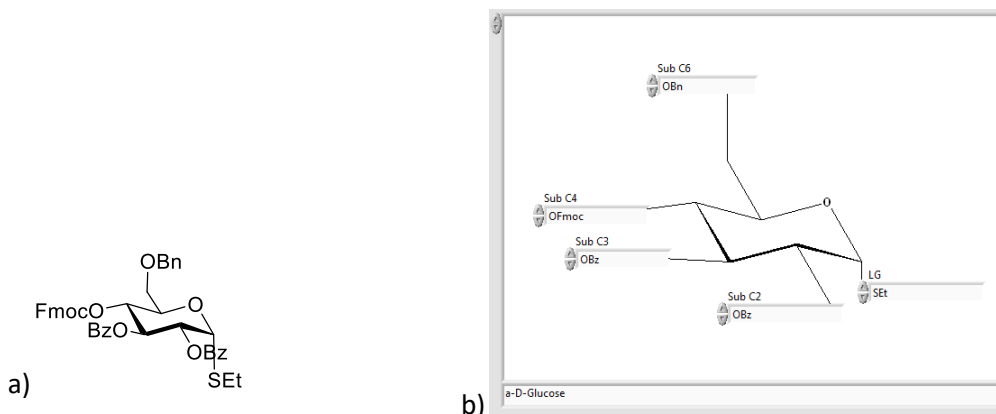


Abbildung 203: a) Zuckerbaustein (Building Block) b) Analog dazu der Zuckerbaustein im Building Block Selektor, einer in LabVIEW programmierten Programmoberfläche zur Eingabe der verwendeten Zuckerbausteine mit wählbarer Abgangsgruppe (LG Leaving Group) und verschiedenen Schutzgruppen (Sub).

Die auswählbaren Substituenten ergeben sich aus den bisher synthetisierten Zuckerbausteinen. Da noch nicht alle kombinatorische Möglichkeiten erfasst sind, lässt sich die Liste vom berechtigten Anwender kontinuierlich erweitern. Die Informationen liegen als Datenbankobjekte organisiert vor und werden als Text-Auswahlobjekt zur Verfügung gestellt. Vergleichbare Auswahlmöglichkeiten stehen für die Abgangsgruppen bereit.

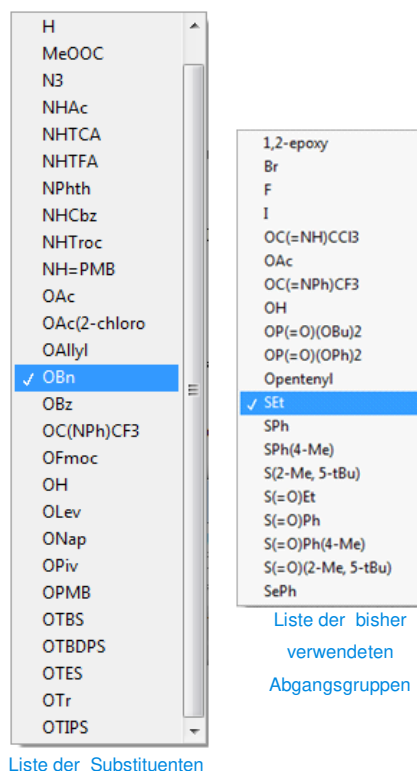


Abbildung 204: Auswahllisten der aktuellen Substituenten und der bisher verwendeten Abgangsgruppen

Der steuerungstechnische Programmcode erstellt sich aufgrund der eingegeben Zuckerbausteine und des angestrebten Zielmoleküls. Dabei gleicht sich die Synthesevorschrift aufgrund der hinterlegten Module, Prozesse und Skripte gleicher oder vergleichbarer Zuckerbausteine ab und generiert so automatisch die Synthesesteuerung und/oder schlägt verschiedene mögliche Varianten zum Syntheseverfahren basierend auf den erfolgten Eingaben vor. Dazu nutzt das Programm in der primären Umsetzungsstufe eine Datenbank, welche die vorhandenen Module, Prozesse und Skripte in Korrelation mit den Zuckerbausteinen und deren typischen Schutzgruppen mit Entschützungsbedingungen verwaltet.

Die Anbindung an ein komplexes, flexibles Datenbanksystem, welches die relevanten Syntheseinformationen möglichst automatisiert, standardisiert, kontrolliert und damit fehlerresistent in der Datenbank abgelegt, gehört zu einem zentralen Punkt im nachfolgend umfassenden Konzept der „chemischen Synthesesteuerung“.

Mit den, in späteren Kapiteln vorgestellten Erweiterungen/Schnittstellen zu verschiedenen Open-Access- Datenbanken, lässt sich in einer fortgeschrittenen Umsetzungsvariante nach strukturellen und funktionellen Analoga suchen und deren Leitstruktur, sowie entsprechende ähnliche Syntheseverfahren auswerten. Gerade in der Medikamentenentwicklung ist es üblich eine große Zahl struktureller Analoga zu einer gegebenen Leitstruktur zu erstellen oder Datenbankanalysen nach strukturellen Analoga zu gegebenen Leitstrukturen durchzuführen.^[285] Die hieraus gewonnenen Erkenntnisse können in die Gesamtkonzeption integriert werden.

Mit den Ähnlichkeitsanalysen verschiedener Analoga zum gewünschten Zielmolekül oder den

verwendeten Zuckerbausteinen, lassen sich – die entsprechende Analyselogik vorausgesetzt – in Zukunft mit der angestrebten Ausbaustufe des Synthesizers zum Labor-Expertensystem, auch systematisch automatisiert nach Varianten und Synthesealternativen suchen und gezielt Synthesevorschläge unterbreiten.

5.3.2.3 Chemiebasiertes Benutzer-Frontend

Bei den bisherigen Automatisierungssystemen wird die Steuerung der Syntheseabläufe programmatisch vorgegeben. Selbst bei den modernen kommerziell verfügbaren Peptid- und DNA-Synthesizer ist die Abstraktionsebene lediglich auf der Ebene einer Sequenzerstellung begrenzt. Die eigentlich notwendige vollständige chemische Synthese/Reaktions-Information auf Molekülebene fehlt bei allen bisherigen Konzepten.

In diesem weiteren, komplexen Konzeptvorschlag wird als Zwischenschritt zur angestrebten chemischen Synthesesteuerung ChemDraw als Standardprogramm in die Synthesizer-Software eingebunden. In der durchgeführten Machbarkeitsanalyse wurde eine rudimentäre Einbindung der ChemDraw Funktionalitäten ausgetestet und die Realisierbarkeit einiger zentraler Kernpunkte überprüft, die für den Betrieb mit dem neuen Synthesizer essenziell sind. Über LabVIEW konnte so bereits der Zugriff per Eigenschaftsknoten und Methodenknoten bis hinunter zu den einzelnen Objekten innerhalb der Reaktionsschritt-kette realisiert werden. Die Fusion eines Standard-Chemie-Programmes mit der hardwarenahen Steuersoftware eines automatischen Synthesegerätes ist weltweit bislang einmalig und könnte in Anlehnung an die Begrifflichkeiten des Industrie 4.0 Konzepts ein Meilenstein in der Entwicklung eines cyber-physischen „Chemiesystems“ darstellen.

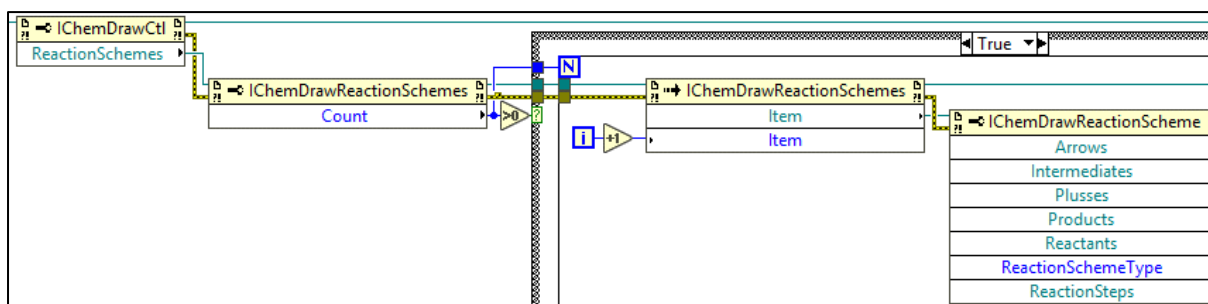


Abbildung 205: LabVIEW - Programmausschnitt, der den Zugriff auf die verschiedenen Eigenschaftsknoten des ChemDraw Objekts bis hinunter zu den einzelnen Objekten innerhalb der Reaktionsschritt-kette zeigt.

Da ChemDraw als Standard-Werkzeug lizenziert sowohl an der Hochschule als auch am Institut vorliegt, ist die vollständige Einbindung in LabVIEW für die neuen Arbeitssynthesizer ohne zusätzliche Lizenzkosten möglich.

Die Verbindung von ChemDraw, als funktionales, EDV-technisches Werkzeug zur grafischen Darstellung von Reaktionskinetik und Molekülvisualisierung, innerhalb von LabVIEW, welches die

Synthese steuert, bietet eine Vielzahl an Synergieeffekten, die in den nachfolgenden realisierten Beispielen vorgestellt werden.

Die ChemDraw Funktionalität wurde durch das Einbinden einer ActiveX –Komponente in das Frontpanel von LabVIEW programmatisch bereitgestellt (Abbildung 206). Damit kann die Strukturformel auch „außerhalb“ der Synthesizer-Software, also in der klassischen ChemDraw-Umgebung oder einer alternativen Chemie-Software, die das ChemDraw – Format bereitstellt (z. B. ACD/ChemSketch, Marvin von ChemAxxon) erstellt werden. Diese Strukturobjekte lassen sich über die Zwischenablage in die LabVIEW Applikation kopieren. Dabei wird nicht nur die reine grafische Bildinformation übernommen, sondern die gesamte Informationsstruktur hinter der chemischen Symbolik als Objekt bereitgestellt.

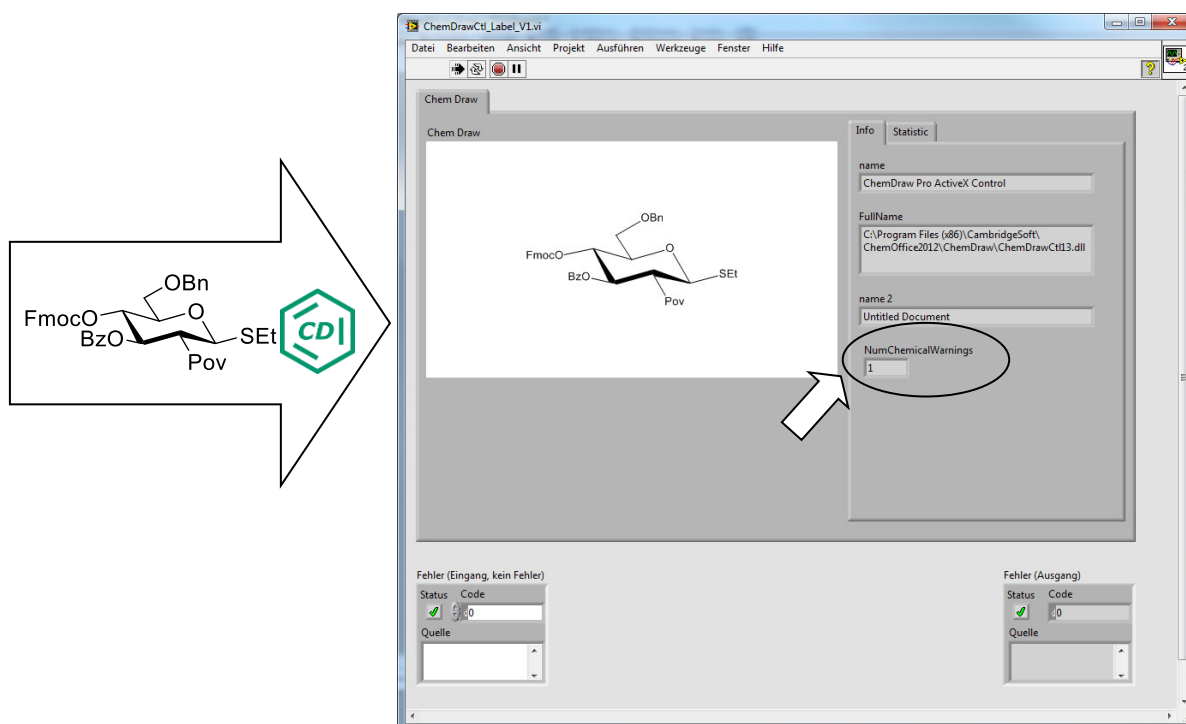


Abbildung 206: ChemDraw Objekte können über die Zwischenablage in die erstellte LabVIEW Applikation kopiert werden und stehen dann mit ihrer vollen Informationsstruktur zur Verfügung. Das LabVIEW-Programm übernimmt die übergebene Visualisierung und Zusatzinformationen.

Sind Objekte nicht korrekt erkannt bzw. eingegeben worden, z. B. durch Tippfehler, kann dies von LabVIEW auch ohne Einblendung der chemischen Warnungen, einer ChemDraw Funktionalität, ausgewertet werden. In Abbildung 206 wird die vorhandene Warnung in der Registerkarte Info ausgegeben. Ist die ChemDraw Funktionalität „Show chemical warnings“ aktiviert, wird das entsprechende fehlerhafte Objekt im Grafikbereich des erstellten Beispielprogrammes als rot umrandet dargestellt (Abbildung 207).

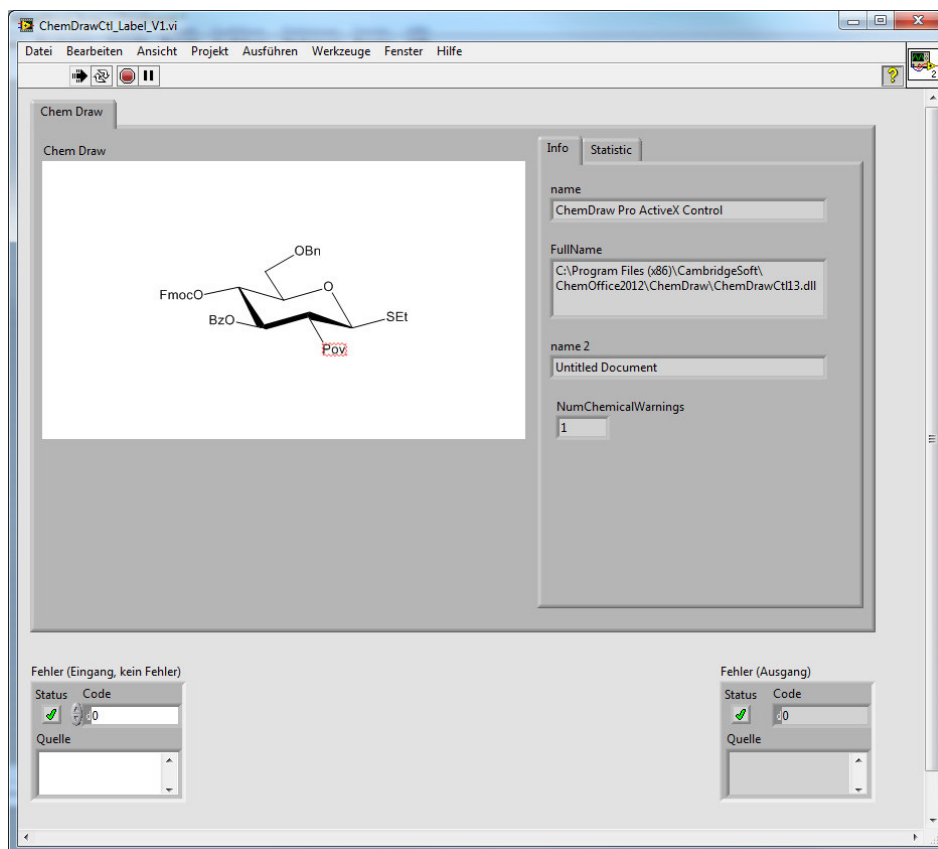


Abbildung 207: Das LabVIEW Programm wertet automatisch Fehler aus. Aufgrund von Tippfehlern nicht korrekt erkannte Objekte werden im chemischen Molekül rot umrandet.

Im LabVIEW-Beispielprogramm kann ohne zusätzliche Programmierung vollwertig über das Kontextmenü des ActiveX Objektes auf die ChemDraw Funktionen zur Visualisierung der Moleküle zugegriffen werden (Abbildung 208).

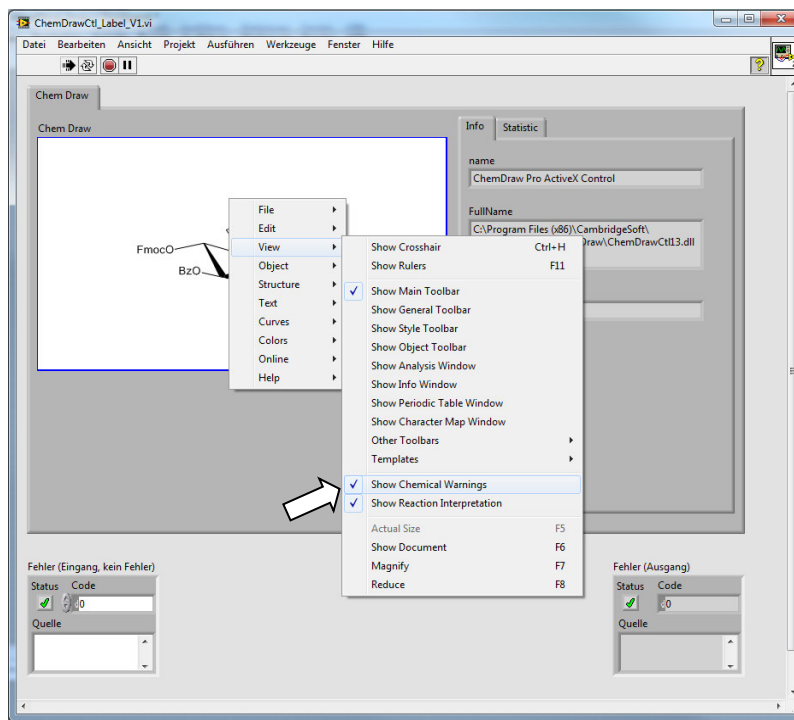


Abbildung 208: Kontextmenü – innerhalb des ChemDraw – Synthesizer VIs. Über das Kontextmenü können auch die chemischen Warnungen angezeigt werden.

ChemDraw hält einen riesigen Funktionsumfang für Chemiker, Biologen und Mediziner bereit, z. B. 3D-Darstellung von Molekülen mit Schnittstellen zu Rechenprogrammen wie Gauss⁹⁷, MOPAC⁹⁸ und Conflex⁹⁹, Simulation von NMR- (CDCl₃ oder DMSO) und MS-Spektren, die „Struct=Name“-Funktion, die z.B. IUPAC-Namen zu Strukturen umwandelt und umgekehrt, sowie Fragmentierungs-Werkzeuge (Massenfragmentierung, Dissoziation, Retrosynthese). ChemDraw versteht in der neuesten Version auch die Hierarchical Editing Language for Macromolecules (HELM), die im nachfolgenden Kapitel Syntheseplanung und Beschreibungssprache eine wichtige Rolle spielt. Mit HELM lassen sich als Beispiel komplexe Strukturformeln von Biomolekülen einfach aus einer persönlichen Bibliothek von Monomeren zusammensetzen.

Zu den zentralen Funktionalitäten zählt jedoch das funktionale Zeichnen biologischer Reaktionspfade (BioDraw), sowie das Zeichnen von Peptid- und Nukleotid-Sequenzen mit 1- und 3-Buchstaben-Notation (Biopolymer-Toolbar). Die Biopolymer-Toolbar könnte damit vollständig als Benutzeroberfläche für die Parametrierung des zukünftigen Molekül-Synthesizers zur Herstellung von Peptiden oder DNA, nachfolgend auch als Betrieb im „Peptid- oder DNA-Modus“ bezeichnet, fungieren.

Für die komplexe Glykanchemie eignet sich BioDraw zum Zeichnen der vollständigen Reaktionspfade. Mit dieser ChemDraw – Funktionalität kann die komplette Synthesen-Parametrierung erfolgen, wodurch diese auch vollständig als implementierte Benutzeroberfläche zum Betrieb des Molekül-Synthesizers im „Glykan-Modus“ genutzt werden kann. Damit erhält der Anwender die Möglichkeit seine Synthese mit einem bekannten Werkzeug „programmieren“, ohne zusätzliche Einarbeitung in die spezielle Synthesizer-Software.

Im folgenden Beispiel wird der prinzipielle Arbeitsablauf und das Steuerungskonzept für eine Synthese am Synthesizer mittels ChemDraw näher erläutert.

Bislang werden Syntheseabläufe im Synthesizer in einem statischen Bild visualisiert (Abbildung 209). Im Gegensatz dazu stellt ChemDraw beim Zeichnen einer Reaktionsgleichung im Hintergrund mittels HotLink Verbindung¹⁰⁰ zur Perkin Elmer Informatics Datenbank zusätzliche chemische Informationen zur Verfügung, z. B. ergänzt es Strukturen, erkennt Fehler, liefert physikalische Eigenschaften, usw.

⁹⁷ GAUSS ist ein kommerzielles Mathematik- und Statistiksystem zur Programmierentwicklung. Es dient vor allem der Lösung numerischer Probleme in den Bereichen Statistik, Ökonometrie, Zeitreihen, Optimierung sowie 2D- und 3D-Visualisierung.

⁹⁸ MOPAC (Molecular Orbital PACKage) ist ein semiempirisches Quantenchemieprogramm, das auf der NDDO-Approximation von Dewar und Thiel basiert.

⁹⁹ CONFLEX ist eine Konformationsuche und Analysesoftware zur Untersuchung des Konformationsraums von kleinen und großen Molekülen. Es enthält auch Algorithmen zur Vorhersage von IR-, NMR-, CD- und UV-Vis-Spektren.

¹⁰⁰ Ein HotLink bezeichnet die Integration extern gespeicherter Medien. Bei ChemDraw wird während dem Zeichnen in Echtzeit in der Perkin Elmer Informatics-Datenbank nach chemischen Strukturen gesucht. Änderungen in Strukturdiagrammen, alle chemischen Bezeichnungen, Formeln, Molekulargewichte und andere zu einem Dokument hinzugefügten physikalischen Eigenschaften werden "live", d. h., automatisch mittels HotLink aktualisiert.

Die Reaktionsgleichung ist somit ein aktives „Chemie“-Objekt, aus dem sich eine aktive Reaktionsumsetzung generieren lässt.

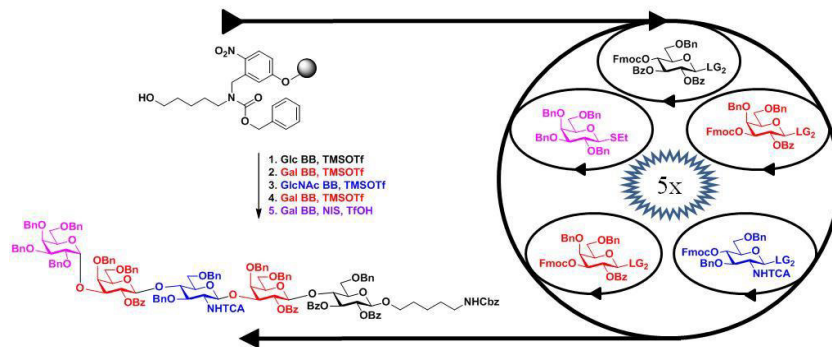


Abbildung 209:
Parametrierung und ihre funktionale Bedeutung

Dieser Machbarkeitsanalyse liegt folgendes Konzept zugrunde: wird eine Reaktionsgleichung in ChemDraw unter Verwendung von beschreibenden Reaktionspfeilen eingegeben, so kann ChemDraw den einzelnen Bestandteilen der Reaktionsgleichung unterschiedliche Objektbedeutung zuweisen (Abbildung 210). Wird die chemische Reaktion formal informationstechnisch korrekt geschrieben, haben die Positionen der Objekte eine funktionale Bedeutung, aus denen informativer Sachverhalt für die Synthese-Steuerung entnommen werden kann.

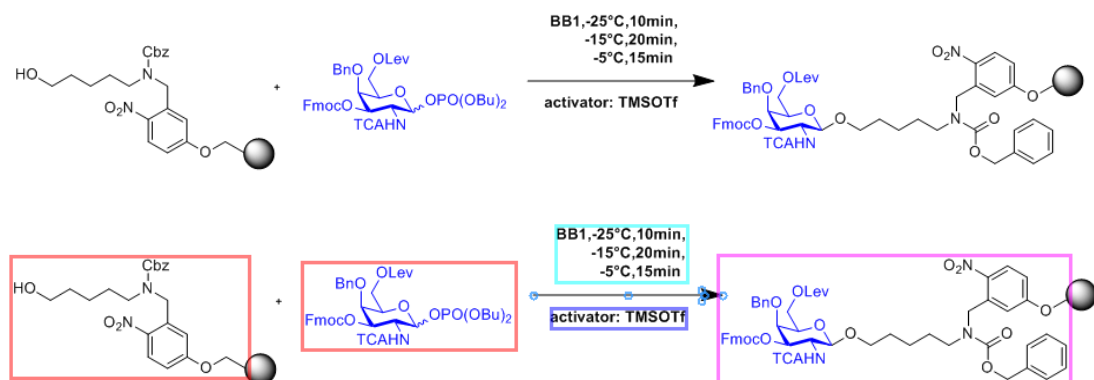


Abbildung 210: a) Die Reaktionsgleichung in ChemDraw kann als funktionaler „chemischer Programmcode“ von LabVIEW genutzt werden. b) Die farbige Zuordnung zur Erkennung der Objektbedeutung der einzelnen chemierelevanten Informationen als Grundlage zur „Funktionssteuerung des Synthesizers“

ChemDraw unterscheidet zwischen Reaktanten, Produkten und Intermedialen (Abbildung 210 b). Reaktanten befinden sich durch + Zeichen getrennt auf der linken Seite des Reaktionspfeils (roter Rahmen). Produkte entsprechend auf der rechten Seite des Reaktionspfeils auch durch + Zeichen getrennt (rosa Rahmen). Die Intermedialen sind oberhalb und unterhalb des Reaktionspfeils angesiedelt. Von ihrer Positionierung her sind diese innerhalb der Grenzen des Reaktionspfeils zu platzieren, um als gültige Objekte zugewiesen/erkannt zu sein.

Dabei unterscheidet ChemDraw zwischen Intermedialen oberhalb des Reaktionspfeils und unterhalb

in einer jeweils eigenen Objekthierarchie. Hierdurch wird es für die zukünftige Ausnutzung möglich, Lösungsmittel und Katalysatoren, wie allgemein üblich, oberhalb des Reaktionspfeils zu platzieren, wohingegen erweiterter funktionaler Anweisungscode unterhalb des Reaktionspfeils zu platzieren ist.

Mit dem erstellten Beispiel – LabVIEW Programm konnten die wesentlichen Informationen aus der ChemDraw Reaktionsgleichung auslesen werden, aus denen sich im folgenden Arbeitsfluss der Zugriff auf die Module, Skripte und Prozesse der Synthesizer-Ansteuerung ergibt. Damit stehen sowohl der Reaktionszyklus für den verwendeten Zuckerbaustein (Temperaturverlauf), als auch das verwendete Lösungsmittel programmatisch im Zugriff (Abbildung 211).

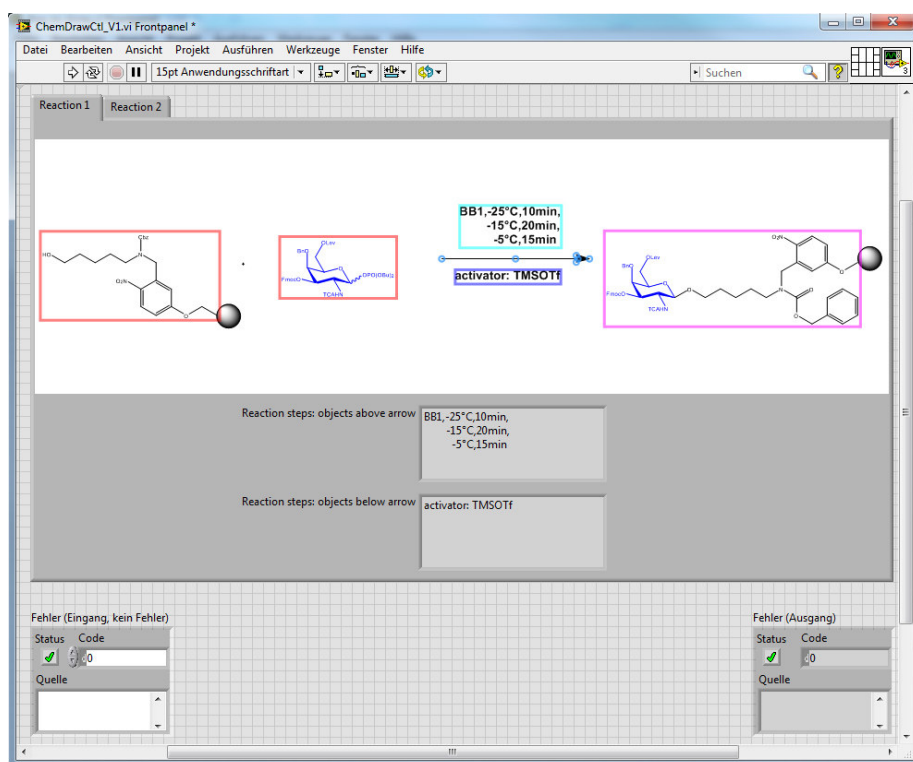


Abbildung 211:
Zuordnung der Reaktionsparameter für die Synthesesteuerung in LabVIEW

Bisher wird der Ansteuerungscode der Automatisierungssysteme rein aus textbasierten Modulen zusammengestellt und parametrisiert oder programmiert. Der Anwender erhält damit Zugriff auf alle Funktionalitäten des Systems, muss sich jedoch auch mit den hardwarenahen Eigenschaften der Systemmodule auseinandersetzen. Mit den Möglichkeiten, die sich aus dem direkten Zugriff auf die in einem reinen Chemieprogramm eingetragenen Informationen ergeben, lassen sich zukünftig die Ansteuerungssequenzen automatisiert erzeugen. Im Übergangszustand müssen die dazu notwendigen Templates¹⁰¹ noch individuell bereitgestellt und auf die jeweilige erstmalige Syntheseführung angepasst werden. Unter Einbeziehung eines möglichst breiten Informationsspektrums anderer Arbeitsgruppen, sowie weiterer externer Quellen, kann mit konsequenter Pflege der Syntheseinformationen in die komplexe, chemische Datenbank des

¹⁰¹ Allgemein: Vorlage in der Datenverarbeitung. Speziell: Vorlagen zur Erstellung des Steuerungscode des Synthesizers

zukünftigen Molekülsynthesizers eine anspruchsvolle Wissensbasis zur Verfügung gestellt werden. Als Quelle für Synthesevorschläge und Empfehlungen bildet dies, mit entsprechenden Auswertelgorithmen und unter Anwendung von Methoden der künstlichen Intelligenz, die Grundlage eines zukunftsweisenden Expertensystem (vgl. Abbildung 214).

5.3.2.4 Molekül-Editor

Noch ist die Wissenschaft weit davon entfernt rein auf Basis der gezeichneten Molekülstruktur eine Synthese vollautomatisch durchführen zu können, doch mit der in dieser Arbeit entwickelten Software- und Hardwarearchitektur des modularen Syntheseautomaten wurden die Grundlagen zur zukunftsweisenden Entwicklung gelegt. Viele verschiedene Entwicklungsschritte sind notwendig, vor allem muss eine grundlegende, breite Datenbasis für unzählige Parameter, Strukturen und Randbedingungen gelegt werden, die standardisiert abgelegt und intelligenten Analysen zur Verfügung steht. Derzeit entwickeln sich die technologischen Voraussetzungen zur Konstruktion eines Molekülsynthesizer-Expertensystems geradezu exponentiell. Die finale Bedienoberfläche für die Synthesesteuerung wird hierbei der Moleküleditor sein. Aufgrund der hardwarenahen Programmierung der Gerätesteuerung ist die Benutzeroberfläche in der Applikationsebene für die Ansteuerung des Syntheseapparates unerheblich. Jedes übliche Programm zur Darstellung chemischer Sachverhalte, wie CheS-Mapper^{102[286]} kann als Schnittstelle zur Synthesedefinition für den Molekülsynthesizer dienen. Da sich Scientific Computing immer mehr in Richtung Web Services und Cloud Computing bewegt, werden Webbrowser zukünftig zur universellen wissenschaftlichen Benutzeroberfläche.^[287] Eine zukunftsichere Variante, die sich aufgrund der Dezentralisierung des Synthesizerkonzepts anbietet, ist deshalb die Verwendung web-basierter Anwendungen, die innerhalb einer integrierten Browseroberfläche als Containeranwendung aufgestartet werden können. Hierdurch werden sowohl Java Script (JSME^{103[288]}, Ketcher^{104[289]}), Java (Jmol^{105[290]}, MarvinTools¹⁰⁶, JChemPaint^{107[291]}) wie auch HTML5 GUI's (ChemDrawDirect¹⁰⁸) direkt nutzbar (Abbildung 212). Da bei einigen der Editoren^[292] direkt die Eingabe von Reaktionsinformationen möglich ist, kann die Applikationsebene des Synthesizers diese Daten analytisch interpretieren und für der Ansteuerung im Molekülsynthesizer verwenden. Hierdurch wird die Möglichkeit geschaffen komplett neue Synthesesequenzen zu designen und diese automatisch am Synthesizer ausführen zu lassen.

¹⁰² Online unter: <http://ches-mapper.org/> [letzter Aufruf: 29.01.2019]

¹⁰³ Online unter: <https://peter-ertl.com/jsme/> [letzter Aufruf: 29.01.2019]

¹⁰⁴ Online unter: <http://lifescience.opensource.epam.com/ketcher/> [letzter Aufruf: 29.01.2019]

¹⁰⁵ Online unter: <http://jmol.sourceforge.net/> [letzter Aufruf: 29.01.2019]

¹⁰⁶ Online unter: <https://sdss-marvin.readthedocs.io/en/stable/tools.html> [letzter Aufruf: 29.01.2019]

¹⁰⁷ Online unter: <https://jchempaint.github.io/> [letzter Aufruf: 06.02.2019]

¹⁰⁸ Online unter: <https://chemdrawdirect.perkinelmer.cloud/js/sample/index.html> [letzter Aufruf: 29.01.2019]

Kürzlich wurde mit Chematica ein Programm vorgestellt, das autonom chemische Synthesewege von medizinisch relevanten Molekülen konstruiert. Bislang konnten die maschinell entworfenen Routen von acht strukturell unterschiedlichen Zielmolekülen erfolgreich im Labor hergestellt und bisherige Synthesewege verbessert werden.^[293] Auch wenn noch viel Forschungsarbeit zu leisten ist, die Entwicklung in den Forschungslaboren geht in Richtung computergestützter chemischer Assistenzsysteme wie sie im Expertensystem des Molekülsynthesizers angestrebt wird.

Durch die Bereitstellung der Ketcher-App innerhalb der Synthesizer-Software können Eingaben für die Synthesen dann auch auf mobilen Endgeräten durchgeführt werden.^[294] Die Daten werden von dort direkt geteilt und stehen im Applikationssystem zur Verfügung. Da die Glykanstrukturen aus einer Grundklasse von Monomeren aufbaubar sind, eignet sich das Anlegen von Templates oder die Verwendung von speziell hierfür entwickelten Editoren (GlycoBlocks^{109[295]}, GlycoViewer¹¹⁰, GlycanBuilder^{111[282,283,296]}, SugarSketcher, ...), deren Betrachtung für die Einbindung in Entwicklungsumgebung des Expertensystems später erfolgt.

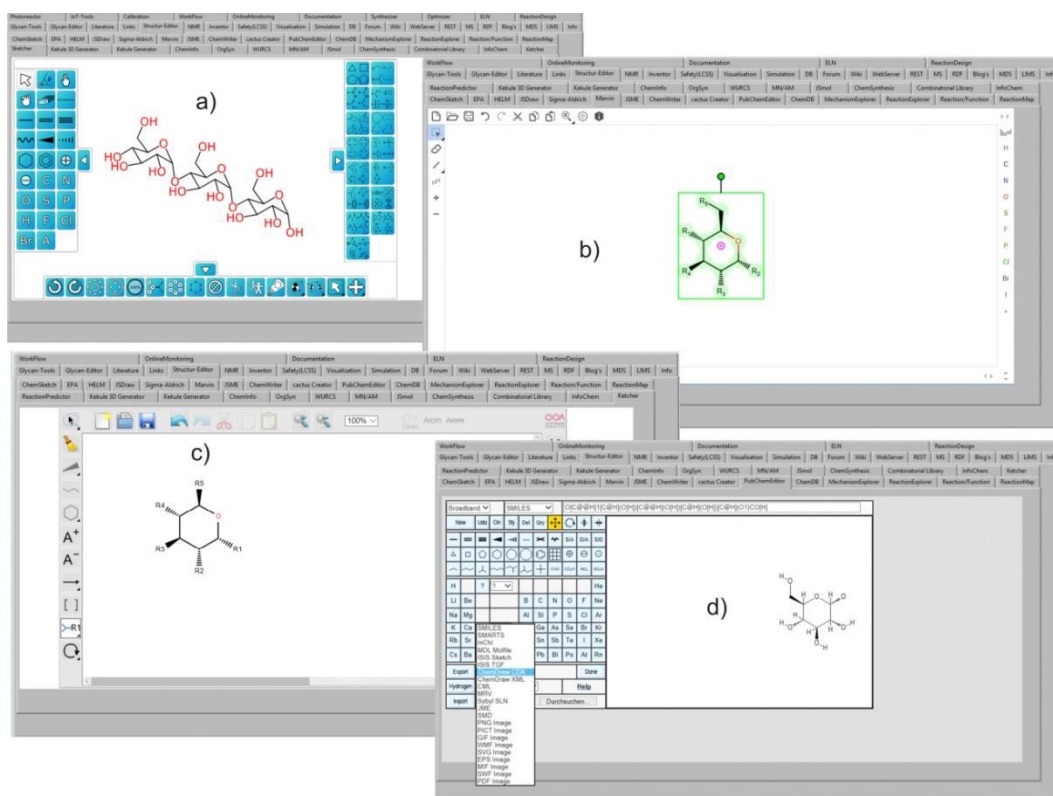


Abbildung 212: Für das „Proof of Konzept“ des zukünftigen Molekülsynthesizers wurden verschiedene Molekül-Editoren in die Synthesizer-Applikation implementiert: a) Sketcher, b) Marvin, c) Ketcher, d) PubChem

¹⁰⁹ Online unter: <http://www.ccp4.ac.uk/> [letzter Aufruf: 29.01.2019]

¹¹⁰ Online unter: <http://www.glycoviewer.babs.unsw.edu.au/> [letzter Aufruf: 29.01.2019]

¹¹¹ Online unter: <http://www.unicarbk.org/> [letzter Aufruf: 29.01.2019]

5.3.2.5 Glykan-Editor

Die Steuerung des Synthesizers aufgrund eines im Molekül-Editor gezeichneten chemischen Produktes, ist durch die Kenntnis des Syntheseweges in internen oder externen Datenbanken oder durch Auswertung von Reaktionsbeschreibungen z. B. in ChemDraw ein realistisches mittelfristiges Ziel. Peptide und Oligonukleotide lassen sich aus einer begrenzten Anzahl von Grundbausteinen linear zusammensetzen. Die Anleitung zur Herstellung eines Peptides oder Oligonukleotides (vgl. Abbildung 4, S. 4) kann somit im Molekül-Editor durch die Eingabe des einheitlichen Buchstabencodes standardisiert nach IUPAC erstellt werden, und ist somit auch von Nichtchemikern vereinfacht zu handhaben.

Zur strukturellen Beschreibung von Kohlenhydraten wurde mit LINUCS^[297,17] ein linearer Textcode realisiert, der die Vielfältigkeit der Glykanstrukturen abbildet. Der Code selbst ist zur eindeutigen Strukturidentifikation vor allem in Datenbanken und für Datenbankabfragen konzipiert. Als Eingabecode zur Herstellung von Glykanen ermöglicht er aufgrund seiner Komplexität für den Anwender keine grundsätzliche Bedienvereinfachung am Synthesizer.

Innerhalb des Zeitrahmens dieser Arbeit etablierte sich jedoch eine neue Beschreibungsstruktur für Kohlenhydrate auf Basis von Monosacchariden: Symbol Nomenclature for Glycans (SNFG)^[20]. Hierbei werden Glykanstrukturen aus Monosacchariden aufgebaut, die jeweils als grafische Symbole veranschaulicht werden (vgl. Abbildung 5, S. 5). Diese Art der Darstellung vereinfacht die Komplexität von Kohlenhydratstrukturen und macht sie für den Benutzer wesentlich intuitiver begreifbar und editierbar. Die SNFG-Nomenklatur bietet sich deshalb als neue und effiziente Eingabeschnittstelle des Expertensystems an.

Mit der SNFG-Nomenklatur lassen sich im Glykan-Editor als Benutzerschnittstelle die gewünschten Zielmoleküle definieren und diese als Syntheseauftrag an den Molekülsynthesizer übergeben. Bekannte Glykanstrukturen können durch die in der Datenbank hinterlegten Reaktionsbeschreibungen direkt anforderungsbezogen hergestellt werden. Für noch nicht am Expertensystem hergestellte Strukturen, lassen sich zudem Daten aus externen Datenbanken heranziehen, sowie aufgrund von Ähnlichkeitsanalysen Synthesewege vorschlagen. Weitere essenzielle Softwareimplementierungen für das Expertensystem werden in nachfolgenden Kapiteln thematisiert.

Durch Integration von Open-Source Quellen können Funktionalitäten innerhalb des Glykan-Editors direkt und schnell bereitgestellt werden. Das Konzept der „offenen“ Benutzerschnittstelle, welches die Einbindung externer Open-Source Werkzeuge, aber auch kommerzieller Software in das System vorsieht, ermöglicht es die Benutzeroberfläche der Synthesizer-Applikation stets auf dem aktuellen Forschungsstand zu halten ohne aufwendige Eigenentwicklungen abseits der Molekül-Synthesizer-

Kernapplikation. Zudem bietet es für den Benutzer den Vorteil, sich jeweils in seiner favorisierten Anwendung zur Moleküldefinition, Visualisierung oder bei der Datenbanksuche zu bewegen. Kommerzielle Softwarelösungen laufen innerhalb des Systems meist stabiler, stellen aber immer eine Insellösung dar, eine Black Box, die nicht direkt vollwertig in das modulare Synthesizer-Konzept überführt werden kann. Open-Source-Anwendungen, die oftmals für spezielle Anwendungsfälle konzipiert sind, besitzen in diesem Gebiet ihre zentralen Vorteile, weisen aber auch Schwächen auf, die von anderen Anwendungen besser administriert werden können. Open-Source Applikationen sind aufgrund ihrer oft begrenzten Projekt-Entwicklungszeit wesentlich fehleranfällig und instabiler, die Weiterentwicklung verläuft nicht kontinuierlich oder wird mit der Zeit sogar ganz eingestellt. Der Vorteil dieser Anwendungen besteht aber in der Möglichkeit einer internen Weiterentwicklung, da sie eben keine „Black Box“-Systeme darstellen. Für die Synthesizer-Applikation werden die Open-Source Anwendungen erst nach Praktikabilitäts- und Stabilitätstests eingebunden, um ihre Nützlichkeit im praktischen Workflow zu untersuchen. Da diese Werkzeuge jedoch nur Übergabestelle zur eigentlichen Hardwaresteuerung des Synthesizers darstellen, sind ihre möglichen Stabilitätsprobleme nicht systemrelevant bei der Durchführung der eigentlichen Synthese. Während der Entwicklungsphase ist die Fusionierung frei verfügbarer externer Funktionalität innerhalb der Synthesizer-Applikation ein wichtiger Prozess-Schritt, da er als technologischer Hebel und als Multiplikator innovativer Ideen, die Realisierungszeit erheblich verkürzen kann.

Für die Eingabeoberfläche des Glykan-Editors eignen sich Tools wie DrawGlycan-SNFG^{112[298]}, SugarSketcher^{113[299]}, einer intuitiven Online-Glykan-Zeichenumgebung, die aufgrund des JavaScript Interface direkt in die Web-Oberfläche der Automatisierungssysteme implementiert werden kann oder der web-basierte GlycoViewer^{114[300]} zur Visualisierung, Zusammenfassung und Vergleichen von Glykan-Strukturen. Weiterführend dienen diese Werkzeuge strukturell für hierarchisch übergeordnete Ansichten von Glykanen aus Proteinen, Zellen, Gewebe bis hin zu einem vollständigen Organismus.

Um dem dreidimensionalen Charakter der Glykane gerecht zu werden, sind die 2D Grafiken auf 3D-SNFG^[301] erweitert worden. Dadurch wird es auch für Nichtchemiker leichter sich die räumliche Ausrichtung der Moleküle zu veranschaulichen.

Bei bioaktiven Molekülen ist die Ausrichtung der Struktur für die Funktion von entscheidender Bedeutung, da damit ihre Wirkungsweise verständlich gemacht werden kann. Zur schematischen dreidimensionalen Darstellung von Glykanen und deren Wechselwirkungen eignen sich u. a.

¹¹² Online unter: <http://www.virtualglycome.org/DrawGlycan/> [letzter Abruf: 07.02.2019]

¹¹³ Online unter: <https://glycoproteome.expasy.org/sugarsketcher/> [letzter Abruf: 06.02.2019]

¹¹⁴ Online unter: <http://www.glycoviewer.babs.unsw.edu.au/> [letzter Abruf: 10.02.2019]

Werkzeuge wie GlycoBlocks^[295], POLYS^[302] oder SweetUnityMol^{[115][303]}. Die Einbindung von CarbBuilder^[304] ermöglicht die direkte Visualisierung dreidimensionaler Strukturen der Zielmoleküle, die wiederum beim Synthesedesign Verwendung finden können.

Auch der Zugriff auf die in Glyko-Datenbanken (GlycoStore^[116]) hinterlegten Glykanstrukturen wird durch die Verwendung der SNFG-Symbolik in der Synthesizersoftware erheblich vereinfacht. Aktuelle Ressourcen der Glykobioinformatik umfassen Informationen über Struktur und Funktion von Glykanen, ihre Wechselwirkung mit Proteinen oder ihre enzymatische Synthese.

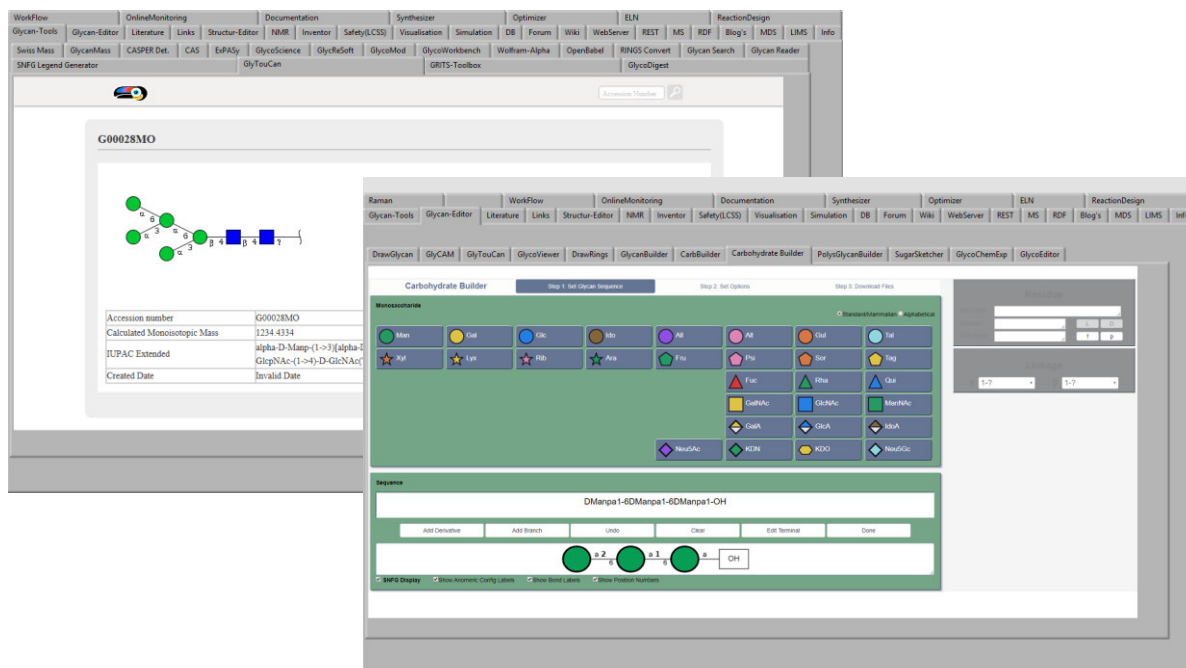


Abbildung 213: Glykan-Editor für den Synthesizer: Für das „Proof of Konzept“ des zukünftigen Molekülsynthesizers wurden verschiedene Open-Source Glykan-Editoren in die Synthesizer-Applikation implementiert: a) GlyYouCan, b) Carbohydrate Builder. Die innerhalb des Glykan-Editors gezeichneten Glykane können als Syntheseanforderung dem Molekülsynthesizer übergeben werden.

Für die Automatisierung der Strukturanalyse von Glykanen sind verschiedene Software-Werkzeuge in der Entwicklung^[305,306], wie z. B. den Glycan Reader^[307], der Zuckertypen und chemischen Modifikationen in Proteindatenbank-Strukturdateien erkennt. Aufgrund der Verknüpfung mit CHARMM-GUI^[117] können hiermit molekulare Simulationssysteme für Kohlenhydrate oder Glykoproteine in Lösungs- oder Membranumgebungen generiert und das elektrostatische Potenzial auf Glykoproteinoberflächen visualisiert werden, um Auswirkungen der Glykosylierung auf die Proteinstruktur und –dynamik zu untersuchen.

Das Expertensystem bleibt „abwärtskompatibel“, d. h. auch „alte“ Glykanbeschreibungen können in der Oberfläche des Expertensystems verwendet werden. Damit bleibt weiterhin der Zugriff auf alle

¹¹⁵ Online unter: <https://glycopedia.eu/resources/sweet-unity-mol-3d-visualization/article/presentation> [letzter Abruf: 11.02.2019]

¹¹⁶ Online unter: <https://glycostore.org/> [letzter Abruf: 06.02.2019]

¹¹⁷ Online unter: <http://www.charmm-gui.org/> [letzter Abruf: 11.02.2019]

bislang erfassten Daten und Informationen bestehen, was einen kontinuierlichen Aufbau der Synthesizer-Expertendatenbank ermöglicht.

5.3.3 Informationsverarbeitung und Wissensmanagement mit ELN und LIMS

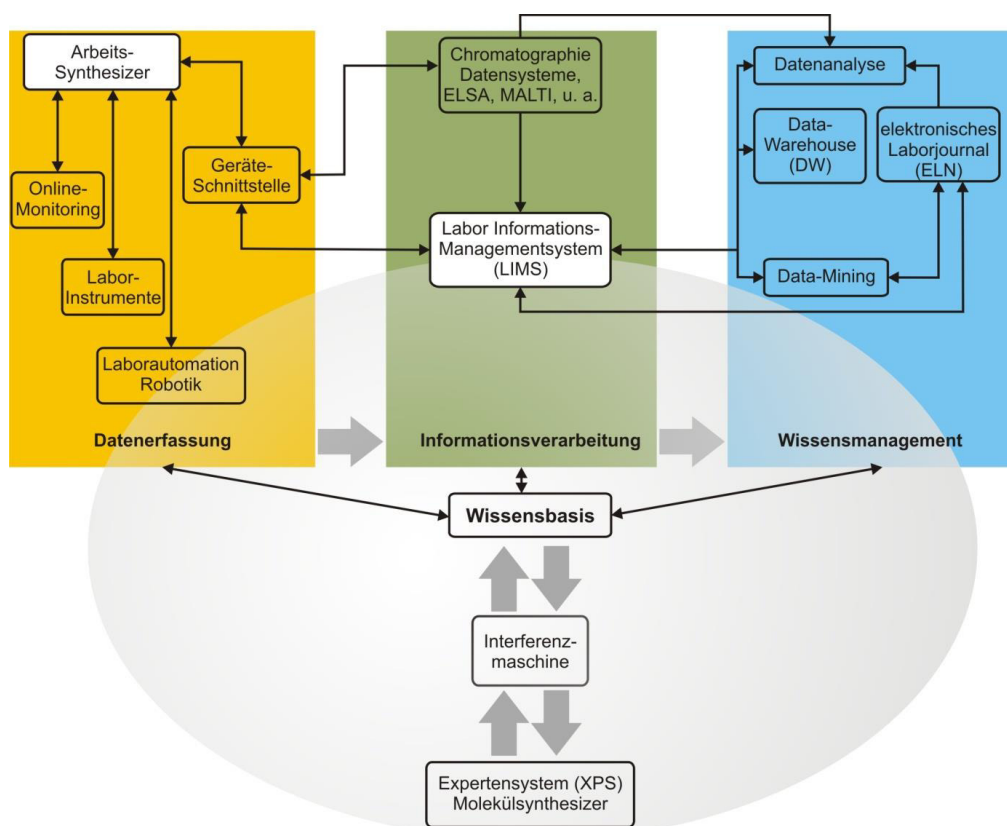


Abbildung 214: Das Labor Informations-Managementsystem (LIMS) im Kontext des Expertensystems

Es besteht eine große Notwendigkeit zur standardisierten Berichterstattung von Metadaten, welches die Wahl des experimentellen Verfahrens, der Materialien und Methoden betrifft, sowie Informationen, die sich auf die chemischen Analyseaspekte einschließlich Probenvorbereitung, experimentelle Analyse, Qualitätskontrolle und Datenvorverarbeitung beziehen. Leider gehen bei vielen Forschungsprojekten nicht publizierte Informationen nach Projektabschluss verloren, da Daten weder allgemeingültig noch wiederverwendbar hinterlegt sind. Das Verknüpfen und Abrufen experimenteller Details und Ergebnisse gestaltet sich damit problematisch und gefährdet die Reproduzierbarkeit wissenschaftlicher Erkenntnisse. Zunehmend beschäftigen sich deshalb aktuelle Studien mit Untersuchungen zur Aufbewahrung und Kuratieren elektronischer Datensätze in elektronischen Labor-Notizbüchern (ELNs)^[308], Open Enventory^[309], sowie der Entwicklung von ELN-Infrastruktur, z.B. Chemotion^{118, [310]} und anderer OpenSource-Software^[311] wie dem in Life-Science-Laboren verbreiteten openBIS^{119, [312]}. Über Mindeststandards bei der Beschreibung von

¹¹⁸ Chemotion-Projekt, Karlsruhe Institute of Technology, online unter <https://www.chemotion.net/> [Abruf: 13.12.2018]

¹¹⁹ Abkürzung für: **Open Biology Information System**;

Online verfügbar unter: <https://wiki-bsse.ethz.ch/display/bis/Home> [letzter Abruf 04.03.2019]

Metabolomics und funktioneller Genomik Datensätzen bezogen auf Daten der Massenspektrometrie (MS) und Kernresonanzspektroskopie (NMR) wird diskutiert.^[313] Unter MIAPE^{120[314]} entwickelt die Proteomics - Initiative der Human Proteom Organization¹²¹ Richtlinienmodule für die standardisierte Berichterstattung von Proteomics-Daten unter Verwendung von Techniken wie Gelelektrophorese und Massenspektrometrie. Seit 2015 wird die standardisierte Verwaltung von Forschungsressourcen für biomedizinische Publikationen in einem Pilotprojekt^{122[315]} getestet, welches nach vorläufigen Erkenntnissen die Reproduzierbarkeit der Forschungsressourcen erheblich verbessert hat. Glykomics-Daten werden unter Verwendung verschiedenster Methoden generiert, darunter MS, Chromatographie, Glykan-Array-Bindungsassays, NMR und andere sich rasch entwickelnder Technologien. Zur Zeit wird mit MIRAGE^{123[316,317]} eine dem MIAPE angelehnte Struktur erarbeitet, um die Mindestanforderungen zur Berichterstattung für ein Glykomics – Experiment zu definieren. 2013 wurden MIRAGE-Richtlinien für die Glykananalyse durch MS veröffentlicht^[316], 2016 für die Probenvorbereitung, inklusiver aller Aspekte der Probenerzeugung, -reinigung und -modifizierung aus biologischem und / oder synthetischem Kohlenhydratmaterial.^[318] Für die Übermittlung von MIRAGE-kompatiblen MS-Informationen in die UniCarb-DB¹²⁴, einem öffentlichen Versuchsarchiv, wurde kürzlich mit dem etablierten Glykomics-Aufzeichnungsformat GlycoWorkbench^{125[319,283]} eine Glykomics e-Infrastruktur vorgestellt.^[320]



Abbildung 215: Schrittkette für die Automatisierung von Aufgaben

Mit den FAIR¹²⁶- Prinzipien^[321] für die Verwaltung von Forschungsdaten bzgl. Auffindbarkeit, Zugänglichkeit, Interoperabilität und Wiederverwendbarkeit, werden aktuell Strukturen für die Automatisierung geschaffen, die bei einer breiten Basis von Interessengruppen im Life Science-Bereich Unterstützung finden.^[322] Wissenschaftliche Verlage spielen eine Rolle bei der Anforderung und Durchsetzung der gemeinsamen Nutzung von Daten. Für Biologen wurde deshalb aktuell eine Checkliste mit einigen Schlüsselementen veröffentlicht, die Autoren für ihre Publikation berücksichtigen sollten^[323] und auch für Wissenschaftler anderer Forschungsgebiete hilfreich ist, da Datenmanagementpläne nach dem FAIR-Prinzip inzwischen für immer mehr Förderprogramme

¹²⁰ Abkürzung für: „The minimum information about a proteomics experiment“

¹²¹ Online unter: <https://www.hupo.org/> [letzter Abruf: 15.01.2019]

¹²² Online unter: <https://scicrunch.org/resources> [letzter Abruf 10.01.2019]

¹²³ Abkürzung für das Projekt „Minimum Information Required for A Glycomics Experiment“ des Beilstein Institutes und der Universität Georgia; Online Informationen unter: <https://www.beilstein-institut.de/en/projects/mirage> [letzter Abruf: 09.01.2019]

¹²⁴ Online unter: <http://unicarb-db.expasy.org/> [letzter Abruf: 16.01.2019]

¹²⁵ Freier Download unter: http://de.download.cnet.com/GlycoWorkbench-64-bit/3000-2383_4-75758804.html [letzter Abruf: 16.01.2019]

¹²⁶ **FAIR** steht für Findable, Accessible, Interoperable, Reusable

verpflichtend geworden sind. Das standardisierte Verwalten von Daten, stellt nur den ersten wichtigen Schritt in der Schrittkette zur Automatisierung von Prozessen dar, und wird im Kapitel Datenarchivierung vertieft.

In der vorliegenden Konzeptanalyse wurden die möglichen Synergieeffekte bei der Implementierung eines ELN Standards in das Synthesizer-Konzept evaluiert. Mit LabVIEW wurde auf die Objekte der Software E-Notebook™ von PerkinElmer® zugegriffen, um ihre potentielle Verwendung innerhalb eines Synthese-Workflows zu testen. Dabei ist ChemDraw™ eine im E-Notebook eingebettete Komponente, welches die Erstellung von Reaktionsschemata mit Ausgangsmaterialien, Katalysatoren, Reagenzien und Produkten ermöglicht und sich deshalb auch als fortschrittliche Benutzeroberfläche für die zukünftige Synthesesteuerung eignet (vgl. Kapitel 5.3.2.3, S. 216). Die Reaktionsgleichung wurde in der Konzeptanalyse über benutzerdefinierte Autotext-Elemente umgesetzt (Abbildung 216). Dies ermöglicht bereits bei der Syntheseplanung eine Bedeutungszuordnung der einzelnen Elemente für die Verwendung als Informationsbasis einer automatischen Parametrierung von Synthesen am Synthesizer. Alternativ kann die Zuordnung der verschiedenen relevanten Objekte auch über das Kontextmenü des E-Notebooks erfolgen.

The screenshot displays the E-Notebook software interface for a chemical reaction. The main window shows a reaction scheme where a resin with Fmoc and BnO protecting groups reacts with TMSOTf and Et3N in DMF, followed by photocleavage to yield a resin with a hydroxyl group and a BnO group. The reaction conditions are listed as: 1) 6 times: a) 5 eq Donor, TMSOTf; b) 5 eq Donor, TMSOTf; c) Et3N, DMF; 2) Photocleavage. Below the reaction scheme, the 'Reactants & Products' table is visible:

Reactant	MF	Limit?	MW	Eq	Moles
1 (2S,3R,4S,5R)-5-(((9H-fluoren-9-yl)methoxy)carbonyloxy)-4-(benzyloxy)-2-((dibutoxyphosphoryl)oxy)tetrahydro-2H-pyran-3-yl benzoate	C42H47O11P	<input type="checkbox"/>	758,790	5	62,5 µ
2 Resin		<input checked="" type="checkbox"/>		1,000	12,5 µ

The 'Preparation' section includes the following steps:

- Glycosylation: -40 (5 min) -30 (15 min) to -15 (25 min)
- TMSOTf: Dissolve Trimethylsilyl Trifluoromethanesulfonate (900 µL) in anhydrous DCM (40 mL) and purge the resulting solution with Ar for 1-2 minutes. (125 mM TMS-OTf)
- Fmoc deprotection: Dissolve Et3N (20 mL) in anhydrous DMF (80 mL) and briefly purge the solution with Ar.

The 'Solvents' table is also visible:

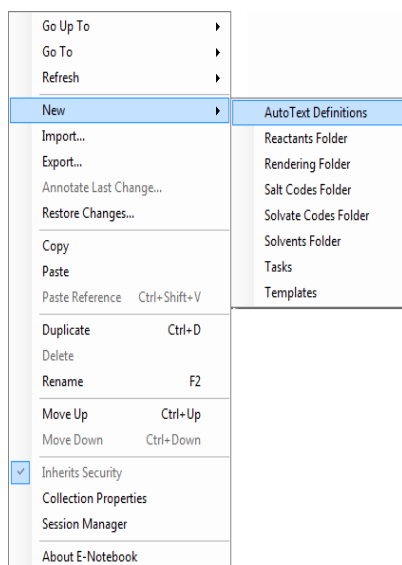
Name	Ratio	Volume

The 'Reaction Conditions' table is also visible:

Reaction Molarity
Pressure
Temperature
Reaction Time

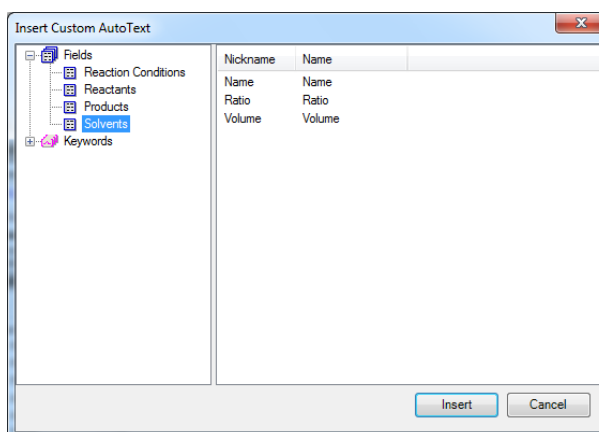
Abbildung 216: Syntheseplanung im E-Notebook

Durch den programmatischen Zugriff auf diese Informationen ergibt sich die Möglichkeit der Standardisierung und weitgehenden Automatisierung der kompletten Synthese-Protokollierung. Die Daten werden in der Synthesizer-Datenbank unter Felddefinitionen gespeichert und stehen damit vollwertig für Datenbank- Abfragen zur Verfügung. Die dazu notwendige Definition erfolgt direkt in E-Notebook als AutoText Definition.



Kontextmenü

Text – Definition:



Parametrierung:

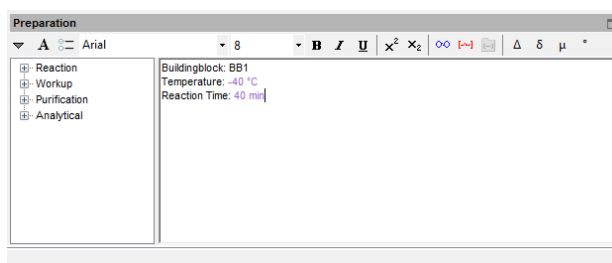


Abbildung 217: Menü E-Notebook: Eingabe von Zusatzinformationen (Metadaten) im E-Notebook unter AutoText Definitions

Das Ergebnis einer automatisiert durchgeführten Synthese sollten Datenstrukturen sein, die nach dem Standard des OAIS-Referenzmodells¹²⁷ hinterlegt sind. Die Definition neuer Synthesen, die Dokumentation aktuell durchgeführter Synthesen, sowie die Suche nach übergreifenden Informationen, kann damit direkt als Funktionalität bereitgestellt werden. Dies alles sind Aufgaben eines modernen elektronischen Laborbuchs.

Online stehen mit SciNote¹²⁸, Benchling¹²⁹, RSpace¹³⁰ und Labarchives¹³¹ integrative Plattformen des elektronischen Labor-Notizbuchs zur Verfügung. Sinnvoller Weise sollten die Daten hier in einem weitestgehend standardisiertem Format hinterlegt werden, damit auch andere Forschungsgruppen ohne Schnittstellenverluste darauf zugreifen können.

¹²⁷ Abkürzung für Open Archival Information System; Referenzmodell eines offenen Archiv-Informationssystem; Online-Spezifikation in deutscher Übersetzung:

<https://nbn-resolving.org/urn/resolver.pl?urn=nbn:de:0008-2013082706> [letzter Abruf: 09.01.2019]

¹²⁸ Online unter: <https://scinote.net/> [letzter Abruf: 04.03.2019]

¹²⁹ Online unter: <https://benchling.com/> [letzter Abruf: 04.03.2019]

¹³⁰ Online unter: <https://www.researchspace.com/> [letzter Abruf: 04.03.2019]

¹³¹ Online unter: <https://www.labarchives.com/> [letzter Abruf: 04.03.2019]

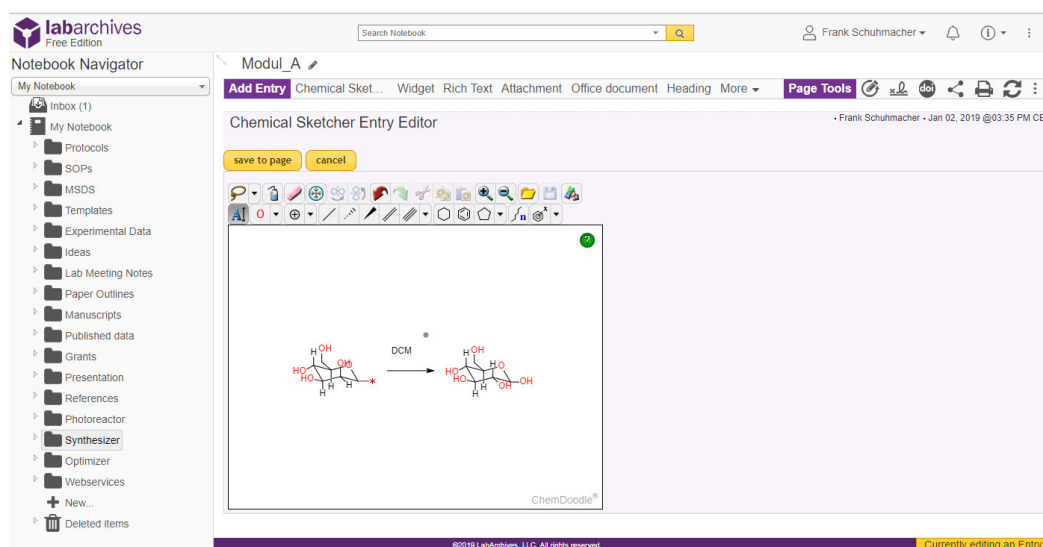


Abbildung 218: Labarchives als Beispiel einer plattformübergreifenden Labor-Notebooks

Gerade für forschungsgruppenübergreifende Datenhaltung bieten sich die Server der jeweiligen Institutionen an. Für die FU Berlin steht mit ReFUBium¹³² hier eine Austauschplattform für Publikationen und Forschungsdaten bereit, in dem Metadaten frei recherchiert werden können. Vergleichbares ist bei der Max-Planck-Gesellschaft über die „digitale library“¹³³ und als Tool zur verteilten Zusammenarbeit <<collective>>¹³⁴ auf Basis von Open Atrium¹³⁵ verfügbar. Zu Projektbeginn wurde unter [workspaz] eine Arbeitsplattform bereitgestellt, mit welcher eine Vielzahl von Interaktionen von Forschern angetestet wurde. Aus dem 2014 aufgesetzten [workspaz] wurde während der Projektlaufzeit <<collective>>. Durch die Verwendung von Open Atrium sind bereits Funktionen wie Blogs, Wikis, Kalender, Aufgabenlisten Chats und Übersichtsseiten integrierbar.

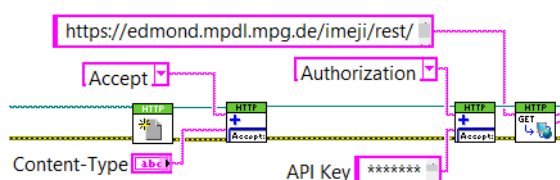


Abbildung 219: LabVIEW-VI zum Zugriff auf das Open Access Data Repository Edmond

Die Max Planck Digital Library stellt mit EDMOND eine Open Access Data Repository¹³⁶ auf der Basis von imeji/Open Atrium bereit. Hier können alle Fachbereiche ihre wissenschaftlichen Daten hosten und austauschen, vergleichbar zum [workspaz] Service, der als Vorläufer zur dieser Webservice-Anwendung verwendet wurde. Durch das Bereitstellen auf dieser Plattform werden die Daten zu

¹³² Online unter: <https://refubium.fu-berlin.de/> [letzter Abruf: 09.01.2019]

¹³³ Online unter: <https://www.mpdl.mpg.de/> [letzter Abruf: 09.01.2019]

¹³⁴ Online unter: <https://collective.mpdl.mpg.de/> [letzter Abruf: 09.01.2019]

¹³⁵ Open-Source-Software basierend auf Drupal. Online unter: <https://www.drupal.org/project/openatrium> [letzter Abruf 09.01.2019]

¹³⁶ Daten-Repository ist in ein allgemeiner Begriff für eine zielbezogene Datenspeicherung z. B. in einer Gruppe von Datenbanken zur Datenaufbewahrung; wird auch als Data-Warehousing (DW) bezeichnet.

zitierbaren Objekten. Da zu allen Daten umfangreiche Metainformationen verknüpfbar sind, können umfangreiche Suchen und Auswertungen realisiert werden.

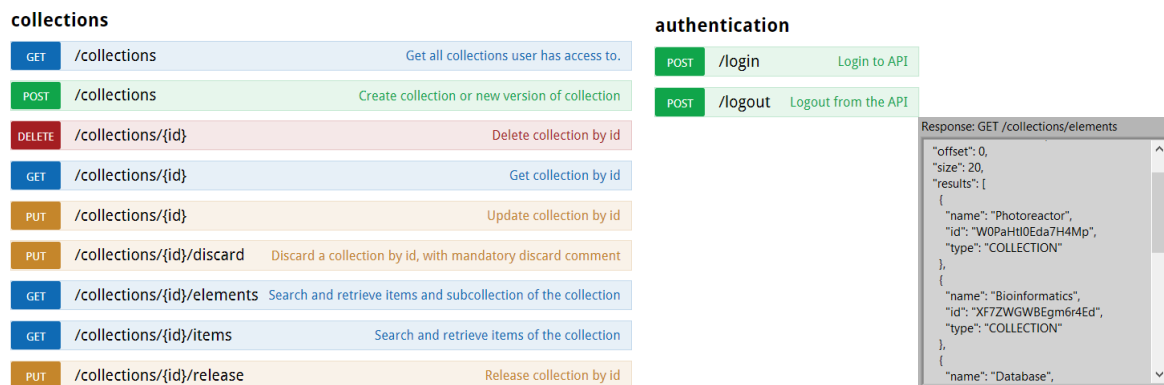


Abbildung 220: API-Funktionen des Open Access Data Repositorys EDMOND und Ausgabe einer GET – Anfrage in LabVIEW (rechts).

Die Informationen der jeweiligen Eintragungen im ELN stehen programmatisch über REST-API¹³⁷ Funktionen im Zugriff, die als etabliertes Mittel bei der Realisierung verteilter Systeme im Kontext von Cloud Computing¹³⁸, Internet of Things¹³⁹ und Microservices¹⁴⁰ zukünftig noch größere Bedeutung gewinnen werden. Damit können die Informationen auch aus alternativen ELN's, bei vollständig ausgefüllten Protokollen, zur Ansteuerung der zukünftigen Synthesizer-Systeme verwendet werden. Leider sind viele vorhandene REST-APIs mangelhaft dokumentiert oder nicht qualitativ gut entwickelt. Mit SpyREST¹⁴¹ wurde deshalb Technik entwickelt, welche die Dokumentation der API Funktionen automatisch generiert und somit den manuellen Prozess der Dokumentationswartung durch einem automatisierten Prozess ersetzt.^[324] Zur Überprüfung der strukturellen Merkmale hinsichtlich der Einhaltung der REST-Architektur wurde kürzlich ein Validierungsansatz vorgestellt.^[325] Damit lassen sich mit Hilfe des REST/API-Frameworks, wie Swagger¹⁴² oder SpyREST, die Webservices^[326] erschließen, um diese im Anschluss fehlerfrei in LabVIEW für die neue Synthesizer-Software implementieren zu können. Seit 2016 liegen die Swagger Spezifikationen als Open-API Spezifikationen auf GitHub¹⁴³ bereit (Abbildung 221). Die Funktionalitäten des zukünftigen Synthesizer-Systems sollten sich an diese Spezifikationen halten,

¹³⁷Abkürzung für REpresentational State Transfer - Application Programming Interface; Programmierschnittstelle zur Kommunikation zwischen Client und Server in Netzwerken

¹³⁸ Dynamische Bereitstellung von Anwendungen auf Abruf und entsprechenden IT-Ressourcen über das Internet bzw. Netzwerken

¹³⁹ Bezeichnet die Vernetzung physischer und virtueller Gegenstände durch Informations- und Kommunikationstechniken

¹⁴⁰ Bezeichnet kleine entkoppelte Prozesse, die einzelne Funktionen für eine Anwendungssoftware bereitstellen.

¹⁴¹ Online unter: <http://www.spyrest.com/> [letzter Abruf: 17.01.2019]

¹⁴² Open-Source-Software-Framework zum Entwurf, Erstellung, Dokumentation und Nutzung von RESTful Webservices. Online unter: <https://swagger.io/> [letzter Abruf: 17.01.2019]

¹⁴³ Online unter: <https://github.com/swagger-api/swagger-ui> [letzter Abruf: 10.01.2019]

um einen weltweiten Austausch zu ermöglichen. Die dadurch vorhandene Client Software Development Kit's (SDK's)¹⁴⁴ vereinfacht die Implementierung für den Anwender erheblich.

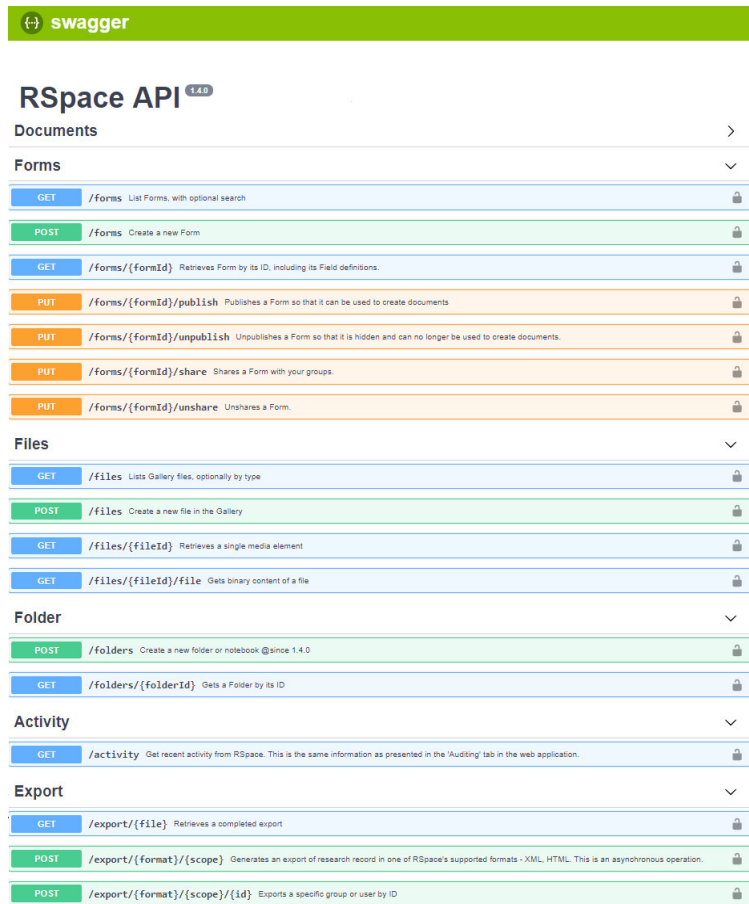


Abbildung 221: Rspace API Funktions-Referenz für das Framework Swagger

Durch die Verwendung von Wikidata Quickstatements wird die automatisierte Aktualisierung von Informationen wesentlich vereinfacht. Mit Hilfe der Skriptsprache können die Daten direkt funktionalen Elementen automatisiert zugewiesen werden. Da prinzipiell alle browserfähige Applikationen den webbasierten Zugriff ermöglichen, ist die automatisierte Editierung von den unterschiedlichsten Plattformen und Anwendern möglich. Somit kann ein schnell wachsendes, jedoch strukturiert bereitgestelltes Datenmanagement erreicht werden. In vielen Anwendungsfällen wird mit JavaScript Object Notation (JSON) eine allgemeingültige Schnittstelle bereitgestellt (Abbildung 222).

¹⁴⁴ SDK bezeichnet eine Sammlung von Programmierwerkzeugen und Programmibliotheken zur Entwicklung von Software.

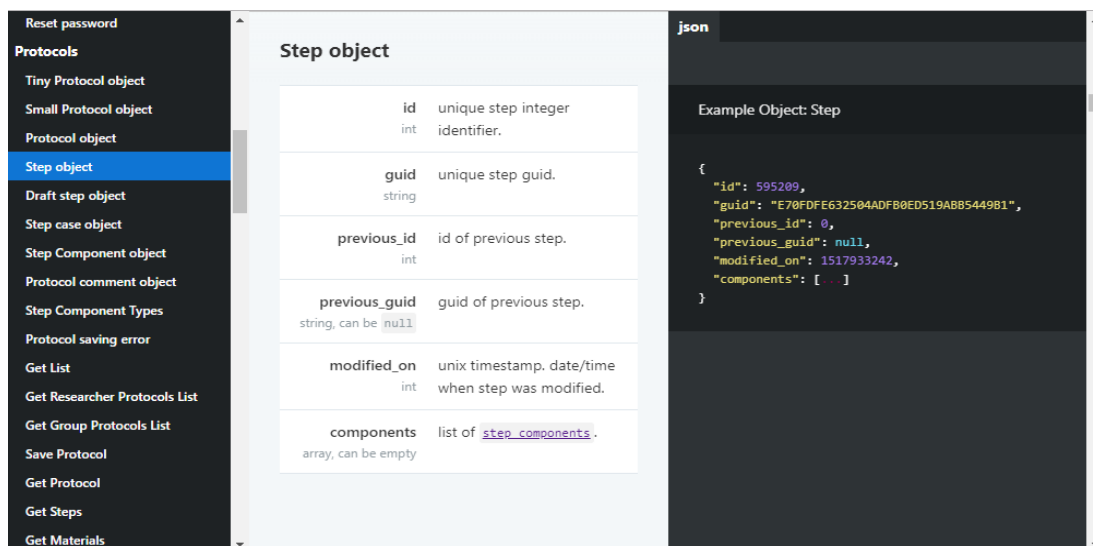


Abbildung 222: API-Objektreferenz für JavaScript Object Notation (JSON), einem kompakten Datenformat in einfach lesbarer Textform für den Datenaustausch zwischen Anwendungen.

Über das elektronische Labor-Notizbuch hinaus ermöglicht ein Labormanagement- und Informationssystem (LIMS) auch die Verwaltung von vorhandenen Substanzen und erhaltenen Produkten. Das LIMS kann den Chemiker kontinuierlich bei der Umsetzung der angestrebten automatisierten Synthese begleiten. Durch Unterstützung bei den verschiedenen Arbeitsschritten und Entlastung von den parallel anfallenden Standardaufgaben wie Recherche, Bestellung, Protokollierung, Beschriftungen und Datenspeicherung, kann sich der Forscher exakt auf die Synthesedurchführung konzentrieren. Ein LIMS sollte sich hierbei möglichst ohne Einschränkungen in den Laboralltag integrieren lassen und umfangreiche Schnittstellen für zukünftige Erweiterungen wie die Chemometrie bereithalten. Wendl et al. hat für die Bioinformatik 2007 ein allgemeines Modell zur Verarbeitung von Labordaten und deren Implementierung als Informationsmanagementsystem vorgestellt, welches sich auch auf das chemische Forschungsumfeld übertragen lässt. Das Modell umfasst viele aktuelle Abstraktionstechniken wie Vererbung und Metadaten, sowie verschiedene reguläre Entitäts¹⁴⁵- und Ereignisschemas, die vollständig über eine standardisierte Schnittstelle integriert werden.^[327]

Ein funktionales LIMS sollte als verteiltes System realisiert werden, damit an allen notwendigen Informationspunkten im Labor eine Datenübergabe und Auskopplung ohne überdimensionalen Aufwand möglich wird. Gerade im Bereich der Cloudanwendungen und lokaler Mesh-Netze ist hier umfangreiches Potential für den Labor-Workflow vorhanden. Um die bereits vorhandenen Geräte und Analyse-Systeme in diese Welt integrieren zu können werden Adapter benötigt. Diese können kostengünstig auf Basis von Makerboards (Raspberry, Arduino, Edison) realisiert werden. Bei einer

¹⁴⁵ Entität = Informationsobjekt, bezeichnet in der Datenmodellierung ein eindeutig zu bestimmendes Objekt (materiell oder immateriell, konkret oder abstrakt), über das Informationen gespeichert oder verarbeitet werden soll.

Vielzahl von Systemen fallen binäre Daten an, die heute, wenn überhaupt, lokal gespeichert werden. Oft sind nur die Endergebnisse in Form von *.csv/*.pdf-Dateien archiviert und stehen somit nur eingeschränkt einem umfangreichen automatisierten Analyse-Verwertungsprozess zur Verfügung. In solchen Fällen sollten die Informationen mit zusätzlichen EDV-technisch nutzbaren Metadaten ins LIMS eingepflegt werden.

5.3.4 Datenarchivierung & Datenmanagement

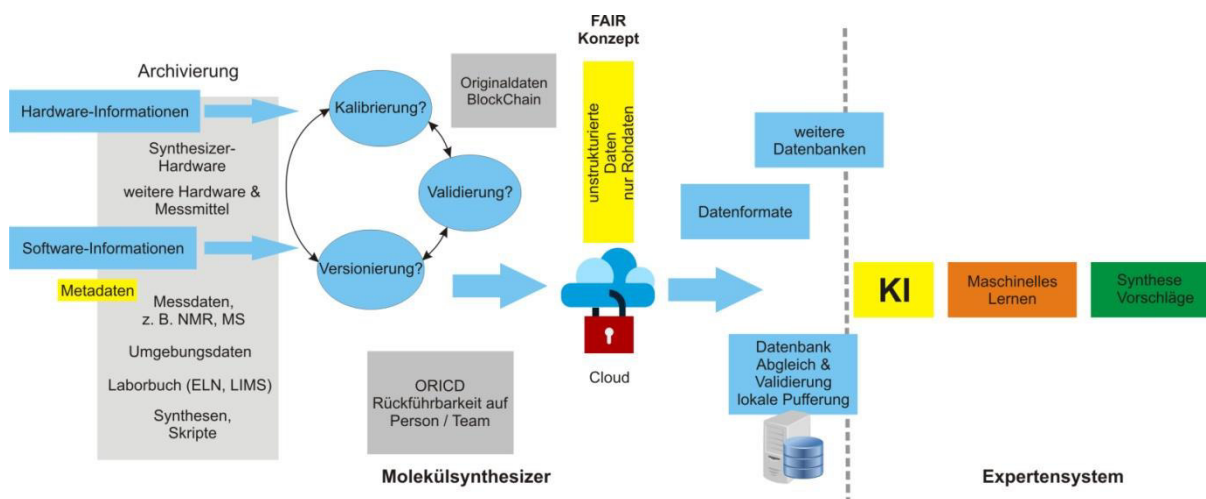


Abbildung 223: Konzeptüberblick des Themenkomplexes Datenarchivierung, Datenformate und Datenbanken vom Molekülsynthesizer bis zum Expertensystem.

Archivierung und Verwaltung elektronischer Daten ist bei Automatisierungsprozessen neben der Standardisierung ein fundamentales Thema. Bei der chemischen Synthese am Synthesizer fallen zahlreiche Datensätze in vielen verschiedenen Arbeitsschritten an. Der zukünftige Molekülsynthesizer wird durch zusätzliche Datenerfassungskomponenten, seiner Vernetzung mit Labormessgeräten, sowie der Anbindung an ein elektronisches Laborbuch bzw. Labor-Management-System, eine noch weit umfangreichere Datenmenge verwalten als bisher. Diese Datensammlung ist essenziell für technologische Weiterentwicklungen und nachgelagerte Prozessoptimierungen, aber inzwischen auch Thema für den Zugang zu Forschungsgeldern im Kontext eines qualitativen Forschungsdaten-Managements (vgl. DFG: „Leitlinien für den Umgang mit Forschungsdaten“¹⁴⁶). Die FAIR^{147[328]}-Leitprinzipien für die Verwaltung von Forschungsdaten: Auffindbarkeit, Zugänglichkeit, Interoperabilität und Wiederverwendbarkeit, dürften zu einem Eckpfeiler der Forschung in den Biowissenschaften werden^[329], und umfassen dabei nur einen kleinen Teil der Anforderungen an ein vollwertiges Qualitätsmanagementsystem nach der DIN EN ISO 9000:2015-11¹⁴⁸. Die grundlegenden

¹⁴⁶ Online unter:

https://www.dfg.de/download/pdf/foerderung/antragstellung/forschungsdaten/richtlinien_forschungsdaten.pdf
[letzter Abruf: 13.02.2019]

¹⁴⁷ Akronym für engl.: Findable, Accessible, Interoperable and Reusable

¹⁴⁸ Kostenpflichtiger Download online unter:

<https://www.din.de/de/mitwirken/normenausschuesse/nqsz/wdc-beuth.din21:235671251> [letzter Abruf: 13.02.2019]

Prinzipien der ISO 9000 für die lineare Rückführbarkeit von Daten, dem systemkonformen kontinuierlichen Verbesserungsprozess, welches die Grundvoraussetzung zur Realisation selbststabilisierender Systeme darstellt, und die Dokumentationsnotwendigkeit an allen systemkritischen Stellen, sei es hardware-, software- oder chemieseitig, sind in der Umsetzung für den Molekülsynthesizer und in Hinblick auf die Weiterentwicklung zum Expertensystem strikt umzusetzen.

Gleiche Bedingungen müssen bei einem beherrschten automatischen Prozess immer zu gleichen Ergebnissen führen. Dies gilt auch für die Validierbarkeit von Forschungsergebnissen, die durch eine eindeutige Variante der Reproduzierbarkeit überhaupt erst möglich ist. Deshalb sind jegliche Veränderungen im Prozess, sei es durch den Benutzer oder durch Änderung am Arbeitsmittel für jede Synthese oder Messung zu archivieren. Dabei sind auch die Kalibrierdaten der jeweiligen Messmittel zu protokollieren und auf Validität zu prüfen. Softwareveränderungen, auch von externen Datenquellen, sind zu versionieren, um bei iterativen Verfahren die Rückführbarkeit zu gewährleisten. Hierfür ist der Einsatz eines Versionskontrollprogramm sinnvoll, wie z. B. Git¹⁴⁹, eine freie Software zur verteilten Versionskontrolle, welches auf der Online-Plattform GitHub¹⁵⁰ Projekte in Repositorien¹⁵¹ organisiert und bereits innerhalb der Chemo- und Bioinformatik^[330] Verbreitung findet. Die Protokollierung jeder Weiterverarbeitung der Originaldaten mittels Blockchain-Konzept^[331] ist die komplexere, aber weitaus umfassendere Lösung mit hohem technologischen Potential, welche eine 100%ig gesicherte Rückführbarkeit der Daten gewährleistet. Bei diesem Konzept führt jede Veränderung im Prozess zu einem Datensatz, im Konzept als „Block“ bezeichnet. Die Datensätze werden mittels kryptographischer Verfahren in einer chronologisch linear aufgebauten Datenbank miteinander verkettet. Jeder Block beinhaltet eine Prüfsumme des vorhergehenden Blocks, sowie Zeitstempel und Transaktionsinformationen. Mit diesem Konzept lassen sich große Datenmengen mittels Verschlüsselung und Zugriffsverwaltung schützen, dezentral sammeln und analysieren, Datenbezugspunkte verifizieren und automatisch Schwachstellen in der Prozesskette identifizieren. Deshalb ist die Blockchain nicht nur ein großes Thema im Finanzsektor, sondern auch zunehmend für dezentrale, sicherheitsrelevante Entwicklungen im IT-Sektor und somit auch für komplexe Automatisierungs-Systeme im chemischen Umfeld.

Zur Vermeidung von Datenverlust sind sinnvolle Backup- Mechanismen, wie Spiegelungen, verteilte Synchronisationen und Cloudspeicher, im Vorfeld zu planen. Da die Originaldaten langfristig über mindestens 10 Jahren hinaus bereit gehalten werden sollten (vgl. „DFG: Denkschrift zur Sicherung

¹⁴⁹ Online verfügbar unter <https://git-scm.com/> [letzter Abruf: 18.02.2019]

¹⁵⁰ Online unter <https://github.com/> [letzter Abruf: 18.02.2019]

¹⁵¹ Verwaltetes Verzeichnis zur Speicherung und Beschreibung von digitalen Objekten für ein digitales Archiv.

guter wissenschaftlichen Praxis“^[332]), sind hier dynamische Erweiterungsmöglichkeiten der Speicherkapazität vorzusehen.

Zur Optimierung der automatisch archivierten Datenmengen, werden abgeschlossene Projekte am Molekülsynthesizer in komprimierter Form verwaltet. In offenen Datenbank-Konzepten kann der Zugriff auf diese Daten z. B. für eine Online Kooperation^[333] über einen Hyperlink erfolgen. Der Ansteuerungscode ist für die Reproduzierbarkeit der Synthesen an den Synthesizern von herausragender Bedeutung und, nach dem FAIR-Konzept für die Veröffentlichung in wissenschaftlichen Zeitschriften, Forschern und Entwicklern zur Verfügung zu stellen, z. B. über die cloudbasierte Plattform Codeocean¹⁵².

Die Implementierung plattformübergreifender Strukturen eignet sich für die Automatisierung, da dabei Wiederverwendbarkeit, Transparenz, Überprüfbarkeit und Reproduzierbarkeit im Vordergrund stehen. Gerade Open Source Repositorien, wie GitHub^[334], Zenodo¹⁵³, BitBucket¹⁵⁴, und kostenlose Cloud Speicher, z. B. Google Drive¹⁵⁵, OneDrive¹⁵⁶, DropBox¹⁵⁷, ermöglichen verteiltes Arbeiten, ohne dass der Anwender in einem Institutsaccount angemeldet sein muss. Der kommerzielle Datenspeicherdienst Figshare¹⁵⁸ vergibt individuelle digitale Objekt Identifier (DOI¹⁵⁹) dateispezifisch und bietet sich damit für die Veröffentlichung von Forschungsdaten an. Der Zugriff und die Verwaltung hierzu erfolgten über API Funktionen. Ähnliche Funktionalität steht mit EDMOND¹⁶⁰ auf Basis der Open Source Repository imeji¹⁶¹ innerhalb der Max-Planck-Gesellschaft bereit. Web-basierte Plattformen für die Teamarbeit wie Protocols.io¹⁶², Slack¹⁶³, Microsoft Team¹⁶⁴, RSpaceCommunity¹⁶⁵ und ResearchGate¹⁶⁶ vereinen die Möglichkeiten der wissenschaftlich-übergreifende Zusammenarbeit von Arbeitsgruppen, aber auch von Instituten und Forschungseinrichtungen weltweit.

Chemoinformatik-Webdienste erleichtern den Zugriff auf die digitalen, chemischen Informationen und bieten Techniken zur Organisation und intelligenten Informationsnutzung.^[335]

Durch die Zuweisung von Metadaten, insbesondere logischer Relationen, kann die

¹⁵² Online unter: <https://codeocean.com/> [letzter Abruf: 18.02.2019]

¹⁵³ Online unter: <https://zenodo.org/> [letzter Abruf: 18.02.2019]

¹⁵⁴ Online unter: <https://bitbucket.org/> [letzter Abruf: 18.02.2019]

¹⁵⁵ Online unter: <https://www.google.com/drive/> [letzter Abruf: 18.02.2019]

¹⁵⁶ Online unter: <https://onedrive.live.com/about/de-de/> [letzter Abruf: 18.02.2019]

¹⁵⁷ Online unter: <https://www.dropbox.com/de/> [letzter Abruf: 18.02.2019]

¹⁵⁸ Online unter: <https://figshare.com/> [letzter Abruf: 18.02.2019]

¹⁵⁹ Der Digitale Objekt Identifier (DOI) nach ISO 26324 ist ein eindeutiger, dauerhafter Identifikator für physische, digitale oder abstrakte Objekte, hauptsächlich verwendet für Online-Artikel wissenschaftlicher Fachzeitschriften.

¹⁶⁰ Online unter: <https://edmond.mpdl.mpg.de/imeji/> [letzter Abruf: 18.02.2019]

¹⁶¹ Online unter: <https://github.com/imeji-community/imeji> [letzter Abruf: 18.02.2019]

¹⁶² Online unter: <https://www.protocols.io/> [letzter Abruf: 18.02.2019]

¹⁶³ Online unter: <https://slack.com/intl/de-de/> [letzter Abruf: 18.02.2019]

¹⁶⁴ Kommerzielle Software für Teamarbeit unter Microsoft Office 365

¹⁶⁵ Online unter: <https://community.researchspace.com> [letzter Abruf: 18.02.2019]

¹⁶⁶ Online unter: <https://www.researchgate.net/> [letzter Abruf: 18.02.2019]

bereichsübergreifende analytische Nutzung der Rohdaten gewährleistet werden, von denen das aufkommende chemische semantische Web¹⁶⁷ abhängt.^[336] Die Bedeutung der Informationen wird damit für Maschinen leichter auswertbar. Standard für die Beschreibung von Metadaten ist das Resource Description Framework¹⁶⁸ (RDF), für das von verschiedenen Forschungsbereichen bereits Mindestanforderungen und Empfehlungen vorliegen, z. B. von der HCLS¹⁶⁹ - Interessengruppe für Daten aus dem Gesundheitswesen und der Lebenswissenschaften^[337] oder von den Entwicklern der wichtigsten Glykomics-Datenbanken zur standardisierten Darstellung von Glykan-Strukturen und Annotationsinformationen.^[338,339]

Zur Beantwortung komplexer biologischer Abfragen und Wirkungsweisen molekularer Interaktionen, müssen Mechanismen zur analytischen Suche in den Synthesizer implementiert werden. Abfragesprachen wie SPARQL und Webservices nach W3C Standard¹⁷⁰ eignen sich zum individuellen Navigieren in den Datenquellen. Bio2RDF^[340] und Chem2Bio2RDF^[341] nutzen die Funktionalitäten der komplexen Abfragemöglichkeiten über SPARQL, um Verknüpfungen für Life-Science-Daten und Data-Mining¹⁷¹ von chemogenomischen und systembiologischen Daten herzustellen. Informationen aus verschiedenen Datenbanken wie UniCarbKB¹⁷², GlyTouCan¹⁷³ und JCGDB¹⁷⁴ können in einer einzigen SPARQL-Abfrage verknüpft werden, um übergreifend Zielinformationen zu erhalten. Proteomics-Daten lassen sich mit Glykomics-Daten verknüpfen, wodurch flexiblere Abfragen in diesen Bereichen möglich sind.^[338]

Mit der grafischen Darstellung der in Lebewesen ablaufenden chemischen Reaktionen und daran beteiligter chemischer Verbindungen, Enzyme, u.a. Einflussgrößen, sowie der Vernetzung der einzelnen Reaktionen miteinander, befassen sich eine Reihe autarker Entwicklungen, die biochemische Reaktionswege, sogenannte „Pathways“, innerhalb des „semantischen Web“-Konzepts umsetzen und deshalb miteinander interagieren können. Pathway – Datenbanken wie Reactome¹⁷⁵ und Pathway Commons¹⁷⁶ für biologische Wirkmechanismen, sowie KEGG¹⁷⁷ oder WikiPathWays¹⁷⁸, die auch die Komplexität der Wirkungsweisen von Glykanen übersichtlich abbilden,

¹⁶⁷ Das Semantic-Web wird auch als Web 3.0 bezeichnet; Standard zur Vereinfachung der Wissensintegration.

¹⁶⁸ sinngemäß „System zur Beschreibung von Ressourcen“

¹⁶⁹ Akronym für Health Care and Life Sciences;

Online Dokument: <https://www.w3.org/2011/05/HCLSIGUseCases> [letzter Abruf: 18.02.2019]

¹⁷⁰ Online unter: <https://www.w3.org/blog/hcls> [letzter Abruf: 15.02.2019]

¹⁷¹ Systematische automatische Analyse großer Datenmengen unter Anwendung statistischer Methoden zur Bestimmung von Regelmäßigkeiten, Gesetzmäßigkeiten und verborgener Zusammenhänge.

¹⁷² Online unter: <http://www.unicarbk.org/> [letzter Abruf: 19.02.2019]

¹⁷³ Online unter: <https://glytoucan.org/> [letzter Abruf: 19.02.2019]

¹⁷⁴ Online unter: https://jcgdb.jp/database_en.html [letzter Abruf: 19.02.2019]

¹⁷⁵ Online unter: <https://reactome.org/> [letzter Abruf: 19.02.2019]

¹⁷⁶ Online unter: <http://www.pathwaycommons.org/> [letzter Abruf: 19.02.2019]

¹⁷⁷ Online unter: <https://www.genome.jp/kegg/> [letzter Abruf: 19.02.2019]

¹⁷⁸ Online unter: <https://www.wikipathways.org/index.php/WikiPathways> [letzter Abruf: 19.02.2019]

vereinfachen das Verständnis komplexer Zusammenhänge erheblich und ermöglichen es effizient zu den Informationen zu navigieren. Damit ist die Interpretation der Daten vollständig von der Datenerfassung entkoppelt.

Mit der Einführung neuer RDF-Exporte kann Wikidata¹⁷⁹ als zentrale Datenverwaltungsplattform von Wikipedia¹⁸⁰ mit dem Linked Data Web¹⁸¹ verbunden werden.^[342] Wikidata stellt eine große, offene Wissensbasis mit vielen interessanten Anwendungen für die wissenschaftliche Arbeit bereit, wie Scholia^{182[343]}, einem Werkzeug zum Umgang mit wissenschaftlichen bibliographischen Informationen, oder WikiPathWays¹⁸³, mit einem enormen Content-Wachstum in den letzten Jahren. WikiPathWays erleichtert die plattformübergreifende Wissensintegration mithilfe von Anweisungen, die als Tripel aufgebaut sind und im Semantic Web über einen SPARQL¹⁸⁴-Endpunkt verfügbar sind.^[344] WikiPathWays-Tripels, in denen jedes Gen, Protein und Metabolit in einem bestimmten Pfad mit einem Standardsatz von Bezeichnern definiert wird, finden zusammen mit weiteren Ressourcen von Drittanbietern wie z. B. Open PHACTS¹⁸⁵, in verschiedenen Tools für die Wirkstoffentwicklung Verwendungen, um neue Verknüpfungen über verschiedene biologische Daten hinweg zu verknüpfen.^[345] Open PHACTS soll als eine Open Source - Entwicklungsplattform der pharmazeutischen Industrie und Wissenschaft helfen Hindernisse bei der Wirkstoffsuche abzubauen. Aktuell wurden der Open PHACTS API neue Methoden in Form von WikiPathWays - Daten hinzugefügt, die Informationen über biologische Wechselwirkungen wie Stimulation und Hemmung enthalten.^[346]

Durch die Aufbereitung, der an den Synthesizern gewonnen Daten mit Metadaten gemäß GlycoRDF^[339], der standardisierten Repräsentation für Glykan-Strukturen, sowie dem implementierten strukturierten Zugriff auf Pathway-Datenbanken, lassen sich Schrittketten erheblich einfacher konstruieren und evaluieren, und können bei entsprechend breiter Datenbasis in nachfolgenden Anwendungsszenarien automatisiert vom Expertensystem vorgeschlagen werden. Die Einarbeitung in die Synthesizer-Software und deren modularen Softwarekomponenten kann unter Verwendung virtueller Lernportale wie Moodle^[347] vereinfacht werden, die bereits in der weiterführenden Ausbildung umfangreiche Nutzung finden und direkt auf SQL-Datenbanken wie MySQL aufbauen. Durch die dynamischen Lerninhalte lässt sich für jeden Anwender ein individuell angepasster Funktionslevel bereitstellen, erlernen und validieren. Neben der Reduzierung der

¹⁷⁹ Online unter: <https://www.wikidata.org/> [letzter Abruf: 16.01.2019]

¹⁸⁰ Online unter: <https://www.wikipedia.org/> [letzter Abruf: 16.01.2019]

¹⁸¹ Linked Data („Verknüpfte Daten“): Methode zum Veröffentlichen strukturierter Daten, die aufbauend auf Standard-Webtechnologien wie HTTP, RDF und URIs durch semantische Abfragen miteinander verknüpft werden können.

¹⁸² Online unter: <https://tools.wmflabs.org/scholia/> [letzter Abruf: 16.01.2019]

¹⁸³ Online unter: <http://www.wikipathways.org> [letzter Abruf: 16.01.2019]

¹⁸⁴ SPARQL ist eine grafenbasierte Abfragesprache für RDF.

¹⁸⁵ Online unter: <http://www.openphacts.org/index.php> [letzter Abruf: 12.04.2019]

Einarbeitungszeiten, kann auch der aktuell vorliegende Wissenstand erfasst und protokolliert werden, um auch als Entscheidungsgrundlage für Syntheseempfehlungen zu fungieren.

Auch die Verwendung von videounterstützten Informationen, z.B. YouTube¹⁸⁶, helfen neuen Anwendern bei der Fehlervermeidung und der Systemwartung. Gerade im Bereich sporadisch auftretenden Wartungs- und Reparaturarbeiten ist hierdurch ein effizienter, qualitativer Informationstransfer möglich.

Die jeweiligen angepassten Erweiterungen und Schnittstellentools der Open-Source Anwendungen können auf GitHub hochgeladen und verwaltet werden. Zur Vermeidung von unbemerktem Datenverlust oder Datenkorruption, sollten sowohl z. B.: MD5¹⁸⁷ Prüfsummen Kontrollen, als auch Verschlüsselung & Sicherung der Informationen erfolgen, u. a. mit WinZip¹⁸⁸ oder 7zip¹⁸⁹.

Mit GlycoPOST¹⁹⁰ steht seit 2018 ein Repository zum Upload von Glykoproteomics Daten bereit. Die Daten lassen sich hierbei mit einer Offenlegungsbeschränkung verwalten, die mit der Publikation aufgehoben oder verwaltet werden kann.

Die Empfehlungen zu MIRIAM^{191[348]} führten zur Toolbox AnIML¹⁹², einer auf XML¹⁹³ basierenden Auszeichnungssprache, welches die Bereitstellung von herstellerunabhängigen Beschreibungsdaten für Analysegeräte und damit die Reproduzierbarkeit von Versuchen auch laborübergreifend erheblich vereinfacht. Die Implementierung dieser Applikation in die Synthesizer-Software ist effizient möglich, da die Generierung von ASCII Dateien funktionaler Bestandteil jeder Programmiersprache ist. Durch die logisch aufeinander aufbauenden Ebenen lassen sich auch komplexere Szenarien abbilden. Mit dieser Funktionalität wird der Modularität der zusammenarbeitenden Laborgeräte eine weiterer Freiheitsgrad hinzugefügt.

¹⁸⁶ Online unter: <https://www.youtube.com/?gl=DE&hl=de> [letzter Abruf: 19.02.2019]

¹⁸⁷ Akronym für **M**essage-**D**igest **A**lgorithm **5**; Verschlüsselungsalgorithmus, der einen 128-Bit-Hashwert erzeugt.

¹⁸⁸ Online verfügbar unter: <https://www.winzip.com/win/de/> [letzter Abruf: 19.02.2019]

¹⁸⁹ Online verfügbar unter: <http://www.7-zip.de/> [letzter Abruf: 19.02.2019]

¹⁹⁰ Online unter: <https://glycopost.glycosmos.org/> [letzter Abruf: 19.02.2019]

¹⁹¹ Akronym für **M**inimum **I**nformation **R**equested in the **A**notation of **M**odels

¹⁹² Online unter: <https://www.animl.org/software> [letzter Abruf: 19.02.2019]

¹⁹³ Abkürzung für Extensible Markup Language; Auszeichnungssprache zur Darstellung hierarchisch strukturierter Daten im Format einer Textdatei

5.3.5 Datenbanken - Datenabgleich und Validierung

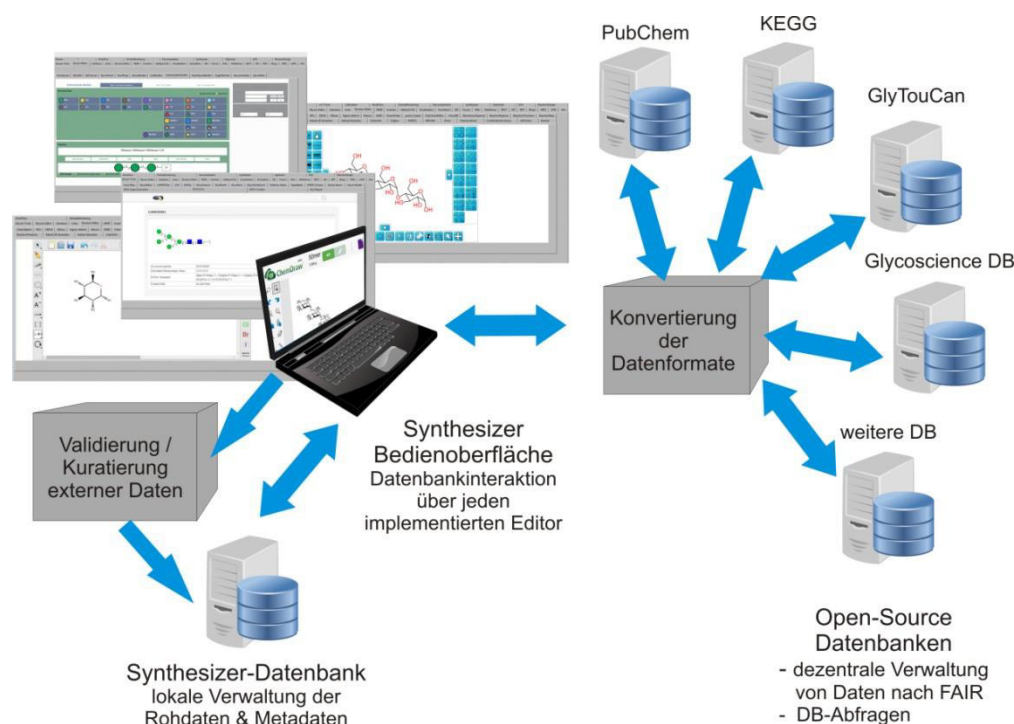


Abbildung 224: Konzept der Datenbankstruktur innerhalb der Synthesizer-Applikation für Molekülsynthesizer als Grundlage zur Entwicklung des Expertensystems.

Der Schlüssel für alle weiteren Entwicklungen für ein cyber-physisches Chemie-System, das über eine KI, mittels maschinellen Lernens zu Synthesevorschlägen führt, liegt in der Qualität, aber auch dem Umfang der Datenbasis. Der aktuelle Synthesizer archiviert Prozessdaten wie Skripte, Module und Synthesen in lokalen Dateien und verwaltet damit die Minimalinformationen zu den maschinellen Vorgängen während der Synthese am Synthesizer. Chemische Daten der Synthese, Reagenzien und Reaktanten, Aufzeichnungen, Rohdaten von NMR, MS, u. a. werden derzeit außerhalb des Synthesegerätes auf lokalen Speichern verwaltet und stehen damit einer automatischen Prozessoptimierung nicht zur Verfügung.

Das Konzept des zukünftigen Synthesizers sieht den Aufbau einer zentralisierten Verwaltung aller, mit der Synthese in Zusammenhang stehender, Daten innerhalb einer Datenbankstruktur vor und geht weit über die reine Datenarchivierung hinaus. Die anfallenden Rohdaten werden dabei mit lokal vorhandenen Metadaten z.B. aus dem elektronischen Laborbuch verknüpft. Strukturinformationen, Eigenschaften, sowie Reaktionsinformationen können durch CML^[349] in der Datenbank hinterlegt werden.

Die gegenwärtigen technischen Fortschritte des Glyko-Engineerings beruhen vor allem auf präzisen und hochdurchsatzfähigen modernen Methoden der Glykan-Analyse^[350], welche wiederum auf die

gestiegene Verfügbarkeit interoperabler Daten der verschiedenen Forschungsbereiche basiert. Die ergänzende Analyse öffentlich zugänglicher Daten in Zusammenhang mit den individuellen Syntheseanforderungen, kann auch für den chemischen Sektor zu dem Evolutionshub führen, wie er sich bereits im biologischen und pharmazeutischen Sektor unter die Einbeziehung von KI-Mechanismen abzeichnet. Das Synthesizer-Konzept sieht deshalb den automatisierten dezentralen Zugriff auf die Open-Source Datenbanken, u. a. auch für den systematischen Aufbau der lokalen Datenbasis des zukünftigen Expertensystems, vor. In der ersten Entwicklungsstufe, dem Molekülsynthesizer, können die verwendeten Reaktanten, Substanzen und Reaktionsergebnisse über den Zugriff auf bestehende dezentrale Datenbanken spezifiziert und nach Validierung zur Dokumentation, Verwaltung und Vorschlagswesen in die lokale Datenbank übernommen werden. In Tabelle 6 (S. 247) werden die derzeit wichtigsten Open-Source-Datenquellen vorgestellt, für die bereits der rudimentäre Zugriff für die Implementierung in die Synthesizer-Applikation getestet wurden. Nicht nur die Daten der Open-Source Quellen, sondern deren jeweilige Schnittstellen über API, HTTP-Webanfragen oder Web-Server, weisen dabei unterschiedliche Strukturen, Fähigkeiten und Komplexitäten auf. Für die nachfolgend vorgestellten Datenbanken wurden testweise Lesezugriffe auf enthaltene Datenbank-Informationen aus LabVIEW realisiert. Bei Abfragefunktionen, die sich aus REST/API erstellen lassen, wurden Prototypen erstellt.

Mit dem Chemical Abstracts Service (CAS)^[351] wurde die Vielfältigkeit und Uneindeutigkeit von chemischen Strukturen durch die Zuweisung eines eindeutigen Identifier für chemische Substanzen beseitigt. Die Implementierung der Webressource¹⁹⁴ Common Chemistry™ des Chemical Abstracts Service in LabVIEW, ermöglicht den Zugriff auf die CAS- Registrierungsnummern von Chemikalien mit allgemein öffentlichem Interesse (Abbildung 225).

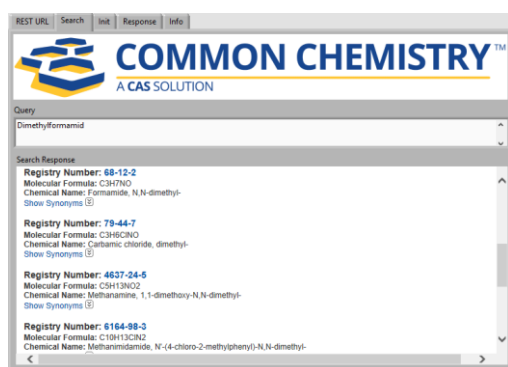


Abbildung 225: Implementierung von Common Chemistry ermöglicht den Zugriff auf die CAS-Registrierungsnummern von Chemikalien aus dem LabVIEW-Programm heraus.

Darüber hinaus ermöglicht PubChem^[352–364] als weitere dezentrale Datenbank den Zugriff auf zusätzliche CAS Nummern, die direkt synthesezugehörig in der lokalen Datenbank verwaltet werden können, was eine wiederholte Durchführung der Synthesen vereinfacht. Die PubChem – Datenbank

¹⁹⁴ Online unter: <http://www.commonchemistry.org/> [Abruf: 06.12.2018]

(Abbildung 226) besteht aus drei miteinander verknüpften Datenbanken: Substance, Compound und BioAssay, die beträchtliche Bedeutung als Informationsquelle für wissenschaftliche Forschungen erlangt haben. Die Substance-Datenbank enthält aktuell (Stand 2018) mehr als 180 Millionen von Forschern bereitgestellte chemische Substanzbeschreibungen. Die Compound-Datenbank speichert ca. 60 Millionen einzigartige chemische Strukturen, die aus der Substance-Datenbank extrahiert wurden und die sowohl textorientiert (Abbildung 226a), als auch grafisch (Abbildung 226b) innerhalb des LabVIEW-Programms abgerufen werden können. In BioAssay sind gegenwärtig über 225 Millionen biologische Aktivitätsdaten von chemischen Substanzen gespeichert, die in Experimenten getestet wurden, und mehr als 9000 einzigartige Protein-Zielsequenzen abdecken.^[357,358,361]

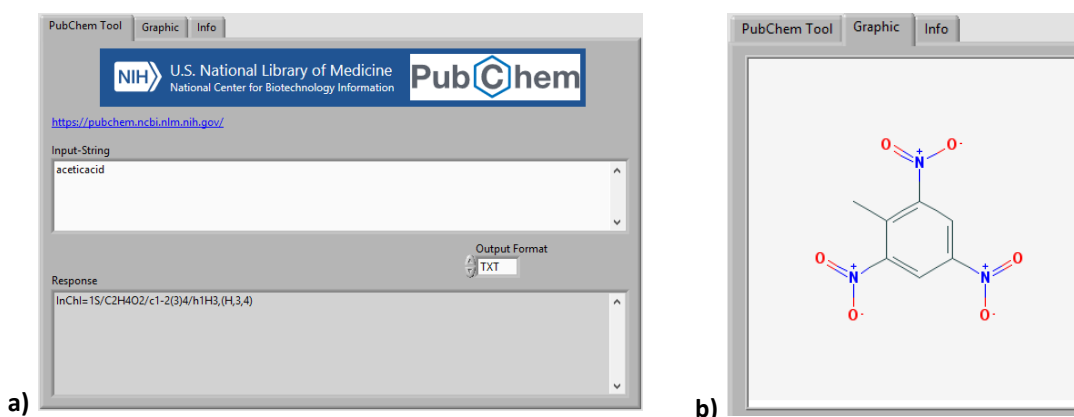


Abbildung 226: LabVIEW – Benutzeroberfläche der Konzeptstudie zur Nutzung der PubChem – Datenbank, die als Web-API/REST-Zugriff in den LabVIEW Programmcode implementiert wurde. Dabei sind nicht nur textorientierte Zugriffe möglich (a), sondern auch die Ausgabe der Strukturformeln (b) und weitergehenden Informationen zu den angefragten Molekülstrukturen.

Da über diese Schnittstelle Zugriff auf eine Vielzahl von substanzbezogener chemischer Parameter und physikalischer Kennwerte besteht, können diese Informationen automatisiert für Syntheseempfehlungen und Plausibilitätsprüfungen verwendet werden. Durch die Funktionalität der Ähnlichkeitssuche erhält der Bediener auch einfachen Zugriff auf Varianten und alternative Reaktanten. Als funktionaler Austauschparameter kann ein nach IUPAC standardisierter chemischer Identifier InChI verwendet werden.

ChEBI^[365–367] ist eine kuratierte, nicht redundante Sammlung chemischer Entitäten und verwandter Daten, die als Ontologie organisiert ist und chemische Verbindungen klassifiziert, von denen ein semantisches Ähnlichkeitsmaß abgeleitet wird, um die biologische Relevanz von Molekülen widerzuspiegeln.^[368] Nachfolgende Abbildung zeigt das LabVIEW-VI für die Implementierung von ChEBI innerhalb der Applikation.

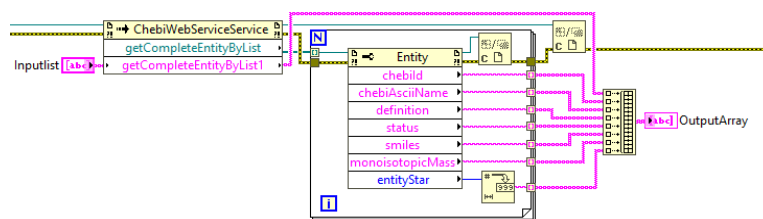


Abbildung 227: LabVIEW-VI zur Implementierung des Webdienstes der Datenbank ChEBI.

Auf dem Gebiet der Wirkstoffforschung und der medizinischen Chemie ist ChEMBL^[369–374] eine etablierte Ressource, die standardisierte Bioaktivitäts-, Molekül-, Ziel- und Arzneimitteldaten aus der medizinisch-chemischen Literatur, aus dem Screening vernachlässigter Krankheiten, agrochemischen Daten, sowie Daten aus klinischen Studien und Patenten speichert.^[373,375] Die ChEMBL-Webdienste können zusammen oder unabhängig voneinander verwendet werden, um Anwendungen und Datenverarbeitungsworkflows zu erstellen, die für die Wirkstoffsuche und chemischen Biologie relevant sind.^[372]

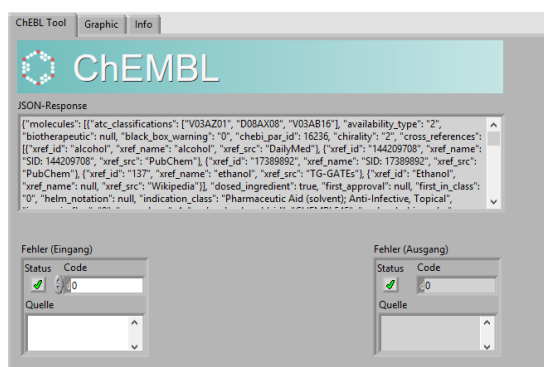


Abbildung 228: Implementierung von ChEMBL in die LabVIEW-Benutzeroberfläche.

BindingDB^[376–380] kuratiert kontinuierlich eine Reihe von Zeitschriften, die nicht in anderen öffentlichen Datenbanken enthalten sind, und verwaltet Bindungsaffinitäten von Protein-Ligand-Komplexen für die medizinische Chemie, Computerchemie und Systempharmakologie.

UniCarbKB^[381–385] enthält aus eukaryotischen Glykoproteinen abgeleitete Kohlenhydratstrukturen und integriert die Kohlenhydratstruktur-Datenbank CarbBank (CCSD¹⁹⁵)^[386,387], sowie EUROCarbDB^[384,385], die Kohlenhydratstrukturen mit experimentellen HPLC-, MS- und NMR-Daten verknüpft. Als Projekt der Kohlenhydratstruktur-Datenbank CSDB^[388,252,389], umfasst die BCSD^[261,390,391,388] bakterielle, pilzliche oder pflanzliche Saccharide. Die ECODAB^[392] Datenbank speichert Antigene von Escherichia coli bestimmten Organismen. Mit PolySac3DB^[393] steht eine kommentierte Datenbank mit dreidimensionalen Polysaccharidstrukturen für die Untersuchung kohlenhydratvermittelter Wirt-Pathogen-Wechselwirkungen und Wechselwirkungen mit anderen

¹⁹⁵ Akronym für Complex Carbohydrate Structure Database

Biomakromolekülen für die Arzneimittelforschung und materialwissenschaftlichen Anwendungen zur Verfügung.

KEGG¹⁹⁶ [394–401] stellt eine wichtige und umfangreiche Glykomik-Informatikressource für Glykanstrukturen dar, die aufgrund der integrierten Wissensbasis mit Daten aus Proteinnetzwerken, Genominformationen und Informationen zu biochemischen Verbindungen und Reaktionen verknüpft sind. In allen KEGG-Ressourcen werden verschiedene Objekte, darunter KEGG-Pfade, chemische Verbindungen sowie Kohlenhydratstrukturen, üblicherweise als Graphen dargestellt,^[394] die sich für weitergehende automatisierte Analysen und die Wirkstoffsuche eignen.

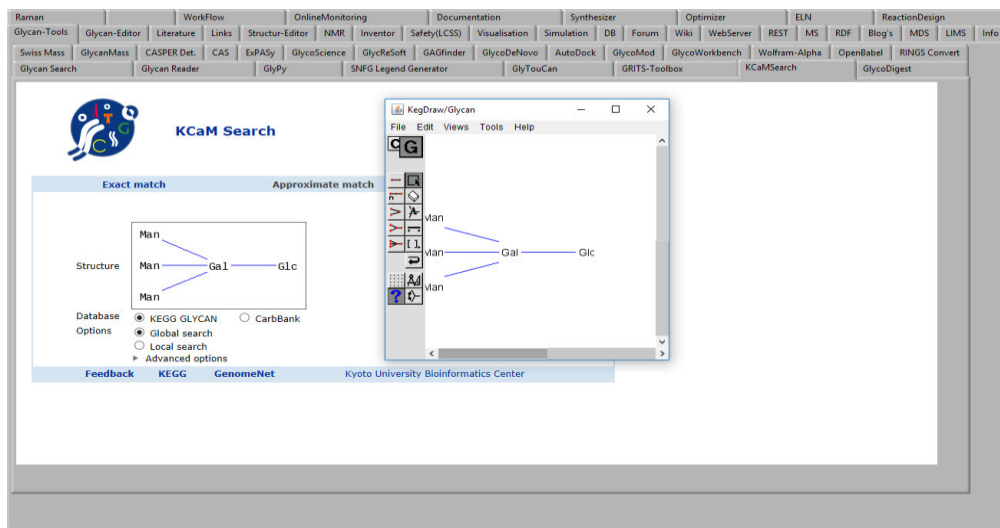


Abbildung 229: In die LabVIEW Applikation implementiertes Suchmasken-Werkzeug KCaM erleichtert Datenbank-Abfragen in der KEGG-DB

GlyTouCan^[402,403] ist ein internationales Repository, die jeder identifizierten Glykanstruktur eine eindeutige Zugangsnummer zuweist und als zentralisierte Ressource für die Ablage von Glykanstrukturen, Kompositionen und Topologien bereitsteht. Die automatische Implementierung von Mechanismen zur Koordinierung von Glykanstrukturen in die Synthesizer-Datenbank, wie sie mit dem GlyTouCan-Repository zur Verfügung stehen, und die Verwendung der eindeutigen Identifikatoren in Kommunikation und Publikationen, ist ein zentraler Faktor bei der Entwicklung und Verbesserung bio- und chemoinformatischer Analysen und Grundvoraussetzung für den Aufbau der Experten-Datenbank. Der Entwicklungshub im Genomik und Proteomik-Bereich nach Einführung eindeutiger Gen- und Protein-Identifikatoren, basiert auf Datenstandards, Datenrepositorien, Modellierungs- und Visualisierungswerkzeugen als Voraussetzungen für systembiologische Forschung,^[404] wie sie derzeit in diesem Umfang für Glykomik - Daten noch fehlen.

Glycosciences^[405–408] verbindet Glykomics und Proteomiks - Ressourcen und enthält Einträge der CarbBank sowie NMR-Daten, theoretische und experimentelle 3D-Strukturen und Molekülmassen.

¹⁹⁶ Kyoto-Enzyklopädie der Gene und Genome

Die Glykan-Fragment-Datenbank (GFDB)^[409] verwaltet glykosidische Torsionswinkel, die von den Glykan-Strukturen in der Proteindatenbank¹⁹⁷ abgeleitet werden. GlycoSuiteDB^[381] ist eine kommentierte und kuratierte relationale Datenbank von Glykanstrukturen aus der Literatur. Die Datenbank des Japanischen Konsortiums für Glykobiologie und Glykotechnologie (JCGGDB) integriert mehrere Datenbanken zu Glykoproteinen, glykome - assoziierten Erkrankungen und analytischen Daten.^[410-412]

5.3.6 Format - Konverter & Kontrollen

Chemische Informationen werden aktuell in vielen verschiedenen Dateitypen und Datenbanken mit unterschiedlichen Datenbankkennungen, oder sogar unter ihren Trivialnamen, gespeichert. Die Limitierung der Wiederverwendbarkeit und Austauschbarkeit chemischer und biologischer Informationen, behindert jedoch fachübergreifende Entwicklungen und Synergieeffekte zwischen den einzelnen Forschungsbereichen. Mit dem Aufkommen öffentlich zugänglicher Bioaktivitäts- und Chemie-Datenbanken hat sich das Bewusstsein für die Themen Datenaufbereitung, Qualität und Integrität auch in der Chemie geschärft.^[373] Zur spezifischen Zieldefinition der Chemo- und Bioinformatik zählt zunehmend die Maximierung der Vergleichbarkeit chemischer Daten und die Verbesserung der Datenintegrität.

Der Molekül-Synthesizer stellt mit seinem flexiblen, modularen Aufbau und dem integrativen Softwarekonzept, ein Zukunftsmodell bereit, in dem die verschiedenen fachspezifischen Grenzen aufgelöst und unter einer einheitlichen Benutzeroberfläche auf unterschiedliche Datenkonzepte und Datenquellen zugegriffen werden kann, um so eine fortschrittliche, effizientere Nutzung des existierenden und zukünftigen Datenbestands zu ermöglichen. Die Implementierung von existierenden, externen Datenquellen und Datentyp-Konvertierungsplattformen in das Synthesizer-Konzept, wie sie in der nachfolgenden Machbarkeitsanalyse durch LabVIEW-Programmbeispiele belegt wird, stellt jedoch nur die Basis des umfassenden Gesamtkonzepts dar.

Die nächste Weiterentwicklung der Anwenderebene wird sich mit der Thematik zur Erstellung plattformübergreifender Äquivalenzkriterien^[413] zwischen chemischen Datensätzen verschiedener Quellen und kontextbezogener Dateninterpretation befassen müssen, welches eine uniforme LabVIEW-Benutzeroberfläche für Datenbankabfragen ermöglicht, die autonom, dem Anwendungsfall entsprechend, die optimale Daten-Antwort (Hintergrund-Quelle) auf die gewünschte Benutzer-Anforderung findet.

¹⁹⁷ Online unter: <https://www.rcsb.org/> [letzter Zugriff: 28.02.2019]

DB Name	Bemerkungen	Link	Literatur
BCSDB	Bacterial Carbohydrate Structure Database	http://csdb.glycoscience.ru/bacterial/	[261,390,391,388]
BindingDB		https://www.bindingdb.org/	[414,376–378,380,379]
BRENDA	Comprehensive enzyme information system	http://www.brenda-enzymes.info	[415]
CAZy	Carbohydrate Active Enzymes	http://www.cazy.org	[416]
CCSD	CarbBank	https://www.genome.jp/dbget-bin/www_bfind?carbbank	[387,386]
CFG DB	Consortium for Functional	http://www.functionalglycomics.org	[417]
ChEBI	Chemical Entities of Biological Interest	http://www.ebi.ac.uk/chebi/	[418a,366,365,367,418b,368]
ChEMBL	Bioaktivitäts-DB	http://ebi.ac.uk	[371,369,370,372,374,373,375]
CSDB	Carbohydrate Structure Database	http://csdb.glycoscience.ru	[388,252,389]
ECODAB	Escherichia coli O-antigen DB	http://www.casper.organ.su.se/ECODAB/	[392]
Glycobase		http://glycobase.nibr.ie	[419]
Glycobase (Lille)		http://glycobase.univ-lille1.fr/base/	
GFDB	Glycan fragment DB	http://www.glycanstructure.org	[409]
GlycoMapsDB		http://www.glycosciences.de/m	[407,406,405,420]
GlycoNavi		http://ws.glyconavi.org	
GlycoPOD	GlycoProtocols Online Database	http://jcgdb.jp/GlycoPOD	[410]
Glycosciences.DB	enthält Monosaccharide DB	http://www.glycosciences.de/database/	[405–408]
GlycosideDB		http://jcgdb.jp/search/GlycosideDB.cgi	
GlycoSuiteDB		http://glycosuitedb.expasy.org	[381]
GlyTouCan	enthält GlycomeDB ^[421]	http://glytoucan.org	[402,403]
KEGG GLYCAN	Kyoto Encyclopedia of Genes and Genomes	http://www.genome.jp/kegg/glycan/ http://www.genome.jp/linkdb/ http://www.kegg.jp	[394–401]
JCGGDB	Japan Consortium for Glycobiology and Glycotechnology DataBase	http://www.jcgdb.jp	[410,411]
PubChem		http://pubchem.ncbi.nlm.nih.gov	[352–364]
UniCarb-DB	enthält GlycoSuiteDB, CarbBank, EUROCarbDB	http://www.unicarb-db.com	[381–385]
UniProt		http://www.uniprot.org	[422]

Tabelle 6: Auswahl allgemein zugänglicher chemischer, biologischer und medizinisch relevanter Datenbanken im Internet, mit denen bereits programmatische Interaktionen innerhalb der Synthesizer-Applikation getestet wurden.

In der Machbarkeitsanalyse wurde International Chemical Identifier Code (InChI), einer der ersten, frei verfügbaren Bezeichner für chemische Substanzen in LabVIEW implementiert. Die Grundfunktionalität hierzu wird von IUPAC¹⁹⁸ bereitgestellt. Der chemische Strukturcode berücksichtigt Konnektivität, Isotopie, Stereochemie, Tautomerie und Formalladung.^[423] InChI – Zeichenketten können nun innerhalb des LabVIEW-Programms auf Plausibilität kontrolliert werden. Die InChI Format-Struktur wurde zu RInChI^[424] ergänzt um den aktuellen Anforderungen gerecht zu werden. Hierdurch werden nun auch Informationen zur Reaktion und Reaktionsführung EDV-technisch verwertbar. Durch die vollwertige Interpretation der enthaltenen Daten kann man mit dem Synthesizer diese Reaktionen abbilden und reproduzieren. Die großen Chemie-Datenbanken (Reaxys, SciFinder, ChemSpider) beginnen diese Erweiterung mit in ihre Strukturen zu etablieren.

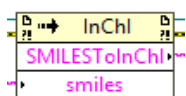


Abbildung 230: LabVIEW-Programmcode aus der Konzept-Studie: Konvertierung eines SMILE Strings in einen InChI String nach IUPAC

Als gängiges Werkzeug der Chemo- und Bioinformatik wurde das LabVIEW-Programm um den Zugriff und die Verwaltung von SMILES¹⁹⁹ als weiteren, wichtigen chemischen Strukturcode ergänzt. Das LabVIEW-Beispiel ermöglicht die Konvertierung von SMILES-Strings in InChI-Strings nach IUPAC (Abbildung 230, Abbildung 231). SMILES^[425] ist der chemische Strukturcode eines kommerziellen Anbieters²⁰⁰. Alternativ existiert mit OpenSMILES²⁰¹ ein Open Source Standard der Blue Obelisk Community.

Moleküle und Reaktionen werden in SMILES unter Verwendung von ASCII-Zeichen angegeben, die Atom- und Bindungssymbole darstellen. In der Peptidwissenschaft dient SMILES-Code zur Beschreibung eingebauter Substituenten bei in-silico-Untersuchungen von Eigenschaften modifizierter Peptide. Ebenso wird SMILES-Code zur Konstruktion von Peptidstrukturen genutzt, die nichtnatürliche und modifizierte Aminosäuren enthalten, auch unter Berücksichtigung glykosylierter Peptide.^[305]

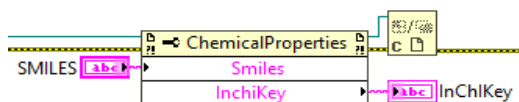


Abbildung 231: Beispiel eines LabVIEW-Eigenschaftsknotens aus der Konzept-Studie: Konvertierung von SMILE in einen InChIKey - String mit der Open-Source-Toolbox OpenBabel. Durch Implementierung von OpenBabel in LabVIEW können aktuell über 110 chemische Datenformate konvertiert und interpretiert werden.

¹⁹⁸ Abkürzung von International Union of Pure and Applied Chemistry; online verfügbar unter: <https://www.inchi-trust.org/> [Abruf: 03.12.2018]

¹⁹⁹ SMILE steht für Simplified Molecular Input Line Entry System

²⁰⁰ DAYLIGHT Chemical Information Systems, Inc.

²⁰¹ Sprachspezifikation online verfügbar unter: <http://opensmiles.org/> [letzter Abruf: 06.12.2018]

Derzeit sind zahlreiche funktional unterschiedliche Programme und Datenbanken verfügbar, die den SMILES-Code verwenden. Ein aktuelles Übersichtspaper speziell für den Peptidsektor wurde 2017 von Minkiewicz et al. verfasst, welches informativ die Verbreitung und Akzeptanz dieser Beschreibungsart aufzeigt.^[426]

Mit OpenBabel^[427] wurde eine chemische Open-Source-Toolbox in den LabVIEW-Programmcode der Machbarkeitsanalyse implementiert, die in der aktuellen Version 2.4. über 110 Formate chemischer Daten beherrscht. Damit können Daten der molekularen Modellierung, Chemie, Festkörpermateriale, Biochemie oder verwandten Bereichen durchsucht, konvertiert, analysiert oder gespeichert werden. Die Darstellung dieser Vielzahl chemischer und molekularer Daten wird durch eine Bibliothek ermöglicht, die einen breiten Bereich chemoinformatischer Algorithmen implementiert. Hierin enthaltene Funktionen reichen von der Teilbeladungszuordnung über die Erfassung der Aromatizität bis zur Abbildung der Bindungsordnung und Kanonisierung²⁰².

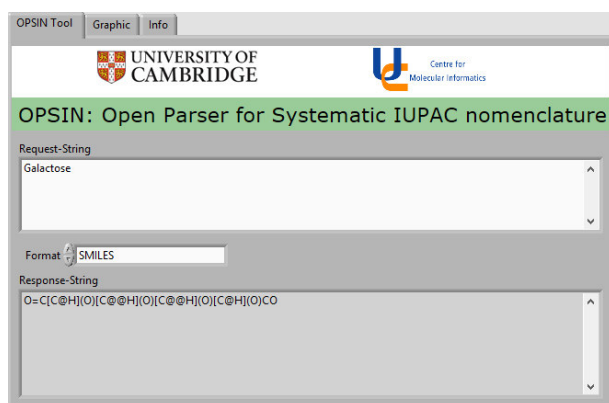


Abbildung 232: LabVIEW-Programmoberfläche zeigt die Implementierung von OPSIN, einem Open Source Konvertierungswerkzeug für chemische Strukturen

Die Vielzahl der unterschiedlichen Strukturbeschreibungen hat gerade bei den großen Datenbanken zur Bereitstellung von Konverter-Funktionen, bis hin zum Abgleich der Trivialnamen, geführt, wie z.B. OPSIN^{203[428]}, das die Mehrheit der organischen chemischen Nomenklatur schnell und präzise interpretieren kann (Abbildung 232). Diese Funktionalitäten werden als Web- REST²⁰⁴/API²⁰⁵ -Zugriffe für die Benutzer zur Verfügung gestellt, also einer auf dem Web-Protokoll basierenden URL²⁰⁶-Abfrage.

²⁰² Kanonisierung ist in der Chemoinformatik die Erstellung eines eindeutigen Satzes von Atomkennungen in einer Strukturdatenbank

²⁰³ Open Parser for Systematic IUPAC nomenclature, Open Source Webservice der University of Cambridge, UK, Online unter: <https://opsin.ch.cam.ac.uk/> [letzter Abruf: 06.12.2018]

²⁰⁴ Programmierparadigma zum Kommunikationsverhalten verteilter Systeme, besonders Webservices; **REST** steht für **Representational State Transfer**

²⁰⁵ Schnittstelle zur Anwendungsprogrammierung, **API** steht für **Application Programming Interface**

²⁰⁶ Uniform Resource Locator (URL) identifiziert und lokalisiert eine Ressource, beispielsweise eine [Webseite](#) und die zu verwendende Zugriffsmethode (z. B. [Netzwerkprotokolle](#) [HTTP](#) oder [FTP](#))

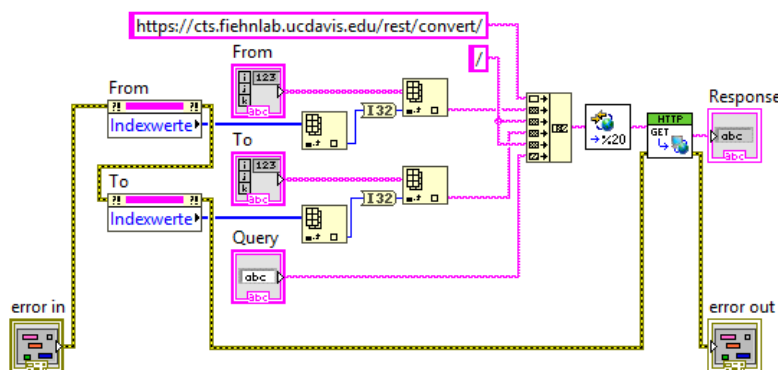


Abbildung 233: Beispiel der Implementierung einer webbasierten Konvertierungsplattform in LabVIEW-Programmcode. Über Listenstrukturen (Quellformat, Zielformat) lässt sich die Konvertierung mittels Web-API/REST - Zugriffe ausführen. Das LabVIEW Programm stellt die nötigen URL-Strings zusammen, und stellt im Response die angeforderte-Antwort bereit.

Damit sich der Anwender der Automatisierungssysteme nicht mit der Konfiguration dieser Anfragen und der erhaltenen Antwort-Zeichenfolgen (Response-Strings) in ihrer Komplexität befassen muss, ist diese Kommunikationsvariante bereits im LabVIEW-Programmbeispiel der Machbarkeitsanalyse umgesetzt (Abbildung 233). Dieser Mechanismus ermöglicht es, die eingebundenen Datenbanken direkt mit den neuen Informationen anzufragen und die erhaltenen Daten zu synchronisieren. Dies führt zu erheblichen Vereinfachungen für die Protokollierung und das Auffinden von Strukturen. Alle nachfolgenden Beispiele wurden über das gezeigte Programmierparadigma Web-REST/API in LabVIEW implementiert.

Mit dem CIR (Chemical Identifier Resolver)²⁰⁷ können chemische Identifier in verschiedene Namensräume konvertiert werden. Hierdurch ergibt sich die Möglichkeit verschiedene Struktureditoren, Datenbanken und Analyseprogramme übergreifend zu verwenden, da oftmals nur unterschiedliche Strukturbeschreibung als Eingangs- oder Ausgangsfunktion bereitgestellt sind. Der CIR bietet derzeit Zugriff auf über 120 Millionen Strukturdatensätze (Abbildung 234).^[429]

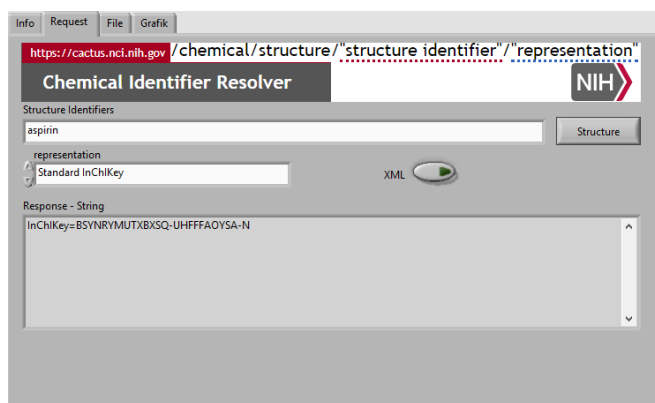


Abbildung 234: LabVIEW – Benutzeroberfläche zur Abfrage und Ausgabe von Strukturen mit dem Chemical Identifier Resolver (CIR). Die Implementierung in das LabVIEW – Programm erfolgt über eine URL-Anfrage basierend auf dem HTTP-Protokoll.

²⁰⁷ Netzwerkdienst der NCI / CADD-Gruppe, der Computer-Aided Drug Design-Forschungsgruppe innerhalb des Chemical Biology Laboratory am National Cancer Institute, United States. Online unter <https://cactus.nci.nih.gov/chemical/structure> [letzter Abruf: 06.12.2018]

The Chemical Translation Service (CTS)^[430] erlaubt als webbasiertes Werkzeug zur Verbesserung der Standardisierung von Metabolomik-Berichten auch die gleichzeitige Konvertierung einer größeren Anzahl von Dateien mittels Stapelverarbeitung, sog. Batch-Konvertierung. Es kann die gebräuchlichsten Verbindungskennungen durchführen, einschließlich CAS, CHEBI, Verbindungsformeln, Human Metabolome Database HMDB, InChI, InChIKey, IUPAC-Namen, KEGG, LipidMaps, PubChem CID und SID, SMILES und chemische Synonymnamen.

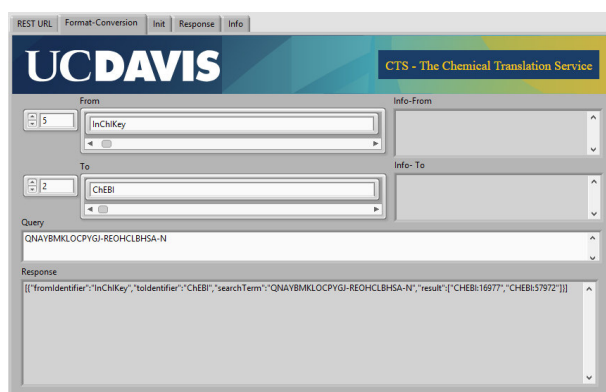


Abbildung 235: Implementierung des Chemical Translation Service in die LabVIEW-Oberfläche. Der Webservice erlaubt Batch-Konvertierung und kann damit eine größere Anzahl von Datei-Konvertierungen der gebräuchlichsten Verbindungskennungen gleichzeitig durchführen.

CML oder auch ChemML ist eine auf XML²⁰⁸ basierendes Textaustauschformat strukturierter Chemiedaten. Mit der „Chemical Markup Language“ steht ein chemisches Dokumentenformat für interoperable Dokumente zur Verfügung. CML unterstützt eine Vielzahl chemischer Konzepte, wie Moleküle, Reaktionen^[431], Spektren und analytische Daten^[432], sowie die Grundkonzepte der Computerinterpretierbaren Chemie.^[433] Viele der Open-Source-Quellen für Bio- und Chemoinformatik, wie z.B. OpenBabel, das Chemistry Development Kit(CDK)^[434] oder Programme wie Jmol²⁰⁹ und Bioclipse^{210[435]} unterstützen das CML-Format. Durch die von LabVIEW bereitgestellten Möglichkeiten XML-Dateien zu schreiben und zu interpretieren lassen sich damit alle Informationen effizient und für das World Wide Web kompatibel bereitstellen. Solange eindeutige Tag's²¹¹ Verwendung finden, ist auch eine nachträgliche Anpassung oder Konvertierung ohne jeglichen Informationsverlust durchführbar.

²⁰⁸ Extensible Markup Language: Auszeichnungssprache zur Darstellung hierarchisch strukturierter Daten, als ein von Menschen und Maschinen lesbares Textdatei-Format.

²⁰⁹ Online unter <https://sourceforge.net/projects/jmol/files/Jmol/Version%2014.29/> [Abruf: 06.12.2018]

²¹⁰ Online unter <http://www.bioclipse.net/> [Abruf: 06.12.2018]

²¹¹ Bezeichnet in der Informatik die Auszeichnung eines Datenbestandes mit zusätzlichen Informationen.

Aufgrund der Vielzahl der für die XML-Welt entstandenen Werkzeuge, ist dieses Format gerade für die Übergangszeit ein sehr sinnvoller Kompromiss, denn im Bereich der Glykan-Datenformate befinden sich aktuell mit RDF²¹² Varianten als auch mit MIRAGE²¹³ ganz neue Strukturen in der Realisierung.

5.3.7 Arbeitssicherheit

Grundsätzlich erfordern die neuen technologischen Entwicklungen im Labor und darin eingeschlossen die Möglichkeiten eines hoch funktionalen Gerätes, wie den zukünftigen Molekülsynthesizer, der „universelle“ Chemie herstellen und zudem auch per „Fernsteuerung“ bedienbar sein wird, ein dem aktuellen Stand der Technik angepasstes Sicherheitskonzept. Heute ist es nicht mehr akzeptabel, wenn mit geringen technologischen Aufwand vermeidbare Risiken der Verantwortung des Benutzers überlassen werden. Erklärtes Ziel des Synthesizerkonzeptes: die Entwicklung eines universell für die chemische Synthese einsetzbares Laborgerät, welches mit geringem Einarbeitungsaufwand auch für Nichtchemiker möglichst intuitiv zu bedienen sein soll, bedeutet im Umkehrschluss, auch dann betriebssichere Prozesse zu gewährleisten, wenn der Benutzer über kein umfassendes chemisches Hintergrundwissen verfügt. Die gerätetechnischen & hardwarenahen Sicherheitssysteme für den Synthesizer wie SPS und lokale Überwachung von Prozessen wurden bereits im Kapitel Hardware-Entwicklung angesprochen. Dieser Abschnitt beschäftigt sich mit den softwaretechnischen Erweiterungen zur Gefährdungsbeurteilung am Prozess beteiligter Substanzen und deren Endprodukte. Aufgrund der Anzahl chemischer Substanzen, ist es auch für den Chemiker unmöglich direkt in allen Labor-Situationen eine konkrete Gefahrenabschätzung abzugeben. Nach der Laborrichtlinien-Verordnung müssen für alle im Labor verwendeten Substanzen die Gefahrenstoff-Datenblätter vorliegen. Damit besteht zwar eine Möglichkeit der grundsätzlichen Einschätzung der Synthese, jedoch können aus dem geplanten Synthesepfad temporär potenziell gefährliche Situationen resultieren, die der erhöhten Aufmerksamkeit des Anwenders bedürfen. Bereits im Vorfeld der Synthese ist deshalb eine Sicherheitsabschätzung / Bewertung aller Eingangs-Chemikalien, der Zwischenreaktionsprodukte, sowie des gewünschten Endproduktes erforderlich. Dem Anwender sind Sicherheitshinweise zur Verfügung zu stellen, damit bei Eintritt einer Notsituation die benötigten Substanzen, z. B. zur Neutralisation bereitstehen, und schnellstmöglich die erforderlichen Schutzmaßnahmen ergriffen werden können.

²¹² Resource Description Framework, Standard für das Codieren von Metadaten und anderen strukturierten Informationen im Semantic Web.

²¹³ MIRAGE steht für „Minimum Information Required for A Glycomics Experiment“: Projekt des Beilstein Institutes in Zusammenarbeit mit Wissenschaftlern der Universität Georgia zur Qualitätsverbesserung experimenteller Daten von Zuckerverbindungen in der Fachliteratur. Online unter: <https://www.beilstein-institut.de/projekte/mirage> [Abruf: 09.01.2019]

Eine Möglichkeit hierzu wird von der Pistoia Alliance in Form der Chemical Safety Library (CLS)²¹⁴ bereitgestellt, die Informationen zu unerwünschten und gefährlichen Nebenreaktionen verwaltet. Die Daten werden unter einer Open Source Lizenz als *.csv – Datei kontinuierlich gepflegt. Zum effizienten Aufbau der Datenbasis steht ein Google-Docs-Formular²¹⁵ als Übermittlungsplattform bereit. Die dort hinterlegten Daten werden moderiert übernommen. Aufgrund der Relevanz dieser Hilfsinformationen, sind die Daten der CSL direkt in PubChem integriert.

Seit Inkrafttreten der EU-Chemikalienverordnung Nr. 1907/2006 (REACH²¹⁶) am 1. Juni 2007, in dem die Unternehmen verpflichtet wurden Stoffsicherheitsbeurteilungen für in Verkehr gebrachte Chemikalien durchzuführen und gegenüber der ECHA²¹⁷ aufzuzeigen, wie der Stoff sicher verwendet werden kann, muss dem Anwender zu jeder Substanz eine Risikomanagementmaßnahme mitgeteilt werden. Mit einem Zugriff auf die ECHA – Datenbank kann die Klassifizierung nach GHS²¹⁸ am Synthesizer verwendeter Substanzen zur grundlegenden Beschriftung der Behältnisse genutzt werden, und zugleich den schnellen Zugriff als digitales Gefahrenstoff- Datenblatt für die durchzuführende Synthese gewährleistet werden.

Die große Anzahl frei verfügbarer Toxizitätsdaten für Industriechemikalien in der ECHA Datenbank, hat zu regulatorischen Rahmenbedingungen für Read-Across²¹⁹, wie das ECHA Read-Across Assessment Framework, Softwaretools und öffentliche Datenbanken für einen automatisierten Prozess zur Lückenfüllung im Rahmen der Sicherheitsbewertung geführt.^[436] Zudem stehen weitere interessante computerchemische Lösungen zur Unterstützung der Sicherheitsbewertung^[437] bereit, die in die aktuelle Synthesizer-Applikation implementiert werden können.

Die eTOX²²⁰ -Datenbank integriert Bioinformatik- und Chemoinformatik-Daten und hat sich, mit derzeit mehr als 1900 verschiedenen chemischen Strukturen und mehr als 8000 Datensätzen für In-vivo-Toxizitätsstudien, zur größten präklinischen Toxizitätsdatenbank für Arzneimittel und Wirkstoffkandidaten entwickelt.^[436] Das webbasierte OpenTox-Framework ist die weltweit führende offene Plattform für prädiktive Toxikologie. Es unterstützt den Zugang zu experimentellen Daten, quantitativen Struktur-Aktivitäts-Beziehungsmodellen und toxikologischen Informationen über eine Integrationsplattform, die den regulatorischen Anforderungen von REACH und OECD-Validierungsgrundsätzen entspricht.^[438] Das AMBIT-Webservicepaket²²¹ ist eine Implementierung der OpenTox - API.^[439] Das CompTox Chemistry Dashboard²²² greift auf eine kuratierte Datenbank der

²¹⁴ Online unter: <https://www.pistoiaalliance.org/projects/chemical-safety-library/> [letzter Abruf: 11.03.2019]

²¹⁵ Online unter: <https://docs.google.com/forms/d/e/1FAIpQLScieVOhngWMAKIKnsMBSM89pp-VSVSdjxlnQlpR9KApZCEA/viewform> [letzter Abruf: 11.03.2019]

²¹⁶ Akronym für Registration, Evaluation, Authorisation and Restriction of Chemicals

²¹⁷ Akronym für European Chemicals Agency; Online unter: <https://echa.europa.eu/de/> [letzter Abruf: 24.02.2019]

²¹⁸ Akronym für Globally Harmonized System of Classification and Labelling of Chemicals

²¹⁹ Stoffgruppenbetrachtung zur Ableitung intrinsischer Stoffeigenschaften.

²²⁰ Initiative der European Innovative Medicines; Online unter: <http://www.etoxproject.eu> [letzter Abruf: 24.02.2019]

²²¹ Online unter: <https://ambitlri.ideaconsult.net/tool2/ui> [letzter Abruf: 25.02.2019]

²²² Online unter: <https://comptox.epa.gov/dashboard> [letzter Abruf 25.02.2019]

EPA²²³ mit derzeit ~760.000 Substanzen zu und ermöglicht den schnellen Zugriff auch auf physikochemische Daten, Verhalten und Transport in der Umwelt, Exposition, Verwendung, In-vivo-Toxizität und In-vitro-Bioassay-Daten, die über einen Integrationshub mit Verknüpfungen zu zusätzlichen EPA-Daten und öffentlich zugänglichen Online-Ressourcen dokumentiert sind.^[440] Mit der Implementierung von Open PHACTS^[441], einer Plattform für die übergreifende Wirkstoffsuche, kann zudem über die API-Funktionen Zugriff auf weitere Datenbanken wie ChEBI, ChEMBL, SureChEMBL, ChemSpider, ConceptWiki, DisGeNET, DrugBank, Gene Ontology, neXtProt, UniProt und WikiPathways genommen werden.

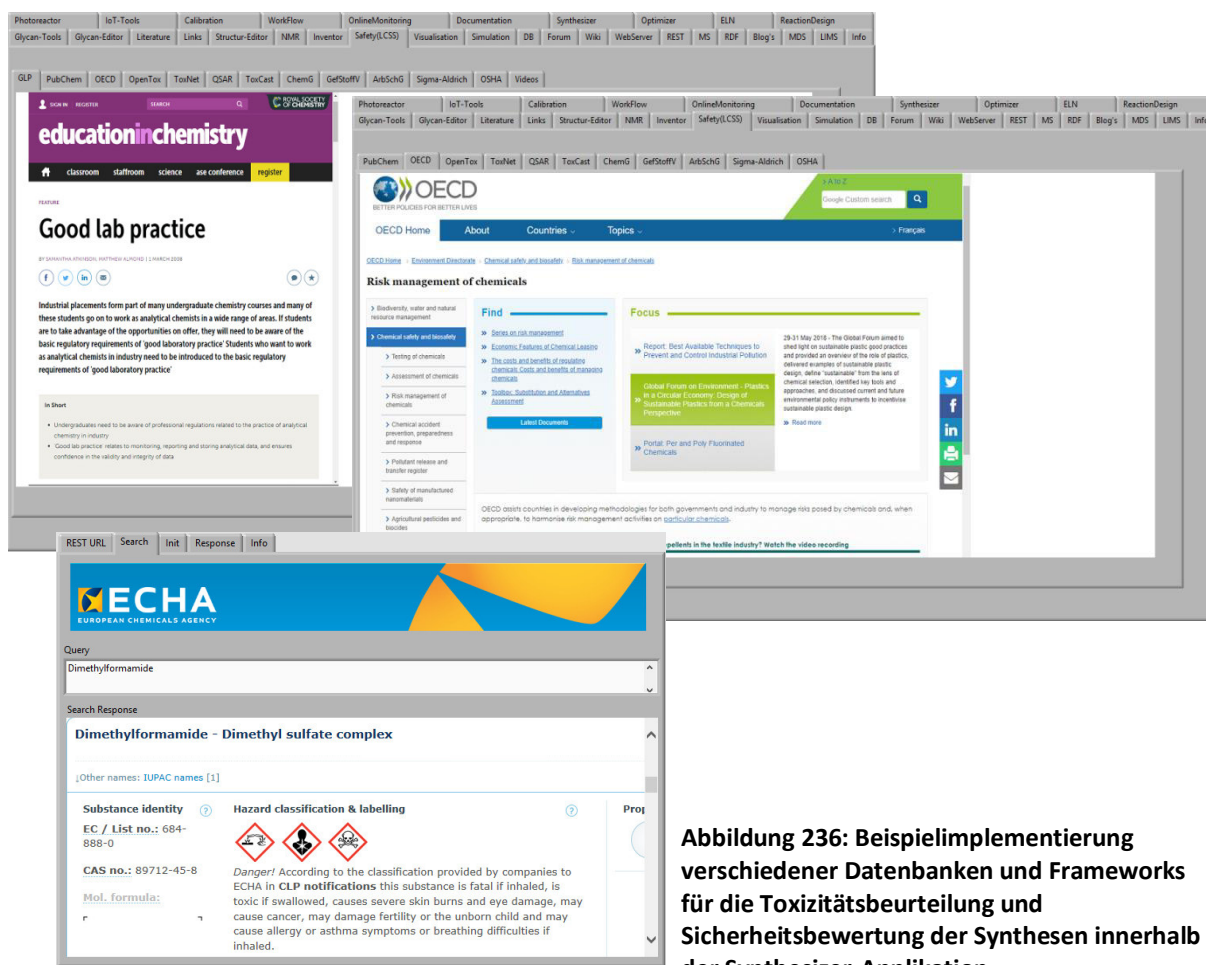


Abbildung 236: Beispielimplementierung verschiedener Datenbanken und Frameworks für die Toxizitätsbeurteilung und Sicherheitsbewertung der Synthesizer-Applikation

Bei der Sicherheitsbewertung für ein neues chemisches Molekül kommen komplexe Blackbox-Techniken wie neuronale Netzwerke²²⁴ sowie Modelle und Methoden des maschinellen Lernens zum Einsatz.^[442] Aufgrund der unterschiedlichen Vorhersagemodelle^[443], können dabei Empfehlungen und Einstufungen in ihrer jeweiligen Gewichtung angepasst werden.

²²³ Akronym für „Environmental Protection Agency“; Unabhängige US-Behörde zum Umweltschutz und zum Schutz der menschlichen Gesundheit.

²²⁴ Künstliche neuronale Netzwerke (KNN) aus der Neuroinformatik sind Teil der „künstlichen Intelligenz“ (KI)

Mit einem Mechanismus zur eindeutigen Identifizierung einer Person, lassen sich benutzerindividuell Daten, Informationen, sowie Funktionen am Synthesizer bereitstellen. Da die eindeutige Rückführbarkeit ebenso Bestandteil des Datenmanagements nach dem FAIR-Konzept ist, wurde der Zugriff über ORCID²²⁵ getestet und rudimentär in die Sub-VI's der Beispielapplikation für die Konzeptstudie implementiert. ORCID ist ein eindeutiges Kennzeichnungssystem, das den Autor mit Veröffentlichungen, Patenten, klinischen Studien, Datensätzen und anderen Forschungsaktivitäten verknüpft und somit das Problem der Mehrdeutigkeit des Autorennamens löst.^[444] Über diesen Mechanismus wird es möglich, die individuellen Synthesen dem jeweiligen Forscher eindeutig und frühzeitig zuzuordnen.

5.3.8 Syntheseplanung & Beschreibungssprache

Mit dem Zugriff auf bekannte Muster bestehender Reaktionen aus verschiedenen Bio- und Chemoinformatik-Ressourcen, sowie den durchgeführten Synthesen an den automatischen Synthesizern, kann dem Bediener zusätzliche Funktionalität für eine effiziente Syntheseplanung zur Verfügung gestellt werden. Insbesondere die biologische Aktivität und die Interaktion zu Proteinen ist beim pharmazeutischen Design der Wirkstoffe ein unschätzbare Hilfsmittel. Die bioaktiven Zielprodukte sind jedoch komplexe Strukturen, die auf atomarer Ebene aufgrund ihrer Größe und räumlicher Orientierung kaum darstellbar sind und deshalb in allen beteiligten Fachbereichen auf die jeweiligen Grundbausteine abstrahiert und unter neuer Nomenklatur zusammengefasst werden. Neben den rein textorientierten Varianten haben sich aufgrund der fortschreitenden EDV-technischen Entwicklung (HTML5) grafische Symboliken etabliert. Die Beschreibungssprache HELM^{226[445]} wurde speziell für komplexe biomolekulare Strukturen entwickelt und findet Anwendung in der Darstellung von Strukturen wie Antisense-Oligonukleotiden, kurzen Interferenz-RNAs, Peptiden, Proteinen und Antikörper-Wirkstoff-Konjugaten. Für die Moleküldefinition steht der HELM-Editor als Open-Source-Anwendung zur Verfügung, der in die modulare Synthesizer-Applikation integriert werden kann. Neben anderen Open-Source-Quellen, steht auch in kommerziellen Produkten wie ChemDraw ab Version 17 eine HELM-Toolbox bereit, was für viele Anwender die Bedienbarkeit und Verfügbarkeit erleichtert.

In HELM setzten sich die Biomoleküle aus Monomeren zusammen, die an ihren jeweilig reaktiven Einheiten koppeln. Die ursprünglich vorhandenen möglichen zwei reaktiven Kopplungsstellen, wurden inzwischen auf fünf Anknüpfungspunkte erweitert, so dass es nun auch neben Peptiden und RNA möglich ist, durch eigene fachspezifische Monomerdefinition mit der Beschreibung der jeweils reaktiven Substituenten, Glykanverbindungen als Zielmoleküldefinition in HELM darzustellen.

²²⁵ Akronym für **O**pen **R**esearcher and **C**ontributor **I**D; Online unter: <https://orcid.org/> [letzter Abruf: 26.02.2019]

²²⁶ Akronym für **H**ierarchical **E**ding **L**anguage for **M**acromolecules

Speziell für die Darstellung von Kohlenhydraten im Semantic Web wurde jüngst die Beschreibungssprache WURCS^[446] als neue lineare Notation entwickelt, die sich als Open Source Ressource ebenfalls leicht in das modulare Synthesizer-Konzept implementieren lässt.

Aufgrund der Datenstrukturierung bei der Zielmoleküldefinition, welche auf spezifischem Monomer, reaktiven Kopplungsstellen und jeweils erforderlichen Reaktionsbedingungen, unter Aufzeichnung aller Nebenreaktionen, Ausbeuten, sowie begleitender Meta-Informationen, u.a. basiert, kann die Synthese-Datenbank des Molekülsynthesizers strukturiert ergänzt und die Relevanz der synthetisierten Moleküle direkt zugeordnet werden. Zum automatisierten Datenabgleich eignet sich GlyConnect²²⁷, eine Plattform zur Integration von Informationsquellen, um molekulare Komponenten der Proteinglykosylierung zu charakterisieren, und KEGG^[394], einer umfassenden Ressource für die Bioinformatik-Analyse von Glykan-Daten. Der Aufbau der Molekülsynthesizer-Datenbank ist eine Grundvoraussetzung für das Synthese-Vorschlagswesen, sowie nachfolgendem Einsatz in der informatischen Modellbildung.

Für Wirkstoffsuche sind eine Reihe hilfreiche Werkzeuge zur Analyse komplexer Moleküle entwickelt worden, die bei der Suche von Struktur-Funktionsbeziehung^[447] nützlich sein können. Bei der automatisierten Strukturidentifikation von Proteinen, helfen Anwendungen wie z. B. DeepNovo^{228[448]} oder die web-basierte Anwendung GlycoEP^{229[449]}, zur Vorhersage von N-, O- und C-Glykosiden in eukaryotischen Proteinsequenzen. In kürzester Zeit lassen sich aufgrund intuitiver grafischer Software wie VMD^[450], OpenRasmol^[451] oder CCP4mg^[452], unkompliziert komplexe statische und dynamische Repräsentationen makromolekularer Strukturen wie Protein- oder Nukleinsäuremodelle erstellen. Ein integriertes Rendering-Modul erzeugt grafische Darstellungen in Veröffentlichungsqualität. Für Kohlenhydrate ist dies aufgrund der aktuell geringen Automatisierungsdaten noch nicht möglich. Um die strukturelle Glykobiologie auf „Proteinstandards“ zu bringen, ist eine umfassende Überarbeitung der Methodik erforderlich.^[295] Hier könnte die Verwendung von HELM als Beschreibungssprache, sowie die Anwendung von Methoden zur Strukturbestimmung^[453], z. B. von Glykoproteinen^[454], von Glykanen unterschiedlicher Herkunft unter Anwendung hochauflösender 1H-NMR-Spektroskopie^[455], sowie die Einführung neuer Werkzeuge wie Glykan-Microarray-Daten oder Glykan-Microarray-Repositorien^[456] nützlich sein.

Aus der interdisziplinären Datenbasis der Synthese-Datenbank können Syntheserouten für das gewünschte Zielmolekül bzw. datenunterstützte Hilfestellung für den vorgesehenen Reaktionsweg abgeleitet werden. Die Anzahl der möglichen Synthesewege, die zum gewünschten Ziel einer

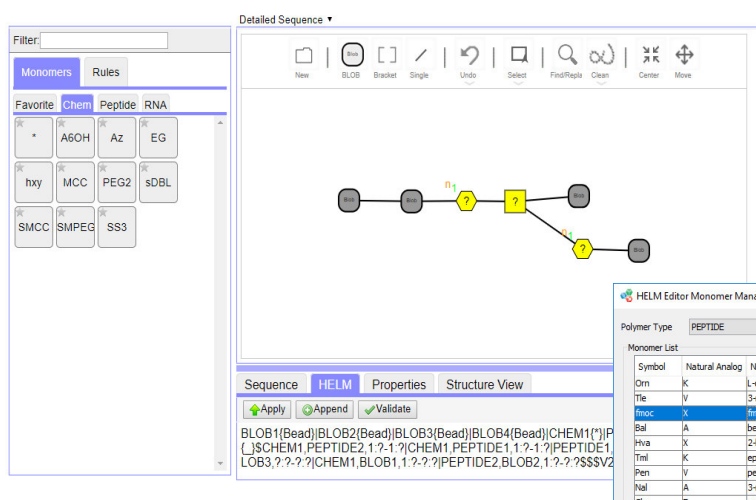
²²⁷ Online unter: <https://glyconnect.expasy.org/> [letzter Abruf: 17.03.2019]

²²⁸ Online unter: <https://github.com/nh2tran/DeepNovo> [letzter Abruf: 17.03.2019]

²²⁹ Online unter: <https://bio.tools/GlycoEP> [letzter Abruf: 17.03.2019]

Synthese führen, kann jedoch astronomisch sein (1019 innerhalb von fünf Syntheseschritten).^[457] Statistische Analysen, die auf Netzwerkkonnektivität basieren, können bei der Optimierung von parallelen Synthesen und bei der Abschätzung der chemischen Reaktivität Verwendung finden.^[458] Deshalb werden in die Syntheser-Applikation Algorithmen integriert, die durch das gesamte bekannte chemisch-synthetische Wissen navigieren, um optimale Synthesewege zu identifizieren. In der Literatur werden Beispiele zur Veranschaulichung von Einzelzieloptimierung und parallelen Optimierung von Synthesen aufgrund von Computeralgorithmen beschrieben, die zu mehreren möglichen Zielpfaden führen^[457] und experimentell mit Hilfe mehrerer zwei-, drei- und sogar vierstufiger Sequenzen validiert wurden. Diese Sequenzen führten zu einer „Neuverdrahtung“ der Synthesewege für populäre und/oder wichtige Verbindungen.^[459]

a)



b)

Symbol	Natural Analog	Name	Structure
Orn	K	L-ornithine	[*]N(C@H)(CCCN[*])C([*])=O [R1];; R2;\$
Val	V	3-methylvaline	CC(C)C(C@H)(N[*])C([*])=O [R1];; R2;\$
fmoc	X	fmoc N-Terminal Protection Group	[*]C(=O)C1=CC=CC=C2C(=C1)C=C(C2)C([*])=O [R1];; R2;\$
Bal	A	beta-Alanine	[*]NCCC([*])=O [R1];; R2;\$
Hva	X	2-hydroxypentanoic acid	CCC(C@H)(O)C([*])=O [R1];; R2;\$
Tml	K	epsilon-N-trimethyllysine	C[N+](C)(C)CCCC(C@H)(N[*])C([*])=O [R1];; R2;\$
Pen	V	penicillamine (3-mercaptopvaline)	CCC(S[*])([*])C([*])=O [R1];; R2;\$
Nal	A	3-naphthylalanine	[*]N(C@H)(C1=CC2=CC=CC=C2C=C1)C([*])=O [R1];; R2;\$
Gla	E	gamma-carboxyglutamic acid	OCC(=O)C(C@H)(N[*])C([*])=O [R1];; R2;\$
Mpa	X	mercaptopropanoic acid	C[C@H](S[*])C([*])=O [R1];; R2;\$
Glc	X	glycolic acid	OCC([*])=O [R1];; R2;\$
Aom	X	2-aminomeclic acid	[*]N(C@H)1CCCCC1[*]C([*])=O [R1];; R2;\$

c)

Abbildung 237: a) Verknüpfung von Monomeren im HELM-Editor; b) Definition der reaktiven Kopplungsstelle; c) Definition eigener Monomere mit bis zu fünf verschiedenen reaktiven Kopplungsstellen

Die Berechnungen für die Datenmodellierung kann aufgrund der Komplexität der gewünschten Zielmoleküle sehr umfangreich sein. Mit der Mathematikplattform Mathematica von Wolfram Research lassen sich auch komplexere Differentialgleichungen symbolisch²³⁰ lösen. Alternative Mathematik-Programme wie MATLAB, OCTAVE²³¹ and SCILAB^[460] können für eine numerische Parameterdiskussionen zur Verifikation der Modellbildung eingesetzt werden. Die gewonnenen und analysierten Daten der Abhängigkeiten lassen sich damit in Gleichungssysteme überführen, die als Grundlage zur Aufstellung komplexer Differentialgleichungen mit eindeutig definierten Störfunktionen dienen, die anschließend z. B. Mathematica^[461] geschlossen symbolisch gelöst werden können und daraus die Grundlagen für die Schrittkette vom maschinellen Lernen zum KI-System darstellen (s. Kapitel Expertensystem). Ergänzend können Anwendungen wie der webbasierte AtomicChargeCalculator^{232[462]} zur Berechnung und Analyse von Atomladungen, welche auf Änderungen der molekularen Konformation und der chemischen Umgebung reagieren, GLYCAM^[463] für Molekulardynamiksimulationen von Kohlenhydraten oder Ambit-SMIRKS^[464], als Software-Werkzeug zur Reaktions-Suche und Strukturtransformation, zum Einsatz kommen. Für die Parameteroptimierung der verschiedenen Synthesewege, kann der Verwendung von parallel arbeitenden virtuellen Maschinen und CPU's zu einer Zeitersparnis für die umfangreichen Berechnungen der Datenmodellierung führen. Zu diesem Zweck findet Open Source Software wie z. B. PML^{233[465]} ihren Einsatz.

Mit der automatischen Dokumentation des „Erfolgs“ oder „Misserfolgs“ einer experimentell verfolgten Strategie u. a. anhand der Analysen, Syntheseparametern und ergänzender Informationen durch den Forscher, wird die Qualität des Vorschlagswesens durch eine sich selbst optimierende Struktur gesichert. Die Synthese-Datenbank wächst damit automatisch zu einer aussagekräftigen Datenbasis heran, aus der in der weiteren Entwicklung das Expertensystem seine Anwendung findet.

²³⁰ Bei der symbolischen Mathematik werden Operation und Kalkulation von mathematischen Ausdrücken mit Variablen auf einem Computeralgebrasystem berechnet. Das Ergebnis symbolischer Mathematik ist im Gegensatz zur numerischen Mathematik universell gültig.

²³¹ Online unter: <https://www.gnu.org/software/octave/> [letzter Abruf: 20.03.2019]

²³² Online verfügbar unter: <http://ncbr.muni.cz/ACC> [letzter Abruf: 18.03.2019]

²³³ Online verfügbar unter: <https://github.com/lmlcic/PML> [letzter Abruf: 18.03.2019]

5.4 Expertensystem – „Maschinelles Lernen“, Simulation, Syntheseplanung

5.4.1 Einleitung

Die Methoden des maschinellen Lernens (ML) sind für die Datenanalyse im Hinblick auf „Big Data“^[234, 466] von großer Bedeutung^[467, 468]. Während sie in der Sprach- oder Bilderkennung^[469], der medizinischen Diagnostik^[470], der Bioinformatik^[471] und der Wirtschaft^[472] erfolgreich Einsatz finden, stecken entsprechende Anwendungen in der Chemie, wie das computergestützte Reaktionsdesign^[473], noch in der frühen Entwicklungsphase.^[474] Dabei fehlt die wesentliche Grundlage: eine qualitativ aussagekräftige, umfangreiche Datenbasis.

Die Anzahl von Literaturbeispielen, an denen ML-Modelle in der Chemie trainiert werden können, ist vollkommen unzureichend. Geschätzt gibt es in der Größenordnung von 10 Millionen bekannte Reaktionen,^[475] aber bis zu 20.000 – 30.000 verschiedene Reaktionstypen, was bedeutet, dass der analytischen Statistik für das maschinelle Lernen nur wenige hundert Beispiele pro Reaktionstyp zur Verfügung stehen, was im Allgemeinen nicht ausreichend ist, um die Kombinationen möglicher Substituenten, sterische und elektronische Effekte usw. abzudecken. Zudem werden organische Reaktionen aufgezeichnet, ohne alle Reagenzien oder Nebenprodukte, Aufreinigungsschritte, Zeiteffekte und Qualitätskontrollen zu berücksichtigen. Das Problem der unvollständigen Stöchiometrie wird durch die inhärente Mehrdeutigkeit der berichteten Reaktionsergebnisse noch verschärft (dh. selbst bei Reaktionen, die von demselben Team durchgeführt werden, können die Ausbeuten erheblich variieren, was eine systematische Überprüfung und „Reinigung“ von Reaktionsablagen wie Reaxys²³⁵ oder SciFinder²³⁶ erfordert.^[476] Die zusätzliche Problematik fehlender Interoperabilität der Daten unterschiedlicher Forschungsbereiche, dem Fehlen umfassender Beschreibungen relevanter experimenteller Parameter, Berechnungsmethoden und zugrunde liegender Annahmen, wurde bereits im Kapitel Datenarchivierung / Datenmanagement, sowie ELN/LIMS für das Konzept des Molekülsynthesizers thematisiert.

Der Aufbau einer für das „maschinellen Lernens“ geeigneten Datenbasis bildet die Grundlage für das nachfolgend vorgestellte Konzept des Expertensystems. Der Molekülsynthesizer ist prädestiniert als ausführendes System für kombinatorische Untersuchungen, da er eine schnellere Durchführung chemischer Reaktionen und Analysen liefert, als dies manuell der Fall wäre und aufgrund des modularen, flexiblen Aufbaus in der Lage sein wird, sowohl Peptide, Oligonukleotide, als auch Oligosaccharide herzustellen. Die unmittelbare Verknüpfung von in silico generierten Ansätzen mit

²³⁴ Big Data bezeichnet große Mengen strukturierter und unstrukturierter Daten, die aufgrund ihres Umfangs, ihrer Komplexität oder ihrer Veränderlichkeit extrem anspruchsvoll für die analytische Datenauswertung sind und deshalb nicht mit herkömmlichen Methoden der Datenverarbeitung auszuwerten sind.

²³⁵ Online unter: <https://www.reaxys.com> [letzter Abruf: 20.03.2019]

²³⁶ Online unter: <https://scifinder.cas.org> [letzter Abruf: 20.03.2019]

experimentellen Daten ist ein äußerst vorteilhafter Aspekt beim Aufbau und Verifizierung der qualitativen, strukturierten und umfassenden Datenbasis, wodurch selbst komplexe Reaktionen, wie Glykosylierungen, schnell, reproduzierbar und systematisch abgesucht werden können, um z. B. zuverlässig die stereo- und regioselektive Installation glykosidischer Bindungen zu ermöglichen. In einer aktuellen Studie konnten dadurch bereits Glykosylierungen mit den automatischen Synthesizer-Systemen, an denen zwei spezifische Partner beteiligt waren, so eingestellt werden, dass sich 11:1 - Selektivität eines Stereoisomers oder 9:1 des anderen durch bloße Veränderungen der Reaktionsbedingungen erzeugen ließen.^[477]

Die im Konzept des Molekülsynthesizers vorgesehene vollkommene Standardisierung von Prozessen und umfassende Echtzeit-Reaktionsüberwachung z. B. durch Verwendung von Kernspinresonanz- und Infrarotspektroskopie, sowie einer Vielzahl zusätzlicher weiterer Prozessparameter, ermöglicht die Sammlung und deren standardisierte Verarbeitung wertvoller Daten für die automatische Analyse über die gesamte Reaktionszeit und den vollständigen Reaktionspfad hinweg. Aufgrund des integrierten Workflows werden zusätzlich experimentelle Daten von Teilreaktionen, aber auch die lückenlose Dokumentation von Negativdaten gewonnen, also Experimenten, die nicht zum gewünschten Erfolg führten, jedoch für das maschinelle Erlernen von Evidenzen ebenso notwendig sind wie Positivdaten. Aufgrund der gewonnenen, strukturellen Experimentaldaten an den verschiedenen vernetzten automatischen Molekülsynthesizern lässt sich so, unabhängig vom aktuellen Standardisierungs-Stand der verschiedenen Forschungsbereiche bezüglich des Datenmanagements, ein eigener valider Datenbestand aufbauen, der sich für das synthesebezogene maschinelle Lernen eignet.

Mit fortschreitender Datenbasis und unter Einführung regelungstechnischer Konzepte zur Optimierung chemischer Synthesen während der Prozessführung und der Prozessmodellierung^[13], eignet sich der Molekülsynthesizer zum Einsatz als cyber-physisches „Chemie-System“ für die anforderungsbezogene Synthese.

Das Expertensystem definiert sich als die artifizielle Intelligenz des Molekülsynthesizers. Eine Rückkopplung der am Expertensystem gewonnenen Erkenntnisse kann nachfolgend in einer verbesserten Software für den Anwender resultieren. Die Molekülsynthesizer, die als verteilte Systeme vernetzt sein können, dienen wiederum als wertvolle Datenlieferanten für die virtuellen „Analyse- und Simulations-Module“ des Expertensystems, wenn die Daten zur Veröffentlichung frei gegeben sind, und stellen so eine kontinuierliche Qualitätsverbesserung für die Modellbildung und Prozessoptimierung sicher.

5.4.2 Datenaufbereitung, Maschinelles Lernen & Modell-Entwicklung

Für das Expertensystem ist ein Programm zu entwickeln, welches aus einer Vielzahl von Beispielen Muster und Gesetzmäßigkeiten ableiten kann. Dieses chemische „KI“-Programm implementiert den Prozess des „maschinellen Lernens“ (ML).^[478] Dieser Lernprozess speist sich einerseits aus Basisinformationen aus den Open-Access-Datenbanken, andererseits aus einer großen Anzahl von Beispieldaten, die durch die Synthesen und über Online-Analysen und Echtzeit-Überwachung am Synthesizer gewonnen werden. Dies stellt die Grundlage des „automatischen Lernens“ bereit, aus denen, mit Hilfe von Mathematica und Matlab in Kombination mit chemischen Workflow-Systemen die komplexen Algorithmen zur Modellbildung des selbstoptimierenden Expertensystems entwickelt werden. Hierbei gilt: je größer die zur Verfügung stehende Datenmenge, desto höher der Lerneffekt. Im Regelfall werden 80% der Basis- und Beispieldaten für die Mustererkennung und das Modelkonzept ausgewertet. Nach Beendigung der umfangreichen Lernphase, erfolgt die Verallgemeinerung der Erkenntnisse in einer Modellbildung. Dabei helfen die Workflow-Programme künstliche Prozesse abzubilden. Mit den restlichen 20% der Basis- und Beispieldaten erfolgt anschließend die Validierung des gebildeten Modells. Bei einem selbstlernenden System ist die Software aufgrund der Modellbildung aus der Datenanalyse imstande selbstständig Aufgaben auszuführen, anstatt für die Ausführung dieser Aufgabe manuell programmiert zu werden.

Nachfolgend werden die Werkzeuge vorgestellt, mit denen die Bedienung des Expertensystems optimieren lässt und die zum Aufbau der systeminternen Geräte-KI für die Simulationsprozesse bezüglich Reaktionsvorhersage oder kombinatorischen Berechnungen dienen. Dabei kommen Methoden der Ähnlichkeitssuche (Similarity Learning)^[479a,479b,480,479c,479d] zum Einsatz, wie der k-Nearest-Neighbor-Algorithmus (kNN)^[481], das Bipartite Local Model (BML)^[482,483] und regularisierte Matrix-Faktorisierung^[484], sowie ML-Algorithmen, wie die logistische Regression^[485], Support Vector Machines (SVM)^[486], neuronale Netze^[487], extrem randomisierte Bäume (ERT)^[488] und Random-Forests (RF)^[489,490]. Wie bereits im Kapitel Synthesepaltung beschrieben, handelt es sich dabei um sehr aufwendige Rechenprozeduren, für die aus Performancegründen ein parallel arbeitendes Netzwerk an Prozessoren und / oder verteilten virtuellen Maschinen notwendig sein wird.

Die durch modularisierte Datenerfassung erhaltenen ungefilterten Daten müssen durch geeignete Verfahren analysiert und verwertbar aufbereitet werden. Organisation und Analyse der riesigen Datenmengen stellt eine immense Herausforderung dar, für die neue, effektive Methoden des Text-Data-Mining²³⁷ zu installieren sind. So ermöglicht das Open-Source-Chemoinformatik-Toolkit RDKit²³⁸

²³⁷ Algorithmus-basiertes Analyseverfahren zur Entdeckung von Bedeutungsstrukturen aus un- oder schwachstrukturierten Textdaten.

²³⁸ Online unter: <https://www.rdkit.org/> [letzter Abruf: 23.03.2019]

mit der Methode „CheTo“, große Molekülsätze "chemischen Themen" zuzuordnen und die Beziehungen zwischen diesen zu untersuchen.^[491] Das RDKit ist im pharmazeutischen Bereich weit verbreitet und auch Bestandteil der Workflow-Software KNIME^{239[492]}, die generell mit einer Reihe von Erweiterungen Methoden für Text Mining, Image Mining, sowie Zeitserienanalyse, eine mächtige Plattform für die Big-Data-Analyse inklusive Methoden des maschinellen Lernens bereitstellt. Der Quell-Nodes²⁴⁰ des „PDB-Connectors“ bieten beispielsweise Verbindungen zu RESTful-Webdiensten für das Suchen und Abrufen von Informationen aus der Proteindatenbank. Seit Juni 2013 sind die KNIME-Nodes von Vernalis²⁴¹ von einem einzelnen Node ("PDB-Connector") auf etwa 126 Nodes (November 2017; Version 1.12.0) gestiegen und beinhalten Nodes, die über Web-Services auf öffentlich zugreifbare Informationen zugreifen, Nodes, die Cheminformatik-Funktionalität bereitstellen, ohne auf ein Cheminformatik-Toolkit zurückzugreifen, und Nodes, die eines der in KNIME vorhandenen Cheminformatik-Toolkits verwenden.^[493] Für KNIME und das RDKit²⁴² stehen mit Open3DQSAR^[494] und Open3DALIGN^[495] auch 3D Werkzeuge zur Verfügung.^[496]

Mit dem semi-automatisierten Mehrfach-Massenspektrometrie-Dateninterpretierer GELATO²⁴³ kann die Funktionalität der GlycoWorkbench^[319,283] und Glycome-DB erweitert und automatisiert werden. Das maschinelle Lernmodell SAGE²⁴⁴ trainiert das Verhalten von Glykoanalysten, um zur Auswahl von GELATO erzeugten Annotationen eine menschliche Interpretation der Spektren nachzuahmen.^[497] Die Anwendung Sweet-Heart^{245[498]} kann den Workflow zur automatischen Identifizierung intakter Glykopeptide aus LC-MS 2-Spektraldaten unterstützen. Hierzu wurde für die Modellbildung mittels maschinellen Lernens ein überwachter Random-Forest-Algorithmus^{246[489]} in den automatisierten Workflow implementiert. Nach Modelloptimierung konnte damit im Split-Sample-Testverfahren²⁴⁷ aus 577 hochzuverlässigen N-Glykopeptid-Spektraldaten, eine optimale True-Positiv-Rate, Präzision und False-Positiv-Rate von 73, 88 bzw. 10% für das gesamte N-Glykopeptid erreicht werden.^[499]

Die Anwendungen des maschinellen Lernens zur Erstellung von Vorhersagemodellen bei der Suche nach neuen Medikamenten auf dem Gebiet der niedermolekularen Wirkstoffforschung^[500] reichen

²³⁹ Abkürzung für „Konstanz Information Miner“, freie Software für die interaktive Datenanalyse.

Online unter: <https://www.knime.com/> [letzter Abruf: 20.03.2019]

²⁴⁰ Nodes bezeichnet in KNIME die kleinste Programmierereinheit. Jedes Node erfüllt eine dedizierte Aufgabe.

²⁴¹ Vernalis Research entwickelt und wendet fragment- und strukturbasierte Methoden für die Wirkstoffforschung an und hat zellaktive Leitverbindungen und Entwicklungskandidaten gegen biologische Ziele in der Onkologie, Neurodegeneration,

Antibiostatika und Entzündungen entwickelt. Online unter: <https://www.vernalis.com/> [letzter Abruf: 23.03.2019]

²⁴² Online unter: <http://www.rdkit.org/> [letzter Abruf: 24.03.2019]

²⁴³ Abkürzung für engl. „Glycomic Elucidation and Annotation“

²⁴⁴ Abkürzung für eng. „Smart Annotation Enhancement Graph“

²⁴⁵ Online unter: <http://sweet-heart.glycoproteomics.proteome.bc.sinica.edu.tw/> [letzter Abruf: 23.03.2019]

²⁴⁶ Statistisches Klassifikationsverfahren, bestehend aus mehreren unkorrelierten Entscheidungsbäumen.

²⁴⁷ Testmethode zur Bewertung zweier Varianten eines Systems, bei der die Originalversion gegen eine leicht veränderte Version getestet wird

von der Vorhersage von ADME²⁴⁸-Eigenschaften, Zielaktivitäten^[501] bis hin zu Aufgaben, wie das molekulare Design und die Vorhersage chemischer Reaktionen.^[483] Die jüngsten Fortschritte bei systemorientierten Ansätzen in der Chemoinformatik^[502], Systembiologie, Pharmakologie und allgemein Omics²⁴⁹-Technologien, haben zu einer Fülle neuer Computertools geführt, die eine fundiertere und erfolgreichere Implementierung des reduktionistischen Ziels der zielorientierten Wirkstoffsuche,^[503] bis zur personalisierten Individualmedizin ermöglichen können. Methoden, die die Integration von Glykomics-Messungen zusammen mit anderen Omics-Datensätzen erleichtern, können zu einem tieferen Verständnis und umfassenderen Einsichten in die Natur der Glykosylierung als komplexen zellulären Prozess führen.^[504]

Mit der Implementierung pharmakologischer Werkzeuge unter der einheitlichen Synthesizer-Plattform sind auch für die interdisziplinäre und verwandte Forschung Vorteile zu erwarten. In den mehrheitlichen Open-Source Anwendungen können zudem fachspezifisch Funktionalitäten ergänzt werden, wodurch sie eine effiziente Alternative zur kompletten Neuentwicklung darstellen, um zielgerichtet und schnell belastbare Resultate zu erzielen.

Diese Anwendungen sollten jedoch möglichst funktionale Schnittstellenstandards enthalten, um sich in die verschiedensten Werkzeuge zu integrieren. Eine geringe Anlernphase für den Forscher, sowie der Zugriff auf Daten und Funktionen unter Verwendung der fachspezifisch üblichen Nomenklatur, ist vorteilhaft. Das Konzept von OpenPHACTS^[441] unterstützt diese Philosophie und stellt eine öffentlich zugängliche Infrastruktur pharmakologischer, physikochemischer und biomolekularer Daten zur Verfügung, die in Verbindung mit Workflow-Tools zur Lösung pharmakologischer Fragestellungen, die Interoperabilität zwischen Ziel-, Substanz- und Wegedaten verfügbar machen soll.^[505] 3D-e-Chem-VM²⁵⁰ ist eine frei verfügbare, quelloffene virtuelle Maschine für die strukturelle Chemoinformatik-Forschung, welche die Entwicklung neuer Ansätze für das virtuelle Ligandenscreening, die ligandenbasierte Metabolismusvorhersage, sowie den Vergleich von strukturellen Proteinbindungsstellen und den Ersatz von bioisosterischen Substanzen für Ligandendesign ermöglicht.^[506] BALLView^[507] kann verschiedene Aspekte von Wirkung und Funktion biologisch aktiver Substanzen effizient verwalten und präsentieren. Es kombiniert modernste Visualisierungsfunktionen mit leistungsstarken Modellierungsfunktionen, einschließlich der Implementierung von Kraftfeldmethoden und Kontinuumselektrostatikmodellen.

²⁴⁸ Abkürzung für „Absorption, Verteilung, Metabolismus und Ausscheidung“; Beschreibt in der Pharmakologie die Disposition einer pharmazeutischen Verbindung innerhalb eines Organismus.

²⁴⁹ Bezeichnet in der Biologie molekularbiologische Methoden, die auf "-omik" (engl. "-omics") enden, wie Genomik, Proteomik, Metabolomik u.a.

²⁵⁰ Online unter: <http://3d-e-chem.github.io/3D-e-Chem-VM> [letzter Abruf: 23.03.2019]

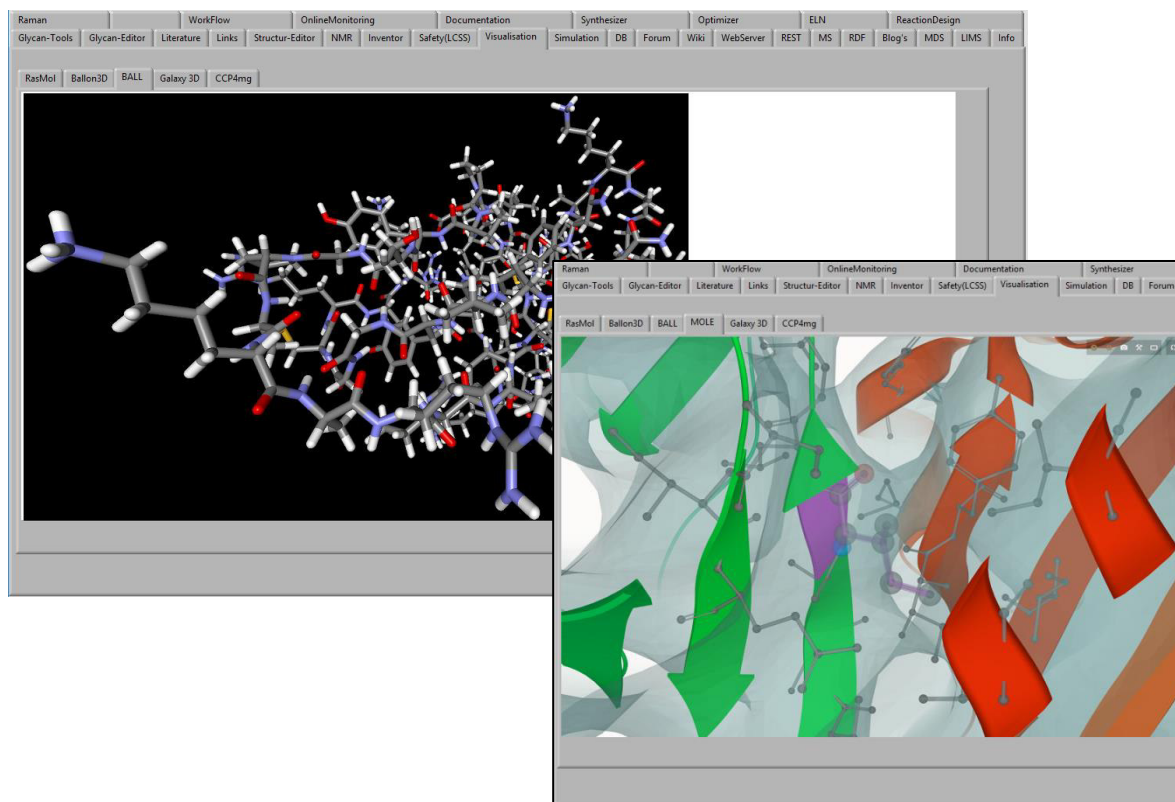


Abbildung 238: Implementierung von BallVIEW (ob.) oder MOLE (re.), einem leistungsstarken Visualisierungs- und Modellierungswerkzeug für die Konzept-Studie des Molekülsynthesizers.

Dynamische Simulationsmodelle für die molekulare Vorhersage, wie das kommerzielle Jaguar der Firma Schröder²⁵¹, können auf Basis dynamischer Energiebetrachtung durchgeführt werden. Das ab-initio Quantenchemie²⁵² - Programm ist auf schnelle elektronische Strukturvorhersagen für molekulare Systeme mittlerer und großer Molmasse spezialisiert ist und findet bei der Modellierung von biomolekularen Berechnungen Einsatz. Anwendungen finden sich in der anorganischen und bioanorganischen Chemie, in der Wirkstoffforschung, Materialforschung, sowie der Kraftfeld-Parametrisierung und anderen Bereichen der Computerforschung.^[508]

Hierbei sind Aussagen zu komplexen Systemen aus dem Lösungsmittelverhalten (GAMESS^[509–511], MOPAC^{253[512]}), den Faltungsvarianten von Biomolekülen, sowie den Kopplungsmöglichkeiten von Proteinen herleitbar (QSAR^{254[513,514]}, CHARMM^[515,516]), z. B. aufgrund von QSARs, quantitativen Struktur-Aktivitäts-Beziehungen, die chemische Strukturen unter Verwendung statistischer Ansätze mit der Aktivität korrelieren und sich für verschiedene Zwecke, einschließlich der Vorhersage von Aktivitäten ungeprüfter Chemikalien eignen,^[514] die für die Registrierung neuer chemischer Substanzen nach der europäischen REACH Verordnung²⁵⁵ erforderlich ist.

²⁵¹ Online Informationen: <https://www.schrodinger.com/jaguar> [letzter Abruf: 19.03.2019]

²⁵² Begriff aus der Quantenchemie, bezeichnet ein Verfahren der Elektronentheorie von Atomen, Molekülen und Festkörpern, die keine an die experimentellen Daten angepassten Parameter verwenden.

²⁵³ Online unter: <http://openmopac.net/> [letzter Abruf: 19.03.2019]

²⁵⁴ Online Toolbox verfügbar unter: <https://qsartoolbox.org/> [letzter Abruf: 19.03.2019]

²⁵⁵ Online Informationen unter: <https://echa.europa.eu/de/regulations/reach/understanding-reach> [letzter Abruf: 19.03.2019]

Posttranslationale Modifikationen kommen in der überwiegenden Mehrheit der Proteine vor. Die Glykosylierung ist eine wichtige posttranslationale Modifikation, die mit der Proteinfaltung, dem intrazellulären Transport und biologischen Funktionen in Verbindung stehen. Die Applikation GPP²⁵⁶ verwendet den Random-Forest-Algorithmus^[489] und den „Paarweisen Vergleich“^[257], um Glykosylierungsstellen vorherzusagen. Das Programm prognostizierte Glykosylierungsstellen mit einer Genauigkeit von 90,8% für Serin -Zentren, 92,0% für Threonin -Zentren und 92,8% für L-Asparagin -Zentren.^[517] Indigo^{258, [518]} ist eine universelle portable Open-Source-Bibliothek basierend auf C++, die wichtige chemische Eigenschaften unterstützt und u. a. Werkzeuge zur Ähnlichkeitsberechnung und Kombinatorik bereitstellt. Mit dem web-basierten Chemical Kinetics Simulator²⁵⁹ können stochastische Simulationen chemischer Reaktionen aufgrund kinetischer Approximationen und die Modellierung oszillatorischer Reaktionen durchgeführt werden.^[519] Die Software CHARMM verwendet empirische Energiefunktionen zur Modellierung makromolekularer Systeme,^[520,515] einschließlich der Modellierung von Phosphat und Sulfat, das an Kohlenhydrate gebunden ist.^[516] GROMACS ist ein sehr schnelles, vielseitiges Programm für die Simulation der Molekulardynamik, welches Schnittstellen zu den gängigen quantenchemischen Paketen (MOPAC^[512], GAMESS^[509–511], GAUSSIAN^{260, [521]}) bereitstellt, um gemischte Quantenmechanik / Molekularmechanik - Simulationen durchzuführen.^[26] Mit der NWChem Computational²⁶¹ - Chemiesoftware und Avogadro Molecular Editor zur Analyse und Visualisierung^[522], lassen sich vielfältige Datensätze aus einer Reihe komplexer Experimente und Computersimulationen innerhalb der multidisziplinären integrierten Forschung koppeln.

Für die Implementierung des „maschinellen Lernens“ in die Expertensystem-Applikation kommen Open-Source Ressourcen zur Anwendung, wie die Workflow-Software KNIME oder RapidMiner²⁶², die Programmbibliothek ML.NET inklusive infer.NET für .NET-Sprachen²⁶³ von Microsoft, TensorFlow²⁶⁴ von Google, PyTorch²⁶⁵ für Python, OpenNN²⁶⁶ für C++ und Python, die in Java programmierten Data-

²⁵⁶ Abkürzung für Glycosylation Prediction Program;

Online unter: <http://comp.chem.nottingham.ac.uk/glyco/> [letzter Abruf: 23.03.2019]

²⁵⁷ Statistik Methode zur Qualitätsbewertung.

²⁵⁸ Online unter: <http://lifescience.opensource.epam.com/indigo/> [letzter Abruf: 24.03.2019]

²⁵⁹ Online unter: <https://www.chem.uci.edu/undergraduate/applets/sim/simulation.htm> [letzter Abruf: 20.03.2019]

²⁶⁰ Online unter: <http://gaussian.com/gaussian16/> [letzter Abruf: 19.03.2019]

²⁶¹ Online unter: http://www.nwchem-sw.org/index.php/Main_Page [letzter Abruf: 19.03.2019]

²⁶² Online unter: <https://rapidminer.com/> [letzter Abruf: 20.03.2019]

²⁶³ Online unter: <https://www.nuget.org/packages?q=Microsoft.ML.Probabilistic> [letzter Abruf: 14.03.2019]

²⁶⁴ Online unter: <https://www.tensorflow.org/> [letzter Abruf: 14.03.2019]

²⁶⁵ Online unter: <https://pytorch.org/> [letzter Abruf: 14.03.2019]

²⁶⁶ Online unter: <http://www.opennn.net/> [letzter Abruf: 14.03.2019]

Mining Applikationen ELKI²⁶⁷ und WEKA²⁶⁸, die Statistiksoftware GNU R²⁶⁹ oder das Framework Caffe²⁷⁰ und die Programmbibliothek für Java Deeplearning4²⁷¹ zur Anwendung neuronaler Netze.

5.4.3 Simulation & KI

Die Entdeckung neuer chemischer Reaktionen ist von Natur aus ein unvorhersehbarer und zeitaufwändiger Prozess^[523], der jedoch mit Hilfe automatisierter Verfahren systematisch erkundet werden kann. Es erfordert das Auffinden passender Reaktionstypen, die Berücksichtigung potenzieller Reaktivitätskonflikte, Nichtselektivitäten, Nebenreaktionen und Schutzanforderungen, sowie die Literaturrecherche nach ähnlichen Reaktionsvoraussagen.

Unterstützt von den Strategien, die auf der Intuition von Chemikern basieren^[524] und den Methoden des maschinellen Lernens, können Algorithmen^[468] einen Großteil der vorhandenen chemischen Daten analysieren und daraus Modelle über verschiedene Aspekte der chemischen Reaktivität oder für die Entwicklung und Bewertung chemischer Routen^[525] ableiten.

Häufig basiert die Modellbildung für chemische Reaktionen auf der Reaktionsnetzwerktheorie, die das qualitative Verhalten der Steady-State²⁷² Konzentration eines chemischen Reaktionsnetzwerkes ohne Verwendung der kinetischen Parameter untersucht. Dabei werden mathematisch grafische und algebraische Strukturen eines Reaktionsnetzwerks mit den qualitativen Eigenschaften eines komplizierten Systems nichtlinearer Differentialgleichungen in Beziehung gesetzt.^[526] Dieser Ansatz ist in der Lage die Existenz von multiplen Steady-States für chemische Reaktionen, die dem Massenwirkungsgesetz unterliegen, wie z. B. alle Reaktionen der Glykose, vorherzusagen oder auszuschließen und kommt ohne die Verwendung numerischer Verfahren aus.

Vereinfacht lässt sich das dynamische Verhalten von chemischen Reaktionsnetzwerken durch Lösen der aus ihrer Struktur abgeleiteten Polynomgleichungen untersuchen. Die daraus resultierenden nichtlinearen Polynome sind jedoch schwer aufzulösen, weshalb als vereinfachtes analytisches Verfahren eine Ermittlung der stationären Zustände durch algebraische Geometriemethoden vorgeschlagen wurde.^[527]

Mit der Mathematikplattform Mathematica von Wolfram Research lassen sich komplexe nichtlineare Differenzialgleichungen elegant lösen. Hierbei ist es von Vorteil, dass Mathematica in der aktuellen Version umfassend und direkt auf die physikalischen und chemischen Eigenschaften der Moleküle/Substanzen zugreifen kann. Es stehen dadurch Werte wie die Entropie, freie Energie,

²⁶⁷ Online unter: <https://elki-project.github.io/> [letzter Abruf: 14.03.2019]

²⁶⁸ Online unter: <https://www.cs.waikato.ac.nz/ml/weka/> [letzter Abruf: 14.03.2019]

²⁶⁹ Online unter: <https://www.r-project.org/> [letzter Abruf: 14.03.2019]

²⁷⁰ Online unter: <http://caffe.berkeleyvision.org/> [letzter Abruf: 14.03.2019]

²⁷¹ Online unter: <https://deeplearning4j.org/> [letzter Abruf: 14.03.2019]

²⁷² Zustand chemischen Gleichgewichts, wenn die chemischen Reaktionen gleich schnell in beide Richtungen ablaufen.

chemisches Potential und vieles mehr als kalkulatorische Parameter und Variablen zur Verfügung. Es wurde auch bereits ein Programmpaket zur Berechnung der Reaktionskinetik²⁷³ entwickelt.^[528]

In der Literatur finden sich Anwendungsbeispiele zum Lösen chemischer Problemstellungen mit Mathematica, z. B. die Beschreibung eines Verfahrens zum Lösen von Reaktionsgleichungen, welches auf Systeme beliebiger Komplexität anwendbar ist^[529] oder zur theoretischen Untersuchung kinetischer Mechanismen oszillierender und chaotischer Reaktionen^[530].

Zur ressourcenschonenden Verwaltung der an den automatisierten Systemen entstandenen Experimentaldaten, bietet sich ein eigenständiger Web-Services an. Der Vorteil gegenüber einer reinen Datenbank ergeben sich aus der Flexibilität und der zukünftigen Wiederverwendbarkeit der benannt abgelegten Daten. Durch die Standardisierung der Struktur ist die Interaktion von unterschiedlichen Applikationen sowohl hersteller- wie auch systemübergreifend möglich. Über den weltweit möglichen Zugriff mit Hilfe einer URL²⁷⁴ werden die Daten/Informationen unter Verwendung von HTTP²⁷⁵ austauschbar. Durch diese Mechanismen sind die Nutzer sowohl von der Software als auch von der Plattform unabhängig. Innerhalb der jeweiligen Programmierumgebung können die optimalen Methoden zur Kommunikation verwendet werden. Der Vorteil von Webservices ist die programmatische Flexibilität. Da nur Funktionen und Daten zur Verfügung gestellt werden, können diese in der weiterführenden Verwendung beliebig kombiniert und kaskadiert werden. Hierdurch lassen sich Abfragen und Strukturinterpretationen realisieren, die zum Zeitpunkt der Erstellung des Service noch nicht absehbar waren. Da die Kommunikation des Client zu dem jeweiligen Service auf Basis von HTTP erstellt werden, kann textgeneriert und einfachen Übertragungsprotokollen gearbeitet werden. Die Visualisierung abgefragter Daten kann z. B. mit Rasmol²⁷⁶ oder PDB2multiGIF²⁷⁷ in einer stereoskopischen Darstellung intuitiver visualisiert werden.

Die Reaktionsvorhersage, die auf quantenchemische Methoden auf hohem Niveau beruht, ist selbst für einfache Moleküle komplex^[531] und die Anwendbarkeit des maschinellen Lernens bleibt noch auf verschiedene Arten von Chemikalien begrenzt. Insbesondere bei den derzeit verfügbaren chemischen Deskriptoren²⁷⁸ setzen die grundlegenden mathematischen Theoreme der Genauigkeit, mit der Reaktionszeiten und -Ausbeuten vorhergesagt werden können, Obergrenzen. Um die Leistungsfähigkeit der Methoden des maschinellen Lernens zu verbessern, müssen basisorientierte neue chemische Deskriptoren entwickelt werden, die nicht nur die Konnektivität molekularer

²⁷³ Online Demo: <http://demonstrations.wolfram.com/DescriptiveReactionKinetics/> [letzter Abruf: 24.03.2019]

²⁷⁴ Abkürzung von engl. Uniform Resource Locator

²⁷⁵ Abkürzung von engl. Hypertext Transfer Protocol

²⁷⁶ Online unter: <http://www.openrasmol.org/> [letzter Abruf: 22.03.2019]

²⁷⁷ Online unter: <http://www.glycosciences.de/modeling/pdb2mgif/> [letzter Abruf: 22.03.2019]

²⁷⁸ Deskriptoren bezeichnen in der chemischen Nomenklatur den Präfix vor dem systematischen Substanznamen, der die Konfiguration oder die Stereochemie des Moleküls beschreibt.

Diagramme berücksichtigen, sondern auch die stereoelektronischen Eigenschaften und dreidimensionalen Konformationen von Molekülen.^[476] Auch die Vorhersageleistung bei Wirkstoff-Ziel-Wechselwirkungen hängt stark von den Deskriptoren und Ähnlichkeitsmaßen von Arzneimitteln und Proteinen ab.^[501] Zudem ist aufgrund der geringen Anzahl bekannter Wirkstoff-Zielpaare der aktuelle Datensatzstamm unausgewogen. Einige Modelle wie Entscheidungsbäume²⁷⁹ und Support-Vektor-Maschinen²⁸⁰ haben eine große Neigung zur Erkennung der Majoritätsklasse und führen daher zu einer schlechten Leistung.^[480] Kürzlich wurde mit einer Kombination aus 15000 experimentellen Reaktionsaufzeichnungen aus der US-Patentliteratur und maschinellen Lernens ein Vorhersagemodell für organische Reaktionsprodukte trainiert. Bei einer fünffachen Kreuzvalidierung²⁸¹ ordnete das trainierte Modell in 71,8% der Fälle den Hauptproduktang 1, in 86,7% der Fälle ≤ 3 und in 90,8% der Fälle den Rang ≤ 5 zu.^[532] Thermodynamische Modelle^[533] erreichten in einer Untersuchung nur eine Vorhersagegenauigkeit von $\pm 15\%$ für Reaktionen mit manuell kursorierter Vollstöchiometrie^[476], was größtenteils auf eine für die vollständige Modellbildung unvollkommene Datenbasis zurückzuführen ist.

Für die strukturelle Weiterentwicklung ist die noch mangelhafte Laborautomatisierung und die in weiten Teilen unvollständigen Aufzeichnungen der Versuchsdurchführung und Analyseergebnisse, zunehmend ein Innovationshindernis, da es den Einsatz fortschrittlicher technologischer Verfahren für die strukturelle Forschung behindert. Ein System, welches auf der einen Seite die strukturellen Beschränkungen aufhebt, weil es zugleich die vollständige experimentelle Verifikation aus den theoretischen Daten von Simulation und Modellbildung übernehmen kann und darüber hinaus in der Lage ist, automatisch und systematisch Parametervariationen zum Zwecke der strukturierten, verbesserten Modellbildung unter standardisierter und automatisierter Dokumentation aller am System anfallender Daten zu fahren, würde in dieser Form einmalig sein.

Es gibt Ansätze, die vorhandene, kuratierte Daten in Reaktionsvorhersagen oder Synthesevorschläge überführen^[534], wie das Softwareprogramm Chematica^[293,535], das für die Planung mehrstufiger Synthesen entworfen wurde, und in einer aktuellen Veröffentlichung erfolgreich Synthesen zu acht kommerziell wertvollen und / oder medizinisch relevanten Zielen entworfen hat. Chematica zeigt Routen als Wegkurven, eine Reihe von Kreisen und Linien, an, und der Benutzer kann die Routen anhand bevorzugter Kriterien priorisieren, z. B. minimierte Ausgangsmaterialkosten oder verbesserte Routenneuheiten. Das Programm entwirft und wertet effizient viel mehr Routen aus, als ein Mensch sie gemäß den vom Benutzer festgelegten Kriterien bewerten könnte, und unterbreitet dabei auch unkonventionelle Vorschläge. 2018 wurde Chematica von MilliporeSigma / Merck KGaA

²⁷⁹ Random Forest Klassifikationsverfahren (RF)

²⁸⁰ Support-Vector-Machines (SVM), Klassifikationsverfahren

²⁸¹ Kreuzvalidierung ist ein Standardwerkzeug der Datenanalyse und wichtige Funktion bei der Entwicklung und Feinabstimmung von Data Mining Modellen.

übernommen.^[536] Es gibt Beispiele, die im kleinen Maßstab strukturelle, experimentelle Parametervariationen an begrenzt automatisierten Laboreinheiten durchführen, um den Reaktionsraum zu erkunden. In einer aktuellen Veröffentlichung führte ein durch maschinelles Lernen gesteuerter organischer Syntheseroboter chemische Reaktionen und Analysen durch und konnte damit die Reaktivität von etwa 1.000 Reaktionskombinationen mit einer Genauigkeit von mehr als 80 Prozent vorhersagen, nachdem Ergebnisse von etwas über 10 Prozent des Datensatzes berücksichtigt wurden. Dabei konnten vier neue Reaktionen entdeckt werden.^[537]

Eine multidisziplinäre, interoperable Plattform, wie mit dem Konzept des Molekülsynthesizers beschrieben, ist zur Zeit noch nicht in Sicht, auch wenn Initiativen wie „Dial-a-Molecule“²⁸² in diese Richtung weisen. Das Expertensystem geht über den Wunsch ein Zielmolekül intuitiv und geführt automatisch zu synthetisieren hinaus, es kann durch systematische, experimentelle Grundlagen-Synthesen den Reaktionsraum Schritt für Schritt und mit wachsender Komplexität erkunden und so die „blinden“ Flecken in der Datenbasis für die kybernetische Modellbildung füllen.

²⁸² Online unter: <http://generic.wordpress.soton.ac.uk/dial-a-molecule/> [letzter Abruf: 24.03.2019]

ZUSAMMENFASSUNG

Am Anfang dieser Arbeit stand die Idee, einen Oligosaccharid-Synthesizer zu erstellen, der einfacher zu bedienen ist, wie die zwei existierenden J-KEM-Systeme auf dem damaligen technischen Stand. Im Optimum sollte ein Nichtchemiker in die Lage versetzt werden, eine Synthese nach kurzer Einarbeitung eigenständig durchzuführen. Hierzu sollte ein vollständig neues Bedienkonzept erstellt und umgesetzt werden.

Durch die umfangreichen Validierungsanalysen, bei denen individuell entwickelte Hardware und Software eingesetzt wurde, konnten die Schwachstellen und Konzeptionsfehler in den bisherigen Systemen eindeutig bestimmt und verifiziert werden. Die daraus resultierenden

Instabilitäten/Abbrüche der Gesamtsynthese und somit Schwankungen in der Reproduzierbarkeit oder sogar Verlust der chemischen Resultate konnten auch durch intensive Recherche, sowohl bei den ehemaligen Anwendern sowie im interaktiven Dialog mit aktuellen Benutzern begleitend zu laufenden Synthesen nachvollzogen werden. Die aus diesen Informationen gewonnenen Hauptmerkmale stellen auch heute den Maßstab der Systementwicklung dar.

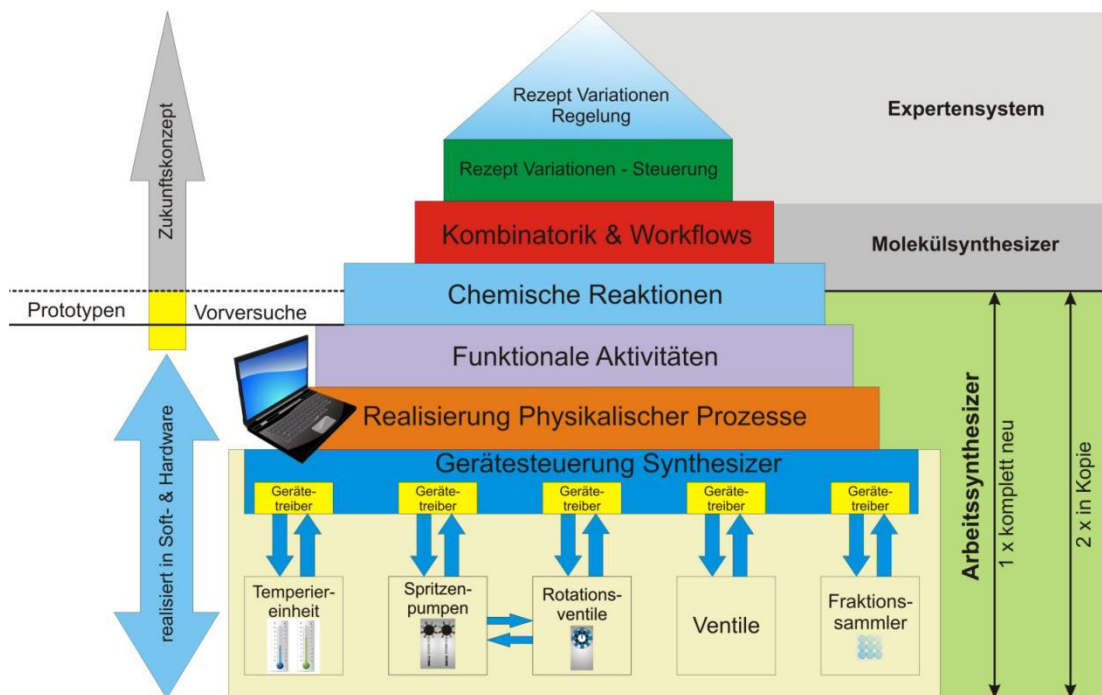


Abbildung 239: Überblick der Abstraktionsebenen der Systementwicklung für die realisierten automatischen Oligosaccharid-Synthesizer neuer Generation (Arbeitssynthesizer) im Kontext zu den in der Konzeptstudie vorgestellten Weiterentwicklungsmöglichkeiten als Molekülsynthesizer und Expertensystem.

Nach der vollständig eigenständigen Konzeption einer modularisierten Systemstruktur, wurde im ersten Schritt ein Synthesizer hardwaretechnisch zu 100% neu aufgebaut. Hierbei sind bereits die modularen Bauweisen getestet worden, um die zukünftige Umsetzung eines kaskadierbaren 19" Profil-Rahmensystems zu eruieren. Die softwareseitig jeweilig benötigten Gerätetreiber wurden in LabVIEW umgesetzt, getestet und als autark arbeitende Zustandsmaschinen für die

Synthesizersoftware in ereignisorientierten Wechselwirkungen zusammengestellt. Durch die realisierte Softwarestruktur lassen sich die bisherig an den Altsystemen durchgeführte Systemabläufe 1:1, das heißt ohne jegliche manuelle Anpassung abspielen. Hierdurch sind alle gewonnen Syntheseinformation vollwertig in der aktuellen Realisierung nutzbar. Durch die flexible Sprachumsetzung können jedoch zusätzlich sowohl Schleifenstrukturen beliebiger Schachtelungstiefe wie auch Bedingungen und Schalterfunktionen mit verwendet werden. Aufgrund von einer Vielzahl von Platzhalterkonstrukten ist die dynamische Modulierung des hinterlegten Codes zur Laufzeit möglich. Daraus erschließen sich sowohl Skalierungs-, Parametrierungs- sowie interpretierbare Rechenfunktionen. Diese Funktionalität erweitert die Flexibilität und Wiederverwendung der jeweiligen Steuerdateien erheblich. Die Steuerdateien sind in unterschiedliche Komplexitätsebenen als Module, Skripte, Prozesse und Bibliotheken mit jeweilig funktionspezifischen Erweiterungen realisiert. Auf allen Ebenen ist der routinierte Anwender berechtigt, Ergänzungen und Anpassungen vorzunehmen und in neuen Objekten abzulegen. Diese Dateien können in das lokale Dateisystem des Synthesizer eingepflegt werden und stehen somit allen Anwendern als zusätzliche Option bei der Syntheseentwicklung zur Verfügung. Dadurch ist eine universelle Weiterentwickelbarkeit bereitgestellt. Durch die zusätzliche Möglichkeit auf der Prozessebene individuelle Benutzerdialoge erstellen zu können, ist auch bei komplexen Synthesesteuerungen eine einfache Verwendung des Endergebnisses möglich. Bei einer gelungenen Zusammenstellung von Prozessen führt der Synthesizer den Anwender komplett dialogorientiert durch die Synthese. Dadurch ist die Einarbeitungszeit erheblich reduziert und die Verwendung nicht mehr auf Experten beschränkt.

Nachdem an diesem Synthesizer-Prototyp der neuen Generation eine Vielzahl von Oligosaccharidsynthesen mit Erfolg reproduzierbar hergestellt wurden, sind die bisherigen zwei J-KEM Systeme komplett in ihre Baugruppen zerlegt, auf die neue modulare Struktur adaptiert und als neu zusammengebaute Klone des Arbeitssynthesizers wieder zusammengefügt worden. Die jeweilig notwendigen Individualanpassungen sind in lokalen Zubringerdateien hinterlegt. Für die Anwender laufen diese Systeme zu 100% identisch. Somit stehen nun drei vollwertig gleiche Arbeitssynthesizer für den Laboralltag zur Verfügung. An diesen wurden vergleichende Synthesen zur Reproduzierbarkeit durchgeführt, die alle erfolgreich durchlaufen wurden. Die Systeme haben sowohl ihre Langzeitstabilität wie ihre effiziente Verwendbarkeit durch eine Vielzahl von Publikationen bewiesen.

Aufgrund der realisierten und belastbaren weiterführenden Prototypentests sind bereits die nächsten umsetzbaren Weiterentwicklungen sowohl in Soft- & Hardware vollwertig im Zugriff. Gerade im Bereich der Software-Interaktionen mit bestehenden Datenbanken und Chemoinformatik-Systemen

sind bereits die Grundlagen bereitgestellt, die eine mittelfristige Revolution der kompletten automatisieren chemischen Labor- Systeme ermöglichen. Durch diese vorgeschlagenen Module wird Industrie 4.0 für die Chemie nutzbar. Aufgrund der Modularität können Teilkomponenten für die aktuellen Felder (FlowChem, PhotoChem...) ausgekoppelt werden. Die im Kapitel 5 aufgezeigten Erweiterungsmöglichkeiten des Molekülsynthesizers erschließen fundamental neue Anwendungsgebiete. Sowohl in der allgemeinen Oligo- Synthese (Peptid, DNA, RNA) wie auch in der Durchführung von zusätzlichen chemischen Reaktionen, sowie enzymatischen, katalysatorischen, elektrochemischen Teilaspekte.

Erstmalig liegt der vollständige Quellcode der Synthesizer- Applikation beim Endanwender vor. Hierdurch lassen sich für alle nachfolgenden Anwendergenerationen die jeweiligen anvisierten Forschungsprojekte durch individuelle Anpassungen vornehmen. Da bereits auf einer hohen Abstraktionsebene entwickelt werden kann, ist die Einarbeitungszeit entsprechend gering. Die vielen Mechanismen der Fehlervermeidung und Qualitätssicherung ermöglichen es reproduzierbare chemische Automatisierungen für jeden Anwender bereitzustellen.

NACHWORT

Im Oktober 2015 wurde in Berlin die Ausstellung „Alles Zucker – Nahrung – Werkstoff – Energie“ eröffnet. Das Zucker-Museum ist seit 1995 Teil des deutschen Technikmuseums in Berlin, war bis 2012 in Berlin-Wedding beheimatet, und ist mit Gründung 1904 das älteste bestehende Spezialmuseum zur Kulturgeschichte des Zuckers.

Die Ausstellung "Alles Zucker!" ist die erste Dauerausstellung zum Thema Zucker, die weit über die Darstellung der Geschichte, Herstellung und Verwendung von Rohr- und Rübenzucker hinausgeht und der Bedeutung der Zucker in der Natur gerecht wird.

Als neue Abteilung im Deutschen Technikmuseum werden auch technische und wissenschaftliche Fragen zu benachbarten Bereichen unter dem Thema „Zucker statt Erdöl“ aufgegriffen und neue Wege in der Materialforschung, Energiegewinnung und der Medizin vorgestellt, die Zucker als Ausgangsstoff nutzen.

Komponenten, die durch die aufgezeigten Verbesserungsmaßnahmen aus den JKEM-Oligosaccharid-Syntheseautomaten ersetzt wurden, konnten zu einem vollwertigen Oligosaccharid-Synthesizer zusammengestellt werden. Hierdurch sind alle Syntheseschritte exemplarisch darstellbar. So entstand ein repräsentatives Schaustück für das Deutsche Technikmuseum Berlin, welches den ersten vollwertigen Oligosaccharid-Syntheseautomaten zeigt.



Abbildung 240: Der Oligosaccharid-Syntheseautomat im Deutschen Technikmuseum



Abbildung 241: Aufbauelemente im Deutschen Technik Museum a) Druckverteiler und Lösungsmittelzufuhr (links), b) Pumpen und Rotationsventile mit Reaktor und Temperierungseinheit(rechts)

Durch ein Informationsterminal wird die Funktion des alten Systems visuell aufbereitet präsentiert. Die aktuellen Systemerweiterungen sowie das modulare Funktionskonzept der Synthesizer werden hierin aufgeführt.



Abbildung 242: Informationsfilm im Deutschen Technik Museum am Schaustück zum historischen JKEM-Oligosaccharid-Synthesizer

LITERATURVERZEICHNIS

- [1] J. M. Berg, J. L. Tymoczko, L. Stryer, G. J. Gatto, *Biochemie*, 7. Aufl., Springer Spektrum, Berlin, **2013**.
- [2] S. Nishihara in *Endo, Seeberger et al. (Hg.) 2014 – Glycoscience*, S. 1–8.
- [3] A. Varki, *Glycobiology* **1993**, 3, 97.
- [4] a) R. A. Dwek, *Chem. Rev.* **1996**, 96, 683; b) H.-J. Gabius, *Anatomia, Histologia, Embryologia* **2001**, 30, 3.
- [5] C. R. Bertozzi, L. L. Kiessling, *Science* **2001**, 291, 2357.
- [6] A. Varki (Hrsg.) *Essentials of glycobiology*, Cold Spring Harbor Laboratory Press, Cold Spring Harbor, NY, **2009**.
- [7] B. F. Gisin, R. B. Merrifield, D. C. Tosteson, *J. Am. Chem. Soc.* **1969**, 91, 2691.
- [8] M. Caruthers, *Science* **1985**, 230, 281.
- [9] J. S. Richardson, "Subunits and Domains: The Anatomy & Taxonomy of Protein Structure", zu finden unter <http://kinemage.biochem.duke.edu/teaching/anatax/html/anatax.2i.html>, **2007**.
- [10] a) D. Voet, J. Voet, C. Pratt (Hrsg.) *Lehrbuch der Biochemie*, Wiley-VCH, Weinheim, **2010**; b) B. Alberts, J. Graw (Hrsg.) *Wiley VCH Lehrbuchkollektion 1*, Wiley-VCH, Weinheim, **2012**.
- [11] IUPAC, *Pure and Applied Chemistry* **1982**, 54.
- [12] K. Forchhammer, A. Böck, *Nachrichten aus der Chemie* **1991**, 39, 966.
- [13] G. Srinivasan, C. M. James, J. A. Krzycki, *Science (New York, N.Y.)* **2002**, 296, 1459.
- [14] K. Munk, C. Abröhl, *Biochemie - Zellbiologie. 39 Tabellen*, 1. Aufl., Georg Thieme Verlag, Stuttgart, **2008**.
- [15] A. Nordheim, *Molekulare Genetik*, 10. Aufl., Thieme, Stuttgart [u.a.], **2015**.
- [16] E. Fischer, *Ber. Dtsch. Chem. Ges.* **1902**, 35, 1095.
- [17] A. Böhne-Lang, E. Lang, T. Förster, von der Lieth, Claus-W., *Carbohydrate research* **2001**, 336, 1.
- [18] T. K. Lindhorst, *Essential of carbohydrate chemistry and biochemistry*, 3. Aufl., Wiley-VCH, Weinheim, **2007**.
- [19] H. A. Harper, G. Löffler, P. E. Petrides, L. Weiss, *Physiologische Chemie: Eine Einführung in die medizinische Biochemie für Studierende der Medizin und Ärzte*, Springer Berlin Heidelberg, **2013**.
- [20] A. Varki, R. D. Cummings, M. Aebi, N. H. Packer, P. H. Seeberger, J. D. Esko, P. Stanley, G. Hart, A. Darvill, T. Kinoshita et al., *Glycobiology* **2015**, 25, 1323.

- [21] T. K. Lindhorst, *Essential of carbohydrate chemistry and biochemistry*, 3. Aufl., Wiley-VCH, Weinheim, **2007**.
- [22] O. Berteau, R. Stenutz, *Carbohydrate research* **2004**, 339, 929.
- [23] Perez S., "The Symbolic Representation of Monosaccharides in the Age of Glycobiology".
- [24] R. R. Schmidt, *Angewandte Chemie* **1986**, 98, 213.
- [25] Villalobo, A., Nogales-González, A., *tr*, 2006, 1.
- [26] R. A. Laine, *Pure and Applied Chemistry* **1997**, 1867.
- [27] H.-J. Gabius, *The Sugar Code*, 2. Aufl., John Wiley & Sons, Hoboken, **2011**.
- [28] R. Dwek, *Science* **1995**, 269, 1234.
- [29] H. J. Gabius, S. Gabius, *Glycosciences. Status and perspectives*, Wiley-VCH, Weinheim, **2002**.
- [30] C. R. Bertozzi, *Essentials in Glycobiology*, Berkeley, **2008**.
- [31] D. E. Levy, *The organic chemistry of sugars*, Taylor & Francis, Boca Raton, Fla., **2006**.
- [32] R. V. Stick, *Carbohydrates: The Sweet Molecules of Life*, Academic, **2001**.
- [33] J. W. Suggs, *Journal of Pharmaceutical Sciences* **1985**, 74, 39.
- [34] S. C. Hung, M.M.L. Zulueta, *Glycochemical Synthesis: Strategies and Applications*, Wiley, **2016**.
- [35] J. M. Frechet, C. Schuerch, *J. Am. Chem. Soc.* **1972**, 94, 604.
- [36] N K Kochetkov, A F Bochkov, *Natural Carbon Compounds*, 77.
- [37] R. R. Schmidt, W. Kinzy, *Advances in carbohydrate chemistry and biochemistry* **1994**, 50, 21.
- [38] S. P. Douglas, D. M. Whitfield, J. J. Krepinsky, *Journal of the American Chemical Society : web edition* **1991**, 113, 5095.
- [39] D. Kahne, S. Walker, Y. Cheng, D. van Engen, *J. Am. Chem. Soc.* **1989**, 111, 6881.
- [40] C. M. Taylor in *Solid Support Oligosaccharide Synthesis and Combinatorial Carbohydrate Libraries* (Hrsg.: P. H. Seeberger), Wiley-Interscience, S.l., **2001**, S. 41–65.
- [41] M. A. Fascione, R. Brabham, W. B. Turnbull, *Chemistry (Weinheim an der Bergstrasse, Germany)* **2016**, 22, 3916.
- [42] R. R. Schmidt, *Pure and Applied Chemistry* **1989**, 61.
- [43] R. R. Schmidt, A. Toepfer, *Tetrahedron Letters* **1991**, 32, 3353.
- [44] S. P. Douglas, D. M. Whitfield, J. J. Krepinsky, *Journal of Carbohydrate Chemistry* **1993**, 12, 131.
- [45] T. Stauch, U. Greilich, R. R. Schmidt, *Liebigs Ann.* **1995**, 1995, 2101.

- [46] L. Knerr, R. R. Schmidt in *Solid Support Oligosaccharide Synthesis and Combinatorial Carbohydrate Libraries* (Hrsg.: P. H. Seeberger), Wiley-Interscience, S.I., **2001**, S. 67–98.
- [47] G. Stork, J. J. La Clair, *J. Am. Chem. Soc.* **1996**, *118*, 247.
- [48] D. Crich, S. Sun, *J. Org. Chem.* **1997**, *62*, 1198.
- [49] T. J. Martin, R. R. Schmidt, *Tetrahedron Letters* **1992**, *33*, 6123.
- [50] T. J. Martin, R. Brescello, A. Toepfer, R. R. Schmidt, *Glycoconjugate J* **1993**, *10*, 16.
- [51] R. B. Andrade, O. J. Plante, L. G. Melean, P. H. Seeberger, *Org. Lett.* **1999**, *1*, 1811.
- [52] K. Routenberg Love, P. H. Seeberger, *Angew. Chem.* **2004**, *116*, 612.
- [53] W.-C. Haase, O. J. Plante, P. H. Seeberger in *Solid Support Oligosaccharide Synthesis and Combinatorial Carbohydrate Libraries* (Hrsg.: P. H. Seeberger), Wiley-Interscience, S.I., **2001**, S. 117–134.
- [54] C.-H. Lai, H. S. Hahm, C.-F. Liang, P. H. Seeberger, *Beilstein J. Org. Chem.* **2015**, *11*, 617.
- [55] S. J. Danishefsky, K. F. McClure, J. T. Randolph, R. B. Ruggeri, *Science* **1993**, *260*, 1307.
- [56] P. H. Seeberger, X. Beebe, G. D. Sukenick, S. Pochapsky, S. J. Danishefsky, *Angewandte Chemie International Edition* **1997**, *36*, 491.
- [57] P. F. Cirillo, S. J. Danishefsky in *Solid Support Oligosaccharide Synthesis and Combinatorial Carbohydrate Libraries* (Hrsg.: P. H. Seeberger), Wiley-Interscience, S.I., **2001**, S. 15–40.
- [58] M. Sasaki, K. Tachibana, H. Nakanishi, *Tetrahedron Letters* **1991**, *32*, 6873.
- [59] G. H. Veeneman, S. H. van Leeuwen, J. H. van Boom, *Tetrahedron Letters* **1990**, *31*, 1331.
- [60] D. Crich, M. Smith, *J. Am. Chem. Soc.* **2002**, *124*, 8867.
- [61] V. Wittmann in *Solid Support Oligosaccharide Synthesis and Combinatorial Carbohydrate Libraries* (Hrsg.: P. H. Seeberger), Wiley-Interscience, S.I., **2001**, S. 99–116.
- [62] Codée, Jeroen D. C., Litjens, Remy E. J. N., van den Bos, Leendert J., H. S. Overkleeft, van der Marel, Gijsbert A., *Chemical Society reviews* **2005**, *34*, 769.
- [63] W. Koenigs, E. Knorr, *Einige Derivate des Traubenzuckers*, Verl.d.K. Akad, München, **1900**.
- [64] E. Fischer, *Über die Glucoside der Alkohole*, Reimer, Berlin, **1893**.
- [65] H. Paulsen, *Angewandte Chemie International Edition in English* **1982**, *21*, 155.
- [66] T. Mukaiyama, Y. Murai, S.-i. Shoda, *Chem. Lett.* **1981**, *10*, 431.
- [67] T. Mukaiyama, M. Suenaga, H. Chiba, H. Jona, *Chem. Lett.* **2002**, *31*, 56.
- [68] K. C. Nicolaou, R. A. Daines, T. K. Chakraborty, Y. Ogawa, *J. Am. Chem. Soc.* **1987**, *109*, 2821.
- [69] H. Ando, S. Manabe, Y. Nakahara, Y. Ito, *Angewandte Chemie International Edition* **2001**, *40*, 4725.

- [70] B. O. Fraser-Reid, K. Tatsuta, J. Thiem, *Glycoscience: Chemistry and Chemical Biology I–III*, Springer Berlin Heidelberg, **2012**.
- [71] B. Fraser-Reid, U. E. Udodong, Z. Wu, H. Ottosson, J. R. Merritt, C. S. Rao, C. Roberts, R. Madsen, *Synlett* **1992**, 1992, 927.
- [72] R. Rodebaugh, S. Joshi, B. Fraser-Reid, H. M. Geysen, *J. Org. Chem.* **1997**, 62, 5660.
- [73] P. H. Toy, Y. Lam, *Solid-Phase Organic Synthesis: Concepts, Strategies, and Applications*, Wiley, **2012**.
- [74] W.-C. Haase, P. H. Seeberger, *COC* **2000**, 4, 481.
- [75] R. B. Merrifield, *J. Am. Chem. Soc.* **1963**, 85, 2149.
- [76] a) J. S. Früchtel, G. Jung, *Angewandte Chemie* **1996**, 108, 19; b) M. Winter in *Combinatorial peptide and nonpeptide libraries. A handbook* (Hrsg.: G. Jung), VCH, Weinheim, New York, **1996**, S. 465–510.
- [77] S. Booth, P. H.H. Hermkens, H. C.J. Ottenheijm, D. C. Rees, *Tetrahedron* **1998**, 54, 15385.
- [78] R. B. Merrifield, *Science* **1965**, 150, 178.
- [79] C. Schuerch, J. M. Frechet, *J. Am. Chem. Soc.* **1971**, 93, 492.
- [80] C. Shuet-Hing Lee, L. Anderson, *Carbohydrate research* **1976**, 50, 227.
- [81] a) R. R. Schmidt, S. Jonke, K.-g. Liu in *ACS symposium series* (Hrsg.: A. V. Demchenko), American Chemical Society, Washington, DC, **2007**, S. 209–236; b) P. H. Seeberger, *Solid support oligosaccharide synthesis and combinatorial carbohydrate libraries*, Wiley, New York, **2001**; c) K. Tanaka, K. Fukase in *Solid-Phase Organic Synthesis*, John Wiley & Sons, Inc, **2011**, S. 489–530.
- [82] L. G. Melean, K. R. Love, P. H. Seeberger, *Carbohydrate research* **2002**, 337, 1893.
- [83] T. Kanemitsu, O. Kanie, C.-H. Wong, *Angewandte Chemie International Edition* **1998**, 37, 3415.
- [84] M. Mogemark, M. Elofsson, J. Kihlberg, *Org. Lett.* **2001**, 3, 1463.
- [85] B. Yan, *Acc. Chem. Res.* **1998**, 31, 621.
- [86] B. J. Egner, G. J. Langley, M. Bradley, *J. Org. Chem.* **1995**, 60, 2652.
- [87] S. Manabe, Y. Ito, *J. Am. Chem. Soc.* **2002**, 124, 12638.
- [88] R. B. Andrade, T. P. Begley in *Wiley Encyclopedia of Chemical Biology*, John Wiley & Sons, Inc, **2007**.
- [89] F. Albericio, P. Lloyd-Williams, E. Giralt, *Methods in enzymology* **1997**, 289, 313.
- [90] P. Hodge, *Chem. Soc. Rev.* **1997**, 26, 417.
- [91] E. Bayer, *Angewandte Chemie* **1991**, 103, 117.
- [92] O. Kanie, I. Ohtsuka, T. Ako, S. Daikoku, Y. Kanie, R. Kato, *Angewandte Chemie* **2006**, 118, 3935.

- [93] H. Shimizu, M. Sakamoto, N. Nagahori, S.-I. Nishimur, *Tetrahedron* **2007**, *63*, 2418.
- [94] P. Pornsuriyasak, S. C. Ranade, A. Li, M. C. Parlato, C. R. Sims, O. V. Shulga, K. J. Stine, A. V. Demchenko, *Chemical communications (Cambridge, England)* **2009**, 1834.
- [95] S. P. Douglas, D. M. Whitfield, J. J. Krepinsky, *J. Am. Chem. Soc.* **1995**, *117*, 2116.
- [96] M. Meldal, F.-I. Auzanneau, O. Hindsgaul, M. M. Palcic, *J. Chem. Soc., Chem. Commun.* **1994**, 1849.
- [97] M. Adinolfi, G. Barone, L. D. Napoli, A. Iadonisi, G. Piccialli, *Tetrahedron Letters* **1996**, *37*, 5007.
- [98] J. J. Krepinski, S. P. Douglas in *Solid Support Oligosaccharide Synthesis and Combinatorial Carbohydrate Libraries* (Hrsg.: P. H. Seeberger), John Wiley & Sons, Inc, New York, USA, **2001**, S. 175–199.
- [99] a) S. Park, M.-R. Lee, I. Shin, *Chemical communications (Cambridge, England)* **2008**, 4389; b) N. Laurent, J. Voglmeir, S. L. Flitsch, *Chemical communications (Cambridge, England)* **2008**, 4400.
- [100] F. Guillier, D. Orain, M. Bradley, *Chem. Rev.* **2000**, *100*, 2091.
- [101] T. W. Greene, Wuts, Peter G. M., *Greene's protective groups in organic synthesis*, 4. Aufl., Wiley-Interscience, Hoboken, NJ, **NJ : WileyInterscience, 2007**.
- [102] P. H. Seeberger, W.-C. Haase, *Chemical reviews* **2000**, *100*, 4349.
- [103] F. Guillier, D. Orain, M. Bradley, *Chemical reviews* **2000**, *100*, 2091.
- [104] Walvoort, Marthe T C, A. G. Volbeda, Reintjens, Niels R M, van den Elst, Hans, O. J. Plante, H. S. Overkleeft, van der Marel, Gijsbert A, Codée, Jeroen D. C., *Org. Lett.* **2012**, *14*, 3776.
- [105] K. R. Love, P. H. Seeberger, *Angewandte Chemie International Edition* **2004**, *43*, 602.
- [106] S. M. Hancock, M. D. Vaughan, S. G. Withers, *Current Opinion in Chemical Biology* **2006**, *10*, 509.
- [107] P. Czechura, N. Guedes, S. Kopitzki, N. Vazquez, M. Martin-Lomas, N.-C. Reichardt, *Chemical communications (Cambridge, England)* **2011**, *47*, 2390.
- [108] N. V. Ganesh, K. Fujikawa, Y. H. Tan, K. J. Stine, A. V. Demchenko, *Org Lett* **2012**, *14*, 3036.
- [109] U. Zehavi, A. Patchornik, *J. Am. Chem. Soc.* **1973**, *95*, 5673.
- [110] S. Eller, M. Collot, J. Yin, H. S. Hahm, P. H. Seeberger, *Angewandte Chemie* **2013**, *125*, 5970.
- [111] B. D. A. Hook, W. Dohle, P. R. Hirst, M. Pickworth, M. B. Berry, K. I. Booker-Milburn, *The Journal of organic chemistry* **2005**, *70*, 7558.
- [112] a) J. Hofmann, H. S. Hahm, P. H. Seeberger, K. Pagel, *Nature* **2015**; b) M. P. Campbell, T. Nguyen-Khuong, C. A. Hayes, S. A. Flowers, K. Alagesan, D. Kolarich, N. H. Packer, N. G. Karlsson, *Biochimica et biophysica acta* **2014**, *1844*, 108.
- [113] M. Wilsdorf, D. Schmidt, M. P. Bartetzko, P. Dallabernardina, F. Schuhmacher, P. H. Seeberger, F. Pfrenge, *Chem. Commun.* **2016**, *52*, 10187.

- [114] P. Sears, *Science* **2001**, 291, 2344.
- [115] a) *Modern Synthetic Methods in Carbohydrate Chemistry*, Wiley-VCH Verlag GmbH & Co. KGaA, **2013**; b) S. H. Khan, R. A. O'Neill (Hrsg.) *Frontiers in natural product research, Vol. 1*, Harwood, Amsterdam, **1996**.
- [116] D. B. Werz, S. Vidal, *Modern synthetic methods in carbohydrate chemistry. From monosaccharides to complex glycoconjugates*, Wiley-VCH, Weinheim, **2014**.
- [117] Y. Ito, S. Hanashima in *Experimental Glycoscience* (Hrsg.: N. Taniguchi, A. Suzuki, Y. Ito, H. Narimatsu, T. Kawasaki, S. Hase), Springer Japan, Tokyo, **2008**, S. 210–216.
- [118] T. Zhu, G.-J. Boons, *Tetrahedron Letters* **1998**, 39, 2187.
- [119] Y. Ito, O. Kanie, T. Ogawa, *Angew. Chem.* **1996**, 108, 2691.
- [120] N. K. Kochetkov, *Russ. Chem. Rev.* **2000**, 69, 795.
- [121] D. R. Mootoo, P. Konradsson, U. Udodong, B. Fraser-Reid, *J Am Chem Soc* **1988**, 110, 5583.
- [122] K. C. Nicolaou, H. UENO in *Preparative Carbohydrate Chemistry*, S. 313–338.
- [123] L. J. Williams, R. M. Garbaccio, S. J. Danishefsky in *Carbohydrates in Chemistry and Biology*, Wiley-VCH Verlag GmbH, **2008**, S. 61–92.
- [124] R. Roy, F. O. Andersson, M. Letellier, *Tetrahedron Letters* **1992**, 33, 6053.
- [125] G.-J. Boons, S. Isles, *Tetrahedron Letters* **1994**, 35, 3593.
- [126] a) Y. Ito, T. Ogawa, *Angewandte Chemie* **1994**, 106, 1843; b) O. Kanie, Y. Ito, T. Ogawa, *J. Am. Chem. Soc.* **1994**, 116, 12073.
- [127] X. Huang, L. Huang, H. Wang, X.-S. Ye, *Angew Chem Int Ed Engl* **2004**, 43, 5221.
- [128] Z. Wang, L. Zhou, K. El-Boubbou, X.-S. Ye, X. Huang, *J Org Chem* **2007**, 72, 6409.
- [129] a) S. Kaeothip, A. V. Demchenko, *Carbohydrate research* **2011**, 346, 1371; b) H. D. Premathilake, A. V. Demchenko, *Topics in current chemistry* **2011**, 301, 189.
- [130] a) T. Zhu, G.-J. Boons, *Tetrahedron: Asymmetry* **2000**, 11, 199; b) C.-H. Wong, X.-S. Ye, Z. Zhang, *J. Am. Chem. Soc.* **1998**, 120, 7137; c) D. Orain, J. Ellard, M. Bradley, *J. Comb. Chem.* **2002**, 4, 1.
- [131] N. Jernberg, R. B. Merrifield, J. M. Stewart, **1970**.
- [132] a) P. G. Katsoyannis (Hrsg.) *The Chemistry of Polypeptides. Essays in Honor of Dr. Leonidas Zervas*, Springer US, Boston, MA, **1973**; b) H. Paulsen, G. Merz, S. Peters, U. Weichert, *Liebigs Ann. Chem.* **1990**, 1990, 1165; c) J. Eichler, *Kombinatorische Chemie. Konzepte und Strategien*, Vieweg+Teubner Verlag, Wiesbaden, **2003**; d) T. Okuda, *Naturwissenschaften* **1968**, 55, 209; e) C. Birr, *Aspects of the Merrifield peptide synthesis*, Springer, Berlin, **1978**.
- [133] R. L. Letsinger, W. B. Lunsford, *J. Am. Chem. Soc.* **1976**, 98, 3655.
- [134] M. D. Matteucci, M. H. Caruthers, *Journal of the American Chemical Society : web edition* **1981**, 103, 3185.

- [135] O. J. Plante, E. R. Palmacci, R. B. Andrade, P. H. Seeberger, *Journal of the American Chemical Society* **2001**, *123*, 9545.
- [136] O. J. Plante, E. R. Palmacci, P. H. Seeberger, *Science* **2001**, *291*, 1523.
- [137] a) P. H. Seeberger, *Chem. Soc. Rev.* **2007**, *37*, 19; b) O. J. Plante, E. R. Palmacci, P. H. Seeberger in *Advances in Carbohydrate Chemistry and Biochemistry*, Academic Press, **2003**, S. 35–54.
- [138] T. Buskas, E. Söderberg, P. Konradsson, B. Fraser-Reid, *J Org Chem* **2000**, *65*, 958.
- [139] a) E. R. Palmacci, O. J. Plante, M. C. Hewitt, P. H. Seeberger, *Helvetica Chimica Acta* **2003**, *86*, 3975; b) T. Kanemitsu, P. H. Seeberger, *Org. Lett.* **2003**, *5*, 4541.
- [140] D. Esposito, M. Hurevich, B. Castagner, C.-C. Wang, P. H. Seeberger, *Beilstein J Org Chem* **2012**, *8*, 1601.
- [141] D. B. Werz, B. Castagner, P. H. Seeberger, *J Am Chem Soc* **2007**, *129*, 2770.
- [142] Codée, Jeroen D. C., L. Kröck, B. Castagner, P. H. Seeberger, *Chem. Eur. J.* **2008**, *14*, 3987.
- [143] a) S. J. Danishefsky, J. R. Allen, *Angew. Chem. Int. Ed.* **2000**, *39*, 836; b) P. H. Seeberger, D. B. Werz, *Nature* **2007**, *446*, 1046; c) S. Hakomori, *Annual review of immunology* **1984**, *2*, 103; d) S. Hakomori, *Annu. Rev. Biochem.* **1981**, *50*, 733; e) S.-i. Hakomori, *Biochimica et biophysica acta* **2008**, *1780*, 325; f) Z. Guo, Q. Wang, *Curr Opin Chem Biol* **2009**, *13*, 608.
- [144] a) T. Zhu, G.-J. Boons, *Angew. Chem.* **1999**, *111*, 3704; b) F. Bosse, L. A. Marcaurelle, P. H. Seeberger, *J Org Chem* **2002**, *67*, 6659; c) J. R. Allen, J. G. Allen, X.-F. Zhang, L. J. Williams, A. Zatorski, G. Ragupathi, P. O. Livingston, S. J. Danishefsky, *Chem. Eur. J.* **2000**, *6*, 1366; d) I. Jeon, K. Iyer, S. J. Danishefsky, *J Org Chem* **2009**, *74*, 8452.
- [145] R. H. Grubbs, S. J. Miller, G. C. Fu, *Acc. Chem. Res.* **1995**, *28*, 446.
- [146] L. Kröck, D. Esposito, B. Castagner, C.-C. Wang, P. Bindschädler, P. H. Seeberger, *Chem. Sci.* **2012**, *3*, 1617.
- [147] a) H. Paulsen, *Angew. Chem. Int. Ed. Engl.* **1990**, *29*, 823; b) P. H. Seeberger, M. Eckhardt, C. E. Gutteridge, S. J. Danishefsky, *J. Am. Chem. Soc.* **1997**, *119*, 10064; c) S. Eller, R. Schuberth, G. Gundel, J. Seifert, C. Unverzagt, *Angew Chem Int Ed Engl* **2007**, *46*, 4173; d) S. Jonke, K.-g. Liu, R. R. Schmidt, *Chem. Eur. J.* **2006**, *12*, 1274.
- [148] K. S. Kim, J. H. Kim, Y. J. Lee, Y. J. Lee, J. Park, *J Am Chem Soc* **2001**, *123*, 8477.
- [149] O. Calin, S. Eller, P. H. Seeberger, *Angewandte Chemie* **2013**, *125*, 5974.
- [150] a) K. Egusa, S. Kusumoto, K. Fukase, *Eur. J. Org. Chem.* **2003**, *2003*, 3435; b) S. Hanashima, S. Manabe, Y. Ito, *Angew Chem Int Ed Engl* **2005**, *44*, 4218; c) H. Ando, S. Manabe, Y. Nakahara, Y. Ito, *J Am Chem Soc* **2001**, *123*, 3848; d) B. Wu, Z. Hua, J. D. Warren, K. Ranganathan, Q. Wan, G. Chen, Z. Tan, J. Chen, A. Endo, S. J. Danishefsky, *Tetrahedron Letters* **2006**, *47*, 5577.
- [151] S. S. Nigudkar, A. R. Parameswar, P. Pornsuriyasak, K. J. Stine, A. V. Demchenko, *Org Biomol Chem* **2013**, *11*, 4068.
- [152] C. S. Bennett, *Org. Biomol. Chem.* **2014**, *12*, 1686.

- [153] a) T. Zhu, G.-J. Boons, *J Am Chem Soc* **2000**, *122*, 10222; b) S. Manabe, Y. Nakahara, Y. Ito, *Synlett* **2000**, *2000*, 1241; c) J. L. de Paz, M. Mar Kayser, G. Macchione, P. M. Nieto, *Carbohydrate research* **2010**, *345*, 565.
- [154] a) J.-Y. Huang, M. Lei, Y.-G. Wang, *Tetrahedron Letters* **2006**, *47*, 3047; b) W. Miao, T. H. Chan, *Acc. Chem. Res.* **2006**, *39*, 897; c) C. K. Yerneni, V. Pathak, A. K. Pathak, *J Org Chem* **2009**, *74*, 6307.
- [155] a) D. P. Curran, R. Ferritto, Y. Hua, *Tetrahedron Letters* **1998**, *39*, 4937; b) T. Miura, Y. Hirose, M. Ohmae, T. Inazu, *Org. Lett.* **2001**, *3*, 3947; c) B. Yang, Y. Jing, X. Huang, *European Journal of Organic Chemistry* **2010**, *2010*, 1290; d) C. Zong, A. Venot, O. Dhamale, G.-J. Boons, *Org. Lett.* **2013**, *15*, 342.
- [156] F. R. Carrel, K. Geyer, Codee, Jeroen D C, P. H. Seeberger, *Org Lett* **2007**, *9*, 2285.
- [157] F. A. Jaipuri, N. L. Pohl, *Org Biomol Chem* **2008**, *6*, 2686.
- [158] Douglas, Pohl, Nicola Lucia in *ACS symposium series, Vol. 990* (Hrsg.: X. Chen, R. Halcomb, P. G. Wang), American Chemical Soc, Washington, DC, **2008**, S. 272–287.
- [159] F. Schuhmacher, *Exposé*, Univ, Berlin, **2010**.
- [160] W. Christ, L. Kröck, O. J. Plante, B. Castagner, P. H. Seeberger, **2011**.
- [161] L. Kröck, *A new linker and new synthesis instrument as key to the automated synthesis of complex oligosaccharide. Automated synthesis of complex oligosaccharides*, Zürich, **2009**.
- [162] J. E. Rumbaugh, *A parallel asynchronous computer architecture for data flow programs*, Massachusetts Institute of Technology. Project MAC, Cambridge (Mass.), **1975**.
- [163] W. R. Ashby, *An Introduction to Cybernetics*, London, Chapman & Hall, **1956**.
- [164] R. Patton, *Software testing*, 2. Aufl., Sams Pub, Indianapolis, IN, **2006**.
- [165] *Qualitätsmanagement DIN EN ISO 9000 ff. Dokumentensammlung; Ausgabe 2016 Einzelplatzversion*, 2016. Aufl., Beuth, Berlin, **2016**.
- [166] A. Gill, *Introduction to the theory of finite-state machines*, McGraw-Hill, New York, **1962**.
- [167] J. E. Hopcroft, R. Motwani, J. D. Ullman, *Einführung in die Automatentheorie, formale Sprachen und Berechenbarkeit. [www.pearson-studium.de companion website]*, 3. Aufl., Pearson Studium, München, **2011**.
- [168] H. Zimmermann, *IEEE Transactions on Communications* **1980**.
- [169] T. Holm, M. Obst, J. Ladiges, L. Urbas, A. Fay, T. Albers, U. Hempten, *Atp-Edition : automatisierungstechnische Praxis ; Organ der GMA (VDI-VDE-Gesellschaft Meß- und Automatisierungstechnik) und der NAMUR (Interessengemeinschaft Automatisierungstechnik der Prozessindustrie)* **2016**, *58*, 82.
- [170] "DIN EN 61010-1 VDE 0411-1:2011-07 - Normen - VDE VERLAG", zu finden unter <https://www.vde-verlag.de/normen/0411044/din-en-61010-1-vde-0411-1-2011-07.html>.

- [171] "Die neuen Laborrichtlinien - BG RCI", zu finden unter <http://www.bgrci.de/fachwissen-portal/start/laboratorien/laborrichtlinien/die-neuen-laborrichtlinien/>.
- [172] J. Kandasamy, M. Hurevich, P. H. Seeberger, *Chem. Commun.* **2013**, 49, 4453.
- [173] M. Badura, "Namur/Cavro Interface Specification", zu finden unter <http://www.namur.net/>.
- [174] AGILENT TECHNOLOGIES, "Cavro Pump Device Driver User Guide".
- [175] W. M. Haynes, D. R. Lide (Hrsg.) *CRC handbook of chemistry and physics. A ready-reference book of chemical and physical data*, CRC Press, Boca Raton, Fla., **2013**.
- [176] a) J. Zhang, A. G. Amin, A. Hölemann, P. H. Seeberger, D. Chatterjee, *Bioorganic & medicinal chemistry* **2010**, 18, 7121; b) S. Khasnobis, J. Zhang, S. K. Angala, A. G. Amin, M. R. McNeil, D. C. Crick, D. Chatterjee, *Chemistry & biology* **2006**, 13, 787; c) J. Zhang, S. K. Angala, P. K. Pramanik, K. Li, D. C. Crick, A. Liav, A. Jozwiak, E. Swiezewska, M. Jackson, D. Chatterjee, *ACS chemical biology* **2011**, 6, 819; d) J. D. Ayers, T. L. Lowary, C. B. Morehouse, G. S. Besra, *Bioorganic & medicinal chemistry letters* **1998**, 8, 437; e) C. Rademacher, G. K. Shoemaker, H.-S. Kim, R. B. Zheng, H. Taha, C. Liu, R. C. Nacario, D. C. Schriemer, J. S. Klassen, T. Peters et al., *J Am Chem Soc* **2007**, 129, 10489.
- [177] a) M. Joe, Y. Bai, R. C. Nacario, T. L. Lowary, *J Am Chem Soc* **2007**, 129, 9885; b) A. Ishiwata, Y. Ito, *J Am Chem Soc* **2011**, 133, 2275.
- [178] O. Calin, S. Eller, P. H. Seeberger, *Angewandte Chemie International Edition* **2013**, 52, 5862.
- [179] J. Kandasamy, F. Schuhmacher, H. S. Hahm, J. C. Klein, P. H. Seeberger, *Chem. Commun.* **2014**, 50, 1875.
- [180] J. M. Trowbridge, R. L. Gallo, *Glycobiology* **2002**, 12, 117R-25R.
- [181] A. M. Halldórsdóttir, L. Zhang, D. M. Tollefsen, *Glycobiology* **2006**, 16, 693.
- [182] a) N. Barroca, J.-C. Jacquinet, *Carbohydrate research* **2000**, 329, 667; b) L. Rochepeau-Jobron, J.-C. Jacquinet, *Carbohydrate research* **1997**, 305, 181; c) J.-C. Jacquinet, P. Sinaÿ, *Carbohydrate research* **1987**, 159, 229; d) A. Marra, X. Dong, M. Petitou, P. Sinaÿ, *Carbohydrate research* **1989**, 195, 39; e) N. A. Karst, R. J. Linhardt, *Current Medicinal Chemistry* **2003**, 10, 1993.
- [183] D. Schmidt, F. Schuhmacher, A. Geissner, P. H. Seeberger, F. Pfrenge, *Chem. Eur. J.* **2015**, 21, 5709.
- [184] P. Dallabernardina, F. Schuhmacher, P. H. Seeberger, F. Pfrenge, *Organic & biomolecular chemistry* **2016**, 14, 309.
- [185] M. P. Bartetzko, F. Schuhmacher, H. S. Hahm, P. H. Seeberger, F. Pfrenge, *Organic letters* **2015**, 17, 4344.
- [186] P. Albersheim, A. Darvill, K. Roberts, R. Sederoff, A. Staehelin (Hrsg.) *Plant Cell Walls: From Chemistry to Biology*, Garland Science, New York, **2010**.
- [187] A. Ebringerova, Z. Hromadkova and T. Heinze, *Adv. Polym.Sci.* **2005**, 1.
- [188] U. P. Scheller HV, *Annu Rev Plant Biol.* **2010**.

- [189] M. K. Kato Y., *Plant Cell Physiol.* **1985**, 437.
- [190] a) N. C. Carpita and D. M. Gibesut, *1993*, 1; b) Carpita, N., and McCann, M., *Biochemistry and Molecular Biology of Plants* **2000**, 52.
- [191] G. Richter, *Stoffwechselfysiologie der Pflanzen: Physiologie und Biochemie des Primär- und Sekundärstoffwechsels*, Thieme, **1998**.
- [192] a) G. J. Seifert, K. Roberts, *Annual review of plant biology* **2007**, 58, 137; b) R. A. Burton, M. J. Gidley, G. B. Fincher, *Nat. Chem. Biol.* **2010**, 724.
- [193] M. D. Serpe, E. A. Nothnagel, *Planta.*, 542.
- [194] A. J. van Hengel, K. Roberts, *The Plant Journal* **2002**, 32, 105.
- [195] M. Gao, A. M. Showalter, *Plant J* **1999**, 19, 321.
- [196] J. P. Knox, P. J. Linstead, J. Peart, C. Cooper, K. Roberts, *The Plant journal : for cell and molecular biology* **1991**, 1, 317.
- [197] E. Nguema-Ona, M. Vicré-Gibouin, M.-A. Cannesan, A. Driouich, *Trends in Plant Science.*, 440.
- [198] a) J. Xia, S. A. Abbas, R. D. Locke, C. F. Piskorz, J. L. Alderfer, K. L. Matta, *Tetrahedron Letters* **2000**, 41, 169; b) J. A. Wright, J. Yu, J. B. Spencer, *Tetrahedron Letters* **2001**, 42, 4033; c) J. Xia, J. L. Alderfer, C. F. Piskorz, K. L. Matta, *Chem. Eur. J.* **2001**, 7, 356.
- [199] M. W. Weishaupt, S. Matthies, P. H. Seeberger, *ChemInform* **2013**, 44, 12497.
- [200] S. Eller, M. Collot, J. Yin, H. S. Hahm, P. H. Seeberger, *Angew. Chem. Int. Ed.* **2013**, 52, 5858.
- [201] Z. Zhang, I. R. Ollmann, X.-S. Ye, R. Wischnat, T. Baasov, C.-H. Wong, *J. Am. Chem. Soc.* **1999**, 121, 734.
- [202] R. Fauré, M. Saura-Valls, H. Brumer, A. Planas, S. Cottaz, H. Driguez, *J Org Chem* **2006**, 71, 5151.
- [203] D. Ghislieri, K. Gilmore, P. H. Seeberger, *Angewandte Chemie International Edition* **2015**, 54, 678.
- [204] P. H. Seeberger, *Acc. Chem. Res.* **2015**, 48, 1450.
- [205] a) P. Kovac, R. B. Taylor, C. P. J. Glaudemans, *J Org Chem* **1985**, 50, 5323; b) T. Ziegler, P. Kováč, C. P. J. Glaudemans, *Liebigs Ann. Chem.* **1990**, 1990, 613; c) F. Kong, *Front. Chem. China* **2009**, 4, 10; d) N. W. McGill, S. J. Williams, *J Org Chem* **2009**, 74, 9388; e) A. Fekete, A. Borbás, S. Antus, A. Lipták, *Carbohydrate research* **2009**, 344, 1434; f) J.-F. Valdor, W. Mackie, *Journal of Carbohydrate Chemistry* **1997**, 16, 429; g) H. Huang, L. Han, Y.-M. Lan, L.-L. Zhang, *Journal of Asian Natural Products Research* **2014**, 16, 640.
- [206] J. Li, Y. Wang, *Synthetic Communications* **2004**, 34, 211.
- [207] a) Y. Wu, D.-C. Xiong, S.-C. Chen, Y.-S. Wang, X.-S. Ye, *Nat Comms* **2017**, 8, 14851; b) S. U. Hansen, G. J. Miller, M. J. Cliff, G. C. Jayson, J. M. Gardiner, *Chem. Sci.* **2015**, 6, 6158; c) M. Islam, G. P. Shinde, S. Hotha, *Chem. Sci.* **2017**, 8, 2033.

- [208] M. Fridman, D. Solomon, S. Yogev, T. Baasov, *Org. Lett.* **2002**, *4*, 281.
- [209] C. Li, A. Faulkner-Jones, A. R. Dun, J. Jin, P. Chen, Y. Xing, Z. Yang, Z. Li, W. Shu, D. Liu et al., *Angew. Chem. Int. Ed.* **2015**, *54*, 3957.
- [210] H. S. Hahm, M. K. Schlegel, M. Hurevich, S. Eller, F. Schuhmacher, J. Hofmann, K. Pagel, P. H. Seeberger, *Proceedings of the National Academy of Sciences of the United States of America* **2017**, *114*, E3385-E3389.
- [211] a) M. Gude, J. Ryf, P. D. White, *Lett Pept Sci* **2002**, *9*, 203; b) J. MEIENHOFER, M. WAKI, E. P. HEIMRE, T. J. LAMBROS, R. C. MAKOSFKE, C.-D. CHANG, *International Journal of Peptide and Protein Research* **1979**, *13*, 35.
- [212] M. Hurevich, J. Kandasamy, B. M. Ponnappa, M. Collot, D. Kopetzki, D. T. McQuade, P. H. Seeberger, *Org Lett* **2014**, *16*, 1794.
- [213] K. Naresh, F. Schumacher, H. S. Hahm, P. H. Seeberger, *Chem. Commun.* **2017**, *53*, 9085.
- [214] a) "Home - ARENA2036", zu finden unter <http://www.arena2036.de/de/>; b) "Industrial Internet Consortium", zu finden unter <http://iiconsortium.org/>; c) "Industrie 4.0 und Digitalisierung 4.0. Tipps zur Umsetzung", zu finden unter <https://www.iph-hannover.de/de/dienstleistungen/automatisierungstechnik/industrie-4.0/>; d) "Landkarte", zu finden unter <http://www.plattform-i40.de/I40/Navigation/DE/In-der-Praxis/Karte/karte.html>; e) ""Made in China 2025" – VDMA", zu finden unter <http://china.vdma.org/article/-/articleview/8593010>; f) "Was ist Industrie 4.0?", zu finden unter <http://www.plattform-i40.de/I40/Navigation/DE/Industrie40/WasIndustrie40/was-ist-industrie-40.html;jsessionid=D7ED4118DED525F176F382A8F939369D>; g) "Perspektivenpapier Forschungsunion", zu finden unter http://www.forschungsunion.de/pdf/forschungsunion_perspektivenpapier_2013.pdf, **2013**; h) *Lecture Notes in Business Information Processing*, Springer International Publishing, **2015**; i) BMBF-Internetredaktion, "Industrie 4.0 - BMBF", zu finden unter <https://www.bmbf.de/de/zukunftsprojekt-industrie-4-0-848.html>; j) Bundesministerium für Wirtschaft und Technologie, Referat Öffentlichkeitsarbeit, "Autonomik. Startseite", zu finden unter <http://www.autonomik.de/>; k) Industrial Value Chain Initiative, "Home - Connected! Manufacturing | Industrial Valuechain Initiative", zu finden unter <https://www.iv-i.org/en/>; l) M. Hermann, T. Pentek, B. Otto, *Design Principles for Industrie 4.0 Scenarios*, **2016**; m) Rainer Schmidt, Michael Möhring, Ralf-Christian Härting, . in *Lecture Notes in Business Information Processing*, Springer International Publishing, **2015**, S. 16–27; n) G. Trade, Invest GmbH, "GTAI – Digitalisierung der Wirtschaft in Korea (Rep.)", zu finden unter <https://www.gtai.de/GTAI/Navigation/DE/Trade/Maerkte/Dossiers/Trends/Digitalisierung/Land-KoreaRep/trend-land-korearep.html#1358932>; o) T. Widmer, "Industrie 4.0@Bosch", zu finden unter http://www.agcs.allianz.com/assets/Global offices assets/Germany/Expert Days 2015/Präsentationen/Expert Days 2015_Thorsten Widmer_final.pdf, **2015**.
- [215] M. Weiser, "The Computer for the 21st Century", zu finden unter <https://www.lri.fr/~mbl/Stanford/CS477/papers/Weiser-SciAm.pdf>.
- [216] a) "HTML5's", zu finden unter <http://statichtml.com/2009/html5-cite-element.html>, **2009**; b) "Spelling HTML5", zu finden unter <http://blog.whatwg.org/spelling-html5>, **2009**; c) "HTML5 Differences from HTML4", zu finden unter <http://www.w3.org/TR/2014/NOTE-html5-diff-20141209/>, **2014**; d) "HTML5, A vocabulary and associated APIs for HTML and XHTML", zu finden unter <http://www.w3.org/TR/2014/REC-html5-20141028/>, **2014**; e) "Open Web Platform Milestone Achieved with HTML5 Recommendation", zu finden unter

- <http://www.w3.org/2014/10/html5-rec.html.en>, **2014**; f) E. Huber, "DRM in HTML5. Protestpetition gegen Kopierschutz im Browser", zu finden unter <http://www.spiegel.de/netzwelt/web/petition-widerstand-im-web-gegen-eme-drm-a-892482.html>, **2013**; g) N. Leenheer, "HTML5 test", zu finden unter <http://html5test.com/>, **2013**; h) M. Pilgrim, "Dive Into HTML5", zu finden unter <http://diveintohtml5.info/>, **2011**.
- [217] E. Geisberger (Hrsg.) *acatech Studie*, Springer-Verl., Berlin, **2012**.
- [218] S. J. Coles, J. G. Frey, C. L. Bird, R. J. Whitby, A. E. Day, *J Cheminf* **2013**, *5*, 52.
- [219] C.-H. Wu, C. Chen, S.-C. Lin, C.-C. Wang, *Synlett* **2018**, *29*, 1430.
- [220] V. Sans, L. Cronin, *Chem. Soc. Rev.* **2016**, *45*, 2032.
- [221] D. Cortés-Borda, E. Wimmer, B. Gouilleux, E. Barré, N. Oger, L. Goulamaly, L. Peault, B. Charrier, C. Truchet, P. Giraudeau et al., *J Org Chem* **2018**.
- [222] J. K. Magtaan, M. Devocelle, F. Kelleher, *Journal of peptide science : an official publication of the European Peptide Society* **2019**, *25*, e3139.
- [223] M. W. Weishaupt, S. Matthies, P. H. Seeberger, *Chem. Eur. J.* **2013**, *19*, 12497.
- [224] M. W. Kessler, *Photochemisch orthogonale Festphasensynthesen*, Cuvillier, **2008**.
- [225] C. G. Bochet, *Synlett* **2004**, *2004*, 2268.
- [226] V. San Miguel, C. G. Bochet, A. del Campo, *Journal of the American Chemical Society* **2011**, *133*, 5380.
- [227] R. J. T. Mikkelsen, K. E. Grier, K. T. Mortensen, T. E. Nielsen, K. Qvortrup, *ACS combinatorial science* **2018**, *20*, 377.
- [228] A. Albini, M. Fagnoni, *Handbook of Synthetic Photochemistry*, Wiley, **2009**.
- [229] a) R. W. Binkley, T. W. Flechtner in *Synthetic Organic Photochemistry* (Hrsg.: W. M. Horspool), Springer US, Boston, MA, **1984**, S. 375–423; b) C. G. Bochet, *J. Chem. Soc., Perkin Trans. 1* **2002**, 125; c) G. Jung, *Combinatorial Chemistry: Synthesis, Analysis, Screening*, Wiley, **2008**; d) K. C. Nicolaou, N. Winssinger, J. Pastor, F. DeRoose, *J. Am. Chem. Soc.* **1997**, *119*, 449; e) K. Lee, D. E. Falvey, *J Am Chem Soc* **2000**, *122*, 9361.
- [230] A. Blanc, C. G. Bochet, *The Journal of organic chemistry* **2002**, *67*, 5567.
- [231] H. Lusic, R. Uprety, A. Deiters, *Organic letters* **2010**, 916.
- [232] T. Voelker, T. Ewell, J. Joo, E. D. Edstrom, *Tetrahedron Letters* **1998**, 359.
- [233] a) A. G. Russell, M.-E. Ragoussi, R. Ramalho, C. W. Wharton, D. Carteau, D. M. Bassani, J. S. Snaith, *The Journal of organic chemistry* **2010**, 4648; b) J. A. Karas, D. B. Scanlon, B. E. Forbes, I. Vetter, R. J. Lewis, J. Gardiner, F. Separovic, J. D. Wade, M. A. Hossain, *Chemistry (Weinheim an der Bergstrasse, Germany)* **2014**, 9549.
- [234] A. Ajayaghosh, Pillai, V. N. R. J., *Org. Chem.* **1987**, 5714.

- [235] A. Gilbert, D. Bryce-Smith, N. S. Allen, *Photochemistry. A review of the literature published between July 1992 and June 1993. A Specialist periodical report*, Cambridge [England], **1994**.
- [236] a) M. Kessler, R. Glatthar, B. Giese, C. G. Bochet, *Organic letters* **2003**, *5*, 1179; b) *Journal of biomedical semantics* **2013**.
- [237] a) Y. Han, V. Agarwal, D. Dodd, J. Kim, B. Bae, R. I. Mackie, S. K. Nair, I. K. O., *J. Biol. Chem.* **287**, 2012; b) M. S. Kim, S. L. Diamond, *Bioorganic & medicinal chemistry letters* **2006**, 4007.
- [238] Makriyannis, A., Biegel, D., *Drug Discovery Strategies and Methods*, Marcel Dekker, New York, **2004**.
- [239] A. Albin, M. Fagnoni, *Photochemically-Generated Intermediates in Synthesis*, Wiley, Hoboken, **2013**.
- [240] G. Barany, R. B. J. Merrifield, *Am. Chem. Soc.* **1977**, 7363.
- [241] K. Korgavkar, M. Xiong, M. A. Weinstock, *Journal of cutaneous medicine and surgery* **2013**, 308.
- [242] *Recommended practice for photobiological safety for lamps and lamp systems--general requirements. Illuminating Engineering Society of North America*, New York, **2005**.
- [243] a) J. C. Colmenares, Y. J. Xu, *Heterogeneous Photocatalysis: From Fundamentals to Green Applications*, Springer Berlin Heidelberg, **2015**; b) P. Mazierski, B. Bajorowicz, E. Grabowska, A. Zaleska-Medynska, *Springer Berlin* **2016**, 211.
- [244] a) E. Danieli, J. Perlo, A. L. L. Duchateau, G. K. M. Verzijl, V. M. Litvinov, B. Blümich, F. Casanova, *Chemphyschem : a European journal of chemical physics and physical chemistry* **2014**, *15*, 3060; b) N. Zientek, *Journal of Magnetic Resonance* **2014**, *249*, 53; c) B. Gouilleux, B. Charrier, E. Danieli, J.-N. Dumez, S. Akoka, F.-X. Felpin, M. Rodriguez-Zubiri, P. Giraudeau, *The Analyst* **2015**, *140*, 7854; d) N. Zientek, C. Laurain, K. Meyer, A. Paul, D. Engel, G. Guthausen, M. Kraume, M. Maiwald, *Magnetic resonance in chemistry : MRC* **2016**, *54*, 513; e) Y. Matviychuk, E. von Harbou, D. J. Holland, *Journal of magnetic resonance (San Diego, Calif. : 1997)* **2017**, *285*, 86; f) S. Kern, S. Guhl, K. Meyer, L. Wander, A. Paul, M. Maiwald (Hrsg.) *P2.07 - Mathematical and Statistical Tools for Online NMR Spectroscopy in Chemical Processes*. AMA Service GmbH, Von-Münchhausen-Str. 49, 31515 Wunstorf, Germany, **2017**.
- [245] a) V. Sans, L. Porwol, V. Dragone, L. Cronin, *Chem. Sci.* **2015**, *6*, 1258; b) K. Meyer, S. Kern, N. Zientek, G. Guthausen, M. Maiwald, *TrAC Trends in Analytical Chemistry* **2016**, *83*, 39; c) B. Blümich, K. Singh, *Angew Chem Int Ed Engl* **2018**, *57*, 6996.
- [246] A. Michalik-Onichimowska, S. Kern, J. Riedel, U. Panne, R. King, M. Maiwald, *Journal of magnetic resonance (San Diego, Calif. : 1997)* **2017**, *277*, 154.
- [247] M. V. Gomez, A. de La Hoz, *Beilstein J Org Chem* **2017**, *13*, 285.
- [248] D. Banfi, L. Patiny, *CHIMIA* **2008**, *62*, 280.
- [249] P. Klukowski, M. Schubert, *Bioinformatics (Oxford, England)* **2018**.
- [250] a) E. Maxwell, Y. Tan, Y. Tan, H. Hu, G. Benson, K. Aizikov, S. Conley, G. O. Staples, G. W. Slys, R. D. Smith et al., *PloS one* **2012**, *7*, e45474; b) J. Klein, L. Carvalho, J. Zaia, *Bioinformatics (Oxford, England)* **2018**, *34*, 3511.

- [251] R. R. Kapaev, P. V. Toukach, *Bioinformatics (Oxford, England)* **2018**, *34*, 957.
- [252] K. S. Egorova, P. V. Toukach in *A Practical Guide to Using Glycomics Databases* (Hrsg.: K. F. Aoki-Kinoshita), Springer, Tokyo, **2017**, S. 75–113.
- [253] C. Steinbeck, S. Krause, S. Kuhn, *J. Chem. Inf. Comput. Sci.* **2003**, *43*, 1733.
- [254] S. Kuhn, N. E. Schlörer, *Magnetic resonance in chemistry : MRC* **2015**, *53*, 582.
- [255] S. Kuhn, B. Egert, S. Neumann, C. Steinbeck, *BMC Bioinformatics* **2008**, *9*, 400.
- [256] A. Sarkar, S. Drouillard, A. Rivet, S. Perez, *Glycobiology* **2015**, *25*, 1480.
- [257] S. J. Chalk, *J Cheminf* **2016**, *8*, 55.
- [258] T. Vosegaard, *Magnetic resonance in chemistry : MRC* **2015**, *53*, 285.
- [259] J. Stähle, G. Widmalm, *CHAPTER 15:NMR Chemical Shift Predictions and Structural Elucidation of Oligo- and Polysaccharides by the Computer Program CASPER*, **2017**, zu finden unter <https://pubs.rsc.org/en/content/chapterpdf/2017/bk9781782623106-00335?isbn=978-1-78262-310-6&sercode=bk>.
- [260] a) R. R. Kapaev, P. V. Toukach, *Journal of chemical information and modeling* **2016**, *56*, 1100; b) R. R. Kapaev, P. V. Toukach, *Anal Chem* **2015**, *87*, 7006.
- [261] R. R. Kapaev, K. S. Egorova, P. V. Toukach, *Journal of chemical information and modeling* **2014**, *54*, 2594.
- [262] a) J. Aires-de-Sousa, M. C. Hemmer, J. Gasteiger, *Anal Chem* **2002**, *74*, 80; b) Y. Binev, J. Aires-de-Sousa, *J. Chem. Inf. Comput. Sci.* **2004**, *44*, 940; c) Y. Binev, M. Corvo, J. Aires-de-Sousa, *J. Chem. Inf. Comput. Sci.* **2004**, *44*, 946; d) Y. Binev, M. M. B. Marques, J. Aires-de-Sousa, *J. Chem. Inf. Model.* **2007**, *47*, 2089.
- [263] J. Meiler, W. Maier, M. Will, R. Meusinger, *Journal of Magnetic Resonance* **2002**, *157*, 242.
- [264] H. Wang, *J. Chem. Educ.* **2005**, *82*, 1340.
- [265] a) T. Claridge, *J. Chem. Inf. Model.* **2009**, *49*, 1136; b) J. Junker, *Journal of cheminformatics* **2011**, *3*, 27.
- [266] A. J. Williams, *Drug Discovery Today* **2008**, *13*, 502.
- [267] S. V. Ley, R. J. Ingham, M. O'Brien, D. L. Browne, *Beilstein J Org Chem* **2013**, *9*, 1051.
- [268] a) H. G. C. Ferreira, E. Dias Canedo, R. T. de Sousa in *Wireless and Mobile Computing, Networking and Communications (WiMob), 2013 IEEE 9th International Conference on. Date 7-9 Oct. 2013*, IEEE, Piscataway, NJ, **2013**, S. 53–60; b) M. Blackstock, R. Lea in *Proceedings of the 5th International Workshop on Web of Things* (Hrsg.: Unknown), ACM, New York, NY, **2014**, S. 34–39.
- [269] a) M. Matijevic, V. Cvjetkovic in *Proceedings of 2016 13th International Conference on Remote Engineering and Virtual Instrumentation (REV). Date and venue: 24-26 February 2016 in Madrid, Spain*, IEEE, [Piscataway, New Jersey], **2016**, S. 56–63; b) S. Ferdoush, X. Li, *Procedia Computer Science* **2014**, *34*, 103; c) A. D. Deshmukh, U. B. Shinde in *Proceedings of the*

- International Conference on Inventive Computational Technologies (ICICT 2016)*. 26-27 August 2016 (Hrsg.: I. C. o. I. C. Technologies), IEEE, [Piscataway, NJ], **2016**, S. 1–6; d) S. Lee, J. Jo, Y. Kim, H. Stephen in *SCC 2014. 2014 IEEE International Conference on Services Computing : proceedings : 27 June-2 July 2013, Anchorage, Alaska*, IEEE Computer Society; IEEE, Los Alamitos, Calif., Piscataway, N.J., **2014**, S. 275–282.
- [270] S. E. Princy, K. G. J. Nigel in *IC-GET 2015. Proceedings of 2015 Online International Conference on Green Engineering and Technologies : Coimbatore, India, 27th November 2015* (Hrsg.: O. I. C. o. G. E. a. Technologies), IEEE, Piscataway, NJ, **2016**, S. 1–4.
- [271] M. S. Bin Bahrudin, R. A. Kassim, N. Buniyamin in *2013 International Conference on Electrical, Electronics and System Engineering (ICEESE). 4th - 5th December 2013, Dorsett Grand Subang, Selangor, Malaysia*, IEEE, [Piscataway, N.J.], **2013**, S. 43–48.
- [272] S. V. Ley, D. E. Fitzpatrick, R. M. Myers, C. Battilocchio, R. J. Ingham, *Angew Chem Int Ed Engl* **2015**, *54*, 10122.
- [273] N. Arbain, N. F. Nordin, N. M. Isa, S. Saaidin in *Electrical, Electronics and System Engineering (ICEESE), 2014 International Conference on. Date, 9-10 Dec. 2014*, IEEE, [Piscataway, N.J.], **2014?**, S. 89–94.
- [274] K. Grgic, I. Speh, I. Hedi in *Proceedings of 2016 International Conference on Smart Systems and Technologies (SST). Faculty of Electrical Engineering, Computer Science, and Information Technology Osijek, Josip Juraj Strossmayer University of Osijek, Osijek, Croatia, October 12-14, 2016* (Hrsg.: D. Žagar), Faculty of Electrical Engineering, Computer Science, and Information Technology Osijek, Osijek, Croatia, **2016**, S. 249–253.
- [275] K. Bougot-Robin, J. Paget, S. C. Atkins, J. B. Edel, *J. Chem. Educ.* **2016**, *93*, 1232.
- [276] D. E. Fitzpatrick, S. V. Ley, *React. Chem. Eng.* **2016**, *1*, 629.
- [277] D. E. Fitzpatrick, C. Battilocchio, S. V. Ley, *Org. Process Res. Dev.* **2016**, *20*, 386.
- [278] R. J. Ingham, C. Battilocchio, J. M. Hawkins, S. V. Ley, *Beilstein J Org Chem* **2014**, *10*, 641.
- [279] M. O'Brien, P. Koos, D. L. Browne, S. V. Ley, *Org. Biomol. Chem.* **2012**, *10*, 7031.
- [280] M. O'Brien, L. Konings, M. Martin, J. Heap, *Tetrahedron Letters* **2017**, *58*, 2409.
- [281] R. J. Ingham, C. Battilocchio, D. E. Fitzpatrick, E. Sliwinski, J. M. Hawkins, S. V. Ley, *Angew Chem Int Ed Engl* **2015**, *54*, 144.
- [282] A. Ceroni, A. Dell, S. M. Haslam, *Source code for biology and medicine* **2007**, *2*, 3.
- [283] D. Damerell, A. Ceroni, K. Maass, R. Ranzinger, A. Dell, S. M. Haslam, *Biological chemistry* **2012**, *393*, 1357.
- [284] H. S. Hahm, C.-F. Liang, C.-H. Lai, R. J. Fair, F. Schuhmacher, P. H. Seeberger, *The Journal of organic chemistry* **2016**, *81*, 5866.
- [285] U. Rester, *Current opinion in drug discovery & development* **2008**, *11*, 559.
- [286] a) M. Gütlein, A. Karwath, S. Kramer, *Journal of cheminformatics* **2012**, *4*, 7; b) M. Gütlein, A. Karwath, S. Kramer, *J Cheminf* **2014**, *6*, 12.

- [287] P. Ertl, *Journal of cheminformatics* **2010**, *2*, 1.
- [288] a) F. Csizmadia, *J. Chem. Inf. Comput. Sci.* **2000**, *40*, 323; b) B. Bienfait, P. Ertl, *J Cheminf* **2013**, *5*, 24.
- [289] B. Karulin, M. Kozhevnikov, *J Cheminf* **2011**, *3*.
- [290] a) M. E. Cass, H. S. Rzepa, D. R. Rzepa, C. K. Williams, *J. Chem. Educ.* **2005**, *82*, 1736; b) A. Herráez, *Biochemistry and molecular biology education : a bimonthly publication of the International Union of Biochemistry and Molecular Biology* **2006**, *34*, 255.
- [291] S. Krause, E. Willighagen, C. Steinbeck, *Molecules* **2000**, *5*, 93.
- [292] a) Z. Li, H. Wan, Y. Shi, P. Ouyang, *J. Chem. Inf. Model.* **2004**, *44*, 1886; b) A. D. Hunter, *J. Chem. Educ.* **1997**, *74*, 905; c) A. Sánchez-Ferrer, E. Núñez-Delicado, R. Bru, *Trends in Biochemical Sciences* **1995**, *20*, 286; d) P. Selvarani, V. Shanthi, C. K. Rajesh, S. Saravanan, K. Sekar, *Nucleic Acids Research* **2004**, *32*, W645-8.
- [293] T. Klucznik, B. Mikulak-Klucznik, M. P. McCormack, H. Lima, S. Szymkuć, M. Bhowmick, K. Molga, Y. Zhou, L. Rickershauser, E. P. Gajewska et al., *Chem* **2018**, *4*, 522.
- [294] a) A. J. Williams, H. E. Pence, *J. Chem. Educ.* **2011**, *88*, 683; b) D. Libman, L. Huang, *J. Chem. Educ.* **2013**, *90*, 320.
- [295] S. McNicholas, J. Agirre, *Acta Cryst D* **2017**, *73*, 187.
- [296] S. Tsuchiya, N. P. Aoki, D. Shinmachi, M. Matsubara, I. Yamada, K. F. Aoki-Kinoshita, H. Narimatsu, *Carbohydr Res* **2017**, *445*, 104.
- [297] Lieth, Claus-Wilhelm von der, A. Bohne-Lang, K. K. Lohmann, M. Frank, *Briefings in Bioinformatics* **2004**, *5*, 164.
- [298] K. Cheng, Y. Zhou, S. Neelamegham, *Glycobiology* **2017**, *27*, 200.
- [299] D. Alocci, P. Suchánková, R. Costa, N. Hory, J. Mariethoz, R. S. Vařeková, P. Toukach, F. Lisacek, *Molecules (Basel, Switzerland)* **2018**, *23*.
- [300] H. J. Joshi, Lieth, Claus-Wilhelm von der, N. H. Packer, M. R. Wilkins, *Nucleic Acids Research* **2010**, *38*, W667-70.
- [301] D. F. Thieker, J. A. Hadden, K. Schulten, R. J. Woods, *Glycobiology* **2016**, *26*, 786.
- [302] a) S. B. Engelsens, S. Cros, W. Mackie, S. Pérez, *Biopolymers* **1996**, *39*, 417; b) S. B. Engelsens, P. I. Hansen, S. Pérez, *Biopolymers* **2014**, *101*, 733.
- [303] S. Pérez, T. Tubiana, A. Imberty, M. Baaden, *Glycobiology* **2015**, *25*, 483.
- [304] M. M. Kuttel, J. Stähle, G. Widmalm, *Journal of Computational Chemistry* **2016**, *37*, 2098.
- [305] M. P. Campbell, R. Ranzinger, T. Lütteke, J. Mariethoz, C. A. Hayes, J. Zhang, Y. Akune, K. F. Aoki-Kinoshita, D. Damerell, G. Carta et al., *BMC Bioinformatics* **2014**, *15*, S9.
- [306] a) K. Berka, O. Hanák, D. Sehnal, P. Banás, V. Navrátilová, D. Jaiswal, C.-M. Ionescu, R. Svobodová Vareková, J. Koca, M. Otyepka, *Nucleic Acids Research* **2012**, *40*, W222-7; b) M.

- Eavenson, K. J. Kochut, J. A. Miller, R. Ranzinger, M. Tiemeyer, K. Aoki, W. S. York, *Glycobiology* **2014**, *25*, 66; c) D. Alocci, J. Mariethoz, O. Horlacher, J. T. Bolleman, M. P. Campbell, F. Lisacek, *PLoS one* **2015**, *10*, e0144578; d) R. Ranzinger, K. J. Kochut, J. A. Miller, M. Eavenson, T. Lütteke, W. S. York, *Perspectives in science* **2017**, *11*, 24.
- [307] a) S. Jo, K. C. Song, H. Desaire, A. D. MacKerell, W. Im, *Journal of Computational Chemistry* **2011**, *32*, 3135; b) S.-J. Park, J. Lee, D. S. Patel, H. Ma, H. S. Lee, S. Jo, W. Im, *Bioinformatics (Oxford, England)* **2017**, *33*, 3051.
- [308] a) C. L. Bird, C. Willoughby, J. G. Frey, *Chemical Society reviews* **2013**, *42*, 8157; b) S. Kanza, C. Willoughby, N. Gibbins, R. Whitby, J. G. Frey, J. Erjavec, K. Zupančič, M. Hren, K. Kovač, *J Cheminf* **2017**, *9*, 31; c) E. M. Riley, H. Z. Hattaway, P. A. Felse, *Journal of biological engineering* **2017**, *11*, 40; d) A. E. Day, S. J. Coles, C. L. Bird, J. G. Frey, R. J. Whitby, V. E. Tkachenko, A. J. Williams, *Journal of chemical information and modeling* **2015**, *55*, 501.
- [309] a) F. Rudolphi, *Nachr. Chem.* **2010**, *58*, 548; b) F. Rudolphi, L. J. Goossen, *Journal of chemical information and modeling* **2012**, *52*, 293.
- [310] a) P. Tremouilhac, A. Nguyen, Y.-C. Huang, S. Kotov, D. S. Lütjohann, F. Hübsch, N. Jung, S. Bräse, *J Cheminf* **2017**, *9*, 54; b) N. Jung, P. Tremouilhac, C. Kramer, J. Potthoff, *Erfassung und Speicherung von Forschungsdaten im Fachbereich Chemie: Bereitstellung moderner Forschungsinfrastrukturen durch ein elektronisches Laborjournal mit Repositorium-Anbindung*, Karlsruhe, **2017**; c) S. Kotov, P. Tremouilhac, N. Jung, S. Bräse, *J Cheminf* **2018**, *10*, 38.
- [311] P. Shrikant, B. Harshul, *J. Eng. Technol. Res.* **2018**, *10*, 1.
- [312] a) A. Bauch, I. Adamczyk, P. Buczek, F.-J. Elmer, K. Enimanev, P. Glyzowski, M. Kohler, T. Pylak, A. Quandt, C. Ramakrishnan et al., *BMC Bioinformatics* **2011**, *12*, 468; b) C. Barillari, D. S. M. Ottoz, J. M. Fuentes-Serna, C. Ramakrishnan, B. Rinn, F. Rudolf, *Bioinformatics (Oxford, England)* **2016**, *32*, 638.
- [313] L. W. Sumner, A. Amberg, D. Barrett, M. H. Beale, R. Beger, C. A. Daykin, T. W.-M. Fan, O. Fiehn, R. Goodacre, J. L. Griffin et al., *Metabolomics* **2007**, *3*, 211.
- [314] C. F. Taylor, N. W. Paton, K. S. Lilley, P.-A. Binz, R. K. J. Jr, A. R. Jones, W. Zhu, R. Apweiler, R. Aebersold, E. W. Deutsch et al., *Nature Biotechnology* **2007**, *25*, 887.
- [315] A. Bandrowski, M. Brush, J. S. Grethe, M. A. Haendel, D. N. Kennedy, S. Hill, P. R. Hof, M. E. Martone, M. Pols, S. Tan et al., *F1000Research* **2015**, *4*.
- [316] D. Kolarich, E. Rapp, W. B. Struwe, S. M. Haslam, J. Zaia, R. McBride, S. Agravat, M. P. Campbell, M. Kato, R. Ranzinger et al., *Molecular & Cellular Proteomics* **2013**, *12*, 991.
- [317] W. S. York, S. Agravat, K. F. Aoki-Kinoshita, R. McBride, M. P. Campbell, C. E. Costello, A. Dell, T. Feizi, S. M. Haslam, N. Karlsson et al., "MIRAGE: The minimum information required for a glycomics experiment", **2014**.
- [318] W. B. Struwe, S. Agravat, K. F. Aoki-Kinoshita, M. P. Campbell, C. E. Costello, A. Dell, T. Feizi, S. M. Haslam, N. G. Karlsson, K.-H. Khoo et al., *Glycobiology* **2016**, *26*, 907.
- [319] A. Ceroni, K. Maass, H. Geyer, R. Geyer, A. Dell, S. M. Haslam, *J. Proteome Res.* **2008**, *7*, 1650.

- [320] M. A. Rojas-Macias, J. Mariethoz, P. Andersson, C. Jin, V. Venkatakrishnan, N. P. Aoki, D. Shinmachi, C. Ashwood, K. Madunic, T. Zhang et al., *e-workflow for recording of glycomic mass spectrometric data in compliance with reporting guidelines*, **2018**.
- [321] M. D. Wilkinson, M. Dumontier, I. J. Aalbersberg, G. Appleton, M. Axton, A. Baak, N. Blomberg, J.-W. Boiten, da Silva Santos, Luiz Bonino, P. E. Bourne et al., *Scientific Data* **2016**, *3*, 160018 EP -.
- [322] M. Boeckhout, G. A. Zielhuis, A. L. Bredenoord, *European Journal of Human Genetics* **2018**, *26*, 931.
- [323] L. Reiser, L. Harper, M. Freeling, B. Han, S. Luan, *Molecular plant* **2018**, *11*, 1105.
- [324] I./A. I. C. o. A. S. Engineering (Hrsg.) *2015 30th IEEE/ACM International Conference on Automated Software Engineering. 9-13 November 2015, Lincoln, Nebraska : proceedings*, IEEE, Piscataway, NJ, **2015**.
- [325] IEEE (Hrsg.) *ICSA 2017. Proceedings : 2017 IEEE International Conference on Software Architecture : 3-7 April 2017, Gothenburg, Sweden*, IEEE, New York, **2017**.
- [326] a) B. Varanasi, S. Belida in *Books for professionals by professionals* (Hrsg.: B. Varanasi, S. Belida), Apress, New York, NY, **2015**, S. 91–104; b) J. Purushothaman, *RESTful Java Web Services - Second Edition*, Packt Publishing, s.l., **2015**.
- [327] M. C. Wendl, S. Smith, C. S. Pohl, D. J. Dooling, A. T. Chinwalla, K. Crouse, T. Hepler, S. Leong, L. Carmichael, M. Nhan et al., *BMC Bioinformatics* **2007**, *8*, 362.
- [328] a) M. D. Wilkinson, M. Dumontier, I. J. Aalbersberg, G. Appleton, M. Axton, A. Baak, N. Blomberg, J.-W. Boiten, da Silva Santos, Luiz Bonino, P. E. Bourne et al., *Scientific Data* **2016**, *3*, 160018 EP -; b) L. Reiser, L. Harper, M. Freeling, B. Han, S. Luan, *Molecular Plant* **2018**, *11*, 1105; c) M. Dumontier, K. Wesley, *The Serials Librarian* **2018**, *74*, 39.
- [329] M. Boeckhout, G. A. Zielhuis, A. L. Bredenoord, *European Journal of Human Genetics* **2018**, *26*, 931.
- [330] J. D. Blischak, E. R. Davenport, G. Wilson, *PLoS computational biology* **2016**, *12*, e1004668.
- [331] a) *acatech HORIZONTE: Blockchain*, München, **2018**; b) D. Drescher, *Blockchain Grundlagen. Eine Einführung in die elementaren Konzepte in 25 Schritten (Übersetzung aus dem Englischen von Guido Lenz)*, mitP Verlags GmbH & Co. KG, Frechen, **2017**; c) Primavera De Filippi, Aaron Wright, *Blockchain and the Law: The Rule of Code*, Harvard University Press, Cambridge, MA, **2018**.
- [332] in *Sicherung Guter Wissenschaftlicher Praxis* (Hrsg.: D. F. DFG), Wiley-VCH Verlag GmbH & Co. KGaA, Weinheim, Germany, **2013**, S. 1–109.
- [333] a) R. van Noorden, *Nature* **2013**, *500*, 243; b) R. van Noorden, *Nature News* **2014**, *512*, 126.
- [334] J. Perkel, *Nature News* **2016**, *538*, 127.
- [335] a) X. Dong, K. E. Gilbert, R. Guha, R. Heiland, J. Kim, M. E. Pierce, G. C. Fox, D. J. Wild, *J. Chem. Inf. Model.* **2007**, *47*, 1303; b) J.-C. Bradley, *Open Notebook Science Using Blogs and Wikis*, **2007**, Nature Publishing Group, zu finden unter <http://precedings.nature.com/documents/39/version/1/files/npre200739-1.pdf>.

- [336] C. L. Bird, J. G. Frey, *Chem. Soc. Rev.* **2013**, *42*, 6754.
- [337] M. S. Marshall, R. Boyce, H. F. Deus, J. Zhao, E. L. Willighagen, M. Samwald, E. Pichler, J. Hajagos, E. Prud'hommeaux, S. Stephens, *Journal of Web Semantics* **2012**, *14*, 2.
- [338] K. F. Aoki-Kinoshita, J. Bolleman, M. P. Campbell, S. Kawano, J.-D. Kim, T. Lütteke, M. Matsubara, S. Okuda, R. Ranzinger, H. Sawaki et al., *J Biomed Sem* **2013**, *4*, 39.
- [339] R. Ranzinger, K. F. Aoki-Kinoshita, M. P. Campbell, S. Kawano, T. Lütteke, S. Okuda, D. Shinmachi, T. Shikanai, H. Sawaki, P. Toukach et al., *Bioinformatics (Oxford, England)* **2015**, *31*, 919.
- [340] W. Hu, H. Qiu, J. Huang, M. Dumontier, *Database : the journal of biological databases and curation* **2017**, 2017.
- [341] B. Chen, X. Dong, D. Jiao, H. Wang, Q. Zhu, Y. Ding, D. J. Wild, *BMC Bioinformatics* **2010**, *11*, 255.
- [342] P. Mika, T. Tudorache, A. Bernstein, C. Welty, C. Knoblock, D. Vrandečić, P. Groth, N. Noy, K. Janowicz, C. Goble et al. (Hrsg.) *Introducing Wikidata to the Linked Data Web. The Semantic Web – ISWC 2014*, Springer International Publishing, **2014**.
- [343] E. Blomqvist, K. Hose, H. Paulheim, A. Ławrynowicz, F. Ciravegna, O. Hartig, F. Å. Nielsen, D. Mietchen, E. Willighagen (Hrsg.) *Scholia, Scientometrics and Wikidata. The Semantic Web: ESWC 2017 Satellite Events*, Springer International Publishing, **2017**.
- [344] M. Kutmon, A. Riutta, N. Nunes, K. Hanspers, E. L. Willighagen, A. Bohler, J. Mélius, A. Waagmeester, S. R. Sinha, R. Miller et al., *Nucleic Acids Research* **2016**, *44*, D488-94.
- [345] A. Waagmeester, M. Kutmon, A. Riutta, R. Miller, E. L. Willighagen, C. T. Evelo, A. R. Pico, *PLoS computational biology* **2016**, *12*, e1004989.
- [346] R. A. Miller, P. Woollard, E. L. Willighagen, D. Digles, M. Kutmon, A. Loizou, A. Waagmeester, S. Senger, C. T. Evelo, *F1000Research* **2018**, *7*.
- [347] E. Costello, *Open Learning: The Journal of Open, Distance and e-Learning* **2013**, *28*, 187.
- [348] N. Le Novère, A. Finney, M. Hucka, U. S. Bhalla, F. Campagne, J. Collado-Vides, E. J. Crampin, M. Halstead, E. Klipp, P. Mendes et al., *Nature Biotechnology* **2005**, *23*, 1509.
- [349] a) P. Murray-Rust, H. S. Rzepa, *Journal of cheminformatics* **2011**, *3*, 44; b) W. Phadungsukanan, M. Kraft, J. A. Townsend, P. Murray-Rust, *Journal of cheminformatics* **2012**, *4*, 15.
- [350] a) P. N. Spahn, N. E. Lewis, *Current Opinion in Biotechnology* **2014**, *30*, 218; b) A. Shubhakar, K. R. Reiding, R. A. Gardner, D. I. R. Spencer, D. L. Fernandes, M. Wührer, *Chromatographia* **2015**, *78*, 321.
- [351] H. L. Morgan, *J. Chem. Doc.* **1965**, *5*, 107.
- [352] E. E. Bolton, Y. Wang, P. A. Thiessen, S. H. Bryant in *Annual Reports in Computational Chemistry* (Hrsg.: D. C. Spellmeyer, R. A. Wheeler), Elsevier Science, Amsterdam, London, **2008**, S. 217–241.

- [353] Y. Wang, J. Xiao, T. O. Suzek, J. Zhang, J. Wang, S. H. Bryant, *Nucleic Acids Research* **2009**, *37*, W623-33.
- [354] W. D. Ihlenfeldt, E. E. Bolton, S. H. Bryant, *Journal of cheminformatics* **2009**, *1*, 20.
- [355] W. D. Ihlenfeldt, *J Cheminf* **2013**, *5*.
- [356] T. Cheng, Y. Pan, M. Hao, Y. Wang, S. H. Bryant, *Drug Discovery Today* **2014**, *19*, 1751.
- [357] S. Kim, P. A. Thiessen, E. E. Bolton, S. H. Bryant, *Nucleic Acids Res* **2015**, *43*, W605-W611.
- [358] S. Kim, P. A. Thiessen, E. E. Bolton, J. Chen, G. Fu, A. Gindulyte, L. Han, J. He, S. He, B. A. Shoemaker et al., *Nucleic Acids Research* **2016**, *44*, D1202-13.
- [359] S. Kim, *Expert opinion on drug discovery* **2016**, *11*, 843.
- [360] Y. Wang, T. Cheng, S. H. Bryant, *SLAS discovery : advancing life sciences R & D* **2017**, *22*, 655.
- [361] V. D. Hähnke, S. Kim, E. E. Bolton, *J Cheminf* **2018**, *10*, 36.
- [362] S. Kim, P. A. Thiessen, T. Cheng, B. Yu, E. E. Bolton, *Nucleic Acids Research* **2018**, *46*, W563-W570.
- [363] S. Kim, J. Chen, T. Cheng, A. Gindulyte, J. He, S. He, Q. Li, B. A. Shoemaker, P. A. Thiessen, B. Yu et al., *Nucleic Acids Research* **2019**, *47*, D1102-D1109.
- [364] S. Kim in *Encyclopedia of bioinformatics and computational biology* (Hrsg.: S. Ranganathan, M. R. Gribskov, K. Nakai, C. Schönbach), Elsevier, Amsterdam, Netherlands, Cambridge, MA, United States, **2019**, S. 628–639.
- [365] K. Degtyarenko, J. Hastings, P. de Matos, M. Ennis, *Current protocols in bioinformatics* **2009**, *Chapter 14*, Unit 14.9.
- [366] J. Hastings, G. Owen, A. Dekker, M. Ennis, N. Kale, V. Muthukrishnan, S. Turner, N. Swainston, P. Mendes, C. Steinbeck, *Nucleic Acids Research* **2016**, *44*, D1214-9.
- [367] N. Swainston, J. Hastings, A. Dekker, V. Muthukrishnan, J. May, C. Steinbeck, P. Mendes, *J Cheminf* **2016**, *8*, 11.
- [368] J. D. Ferreira, F. M. Couto, *PLoS computational biology* **2010**, *6*.
- [369] N. Y. Mok, R. Brenk, *Journal of chemical information and modeling* **2011**, *51*, 2449.
- [370] E. L. Willighagen, A. Waagmeester, O. Spjuth, P. Ansell, A. J. Williams, V. Tkachenko, J. Hastings, B. Chen, D. J. Wild, *Journal of cheminformatics* **2013**, *5*, 23.
- [371] A. P. Bento, A. Gaulton, A. Hersey, L. J. Bellis, J. Chambers, M. Davies, F. A. Krüger, Y. Light, L. Mak, S. McGlinchey et al., *Nucleic Acids Research* **2014**, *42*, D1083-90.
- [372] M. Davies, M. Nowotka, G. Papadatos, N. Dedman, A. Gaulton, F. Atkinson, L. Bellis, J. P. Overington, *Nucleic Acids Research* **2015**, *43*, W612-20.
- [373] G. Papadatos, A. Gaulton, A. Hersey, J. P. Overington, *J Comput Aided Mol Des* **2015**, *29*, 885.

- [374] G. Papadatos, A. Gaulton, A. Hersey, J. P. Overington, *Journal of Computer-Aided Molecular Design* **2015**, *29*, 885.
- [375] A. Gaulton, A. Hersey, M. Nowotka, A. P. Bento, J. Chambers, D. Mendez, P. Mutowo, F. Atkinson, L. J. Bellis, E. Cibrián-Uhalte et al., *Nucleic Acids Research* **2017**, *45*, D945-D954.
- [376] X. Chen, M. Liu, M. Gilson, *CCHTS* **2001**, *4*, 719.
- [377] T. Liu, Y. Lin, X. Wen, R. N. Jorissen, M. K. Gilson, *Nucleic Acids Research* **2007**, *35*, D198-201.
- [378] A. M. Wassermann, J. Bajorath, *Expert opinion on drug discovery* **2011**, *6*, 683.
- [379] M. K. Gilson, T. Liu, M. Baitaluk, G. Nicola, L. Hwang, J. Chong, *Nucleic Acids Research* **2016**, *44*, D1045-53.
- [380] G. Nicola, M. R. Berthold, M. P. Hedrick, M. K. Gilson, *Database : the journal of biological databases and curation* **2015**, 2015.
- [381] C. A. Cooper, *Nucl Acids Res* **2003**, *31*, 511.
- [382] C. A. Cooper, M. J. Harrison, J. M. Webster, M. R. Wilkins, N. H. Packer, *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing* **2002**, 297.
- [383] M. P. Campbell, R. Peterson, J. Mariethoz, E. Gasteiger, Y. Akune, K. F. Aoki-Kinoshita, F. Lisacek, N. H. Packer, *Nucleic Acids Research* **2013**, *42*, D215-D221.
- [384] von der Lieth, C.-W., A. A. Freire, D. Blank, M. P. Campbell, A. Ceroni, D. R. Damerell, A. Dell, R. A. Dwek, B. Ernst, R. Fogh et al., *Glycobiology* **2011**, *21*, 493.
- [385] K. Al Jadda, M. P. Porterfield, R. Bridger, C. Heiss, M. Tiemeyer, L. Wells, J. A. Miller, W. S. York, R. Ranzinger, *Bioinformatics (Oxford, England)* **2015**, *31*, 242.
- [386] S. Doubet, P. Albersheim, *Glycobiology* **1992**, *2*, 505.
- [387] S. Doubet, K. Bock, D. Smith, A. Darvill, P. Albersheim, *Trends in Biochemical Sciences* **1989**, *14*, 475.
- [388] P. V. Toukach, K. S. Egorova, *Nucleic Acids Research* **2016**, *44*, D1229-36.
- [389] K. S. Egorova, A. N. Kondakova, P. V. Toukach, *Database : the journal of biological databases and curation* **2015**, 2015.
- [390] K. F. Aoki-Kinoshita (Hrsg.) *A Practical Guide to Using Glycomics Databases*, Springer, Tokyo, **2017**.
- [391] P. V. Toukach, K. S. Egorova in *Springer protocols, Vol. 1273* (Hrsg.: T. Lütteke, M. Frank), Humana Press, New York, **2015**, S. 55–85.
- [392] M. A. Rojas-Macias, J. Stähle, T. Lütteke, G. Widmalm, *Glycobiology* **2015**, *25*, 341.
- [393] A. Sarkar, S. Pérez, *BMC Bioinformatics* **2012**, *13*, 302.
- [394] K. Hashimoto, S. Goto, S. Kawano, K. F. Aoki-Kinoshita, N. Ueda, M. Hamajima, T. Kawasaki, M. Kanehisa, *Glycobiology* **2006**, *16*, 63.

- [395] K. F. Aoki-Kinoshita, M. Kanehisa, *Methods in molecular biology (Clifton, N.J.)* **2015**, 1273, 97.
- [396] M. Kanehisa, *Nucl Acids Res* **2000**, 28, 27.
- [397] M. Kanehisa, M. Araki, S. Goto, M. Hattori, M. Hirakawa, M. Itoh, T. Katayama, S. Kawashima, S. Okuda, T. Tokimatsu et al., *Nucleic Acids Research* **2008**, 36, D480-4.
- [398] M. Kanehisa, S. Goto, M. Furumichi, M. Tanabe, M. Hirakawa, *Nucleic Acids Research* **2010**, 38, D355-60.
- [399] M. Kanehisa, S. Goto, M. Hattori, K. F. Aoki-Kinoshita, M. Itoh, S. Kawashima, T. Katayama, M. Araki, M. Hirakawa, *Nucleic Acids Research* **2006**, 34, D354-7.
- [400] M. Kanehisa, S. Goto, S. Kawashima, Y. Okuno, M. Hattori, *Nucleic Acids Research* **2004**, 32, D277-80.
- [401] M. Kanehisa, Y. Sato, M. Kawashima, M. Furumichi, M. Tanabe, *Nucleic Acids Research* **2016**, 44, D457-62.
- [402] K. Aoki-Kinoshita, S. Agravat, N. P. Aoki, S. Arpinar, R. D. Cummings, A. Fujita, N. Fujita, G. M. Hart, S. M. Haslam, T. Kawasaki et al., *Nucleic Acids Research* **2015**.
- [403] M. Tiemeyer, K. Aoki, J. Paulson, R. D. Cummings, W. S. York, N. G. Karlsson, F. Lisacek, N. H. Packer, M. P. Campbell, N. P. Aoki et al., *Glycobiology* **2017**, 27, 915.
- [404] R. Eils (Hrsg.) *Computational systems biology. [from molecular mechanisms to disease]*, Elsevier, Academic Press, Amsterdam, Boston, Heidelberg, **2014**.
- [405] T. Lütteke, A. Bohne-Lang, A. Loss, T. Goetz, M. Frank, C.-W. von der Lieth, *Glycobiology* **2006**, 16, 71.
- [406] A. Loss, T. Lütteke, *Methods in molecular biology (Clifton, N.J.)* **2015**, 1273, 87.
- [407] *Databases and tools of GLYCOSCIENCES.de web server*, **2015**.
- [408] M. Böhm, A. Bohne-Lang, M. Frank, A. Loss, M. A. Rojas-Macias, T. Lütteke, *Nucleic Acids Research* **2019**, 47, D1195-D1201.
- [409] S. Jo, W. Im, *Nucleic Acids Research* **2013**, 41, 4.
- [410] M. Maeda, N. Fujita, Y. Suzuki, H. Sawaki, T. Shikanai, H. Narimatsu, *Methods in molecular biology (Clifton, N.J.)* **2015**, 1273, 161.
- [411] T. Shikanai, H. Narimatsu in *Glycoscience: Biology and Medicine* (Hrsg.: T. Endo, P. H. Seeberger, G. W. Hart, C.-H. Wong, N. Taniguchi), Springer Japan, Tokyo, **2012**, S. 1–5.
- [412] K. F. Aoki-Kinoshita, N. P. Aoki, A. Fujita, N. Fujita, T. Kawasaki, M. Matsubara, S. Okuda, T. Shikanai, D. Shinmachi, E. Solovieva et al., *Perspectives in science* **2017**, 11, 18.
- [413] Peter Mika, Tania Tudorache, Abraham Bernstein, Chris Welty, Craig Knoblock, Denny Vrandečić, Paul Groth, Natasha Noy, Krzysztof Janowicz, Carole Goble (Hrsg.) *The Semantic Web – ISWC 2014*, Springer, Cham, **2014**.
- [414] X. Chen, Y. Lin, M. K. Gilson, *Biopolymers* **2001**, 61, 127.

- [415] L. Jeske, S. Placzek, I. Schomburg, A. Chang, D. Schomburg, *Nucleic Acids Research* **2019**, *47*, D542-D549.
- [416] a) B. L. Cantarel, P. M. Coutinho, C. Rancurel, T. Bernard, V. Lombard, B. Henrissat, *Nucleic Acids Research* **2009**, *37*, 8; b) B. A. BURKETT, *COSMOS* **2008**, *04*, 17.
- [417] R. Raman, M. Venkataraman, S. Ramakrishnan, W. Lang, S. Raguram, R. Sasisekharan, *Glycobiology* **2006**, *16*, 82.
- [418] a) P. de Matos, R. Alcántara, A. Dekker, M. Ennis, J. Hastings, K. Haug, I. Spiteri, S. Turner, C. Steinbeck, *Nucleic Acids Research* **2010**, *38*, D249-54; b) A. Labarga, F. Valentin, M. Anderson, R. Lopez, *Nucleic Acids Research* **2007**, *35*, W6-11.
- [419] a) M. P. Campbell, L. Royle, C. M. Radcliffe, R. A. Dwek, P. M. Rudd, *Bioinformatics* **2008**, *24*, 1214; b) M. P. Campbell, L. Royle, P. M. Rudd, *Methods in molecular biology (Clifton, N.J.)* **2015**, *1273*, 17.
- [420] M. A. Rojas-Macias, A. Loss, A. Bohne-Lang, M. Frank, T. Lütteke in *Endo, Seeberger et al. (Hg.) 2014 – Glycoscience*, S. 1–6.
- [421] a) R. Ranzinger, S. Herget, T. Wetter, von der Lieth, Claus-Wilhelm, *BMC Bioinformatics* **2008**, *9*, 384; b) R. Ranzinger, S. Herget, Lieth, Claus-Wilhelm von der, M. Frank, *Nucl Acids Res* **2010**, *39*, D373-6.
- [422] a) R. Apweiler, *Nucl Acids Res* **2004**, *32*, 115D-119; b) A. Bairoch, R. Apweiler, C. H. Wu, W. C. Barker, B. Boeckmann, S. Ferro, E. Gasteiger, H. Huang, R. Lopez, M. Magrane et al., *Nucleic Acids Research* **2005**, *33*, D154-9; c) *Nucleic Acids Research* **2015**, *43*, D204-12.
- [423] S. R. Heller, A. McNaught, I. Pletnev, S. Stein, D. Tchekhovskoi, *J Cheminf* **2015**, *7*.
- [424] G. Grethe, J. M. Goodman, C. H. Allen, *J Cheminf* **2013**, *5*, 45.
- [425] D. Weininger, *J. Chem. Inf. Model.* **1988**, *28*, 31.
- [426] P. Minkiewicz, A. Iwaniak, M. Darewicz, *Molecules* **2017**, *22*, 2075.
- [427] N. M. O'Boyle, M. Banck, C. A. James, C. Morley, T. Vandermeersch, G. R. Hutchison, *J Cheminf* **2011**, *3*, 33.
- [428] D. M. Lowe, P. T. Corbett, P. Murray-Rust, R. C. Glen, *Journal of chemical information and modeling* **2011**, *51*, 739.
- [429] M. L. Peach, M. C. Nicklaus in *Applied chemoinformatics. Achievements and future opportunities* (Hrsg.: T. Engel, J. Gasteiger), Wiley-VCH, Weinheim, Germany, **2018**, S. 385–393.
- [430] G. Wohlgemuth, P. K. Haldiya, E. Willighagen, T. Kind, O. Fiehn, *Bioinformatics* **2010**, *26*, 2647.
- [431] G. L. Holliday, P. Murray-Rust, H. S. Rzepa, *Journal of chemical information and modeling* **2006**, *46*, 145.
- [432] S. Kuhn, T. Helmus, R. J. Lancashire, P. Murray-Rust, H. S. Rzepa, C. Steinbeck, E. L. Willighagen, *Journal of chemical information and modeling* **2007**, *47*, 2015.

- [433] G. V. Gkoutos, P. Murray-Rust, H. S. Rzepa, M. Wright, *J. Chem. Inf. Comput. Sci.* **2001**, *41*, 1124.
- [434] C. Steinbeck, Y. Han, S. Kuhn, O. Horlacher, E. Luttmann, E. Willighagen, *Journal of chemical information and computer sciences* **2003**, *43*, 493.
- [435] a) O. Spjuth, T. Helmus, E. L. Willighagen, S. Kuhn, M. Eklund, J. Wagener, P. Murray-Rust, C. Steinbeck, J. E. S. Wikberg, *BMC Bioinformatics* **2007**, *8*, 59; b) O. Spjuth, J. Alvarsson, A. Berg, M. Eklund, S. Kuhn, C. Mäsak, G. Torrance, J. Wagener, E. L. Willighagen, C. Steinbeck et al., *BMC Bioinformatics* **2009**, *10*, 397.
- [436] T. Steger-Hartmann, F. Pognan, *Basic & clinical pharmacology & toxicology* **2018**, *123 Suppl 5*, 29.
- [437] M. Cronin, F. Bois, E. Fioravanzo, T. Meinl, D. Neagu, A. Worth, C. Yang, A.-N. Richarz, *Toxicology Letters* **2015**, *238*, S169-S170.
- [438] a) B. Hardy, N. Douglas, C. Helma, M. Rautenberg, N. Jeliaskova, V. Jeliaskov, I. Nikolova, R. Benigni, O. Tcheremenskaia, S. Kramer et al., *Journal of cheminformatics* **2010**, *2*, 7; b) O. Tcheremenskaia, R. Benigni, I. Nikolova, N. Jeliaskova, S. E. Escher, M. Batke, T. Baier, V. Poroikov, A. Lagunin, M. Rautenberg et al., *Journal of biomedical semantics* **2012**, *3 Suppl 1*, S7.
- [439] N. Jeliaskova, V. Jeliaskov, *Journal of cheminformatics* **2011**, *3*, 18.
- [440] A. J. Williams, C. M. Grulke, J. Edwards, A. D. McEachran, K. Mansouri, N. C. Baker, G. Patlewicz, I. Shah, J. F. Wambaugh, R. S. Judson et al., *J Cheminf* **2017**, *9*, 61.
- [441] A. J. Williams, L. Harland, P. Groth, S. Pettifer, C. Chichester, E. L. Willighagen, C. T. Evelo, N. Blomberg, G. Ecker, C. Goble et al., *Drug Discovery Today* **2012**, *17*, 1188.
- [442] A. Karim, A. Mishra, M. A. H. Newton, A. Sattar, *ACS Omega* **2019**, *4*, 1874.
- [443] S. Soh, Y. Wei, B. Kowalczyk, C. M. Gothard, B. Baytekin, N. Gothard, B. A. Grzybowski, *Chem. Sci.* **2012**, *3*, 1497.
- [444] a) L. L. Haak, M. Fenner, L. Paglione, E. Pentz, H. Ratner, *Learn. Pub.* **2012**, *25*, 259; b) W. J. Thomas, B. Chen, G. Clement, *The Serials Librarian* **2015**, *68*, 332.
- [445] a) T. Zhang, H. Li, H. Xi, R. V. Stanton, S. H. Rotstein, *Journal of chemical information and modeling* **2012**, *52*, 2796; b) J. Milton, T. Zhang, C. Bellamy, E. Swayze, C. Hart, M. Weisser, S. Hecht, S. Rotstein, *Journal of chemical information and modeling* **2017**, *57*, 1233.
- [446] a) K. Tanaka, K. F. Aoki-Kinoshita, M. Kotera, H. Sawaki, S. Tsuchiya, N. Fujita, T. Shikanai, M. Kato, S. Kawano, I. Yamada et al., *Journal of chemical information and modeling* **2014**, *54*, 1558; b) M. Matsubara, K. F. Aoki-Kinoshita, N. P. Aoki, I. Yamada, H. Narimatsu, *Journal of chemical information and modeling* **2017**, *57*, 632.
- [447] D. Solís, N. V. Bovin, A. P. Davis, J. Jiménez-Barbero, A. Romero, R. Roy, K. Smetana, H.-J. Gabius, *Biochimica et biophysica acta* **2015**, *1850*, 186.
- [448] N. H. Tran, Z. Levine, L. Xin, B. Shan, M. Li, *Protein identification with deep learning: from abc to xyz*, **2017**, zu finden unter <https://arxiv.org/pdf/1710.02765>.
- [449] J. S. Chauhan, A. Rao, G. P. S. Raghava, *PloS one* **2013**, *8*, e67008.

- [450] W. Humphrey, A. Dalke, K. Schulten, *Journal of Molecular Graphics* **1996**, *14*, 33.
- [451] M. Irida, S. Gondo, Y. Fujishima, K. Kakizaki, *Biophysics (Nagoya-shi, Japan)* **2007**, *3*, 57.
- [452] S. McNicholas, E. Potterton, K. S. Wilson, M. E. M. Noble, *Acta crystallographica. Section D, Biological crystallography* **2011**, *67*, 386.
- [453] R. Danne, C. Poojari, H. Martinez-Seara, S. Rissanen, F. Lolicato, T. Róg, I. Vattulainen, *Journal of chemical information and modeling* **2017**, *57*, 2401.
- [454] R. Pendrill, K. H. M. Jonsson, G. Widmalm, *Pure and Applied Chemistry* **2013**, *85*, 1759.
- [455] J. F.G. Vliegthart, *Perspectives in science* **2017**, *11*, 3.
- [456] C. Kettner, M. G. Hicks, *Perspectives in science* **2017**, *11*, 1.
- [457] M. Kowalik, C. M. Gothard, A. M. Drews, N. A. Gothard, A. Weckiewicz, P. E. Fuller, B. A. Grzybowski, K. J. M. Bishop, *Angew Chem Int Ed Engl* **2012**, *51*, 7928.
- [458] B. A. Grzybowski, K. J. M. Bishop, B. Kowalczyk, C. E. Wilmer, *Nature chemistry* **2009**, *1*, 31.
- [459] C. M. Gothard, S. Soh, N. A. Gothard, B. Kowalczyk, Y. Wei, B. Baytekin, B. A. Grzybowski, *Angew. Chem.* **2012**, *124*, 8046.
- [460] A. Vande Wouwer, P. Saucez, C. Vilas, *Simulation of ODE/PDE models with MATLAB®, OCTAVE and SCILAB. Scientific and engineering applications*, Springer, Cham, **2014**.
- [461] a) S. Lynch, *Dynamical systems with applications using Mathematica*, Birkhäuser Springer, Cham, **2017**; b) J. E. House, *Principles of Chemical Kinetics*, 2. Aufl., Elsevier professional, s.l., **2007**; c) D. U. L.K. Doraiswamy, *Chemical Reaction Engineering*, CRC Press, **2013**.
- [462] C.-M. Ionescu, D. Sehnal, F. L. Falginella, P. Pant, L. Pravda, T. Bouchal, R. Svobodová Vařeková, S. Geidl, J. Koča, *J Cheminf* **2015**, *7*, 50.
- [463] K. N. Kirschner, A. B. Yongye, S. M. Tschampel, J. González-Outeiriño, C. R. Daniels, B. L. Foley, R. J. Woods, *Journal of Computational Chemistry* **2008**, *29*, 622.
- [464] N. Kochev, S. Avramova, N. Jeliaskova, *J Cheminf* **2018**, *10*, 42.
- [465] R. Jing, J. Sun, Y. Wang, M. Li, X. Pu, *Chemometrics and Intelligent Laboratory Systems* **2014**, *138*, 1.
- [466] a) D. Howe, M. Costanzo, P. Fey, T. Gojobori, L. Hannick, W. Hide, D. P. Hill, R. Kania, M. Schaeffer, S. St Pierre et al., *Nature* **2008**, *455*, 47 EP -; b) V. Marx, *Nature* **2013**, *498*, 255 EP -; c) B. C. Gibb, *Nature chemistry* **2013**, *5*, 248; d) N. Jones, *Nature News* **2014**, *505*, 146.
- [467] Y. LeCun, Y. Bengio, G. Hinton, *Nature* **2015**, *521*, 436.
- [468] M. I. Jordan, T. M. Mitchell, *Science (New York, N.Y.)* **2015**, *349*, 255.
- [469] G. Hinton, L. Deng, D. Yu, G. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath et al., *IEEE Signal Process. Mag.* **2012**, *29*, 82.
- [470] L. Sajn, M. Kukar, *Computer methods and programs in biomedicine* **2011**, *104*, e75-86.

- [471] D. B. Kell, *The FEBS journal* **2006**, 273, 873.
- [472] D. C. Parkes, M. P. Wellman, *Science (New York, N.Y.)* **2015**, 349, 267.
- [473] W. A. Warr, *Molecular informatics* **2014**, 33, 469.
- [474] P. Raccuglia, K. C. Elbert, P. D. F. Adler, C. Falk, M. B. Wenny, A. Mollo, M. Zeller, S. A. Friedler, J. Schrier, A. J. Norquist, *Nature* **2016**, 533, 73.
- [475] S. Szymkuć, E. P. Gajewska, T. Klucznik, K. Molga, P. Dittwald, M. Startek, M. Bajczyk, B. A. Grzybowski, *Angew Chem Int Ed Engl* **2016**, 55, 5904.
- [476] G. Skoraczyński, P. Dittwald, B. Miasojedow, S. Szymkuć, E. P. Gajewska, B. A. Grzybowski, A. Gambin, *Scientific Reports*, 7, 3582.
- [477] S. Chatterjee, S. Moon, F. Hentschel, K. Gilmore, P. H. Seeberger, *J Am Chem Soc* **2018**, 140, 11942.
- [478] T. Reitmaier, *Aktives Lernen für Klassifikationsprobleme unter der Nutzung von Strukturinformationen*, Kassel University Press, **2015**.
- [479] a) B. Nisius, F. Sha, H. Gohlke, *Journal of Biotechnology* **2012**, 159, 123; b) H. Ding, I. Takigawa, H. Mamitsuka, S. Zhu, *Briefings in Bioinformatics* **2014**, 15, 734; c) G. Huang, F. Yan, D. Tan, *Current protein & peptide science* **2018**, 19, 562; d) A. Ezzat, M. Wu, X.-L. Li, C.-K. Kwoh, *Briefings in Bioinformatics* **2018**.
- [480] Z. Mousavian, A. Masoudi-Nejad, *Expert opinion on drug metabolism & toxicology* **2014**, 10, 1273.
- [481] a) T. van Laarhoven, E. Marchiori, *PLoS one* **2013**, 8, e66952; b) J.-P. Mei, C.-K. Kwoh, P. Yang, X.-L. Li, J. Zheng, *Bioinformatics (Oxford, England)* **2013**, 29, 238; c) Y. Liu, M. Wu, C. Miao, P. Zhao, X.-L. Li, *PLoS computational biology* **2016**, 12, e1004760; d) T. Ban, M. Ohue, Y. Akiyama, *Biochemistry and biophysics reports* **2019**, 18, 100615.
- [482] a) K. Bleakley, Y. Yamanishi, *Bioinformatics* **2009**, 25, 2397; b) K. Buza, L. Peška, *Neurocomputing* **2017**, 260, 284.
- [483] H. Chen, T. Kogej, O. Engkvist, *Molecular informatics* **2018**, 37, e1800041.
- [484] A. Ezzat, P. Zhao, M. Wu, X.-L. Li, C.-K. Kwoh, *IEEE/ACM transactions on computational biology and bioinformatics* **2017**, 14, 646.
- [485] G. James, D. Witten, T. Hastie, R. Tibshirani, *An introduction to statistical learning. With applications in R*, 8. Aufl., Springer, New York, Heidelberg, Dordrecht, London, **2017**.
- [486] C. Cortes, V. Vapnik, *Machine Learning* **1995**, 20, 273.
- [487] a) J. Gasteiger, J. Zupan, *Angew. Chem. Int. Ed. Engl.* **1993**, 32, 503; b) S. S. Haykin, *Neural networks. A comprehensive foundation*, Macmillan, New York, NY, **1995**.
- [488] P. Geurts, D. Ernst, L. Wehenkel, *Machine Learning* **2006**, 63, 3.
- [489] L. Breiman, *Machine Learning* **2001**, 45, 5.

- [490] a) T. Hastie, R. Tibshirani, J. H. Friedman, *The elements of statistical learning. Data mining, inference, and prediction*, 12. Aufl., Springer, New York, NY, **2017**; b) V. Svetnik, A. Liaw, C. Tong, J. C. Culberson, R. P. Sheridan, B. P. Feuston, *J. Chem. Inf. Comput. Sci.* **2003**, *43*, 1947.
- [491] N. Schneider, N. Fechner, G. A. Landrum, N. Stiefl, *Journal of chemical information and modeling* **2017**, *57*, 1816.
- [492] M. R. Berthold, N. Cebron, F. Dill, T. R. Gabriel, T. Kötter, T. Meinl, P. Ohl, K. Thiel, B. Wiswedel, *SIGKDD Explor. Newsl.* **2009**, *11*, 26.
- [493] S. Roughley, *Current Medicinal Chemistry* **2018**.
- [494] P. Tosco, T. Balle, *Journal of Molecular Modeling* **2011**, *17*, 201.
- [495] P. Tosco, T. Balle, F. Shiri, *Journal of Computer-Aided Molecular Design* **2011**, *25*, 777.
- [496] P. Tosco, N. Stiefl, G. Landrum, *J Cheminf* **2014**, *6*, 201.
- [497] K. AlJadda, R. Ranzinger, M. Porterfield, B. Weatherly, M. Korayem, J. A. Miller, K. Rasheed, K. J. Kochut, W. S. York, *GELATO and SAGE: An Integrated Framework for MS Annotation*, **2016**, zu finden unter <https://arxiv.org/pdf/1512.08451>.
- [498] S.-W. Wu, S.-Y. Liang, T.-H. Pu, F.-Y. Chang, K.-H. Khoo, *Journal of proteomics* **2013**, *84*, 1.
- [499] S.-Y. Liang, S.-W. Wu, T.-H. Pu, F.-Y. Chang, K.-H. Khoo, *Bioinformatics (Oxford, England)* **2014**, *30*, 1908.
- [500] a) J.-P. Vert, L. Jacob, *CCHTS* **2008**, *11*, 677; b) S. Agarwal, D. Dugar, S. Sengupta, *Journal of chemical information and modeling* **2010**, *50*, 716; c) F. Buchwald, L. Richter, S. Kramer, *Journal of cheminformatics* **2011**, *3*, 22; d) O. Engkvist, P.-O. Norrby, N. Selmi, Y.-h. Lam, Z. Peng, E. C. Sherer, W. Amberg, T. Erhard, L. A. Smyth, *Drug Discovery Today* **2018**, *23*, 1203.
- [501] R. Sawada, M. Kotera, Y. Yamanishi, *Molecular informatics* **2014**, *33*, 719.
- [502] a) N. Brown, *ACM Comput. Surv.* **2009**, *41*, 1; b) J. Gasteiger, *Analytical and Bioanalytical Chemistry* **2006**, *384*, 57.
- [503] C. Fotis, A. Antoranz, D. Hatzivramidis, T. Sakellaropoulos, L. G. Alexopoulos, *Drug Discovery Today* **2018**, *23*, 626.
- [504] a) S. V. Bennun, D. B. Hizal, K. Heffner, O. Can, H. Zhang, M. J. Betenbaugh, *Journal of molecular biology* **2016**, *428*, 3337; b) L.-X. Wang, J. V. Lomino, *ACS chemical biology* **2012**, *7*, 110; c) S. Jo, Y. Qi, W. Im, *Glycobiology* **2016**, *26*, 19.
- [505] J. Ratnam, B. Zdrzil, D. Digles, E. Cuadrado-Rodriguez, J.-M. Neefs, H. Tipney, R. Siebes, A. Waagmeester, G. Bradley, C. H. Chau et al., *PLoS one* **2014**, *9*, e115460.
- [506] R. McGuire, S. Verhoeven, M. Vass, G. Vriend, I. J. P. de Esch, S. J. Lusher, R. Leurs, L. Ridder, A. J. Kooistra, T. Ritschel et al., *Journal of chemical information and modeling* **2017**, *57*, 115.
- [507] a) A. Moll, A. Hildebrandt, H.-P. Lenhof, O. Kohlbacher, *Journal of Computer-Aided Molecular Design* **2005**, *19*, 791; b) A. Moll, A. Hildebrandt, H.-P. Lenhof, O. Kohlbacher, *Bioinformatics* **2006**, *22*, 365; c) A. Hildebrandt, A. K. Dehof, A. Rurainski, A. Bertsch, M. Schumann, N. C. Toussaint, A. Moll, D. Stöckel, S. Nickels, S. C. Mueller et al., *BMC Bioinformatics* **2010**, *11*, 531.

- [508] A. D. Bochevarov, E. Harder, T. F. Hughes, J. R. Greenwood, D. A. Braden, D. M. Philipp, D. Rinaldo, M. D. Halls, J. Zhang, R. A. Friesner, *Int. J. Quantum Chem.* **2013**, *113*, 2110.
- [509] M. F. Guest *, I. J. Bush, H. J. J. van Dam, P. Sherwood, J. M. H. Thomas, J. H. van Lenthe, R. W. A. Havenith, J. Kendrick, *Molecular Physics* **2005**, *103*, 719.
- [510] B. N. Papas, H. F. Schaefer, *Journal of Molecular Structure: THEOCHEM* **2006**, *768*, 175.
- [511] Y. Alexeev, M. P. Mazanetz, O. Ichihara, D. G. Fedorov, *CTMC* **2012**, *12*, 2013.
- [512] J. J. P. Stewart, *Journal of Computer-Aided Molecular Design* **1990**, *4*, 1.
- [513] M. Roice, I. Johannsen, M. Meldal, *QSAR Comb. Sci.* **2004**, *23*, 662.
- [514] R. Perkins, H. Fang, W. Tong, W. J. Welsh, *Environ Toxicol Chem* **2003**, *22*, 1666.
- [515] B. R. Brooks, C. L. Brooks, A. D. Mackerell, L. Nilsson, R. J. Petrella, B. Roux, Y. Won, G. Archontis, C. Bartels, S. Boresch et al., *Journal of Computational Chemistry* **2009**, *30*, 1545.
- [516] S. S. Mallajosyula, O. Guvench, E. Hatcher, A. D. MacKerell, *Journal of chemical theory and computation* **2012**, *8*, 759.
- [517] S. E. Hamby, J. D. Hirst, *BMC Bioinformatics* **2008**, *9*, 500.
- [518] D. Pavlov, M. Rybalkin, B. Karulin, M. Kozhevnikov, A. Savelyev, A. Churinov, *J Cheminf* **2011**, *3*.
- [519] N. Benteinitis, *J. Chem. Educ.* **2008**, *85*, 1146.
- [520] B. R. Brooks, R. E. Bruccoleri, B. D. Olafson, D. J. States, S. Swaminathan, M. Karplus, *Journal of Computational Chemistry* **1983**, *4*, 187.
- [521] M. J. Frisch, G. W. Trucks, H. B. Schlegel, G. E. Scuseria, M. A. Robb, J. R. Cheeseman, G. Scalmani, V. Barone, G. A. Petersson, H. Nakatsuji et al., *Gaussian~16 Revision B.01*, **2016**.
- [522] W. A. de Jong, A. M. Walker, M. D. Hanwell, *J Cheminf* **2013**, *5*, 25.
- [523] K. D. Collins, T. Gensch, F. Glorius, *Nature chemistry* **2014**, *6*, 859.
- [524] N. Graulich, H. Hopf, P. R. Schreiner, *Chem. Soc. Rev.* **2010**, *39*, 1503.
- [525] P.-M. Jacob, P. Yamin, C. Perez-Storey, M. Hopgood, A. A. Lapkin, *Green Chem.* **2017**, *19*, 140.
- [526] M. Feinberg, *Foundations of Chemical Reaction Network Theory*, Springer International Publishing, Cham, **2019**.
- [527] I. Martínez-Forero, A. Peláez-López, P. Villoslada, *PloS one* **2010**, *5*, e10823.
- [528] A. L. Nagy, d. Papp, J. Tóth, *Chemical Engineering Science* **2012**, *83*, 12.
- [529] W. R. Smith, R. W. Missen, *J. Chem. Educ.* **1997**, *74*, 1369.
- [530] M. M. C. Ferreira, W. C. Ferreira, A. C. S. Lino, M. E. G. Porto, *J. Chem. Educ.* **1999**, *76*, 861.
- [531] R. E. Plata, D. A. Singleton, *J Am Chem Soc* **2015**, *137*, 3811.

- [532] C. W. Coley, R. Barzilay, T. S. Jaakkola, W. H. Green, K. F. Jensen, *ACS central science* **2017**, *3*, 434.
- [533] F. S. Emami, A. Vahid, E. K. Wylie, S. Szymkuć, P. Dittwald, K. Molga, B. A. Grzybowski, *Angew Chem Int Ed Engl* **2015**, *54*, 10797.
- [534] B. A. Grzybowski, S. Szymkuć, E. P. Gajewska, K. Molga, P. Dittwald, A. Wołos, T. Klucznik, *Chem* **2018**, *4*, 390.
- [535] S. Szymku, E. P. Gajewska, T. Klucznik, K. Molga, P. Dittwald, M. Startek, M. Bajczyk, B. A. Grzybowski, *Angewandte Chemie* **2016**, *128*, 6004.
- [536] T. Cernak, *Chem* **2018**, *4*, 401.
- [537] J. M. Granda, L. Donina, V. Dragone, D.-L. Long, L. Cronin, *Nature* **2018**, *559*, 377.

ABBILDUNGSVERZEICHNIS

Abbildung 1: Chitin ist als eines der weitest verbreiteten Glykane für den strukturellen Aufbau von Pflanzen und Tieren verantwortlich, z. B. im Exoskelett der Riesenkrabbe.....	1
Abbildung 2: Schematische Darstellung verschiedener Zell-Interaktionen mit Beispielen von Zuckermolekülen ^[2,3,4a,5,4b,6] , die dabei eine Rolle spielen.....	1
Abbildung 3: Vergleich der wichtigsten Biomoleküle und deren Ankopplungsstellen bei der Synthese von Biopolymeren. Bei den Zuckermolekülen erhöhen bereits die fünf möglichen Kopplungsstellen die Komplexität bei der Synthese, aber zusätzlich auch deren sterische Ausrichtung.....	2
Abbildung 4: Peptide und Oligonukleotide (DNA/RNA) lassen sich aus einer begrenzten Anzahl von Grundbausteinen linear zusammensetzen. Zuckermoleküle werden aus einer Vielzahl von Bausteinen zusammengesetzt. Dafür existieren unterschiedliche Beschreibungs-codes. Die Symbolcodierung Cartoon -Symbolik ermöglicht die anschauliche Darstellung komplexer Zuckermoleküle.....	4
Abbildung 5: Symbole der wichtigsten Monomere mit biologischer Bedeutung ^[18]	5
Abbildung 6: Prinzip der Glykosid-Synthese, bei dem der Zucker in zwei verschiedenen Konfigurationen, als α - und β -Glykosid, entstehen kann.....	7
Abbildung 7: Der anomere Effekt: a) $n-\sigma^*$ - Hyperkonjugation, b) Dipol-Dipol Wechselwirkung, c) Verhältnis der Konfigurationen in Abhängigkeit vom Substituenten am Beispiel D-Glucose	8
Abbildung 8: Im antiperiplanaren β -Anomer findet eine größere elektrostatische Abstoßung zwischen den Ionenpaaren des endocyclischen Sauerstoffs und dem elektronegativen anomeren Substituenten statt, als in der synclinalen Konformation des α -Anomers.	9
Abbildung 9: Beispiele für 1,2- <i>trans</i> und 1,2- <i>cis</i> -Glykoside	10
Abbildung 10: Einfluss des Nachbargruppeneffektes auf die Glykosylierung eines 1,2- <i>trans</i> -Glykosids.	10
Abbildung 11: Prinzip des Festphasensystems.....	14
Abbildung 12: Schematischer Ablauf einer automatisierten Zuckersynthese	15
Abbildung 13: Struktur von Merrifield Harz und TentaGel TM	18
Abbildung 14: Herausforderungen einer Oligosaccharid-Synthese	23
Abbildung 15: a) Donor-basierte Synthesestrategie b) Akzeptor-basierte Synthesestrategie	24
Abbildung 16: Nachbau des ersten automatischen Peptidsynthesizers und schematische Zeichnung des automatischen Syntheseapparates für Peptide aus der Patentschrift von Bruce Merrifield ^[131] ...	26
Abbildung 17: Erste automatische Festphasen-Synthese eines Hexasaccharids	27
Abbildung 18: Automatische Synthese eine Globo-H Hexasaccharids mit 30% Ausbeute.....	28
Abbildung 19: Automatische Synthese einer N-Glykan Zielstruktur unter Verwendung des ersten vollautomatischen Oligosaccharid-Synthesizers	31

Abbildung 20: Automatische Festphasen-Synthese eines linearen 30-mer α -(1,6)- Oligomannosids unter Anwendung der Catch–Release Reinigung	32
Abbildung 21: HPLC-assistierte automatische Oligosaccharid-Synthese.....	34
Abbildung 22: Fluor-Tag unterstützte Synthese im Mikroreaktor.....	36
Abbildung 23: Automatische Synthese von linearen Mannose-Oligosacchariden unter Verwendung von Fluor-Tag Unterstützung und FSPE Filtration	37
Abbildung 24: Aufbau des J-KEM – Systems. Die Flüssigkeitszufuhr erfolgt über Spritzen.	53
Abbildung 25: J-KEM System Oligosaccharid-Syntheseautomat. ^[160]	54
Abbildung 26: Gerätesteuerung des J-KEM Oligosaccharid-Syntheseautomaten mit dem Controller als zentralem Element und steuerungsbezogener Abarbeitung	55
Abbildung 27: J-KEM Benutzeroberfläche.....	56
Abbildung 28: Beispiel eines J-KEM Programmtextes mit farblicher Codierung der Befehle, Kurzbeschreibung zur besseren Lesbarkeit und Vergleichbarkeit.....	57
Abbildung 29: Rückseite der J-KEM Controllereinheit mit aufgestecktem Schnittstellenadapter zur Protokollierung der gerätespezifische Befehls- und Ansteuerungsprozeduren.....	58
Abbildung 30: Selbst entwickelte Adapterstecker zum Protokollieren der RS232 Kommunikation	58
Abbildung 31: Adapterkabel (o. li.) zum Informationsabgriff der Ventilansteuerung (o. re.) über selbst konzipierte und aufgebaute Optokopplerschaltung/Adapterboard (u.)	60
Abbildung 32: Konzeption und Umsetzung der Geräteweiterentwicklung erfolgt im Bottom-up Prozess. Die für die Geräteentwicklung notwendigen Projektschritte sind als V-Modell dargestellt...61	61
Abbildung 33: Dezentrale Gerätesteuerung	63
Abbildung 34: Frontpanel eines LabVIEW Virtual Instruments (VI) mit zwei numerischen Bedienelementen und einem numerischen Anzeigeelement, die sich aus einer umfangreichen Elemente-Palette per Drag & Drop auf der Programmfläche positionieren lassen.	65
Abbildung 35: Darstellungsarten im Blockdiagramm. Datentypen sind über Rahmenfarbe und Symbol gekennzeichnet.	65
Abbildung 36: Programmcode einer einfachen Addition. Die Funktion wird per Drag & Drop im Blockdiagramm positioniert. Nach dem Datenflussprinzip werden die Eingabelemente mit den Eingängen des Additionsobjekts verbunden und das Ergebnis durch Verbinden mit der Kabeltrommel (Werkzeugpalette) ans Anzeigeelement übergeben.	66
Abbildung 37: Nach Eingabe der Zahlen auf der Benutzeroberfläche kann der Programmcode zur Ausführung gebracht werden. Im Blockdiagramm lässt sich mit der Highlight-Funktion parallel dazu der Datenfluss veranschaulichen.....	67
Abbildung 38: Dem VI wird ein Muster für den Anschlussterminal zugewiesen. Aufgrund des Anschlussterminal ist für dieses VI eine Beschreibung in der Kontexthilfe entstanden.	67

Abbildung 39: Das Beispiel VI wird in einem neuen Blockdiagramm mit beliebig wählbaren Eingabe- und Anzeigeelementen.	68
Abbildung 40: Eingabe- und Ausgabeelemente können individuell verändert, ausgetauscht oder durch individuelle Anzeigeobjekte ersetzt werden. Der Programmiercode bleibt dabei unverändert.	68
Abbildung 41: Für die Gestaltung der VI Icons steht der Symbol-Editor zur Verfügung. Aussagekräftige VI – Icons erhöhen die Lesbarkeit komplexer Programmstrukturen.	69
Abbildung 42: LabVIEW verfügt über umfangreiche Dokumentationsmöglichkeiten, die sowohl lokal in der Kontexthilfe als auch global als Online – Hilfe Bestandteil der Entwicklungsumgebung werden und damit Anwender und Entwickler unterstützen.	69
Abbildung 43: Mit demselben Programmcode im Blockdiagramm lassen sich auch Arrays addieren, wenn diese als Ein- und Ausgabeelemente definiert sind.	70
Abbildung 44: Das Beispiel-VI kann nicht mit dem Zahlenarray als Eingabeelement verbunden werden, da diese Operation nicht im VI definiert ist, intern wird jedoch dieselbe Programmstruktur genutzt.	70
Abbildung 45: Für eine Textaddition wird eine neue Funktion genutzt, die mit entsprechend des Datentyps korrespondierender Text-Ein- und Anzeigeelemente verkabelt sind.	71
Abbildung 46: Im polymorphen VI sind mehrere Funktionalitätsinstanzen ineinander vereint. Das VI erkennt den Datentyp der angeschlossenen Werte und wechselt entsprechend zu der Instanz, die damit rechnen kann.	71
Abbildung 47: Das Fehlercluster definiert das Verhalten des VIs im Fehlerfall.	72
Abbildung 48: Das Beispiel VI wird um einen Fehlercluster mit CASE – Struktur erweitert. Die Programmausführung findet nur statt, wenn kein Fehler am Eingang vorliegt.	72
Abbildung 49: Erweiterung des Beispiel-VIs um weitere Mathematikfunktionen über ein ENUM-Element.	73
Abbildung 50: Verknüpfung des ENUM Elements mit dem VI stellt die erweiterte Funktionalität allen Instanzen im VI bereit.	73
Abbildung 51: Gerätetreiber werden als Black Box Elemente umgesetzt. Der Anwender muss sich nur mit der Eingabe der Eingangsgrößen beschäftigen, wobei Kenntnisse über die innere Funktionalität des Black Box Systems nicht nötig sind, um die angestrebte Funktionalität zu nutzen.	74
Abbildung 52: Innerhalb des Gerätetreibers sorgt ein komplexes Fehlermanagementsystem für definierte Systemzustände.	75
Abbildung 53: Das Fehlermanagement ist in Form einer Zustandsmaschine realisiert.	76
Abbildung 54: Befehle werden von der übergeordneten Applikationsebene in eine Warteschlange geschrieben, die linear abgearbeitet wird. Priorisierte Anforderungen können an die aktuelle Stelle der Warteschlange rücken (z. B. Not-Aus).	77
Abbildung 55: Allgemeine Anforderungen an die Gerätetreiber VIs und deren interne Umsetzung bzw. Ausgabefunktionalität.	77

Abbildung 56: Geräteübergreifende Funktionskontrolle der einzelnen Systemkomponenten durch Statusmeldungen der Gerätetreiber als eigenständige Zustandsmaschine.	78
Abbildung 57: Die funktionale Grundstruktur der gesamten Synthesizer-Systems orientiert sich am ISO 7 Schichtenmodell.	79
Abbildung 58: Sensorische (mechanische und optische) Überprüfungsmöglichkeiten stellen fest, ob eine Systemanforderung wie z. B. ein Lösungsmitteltransport in das Reaktionsgefäß ordnungsgemäß erfolgt ist. In Abhängigkeit ob die Anforderung erfolgreich umgesetzt wurde oder ein Fehler erkannt ist, greifen Fehlervermeidungsstrategien zur Syntheseoptimierung.	80
Abbildung 59: a) Beispiel für die konzeptionelle Umsetzung eines allgemeinen Gerätetreiber VIs als Black-Box-Elemente mit interner Zustandsmaschine (Frontpanel) b) Kontexthilfe zu a) zeigt die erstellten Anschlusselemente, die für das allgemeine, konzeptionelle Gerätetreiber-VI angelegt sind.	81
Abbildung 60: Jedes VI in der Programmoberfläche besteht aus einem visuell aussagekräftigem Programm-ICON und einem funktionalen Anschlussterminal.	83
Abbildung 61: LabVIEW Programmbeispiel: Blockdiagramm einer Zustandsmaschine aus sechs statisch aufeinanderfolgenden Schrittkettenelementen.....	83
Abbildung 62: LabVIEW Programmbeispiel: Blockdiagramm einer Zustandsmaschine mit Zugriff über Schieberegister.....	84
Abbildung 63: Für die Kloehn – Pumpe erstellte VIs, die aus verschiedenen SubVIs mit komplexen Schrittkettenanweisungen bestehen.....	85
Abbildung 64: Geräteerweiterungen sind beim dezentralen Steuersystem relativ leicht in das Gesamtsystem zu integrieren. Neben dem Hardwareanschluss muss der neue Gerätetreiber zum Betrieb in der Synthesizer-Applikation eingebunden werden.	85
Abbildung 65: Hardwaresteuerung und die Bediensoftware ergeben zusammen die Gesamtapplikation. Die Hardwaresteuerung ist ein Black Box System, an das die Bediensoftware Anforderungen stellt, die sich aus der Synthesevorschrift ergeben. Die Abstraktionstiefe ist von der Stufe der Sprachentwicklung, die innerhalb der Bediensoftware verwendet wird, abhängig.	86
Abbildung 66: Abstraktionsebenen zur Entwicklung eines „universellen“ Syntheseautomaten. Mit jeder höheren Abstraktionsebene wächst die Komplexität der zur Verfügung stehenden Funktionalitäten. Zugleich nimmt die Komplexität in der Bedienung für den Anwender in jedem Abstraktionslevel ab.	87
Abbildung 67: Schematischer Aufbau des neuen Arbeitssynthesizers mit dezentraler Steuerung durch den Automatisierungsrechner und Standardsystemkomponenten.	90
Abbildung 68: Einbaupositionen der wichtigsten Synthesizer-Komponenten am Neubau des Arbeitssynthesizer III.....	91
Abbildung 69: Gesamtrahmenaufbau vom Arbeitssynthesizer III mit lokaler mobiler integrierter Werkstatt	92
Abbildung 70: Flüssigkeitstransport von Lösungsmitteln, Reagenzien und Zuckerbausteinen im Synthesizer.	93

Abbildung 71: Farbcodierung zur vereinfachten Zuordnung der Flüssigkeitsführung an den Vorratsflaschen der Aktivatoren und den Entschützungsreagenzien	94
Abbildung 72: Pufferloops der spritzenbasierten Flüssigkeitszufuhr und die dazu notwendigen Kloehn-Pumpen	94
Abbildung 73: Ventilkonzept bei druckbasierter Lösungsmittel-Zuführung (Spulenventil) mit Zentralverschluss (Ventil 16) des Reaktors	94
Abbildung 74: Flüssigkeitsführung für Bausteine	95
Abbildung 75: Schematischer Aufbau der Ventilsteuerung zum Lösungsmitteltransport. Die Schaltung der Spulenventile erfolgt über den Ventil-Gerätetreiber mit Hilfe einer PC-Karte, sowie einer sich anschließenden Verstärkerbox (DIO Box).	95
Abbildung 76: Die lösungsmittelresistenten, vollständig mit Teflon ummantelten Spulenventile ermöglichen die Steuerung des schnellen Flüssigkeitstransports im Synthesizer	96
Abbildung 77: DIO Box Prototyp für die Arbeitssynthesizer I + II mit Spezialsteckeranschluss zur Ventilsteuerung. Auf der Geräterückseite finden sich die Anschlussbuchsen für das Schnittstellenkabel zum PC, Spannungsversorgung, sowie Status-LEDs zur lokalen Kontrolle der aktiven Ventilschaltung.....	96
Abbildung 78: DIO Box für den Arbeitssynthesizer III. Anschlüsselemente für Spulenventile auf der Vorderseite sind durch Standardniederspannungsstecker ersetzt und mit Status-LEDs und Hauptschalter komplettiert. An der Rückseite finden sich die PC – Anschlüsse, Spannungsversorgung(5V, 24V), sowie und Netz-Sicherung.	96
Abbildung 79: DIO Box für Arbeitssynthesizer III als modulare Komponente für den 19“ Einbaurahmen in der Draufsicht.....	97
Abbildung 80: Funktionskontrolle an einem „Vierer“- Spulenventil.	97
Abbildung 81: Die Ventilsteuerung erfolgt mittels Gerätetreiber über eine Digital I/O Karte. Die Karte kann in LabVIEW direkt über die Treibersoftware angesprochen werden. Das Signal am Datenausgang der Karte muss für die Ventile auf 24 V verstärkt werden und spricht damit die Aktorik an. Die PC-Karte ist ebenfalls in der Lage parallel Messdaten weiterer Sensorik aufzunehmen.....	98
Abbildung 82: In der Gerätetreiberentwicklung für die Spulenventile wird die unterste Abstraktionsebene der Sprachentwicklung für den Synthesizer administriert: binärer, hardwarekonformer Steuercode.....	98
Abbildung 83: Frontpanel mit 16 LEDs, die über ein Array verwaltet werden. Jede LED kann direkt ausgewählt und auf 0 oder 1 bzw. „An“ oder „Aus“ gesetzt werden.	99
Abbildung 84: Blockdiagramm zur ereignisbasierten Anzeige der Ventilzustände mittels DAQ Assistent. Die Zuordnung der einzelnen LED-Schaltzustände erfolgt durch Addition der Binärwerte des Gesamtsystems. Dieser Wert wird an den DAQ Assistenten übergeben, der die aktuellen Werte des Bedien-LED-Array herausschreibt.	100
Abbildung 85: Auf digitale Ausgänge wird reihenweise (immer 8 Kanäle gleichzeitig) zugegriffen. Im Frontpanel links lässt sich nur die gesamte Reihe (lines) „An“ und „Aus“ schalten. Am Frontpanel rechts können nach Selektion der Reihe auch die Einzelkanäle geschaltet werden.	102

Abbildung 86: Aus dem Blockdiagramm lässt sich die funktionale Schrittkette beim Hardwarezugriff ableiten. Die Programmierung hinter den beiden Benutzeroberflächen aus Abbildung 85 ist dieselbe.102

Abbildung 87: Schrittkette bei einem Hardwarezugriff103

Abbildung 88: In LabVIEW programmierte Benutzeroberflächen lassen sich mit Ein- und Ausgabelementen ohne Veränderung des funktionalen Programmiercodes beliebig anpassen. Für unterschiedliche Datentypen lassen sich verschiedene Verarbeitungsroutinen am Element hinterlegen (stapeln).....103

Abbildung 89: Einfacher Zugriff auf die digitalen Kanäle einer Reihe. Jede Reihe besitzt acht Kanäle, die einzeln mit dem Schalter „An“ oder „Aus“ geschaltet werden können.....104

Abbildung 90: Blockdiagramm für den einfachen Hardwarezugriff auf die digitalen Kanäle der Datenerfassungskarte.104

Abbildung 91: Für die aktuellen Gerätetreiber wird in der Software ein Interpretations-Mechanismus etabliert, der die Synthese-Ansteuerung der Vorgänger-Geräte weiter nutzbar macht. Diese Sprachabstraktion findet bereits auf einer höheren Abstraktions- bzw. Interpretationsebene statt.105

Abbildung 92: Realisierung eines einfachen Befehlsinterpreters. Der vorliegende Befehl wird in Kanalnummer und Status geteilt. Der Textstring „Status“ wird in einen Binärwert umgewandelt und als LED angezeigt.105

Abbildung 93: Befehlsinterpreter mit Befehlsvorgabe und Abarbeitung durch ein Array. Die Zustandsänderung für die einzelnen Ventile erfolgt sich über eine Schleifensteuerung.106

Abbildung 94: Der Befehlsinterpreter ist ein Unterprogramm, das die Kompatibilität mit den Syntheskripten des Altsystems gewährleistet. Die Skripte werden in die Anweisungsliste kopiert und durch Interpretation der übergebenen Texte in funktionalen Code für das neue System umgewandelt.....107

Abbildung 95: Frontpanel für ein Unterprogramm des entsprechenden Gerätetreibers mit dem die Spulventile im Synthesizer auf ihre Funktion getestet werden können.108

Abbildung 96: Blockdiagramm: Spülprogramm testen (Schleifensteuerung) mit Geräteinitialisierung, Gerätezuordnung, Befehlsinterpretation der Synthese-Skripte des J-KEM-Systems, Schaltung der Ventile gemäß des angegebenen Zustandes und abschließender Hardwarefreigabe nach Funktionsabarbeitung.....109

Abbildung 97: Zufuhr von 3 * 3mL DCM zur Überprüfung der geförderten Volumina als Kalibrierung vor der Synthese, um die allgemeine Funktionsbereitschaft zu gewährleisten.....109

Abbildung 98: Die Ansteuerung der Spritzenpumpe findet über ein serielles Bussystem statt. Damit lassen sich neben der Spritzenpumpe noch bis zu 15 weitere Geräte steuern. Alternativ kann für die Gerätesteuerung auch ein CAN-Bus, das Industriestandard-Bussystem der Automobiltechnik, Verwendung finden. Hier existieren keinerlei reale Einschränkungen hinsichtlich der Geräteanzahl.110

Abbildung 99: Spritzenpumpe (li.) und Rotationsventile, Steuerungseinheit der Firma Kloehn (re.) .111

Abbildung 100: Innerhalb von LabVIEW können direkt Befehle an die angeschlossene Spritzenpumpe / Rotationsventil über das Kommunikationsprogramm (MAX) an die serielle Schnittstelle abgeschickt

werden. Zusätzlich zum Befehlsstring, müssen hier Steuerzeichen angegeben werden. Die Rückmeldung des Gerätes wird direkt angezeigt.	111
Abbildung 101: Abstraktionsebene der Software bei der Gerätetreiber-Entwicklung der Spritzenpumpen und Rotationsventile. Die geräteinterne Software ermöglicht eine Direktkommunikation der Komponenten untereinander.....	112
Abbildung 102: Einbinden des Konsolenprogramms Kserial im LabVIEW-Testprogramm enthält alle Elemente zur vollständigen Administration der Programmierschnittstelle.....	113
Abbildung 103: Übersicht der einzelnen VI's zur Kommunikation mit den Kloehn - Geräten.....	114
Abbildung 104: Test der Spritzenpumpen mit der Gerätetreibersoftware und Testprogrammen zur Befüllung und Entleerung.....	115
Abbildung 105: Drücke am Eingang des Druckverteilers. Aufgrund unterschiedlicher Viskositäten bei den Lösungsmitteln und Reagenzien sind die benötigten Drücke individuell anzupassen, um gleiche Volumina über die Ventilsteuerung zu transportieren.	116
Abbildung 106: Technische Zeichnung des Manifolds der Fa. Swagelok, der an Arbeitssynthesizer II eingebaut wurde.....	116
Abbildung 107: Kalibrierung der Drücke und Hysteresemessung	117
Abbildung 108: Druckmessungen am Druckverteiler für die unterschiedlichen Lösungsmittel und Reagenzien. Das „Wegdriften“ eines Signals zeigt an, dass ein Ventil nicht vollständig geschlossen wurde. Die Druckkontrolle lässt sich auf eine Übersichts-Oberfläche vereinfachen.	117
Abbildung 109: Der Fraktionssammler wird über eine rudimentäre RS232 Schnittstelle angesteuert. Die Probenentnahme aus dem Reaktionsgefäß erfolgt über die Spulenventile in den Fraktionssammler. Somit sind für die Bestückung des Fraktionssammlers Anforderungen an die Gerätetreiber der Ventile zum Flüssigkeitstransport, sowie des Fraktionssammlers zur Positionierung der Probe notwendig.	118
Abbildung 110: Im Fraktionssammler-VI können zu Testzwecken auch Einzelkommandos abgeschickt werden, die in der lokalen Anzeige des Zielsystems kontrollierbar sind. Die Positionierungsauflösung der drei vorhandenen Fraktionssammler für die Arbeitssynthesizer ist jeweils unterschiedlich und wird über gerätebezogene *.ini – Dateien eingelesen.	119
Abbildung 111: Die vorhandenen Fraktionssammler ist vollständig vom neuen Synthesizer-Programm ansteuerbar.	119
Abbildung 112: Im Fraktionssammler-VI lassen sich unterschiedliche Rack-Strukturen über den Abstand der jeweiligen Auffangbehältermittelpunkte in x- und y-Richtung, sowie reihen- und spaltenbezogene Anzahl der Behältnisse definieren.....	120
Abbildung 113: Im Testprogramm wurden, die mittels Befehlsinterpreter übersetzten Befehle des Altsystems ausgeführt. Hier erfolgt nach der Geräteinitialisierung mit dem Befehl <i>Advanced</i> eine Direktdefinition zur nächsten gültigen Sammelstellenposition.	120
Abbildung 114: Reaktionsauskopplungen während der Synthese	121

Abbildungsverzeichnis

Abbildung 115: Die Temperiereinheit wird über die RS232 – Schnittstelle (alternativ CAN-Bus-System) angesprochen. Für die Glykosylierungsreaktion ist die Temperierung eine der wichtigsten Einflussgrößen.	121
Abbildung 116: Julabo_LH85plus ist die Temperierungskomponente für die drei aufgebauten Arbeitssynthesizer	122
Abbildung 117: Das Gerätetreiber-VI für das Temperiergerät stellt alle Gerätefunktionen in einer Selektionsliste bereit. Die individuellen Parameter können dann direkt eingegeben werden.....	122
Abbildung 118: Mit den Abfragefunktionen können die aktuellen Geräteparameter aus der Geräteeinstellung ermittelt werden.....	123
Abbildung 119: Testprogramm im Befehlsinterpreter-VI zum Setzen einer Zieltemperierung und anschließender Schwellwertüberwachung, nachdem die weiteren Befehle erst ausgeführt werden, wenn eine untere Schwellwert-Temperatur von -14 ° C erreicht ist.....	124
Abbildung 120: Sprungantwort aufgrund der Vorgabe zum Erreichen der Solltemperatur von 50°C und Ergebnisverlauf bei einem periodischen Temperaturwechsel nach Erreichen der jeweiligen Zieltemperaturen von +50°C auf -40°C.....	125
Abbildung 121: Links: Ergebniskurve bei einem Temperaturanforderung von +40°C auf -40 in 43 min rechts: Sprungantwort bei einem Temperaturverlauf von -65°C auf +40 in 17 min	125
Abbildung 122: Test zur Reproduzierbarkeit der Temperaturführung. Die zeitlich aufgetragenen Verlaufskurven wurden segmentiert und in der weiteren Analyse als Wasserfalldarstellung zur besseren optischen Vergleichbarkeit hintereinander aufgetragen.....	125
Abbildung 123: Aufbauort der neuen Systeme Synthesizer I, II und III in Golm	126
Abbildung 124: Neuaufbau des Rahmensystems für Synthesizer I. Umsetzung des modularen Konzeptes.....	126
Abbildung 125: umgebauter Synthesizer	126
Abbildung 126: Bevorratung der Lösungsmittel in den jeweilig zugeordneten Lösungsmittelsicherheitschränken.....	127
Abbildung 127: 19" Systemanpassung als Zwischenlösung auf dem Weg zum mobilen 19"-Rack.....	127
Abbildung 128: Umbau auf neue Maße am Standort Potsdam-Golm für Synthesizer I, Synthesizer II, Synthesizer III(von links).....	128
Abbildung 129: Bedienkonzept der alten Synthesizer-Software bis 2012. In der Steuerungssoftware wurden die Kommandos einzeln ausgewählt, parametrisiert und so in die Programmliste eingefügt. Aus der Programmliste konnten Befehlslisten herauskopiert und in einem Textfile editiert werden. Aus dem Textfile wurden funktionale Befehlslisten zur Abarbeitung in die Programmliste der Steuerungssoftware zurück kopiert und zur Ausführung gebracht.	129
Abbildung 130: In der Übergangsphase 2012-2013 erfolgte die Verwaltung der Synthese-Programme des Altsystems als funktionale Einheiten in Form von Excel-Tabellen, die mit Hilfe von dialoggeführten Makros einfacher parametrisierbar wurden. Die Programmcodes steuerten gleichermaßen die beiden controllerbasierten Altsysteme, als auch den neuen dezentralen Arbeitssynthesizers mittels LabVIEW.....	130

Abbildung 131: Die funktionalen Programmeinheiten der alten Syntheseskripte wurden in Excel-Tabellen überführt. Zur Vereinfachung der Bedienbarkeit und zur Fehlervermeidung wurden mit VBA dialogbasierte Makros realisiert, z. B. zur Parametrierung der Befehle und zur automatischen Codebereitstellung in die Zwischenablage.....	131
Abbildung 132: Erste Version (links) der Benutzerschnittstelle für den neuen Arbeitssynthesizer zum Test der chemischen Funktionseinheiten.	132
Abbildung 133: In der Benutzerschnittstelle Version 1.6 wird der Abarbeitungszustand der Codeelemente farblich hervorgehoben. Es ist möglich in einer zweiten Excel-Spalte neben dem Code auch Kommentare zu hinterlegen und diese mit in die Anweisungsliste zu kopieren. Alternativ können Kommentare durch Semikolon definiert werden.	133
Abbildung 134: Die erste vollständig lauffähige *.exe Programmversion für die neue Synthesizer-Generation.	133
Abbildung 135: Einstellen des Simulationsmodus für die Synthesizer-Applikation. Somit können Syntheseprogramme ohne Hardwarezugriff ablaufen, um Fehler (Syntax, Schrittkettenfehler) in neuen Syntheseprogrammen zu erkennen.	134
Abbildung 136: Bedienoberfläche der Arbeitssynthesizer. Aus der Kommandoliste lassen sich im Skriptgenerator funktionale Programmeinheiten als Module zusammenstellen und abspeichern. Skripte können sich aus Einzelbefehlen und Modulen zusammensetzen. Die Abarbeitung der Skripte erfolgt im Skriptinterpreter.	135
Abbildung 137: Das Skriptbeispiel einer Fmoc – Entschützung besteht aus einzelnen funktionalen Modulen, die entsprechende Prozess-Schritte abbilden.	137
Abbildung 138: Der Transport des Lösungsmittels DMF in das Reaktionsgefäß erfolgt durch Schaltung der entsprechenden Spulenventile und infolge der Überdruckatmosphäre. Die Mengenzufuhr wird über die Schaltdauer der entsprechenden Ventile gesteuert.....	138
Abbildung 139: Funktionsmodul zum Lösungsmitteltransport in das Reaktionsgefäß bei Raumtemperatur (25°C).....	138
Abbildung 140: Funktionsweise des Moduls zum Spülen der Hauptverbindung mit Entschützungslösung. Zunächst wird Entschützungslösung mit der Spritzenpumpe aufgezogen, die über die Hauptverbindung zwischen dem Zulieferungs-Port und der Transporteinheit zur Unterbindung von Kontaminationen in den Abfallbehälter transportiert wird. Anschließend erfolgt das Aufziehen der Spritzenpumpe mit 2 mL der Entschützungslösung.....	139
Abbildung 141: Funktionsmodul zum Spülen der Verbindung zwischen Spritzenpumpe 2 und Zuführung der Entschützungslösungen.	139
Abbildung 142: Funktionsmodul zum Transport einer definierten Menge Entschützungslösung ins Reaktionsgefäß, das bereits von der Spritzenpumpe aufgezogen war. Anschließend werden die Schläuche geleert und Restmengen in den Abfallbehälter befördert. Im Reaktionsgefäß werden die Reagenzien durch Argonzufuhr (Bubbling) durchmischt. Der Pufferloop wird mit DMF als Entschützungslösung gespült. Anschließend wird eine definierte Menge Entschützungslösung 1 von der Spritzenpumpe aufgezogen, die Argonzufuhr gestoppt und ein Teil der Lösung aus dem Reaktionsgefäß in den Fraktionssammler transportiert.	140
Abbildung 143: Anweisungsliste des Funktionsmoduls: Zulieferung der Entschützungslösung 1 und Transport der Reaktionslösung in den Fraktionssammler	141

Abbildung 144: Schnellzugriff auf Standardfunktionen zur Funktionskontrolle oder für Wartungsarbeiten, die direkt ausgeführt werden.....	142
Abbildung 145: Anlegen von direkt ausführbaren benutzerindividuellen Favoriten-Funktionen auf Schaltflächen innerhalb der Synthesizer-Benutzeroberfläche durch Kopieren des gewünschten Programm-Moduls und Einfügen in das Buttons – Verzeichnis. So lässt sich ohne Programmieraufwand Einfluss auf die individuelle Funktionalität der Bedienoberfläche der Synthesizer-Applikation nehmen.....	143
Abbildung 146: Schematischer Ablauf für eine Lösungsmittelzufuhr zum Reaktionsgefäß. Für jedes der am Synthesizer verwendeten Lösungsmittel ist ein eigenes Ventil zuständig. So leitet Ventil 2 das Lösungsmittel DCM in die Transportleitung zum Reaktionsgefäß und Ventil 5 transportiert DMF. Ventil 16 öffnet die Zufuhr zum Reaktionsgefäß.....	144
Abbildung 147:Die allgemeine Funktionsbeschreibung führt zur spezifizierten Umsetzung als statisches Modul, das individuell für jedes Lösungsmittel und für jede Mengenänderung angepasst werden muss. Daraus abstrahiert sich der dynamische Prozess, bei der Parameter für die jeweiligen Lösungsmittel und Mengenangaben Verwendung finden. Die Parameter dienen als Platzhalter-Variablen für die spezifischen Angaben, die später in den ausführbaren Programmcode ersetzt werden. Dadurch verringern sich die benötigte Module der Lösungsmittelzufuhr für jedes der sieben Lösungsmittel * x mL Mengenangaben auf einen einzigen Prozess.....	145
Abbildung 148: Schematischer Programmablauf für den dynamischen Prozess der Lösungsmittelzufuhr. Lösungsmittelauswahl und Zufuhrmenge erfolgt über einen Benutzerdialog. Eine hinterlegte Zuordnungstabelle verwaltet das dem Lösungsmittel zugeordnete Ventil. Der Benutzer kann direkt die Zufuhrmenge angeben, ein Skalierungsfaktor berechnet die Transport- bzw. Wartezeit. In der Programmabarbeitung werden die „Schlüsselworte“ durch die interpretierten Parameter aufgefüllt und in ausführbaren Programmcode überführt.....	146
Abbildung 149: Konvertieren eines statischen Skripts zu einem dynamischen Prozess durch Austauschen der Prozessparameter mit @@Parameterbezeichnung@@ durch die Funktion „Suchen und Ersetzen“. Der Programmcode für Module / Skripts liegt als ASCII – Datei vor und kann somit in jedem beliebigen Texteditor geöffnet/bearbeitet werden.....	147
Abbildung 150: Das dynamische Prozess-Skript wird mit dem korrespondierenden Benutzerdialog zur Parametereingabe über den gleichlautenden Namen verknüpft. Das Prozess-VI muss dafür im Unterverzeichnis „panels“ des Synthesizer-Verzeichnisses abgespeichert werden. Für diesen Beispielprozess sind sowohl die zugeführten Building Blocks als auch zwei Reaktionstemperaturen und deren dazugehörige Haltezeit variabel.	148
Abbildung 151: Schematisch programmatischer Ablauf einer Disaccharid-Synthese mit Prozessparametern. Die einzelnen Syntheseschritte bilden sich in der Benutzeroberfläche mit jeweils eigenen Prozess-Skripten ab.	148
Abbildung 152: Das Syntheseskript zur automatischen Synthese eines Disaccharids besteht aus den einzelnen dynamischen Prozess-Schritten. Die jeweiligen Prozessparameter erschließen sich direkt aus der Befehlszeile.	149
Abbildung 153: Anweisungsliste des selektierten Syntheseskripts kann direkt mit START ausgeführt werden	150
Abbildung 154: Automatische Synthese linearer Di- und Hexa-Arabinofuranoside unter Verwendung des neuen vollautomatischen Arbeitssynthesizers.	152

Abbildungsverzeichnis

Abbildung 155: Automatische Synthese von verzweigten Tri- und Hexasaccharide unter Verwendung des neuen vollautomatischen Arbeitssynthesizers.	153
Abbildung 156: Automatische Synthese von Dermatan – Oligosacchariden unter Verwendung des Arbeitssynthesizers.	155
Abbildung 157: Entschützung des sulfatierten Disaccharids in der Lösungsphase	156
Abbildung 158: Hemicellulosen sind ein Bestandteil pflanzlicher Zellwände, deren Matrix aus fibrillärer, teilweise kristalliner Cellulose besteht.	157
Abbildung 159: Automatisierte Festphasensynthese von Arabinoxylan – Fragmenten. Der Buchstabencode unter den Strukturen zeigt die jeweilige Synthesesequenz am Arbeitssynthesizer.	159
Abbildung 160: Automatische Synthese von Xyloglucan – Oligosacchariden am Arbeitssynthesizer.	161
Abbildung 161: Automatische Synthese von Type-II Galaktan Fragmenten 83 – 89 am Arbeitssynthesizer. ^[185] Der Buchstabencode unter den Produkten bezeichnet die jeweiligen Syntheseschritte.	163
Abbildung 162: Automatische Synthese von Type-II Galaktan Fragmenten 90 – 96 am Arbeitssynthesizer. ^[185] Der Buchstabencode unter den Produkten bezeichnet die jeweiligen Syntheseschritte.	165
Abbildung 163: Automatische Festphasensynthese von Mannose Oligomeren am neuen Arbeitssynthesizer	167
Abbildung 164: Automatische Festphasensynthese von Mannose Oligomeren am neuen Arbeitssynthesizer	168
Abbildung 165: Zukunftskonzept - Funktionalitäten von Arbeitssynthesizer, konzeptionierten Molekülsynthesizer und Expertensystem	170
Abbildung 166: Arbeitsfluss zur Entwicklung eines service-basierten Labormanagementsystems. Mit den Begriffen aus dem Konzept Industrie 4.0 kann damit die Vorgehensweise beim Entwurf eines cyber-physischen „Chemie – Automatisierungs“ - Systems beschrieben werden.	173
Abbildung 167: Schematische Darstellung des Molekülsynthesizers als cyber - physisches System im Kontext des Zukunftsprojektes Industrie 4.0	174
Abbildung 168: Konzeptioneller Aufbau des Synthesizers als mobiles 19“ Rack in der Seitenansicht mit den drei Systembereichen: Chemie, Elektronik, Lösungsmittel. Elektronik und Lösungsmittelbereich sind auf Vollauszugsschienen montiert.	178
Abbildung 169: Prototyp modularer Systemkomponenten eingebaut in einem 19“ Rack. Maximale Flexibilität im Einbau oder Umbau neuer Komponenten, sowie die leichte Wiederverwendbarkeit in weiteren Systemen ist damit gewährleistet.....	179
Abbildung 170: Übersicht über rückseitige Hardware-Module in kompakter, für die 19“ Gerätevariante optimierter Bauweise.	179
Abbildung 171: Zur Reduzierung der Gesamtgeräteabmaße ist eine platzeffiziente Anordnung der Gerätekomponenten, hier der Datenerfassungskomponenten, notwendig.	179

Abbildung 172: a) Druckversorgung b) Ventilsteuerung c) Rückschlagventilblock d) Filterblock....	179
Abbildung 173: Der Zugriff (Wechsel/Wartung) auf den Anschluss der Spulenventile ist durch den Einbau eines Adapters im Mittelteil des Synthesizers vereinfacht.....	180
Abbildung 174: Schematische Darstellung des elektronisch druckgeregelten Argon-Manifolds	181
Abbildung 175: Prototyp des elektronischen Druckverteilers zur individuellen Einstellung der benötigten Versorgungsdrücke	182
Abbildung 176: Funktionstest mit der herstellereigenen „get red-y“-Software	182
Abbildung 177: Anzeigeelement in der Benutzeroberfläche für die Druckversorgung mit farbiger Hervorhebung der jeweiligen Druckbereiche	183
Abbildung 178: Fotolabile Linker in der automatischen Glykansynthese. ^[113]	184
Abbildung 179: Schematischer automatisierter Synthesesyklus mit fotoselektiver Abspaltung einer oder mehrerer temporären fotolabilen Schutzgruppen, sowie integrierter Abspaltung des fotolabilen Linkers.	186
Abbildung 180: Spektralverteilung der Standard-UV-Strahlungsquelle Hg-Z1 (rot) und Hg-Z4 (blau) von Peschel UV. Diese Lampen werden oft im Labor für die photochemische Prozessentwicklung und Produktion eingesetzt.	187
Abbildung 181: Verschiedene Übertragungsspektren für Schmalbandfilter mit typischer Bandbreite von 20 nm NS245(rot), NS297(blau), NS313(dunkelgrün), NS335(grün). Optischer Bandpassfilter blockieren durch Absorption und/oder Reflexion alle Wellenlängen oberhalb und unterhalb des spezifizierten Übertragungsbereichs.	187
Abbildung 182: Messung der Fotospektren der innerhalb der Belichtungskomponente vorhandenen Lichtquellen, Deuterium (rot), Halogen (weiß)	188
Abbildung 183: Messdaten und Messkurven der Spektren verschiedener Lampen.....	189
Abbildung 184: Spektrum der bisher verwendeten Leuchtstoff-Lampe direkt (blau) und nach Schlauchmaterial (rot)	189
Abbildung 185: Spektrale Veränderung des fotospaltbaren Linkers bei unterschiedlich langen Belichtungszeiten: Referenz keine Belichtung (grün), 5 min (pink), 10 min (türkis), 15 min (lila)	190
Abbildung 186: Die Absorptionsmessungen erfolgten mit UV / VIS-Spektralphotometer von Ocean Optics (USB4000) gesteuert von LabVIEW. Es zeigt die Absorption des Fotolinkers am Anfang (rot) nach 15 min Belichtung mit 311 nm Lampe (blau) Lösungsmittel: DCM.....	190
Abbildung 187: Vorbereitung zur Abspaltung des Linkers vom Resin. Verfahrensweise am alten Fotoreaktor bei Verwendung der Arbeitssynthesizer.....	191
Abbildung 188: Modularer Fotoreaktor – Eigenbau – Typ I – für den integralen Einbau am Entwicklungs-Synthesizer zur automatisieren Linkerabspaltung	191
Abbildung 189: Weiterführende Systemmodularisierung – Fotoreaktor Prototyp II 19“-Bauvariante.	191

Abbildung 190: a) Einlesen von NMR-Messdaten in LabVIEW konformer (x-Achse linear steigend) Darstellungsweise in das Beispielprogramm zur Konzeptstudie für die Weiterentwicklung des Synthesizers.....	193
Abbildung 191: Vergleich gemessener NMR-Daten (jeweils obere Kurve) mit simulierten NMR-Daten des abgebildeten Zuckerbausteins (jeweils untere Kurve)	194
Abbildung 192: Aufbau eines HSQC – Spektrums in LabVIEW aus den Daten von Jey Kandasamy's Arabinofuranosid Synthese. Überlagerung als 2D NMR im LabVIEW Programm als Visualisierungsmöglichkeit der eingelesenen Originaldaten (re.)	195
Abbildung 193: Einbindung GlycoNMRSearch in das LabVIEW-Beispielprogramm.	195
Abbildung 194: Implementierung von ChemDoodleNMR in die LabVIEW-Programmumgebung	196
Abbildung 195: Konzept der notwendigen Entwicklungsstufen für die Bedienebenen und Benutzerschnittstellen von Molekülsynthesizer und Expertensystem, ausgehend vom aktuellen technologischen Entwicklungsstand des Oligosaccharid-Synthesizers.....	204
Abbildung 196: In der Weiterentwicklung des Synthesizer-Konzepts ist eine Benutzer-Assistenz zur einfachen Gerätebedienung vorgesehen.....	206
Abbildung 197: An den Arbeitssynthesizern hergestellte Arabinoxylan- Oligosaccharide. ^[183] Beispiel von Schnellparametrieroberflächen zur Eingabe der Glykosylierungsparameter zur vereinfachten Bedienung am Konzeptsystem.	209
Abbildung 198: Schnellparametrierung von Strukturvariationen über anwenderindividuelle Schaltflächen. Der Zugriff über den „Glykan-Code“ des gewünschten Endproduktes definiert so durch minimale Anpassungen die automatische Synthesesteuerung. a) Strukturvariationen von Cellulose- und XG Oligosaccharid – Fragmenten ^[184] b) Herstellung von Typ II Galaktan Fragmenten aus vier Grundbausteinen. ^[185]	210
Abbildung 199: Ausschnitt aus der Benutzerverwaltung und vereinfachte Anwenderoberfläche zur Selektion der Synthesesteuerung, z. B. von Mannosen- Synthesen. Das Kontextmenü listet die benutzerindividuelle Synthesebibliothek des selektierten Benutzers auf.	211
Abbildung 200: Projektbezogene, individuell parametrierbare Benutzeroberfläche zur Synthesesteuerung einer n-Mannose.....	212
Abbildung 201: Exemplarisch stehen die Monosaccharid – Building Blocks zur automatischen Synthese von Lactotetraosyl (Lc4) und Neoleactotetraosyl (nLc4) – Kernstrukturen für das Projekt „Automated Glycan Assambly of Complex Oligosaccharides Related to Blood Group Determinants“ ^[284] für die Vielfalt möglicher Zuckerbausteine. Dies erfordert eine effektive Datenverwaltung von Prozessinformationen in Relation mit den verwendeten Building Blocks zur Durchführung von System- / Prozessanalysen.	213
Abbildung 202: Definitions-Dialog für den Building Block Selektor	214
Abbildung 203: a) Zuckerbaustein (Building Block) b) Analog dazu der Zuckerbaustein im Building Block Selektor, einer in LabVIEW programmierten Programmoberfläche zur Eingabe der verwendeten Zuckerbausteine mit wählbarer Abgangsgruppe (LG Leaving Group) und verschiedenen Schutzgruppen (Sub).	214

Abbildung 204: Auswahllisten der aktuellen Substituenten und der bisher verwendeten Abgangsgruppen.....	215
Abbildung 205: LabVIEW - Programmausschnitt, der den Zugriff auf die verschiedenen Eigenschaftsknoten des ChemDraw Objekts bis hinunter zu den einzelnen Objekten innerhalb der Reaktionsschritt-kette zeigt.....	216
Abbildung 206: ChemDraw Objekte können über die Zwischenablage in die erstellte LabVIEW Applikation kopiert werden und stehen dann mit ihrer vollen Informationsstruktur zur Verfügung. Das LabVIEW-Programm übernimmt die übergebene Visualisierung und Zusatzinformationen.	217
Abbildung 207: Das LabVIEW Programm wertet automatisch Fehler aus. Aufgrund von Tippfehlern nicht korrekt erkannte Objekte werden im chemischen Molekül rot umrandet.	218
Abbildung 208: Kontextmenü – innerhalb des ChemDraw – Synthesizer VIs. Über das Kontextmenü können auch die chemischen Warnungen angezeigt werden.	218
Abbildung 209: Parametrierung und ihre funktionale Bedeutung.....	220
Abbildung 210: a) Die Reaktionsgleichung in ChemDraw kann als funktionaler „chemischer Programmcode“ von LabVIEW genutzt werden. b) Die farbige Zuordnung zur Erkennung der Objektbedeutung der einzelnen chemierelevanten Informationen als Grundlage zur „Funktionssteuerung des Synthesizers“	220
Abbildung 211: Zuordnung der Reaktionsparameter für die Synthesesteuerung in LabVIEW	221
Abbildung 212: Für das „Proof of Konzept“ des zukünftigen Molekülsynthesizers wurden verschiedene Molekül-Editoren in die Synthesizer-Applikation implementiert: a) Sketcher, b) Marvin, c) Ketcher, d) PubChem.....	223
Abbildung 213: Glykan-Editor für den Synthesizer: Für das „Proof of Konzept“ des zukünftigen Molekülsynthesizers wurden verschiedene Open-Source Glykan-Editoren in die Synthesizer-Applikation implementiert: a) GlyTouCan, b) Carbohydrate Builder. Die innerhalb des Glykan-Editors gezeichneten Glykane können als Syntheseanforderung dem Molekülsynthesizer übergeben werden.	226
Abbildung 214: Das Labor Informations-Managementssystem (LIMS) im Kontext des Expertensystems	227
Abbildung 215: Schrittkette für die Automatisierung von Aufgaben	228
Abbildung 216: Syntheseplanung im E-Notebook.....	229
Abbildung 217: Menü E-Notebook: Eingabe von Zusatzinformationen (Metadaten) im E-Notebook unter AutoText Definitions.....	230
Abbildung 218: Labarchives als Beispiel einer plattformübergreifenden Labor-Notebooks.....	231
Abbildung 219: LabVIEW-VI zum Zugriff auf das Open Access Data Repository Edmond	231
Abbildung 220: API-Funktionen des Open Access Data Repositories EDMOND und Ausgabe einer GET – Anfrage in LabVIEW (rechts).....	232
Abbildung 221: Rspace API Funktions-Referenz für das Framework Swagger.....	233

Abbildung 222: API-Objektreferenz für JavaScript Object Notation (JSON), einem kompakten Datenformat in einfach lesbarer Textform für den Datenaustausch zwischen Anwendungen.	234
Abbildung 223: Konzeptüberblick des Themenkomplexes Datenarchivierung, Datenformate und Datenbanken vom Molekülsynthesizer bis zum Expertensystem.	235
Abbildung 224: Konzept der Datenbankstruktur innerhalb der Synthesizer-Applikation für Molekülsynthesizer als Grundlage zur Entwicklung des Expertensystems.	241
Abbildung 225: Implementierung von Common Chemistry ermöglicht den Zugriff auf die CAS-Registrierungsnummern von Chemikalien aus dem LabVIEW-Programm heraus.	242
Abbildung 226: LabVIEW – Benutzeroberfläche der Konzeptstudie zur Nutzung der PubChem – Datenbank, die als Web-API/REST-Zugriff in den LabVIEW Programmcode implementiert wurde. Dabei sind nicht nur textorientierte Zugriffe möglich (a), sondern auch die Ausgabe der Strukturformeln (b) und weitergehenden Informationen zu den angefragten Molekülstrukturen. ..	243
Abbildung 227: LabVIEW-VI zur Implementierung des Webdienstes der Datenbank ChEBI.	244
Abbildung 228: Implementierung von ChEMBL in die LabVIEW-Benutzeroberfläche.	244
Abbildung 229: In die LabVIEW Applikation implementiertes Suchmasken-Werkzeug KCaM erleichtert Datenbank-Abfragen in der KEGG-DB.	245
Abbildung 230: LabVIEW-Programmcode aus der Konzept-Studie: Konvertierung eines SMILE Strings in einen InChI String nach IUPAC.	248
Abbildung 231: Beispiel eines LabVIEW-Eigenschaftsknotens aus der Konzept-Studie: Konvertierung von SMILE in einen InChIKey - String mit der Open-Source-Toolbox OpenBabel. Durch Implementierung von OpenBabel in LabVIEW können aktuell über 110 chemische Datenformate konvertiert und interpretiert werden.	248
Abbildung 232: LabVIEW-Programmoberfläche zeigt die Implementierung von OPSIN, einem Open Source Konvertierungswerkzeug für chemische Strukturen.	249
Abbildung 233: Beispiel der Implementierung einer webbasierten Konvertierungsplattform in LabVIEW-Programmcode. Über Listenstrukturen (Quellformat, Zielformat) lässt sich die Konvertierung mittels Web-API/REST - Zugriffe ausführen. Das LabVIEW Programm stellt die nötigen URL-Strings zusammen, und stellt im Response die angeforderte-Antwort bereit.	250
Abbildung 234: LabVIEW – Benutzeroberfläche zur Abfrage und Ausgabe von Strukturen mit dem Chemical Identifier Resolver (CIR). Die Implementierung in das LabVIEW – Programm erfolgt über eine URL-Anfrage basierend auf dem HTTP-Protokoll.	250
Abbildung 235: Implementierung des Chemical Translation Service in die LabVIEW-Oberfläche. Der Webservice erlaubt Batch-Konvertierung und kann damit eine größere Anzahl von Datei-Konvertierungen der gebräuchlichsten Verbindungskennungen gleichzeitig durchführen.	251
Abbildung 236: Beispielimplementierung verschiedener Datenbanken und Frameworks für die Toxizitätsbeurteilung und Sicherheitsbewertung der Synthesen innerhalb der Synthesizer-Applikation.	254

Abbildung 237: a) Verknüpfung von Monomeren im HELM-Editor; b) Definition der reaktiven Kopplungsstelle; c) Definition eigener Monomere mit bis zu fünf verschiedenen reaktiven Kopplungsstellen257

Abbildung 238: Implementierung von BallVIEW (ob.) oder MOLE (re.), einem leistungsstarken Visualisierungs- und Modellierungswerkzeug für die Konzept-Studie des Molekülsynthesizers.264

Abbildung 239: Überblick der Abstraktionsebenen der Systementwicklung für die realisierten automatischen Oligosaccharid-Synthesizer neuer Generation (Arbeitssynthesizer) im Kontext zu den in der Konzeptstudie vorgestellten Weiterentwicklungsmöglichkeiten als Molekülsynthesizer und Expertensystem.270

Abbildung 240: Der Oligosaccharid-Syntheseautomat im Deutschen Technikmuseum273

Abbildung 241: Aufbausegmente im Deutschen Technik Museum a) Druckverteiler und Lösungsmittelzufuhr (links), b) Pumpen und Rotationsventile mit Reaktor und Temperierungseinheit(rechts).....274

Abbildung 242: Informationsfilm im Deutschen Technik Museum am Schaustück zum historischen JKEM- Oligosaccharid-Synthesizer274

TABELLENVERZEICHNIS

Tabelle 1: Isomeriemöglichkeiten bei Biopolymeren ohne D- und L- Formen. $[\alpha]_{3' \rightarrow 5'}$ verknüpfte Desoxyribo- oder Ribonukleinsäuren $[b]$ bezogen auf Hexopyranosen (Glc, Man, ...). ^[24] Aufgrund der verschiedenen Verknüpfungsvarianten beinhalten Oligosaccharide weit mehr Information als Peptide oder Nukleinsäuren	6
Tabelle 2: Beispiele von Glykosyl-Donoren bzw. Akzeptoren für die Synthese an der festen Phase	12
Tabelle 3: Beispiele für Linker in der automatisierten Oligosaccharid-Festphasensynthese. Bis auf Linker Nr. 4, sind alle an ein Merrifield – Harz gebunden.....	22
Tabelle 4: Befehlsliste - Funktionsbeschreibung	136
Tabelle 5: Fotochemisch abspaltbare Schutzgruppen	186
Tabelle 6: Auswahl allgemein zugänglicher chemischer, biologischer und medizinisch relevanter Datenbanken im Internet, mit denen bereits programmatische Interaktionen innerhalb der Synthesizer-Applikation getestet wurden.....	247

FORMELVERZEICHNIS

Formel 1: Näherungsformel Dampfdruck entsprechend der Clausius-Clapeyron`schen Gleichung ...	114
--	-----

LEBENS LAUF

Mein Lebenslauf wird aus Gründen
des Datenschutzes in der elektronischen
Fassung meiner Arbeit nicht veröffentlicht.

Mein Lebenslauf wird aus Gründen
des Datenschutzes in der elektronischen
Fassung meiner Arbeit nicht veröffentlicht.

ANHANG

1. Wartungsinformationen.....	4
1.1. Installation von LabVIEW als Entwicklungssystem mit NI-Karte.....	4
1.2. Installation der LabVIEW Runtime Version	8
1.3. DIO Box.....	10
1.4. Kommunikation über RS232 / RS485 - Schnittstellen.....	11
1.5. Verbindung der Kloehn-Komponenten	17
1.6. VBA Zugriff auf die Kloehngeräte	22
1.7. MAX-Zugriff auf die Kloehngeräte	24
1.8. Wartung: Testprogramme für Kloehn – Pumpe.....	25
1.9. Konfiguration der Rotationsventile	27
1.10. Austausch des Pumpenkopfes	29
1.11. Tests über Kloehn Control	30
1.12. Fraktionssammler - Schnittstellenparameter	35
1.13. Testprogramme Fraktionssammler	38
1.14. Temperiereinheit – Julabo – Konfiguration der RS232 Schnittstelle	41
1.15. Test Julabo Hyperterminal Programm	44
1.16. Test Julabo im MAX.....	45
1.17. Schnittstellenparameter Temperiereinheit Julabo	48
1.18. Testprogramme Temperiereinheit Julabo	50
1.19. Testprogramme Ventilschaltungen.....	51
1.20. Testprogramme Synthesizer	53
1.21. Verschaltung des druckgeregelten Massendurchflussmessers (ES)	56
2. LabVIEW VIs	59
2.1. Befehlsinterpreter: Liste_interpretieren VI.....	59

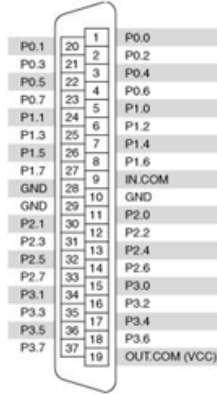
2.2.	Befehlsinterpreter: Gerätezuordnung VI	60
2.3.	Befehlsinterpreter: INI () – Spritzenpumpe / Rotationsventil.....	62
2.4.	Befehlsinterpreter: RATE () – Spritzenpumpe	65
2.5.	Befehlsinterpreter: WITHDRAW() – Spritzenpumpe.....	70
2.6.	Befehlsinterpreter: DISPENSE () – Spritzenpumpe	71
2.7.	Auswerteprogramm Druckmessung am Druckverteiler (Manifold)	72
2.8.	Fraktionssammler: HOME Interpretation.....	73
2.9.	Fraktionssammler: MOVE – Befehl.....	73
2.10.	Fraktionssammler: Testprogramm.....	74
2.11.	Auswerteprogramm Protokolldatei Julabo.....	75
2.12.	Testversion Benutzerschnittstelle Arbeitssynthesizer I	77
2.13.	Unterscheidung der Applikationsumgebung	78
2.14.	Drag & Drop Bedienung	79
2.15.	Aktualisierung bei Wechsel der Registerkarte	81
2.16.	Programmbeispiel-VI: Benutzerkommunikation mit dem elektrischen Manifold	83
2.17.	Einbindung von ChemDraw Funktionalitäten.....	84
2.18.	Zugriff auf ChemFinder aus LabVIEW	92
3.	Fehlerbilder - Ursachen	93
3.1.	Rückschlagventile, Spritzenpumpe	93
3.2.	Fehlerbeseitigung - Kloehngeräte	96
3.3.	Fehler Fraktionssammler	100
3.4.	Fehler / Wartung – Temperiereinheit Julabo	101
4.	Lösungsmittel Beständigkeitstest für den kommerziellen Synthesizer	104
5.	Liste der Syntheseprodukte aus den Arbeitssynthesizern.....	106
6.	Beschreibung der Prozessmodule.....	128

1. Wartungsinformationen

1.1. Installation von LabVIEW als Entwicklungssystem mit NI-Karte

Bei den Arbeitssynthesizern kommen, neben Standarddesktop-PC's, zur Steuerung der Ventile Digital I/O Karten vom Typ: *NI PCI 6519* der Firma National Instruments zur Anwendung. Diese Karten werden vollständig von dem DAQMX Treiber unterstützt und bieten somit eine sehr effiziente Möglichkeit die Steuergrößen für die chemischen Ventile (24 V, 100 mA) bereit zu stellen.

Aufgrund der Spezifikationsbeschreibung der PC Karte müssen die Karten mit einer externen Verstärkerbox (DIO Box) für die verwendeten Chemieventile betrieben werden. Über die nachfolgend aufgezeigte PIN-Belegung ergibt sich die Zuordnung zwischen der jeweiligen PIN – Nummer und dem dazugehörigen Signalnamen, der sich innerhalb der Programmierung wiederfindet.



Pin Number	Signal Name	Input or Output*	Signal Description
1, 2, 3, 4, 20, 21, 22, 23	P0.<0..7>	Input	Data lines for port 0
5, 6, 7, 8, 24, 25, 26, 27	P1.<0..7>	Input	Data lines for port 1
11, 12, 13, 14, 30, 31, 32, 33	P2.<0..7>	Output	Data lines for port 2
15, 16, 17, 18, 34, 35, 36, 37	P3.<0..7>	Output	Data lines for port 3
9	IN.COM	Input	Common line for all input channels
10, 28, 29	GND	Input	Ground for all ports
19	OUT.COM (VCC)	Input	Power line for all output channels

* The NI 6519 inputs are isolated from the outputs.

Abbildung 1: Pinbelegung NI-PCI-6519 und Zuordnungstabelle Pin und Signal nach Bedienungsanleitung

Zum Betrieb der Karte ist die Installation der *DAQMX-Treiber* für LabVIEW notwendig. Da gerade bei den zu erwartenden Wechslen des Betriebssystems zur Laufzeit der Arbeitssynthesizer alle Informationen hilfreich sind, nachfolgend die wichtigsten Schritte der Inbetriebnahme und der Funktionstests. Hierbei werden die Screenshots eingebunden, um möglichst umfassend alle Sachverhalte aufzeigen zu können.

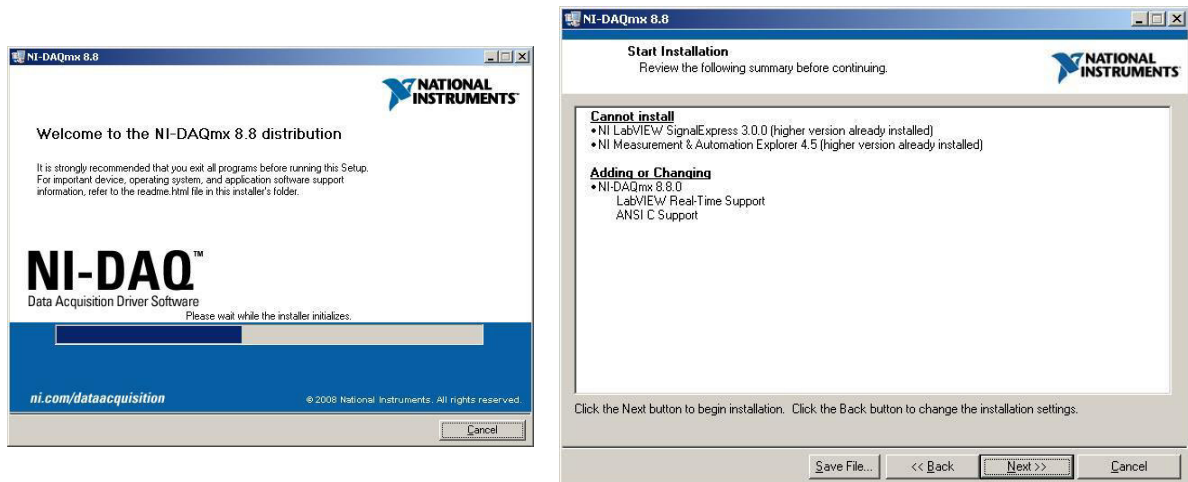


Abbildung 2: li: Startfenster bei der Installation des DAQmx-Treiber, re.: Statusrückmeldung

Durch die bereits installierte aktuellere Entwicklungsumgebung, können einige Dateien nicht upgedatet werden. Die kartenindividuelle Software wird installiert und von Seiten des Betriebssystems nach der Detektierung als *PCI Device* als *PCI 6519* registriert. Damit ist die zum Betrieb notwendige Treibersoftware vervollständigt.

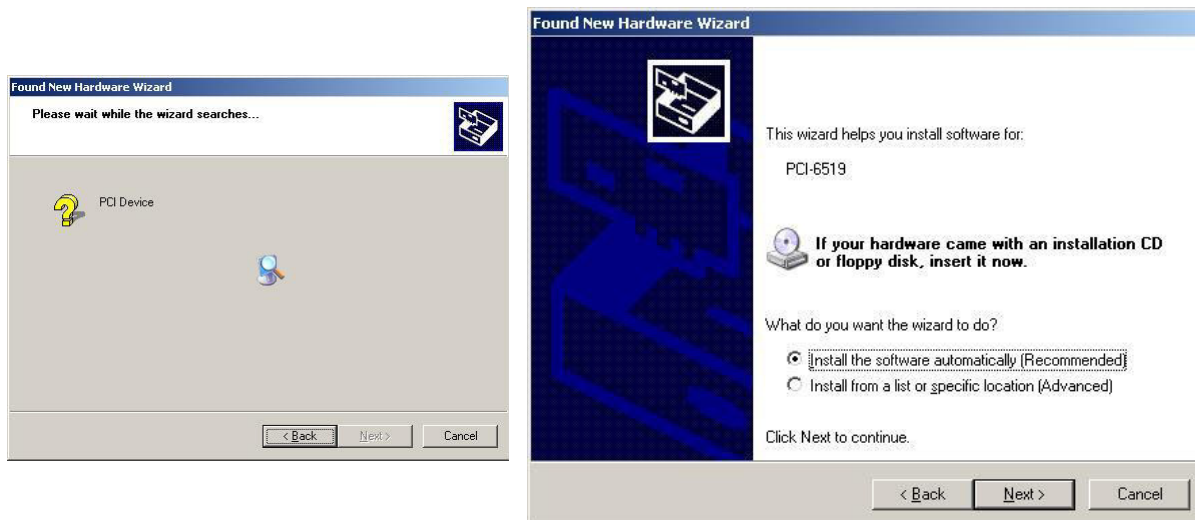


Abbildung 3: li.: Installation des Treibers für PCI Karte, re.: Gefundene Hardware

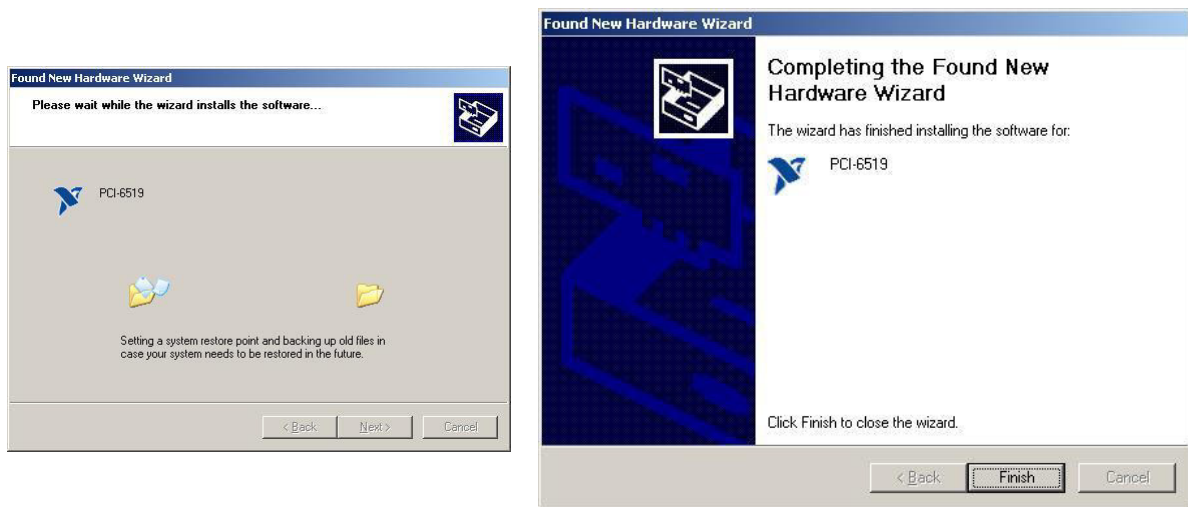


Abbildung 4: li: Kopieren der individuell benötigten Dateien, re.: Abschluss der Treiberinstallation
Zur Funktionskontrolle kann der Zugriff über den Gerätemanager auf die dort vorhandene DAQMX Geräte Eintragungen erfolgen. Über das Kontextmenü der Geräteeigenschaften, stehen detaillierte Informationen bereit.

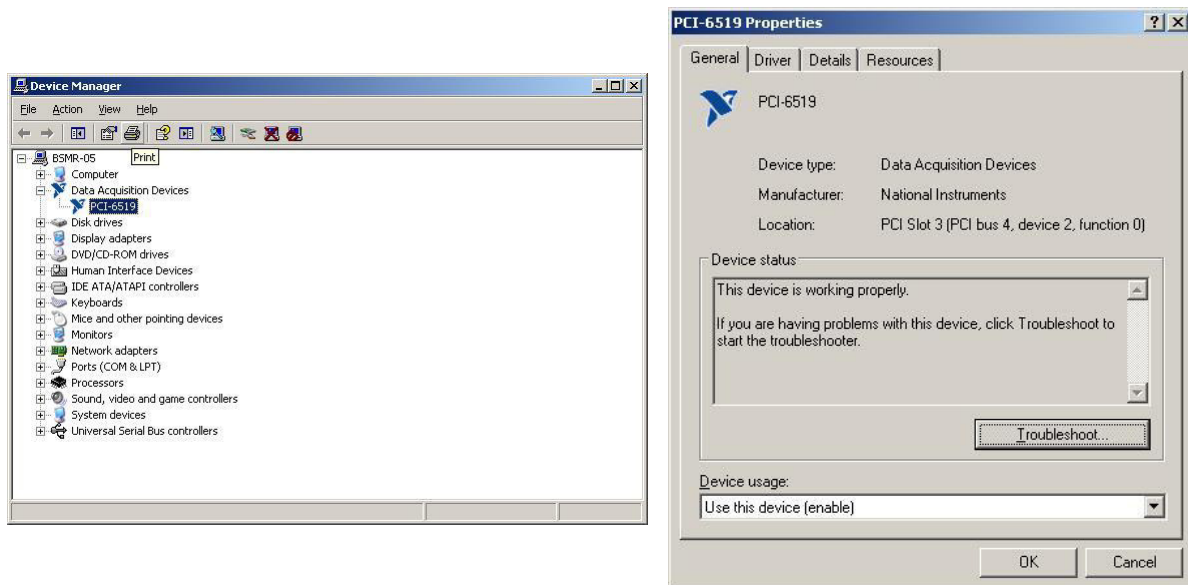


Abbildung 5: li.: Kontrolle der Hardware im Gerätemanager, re.: Statusinformation der NI-Karte

Der abschließende Funktionstest der Karte kann unter *NI-MAX* durchgeführt werden. Im Verzeichnisbaum der detektierten Geräte und Schnittstellen, befindet sich nun unter *DAQmx* die neu installierte Hardware.

Nach erfolgreichem Selbsttest der Karte, kann diese nun voll genutzt werden.

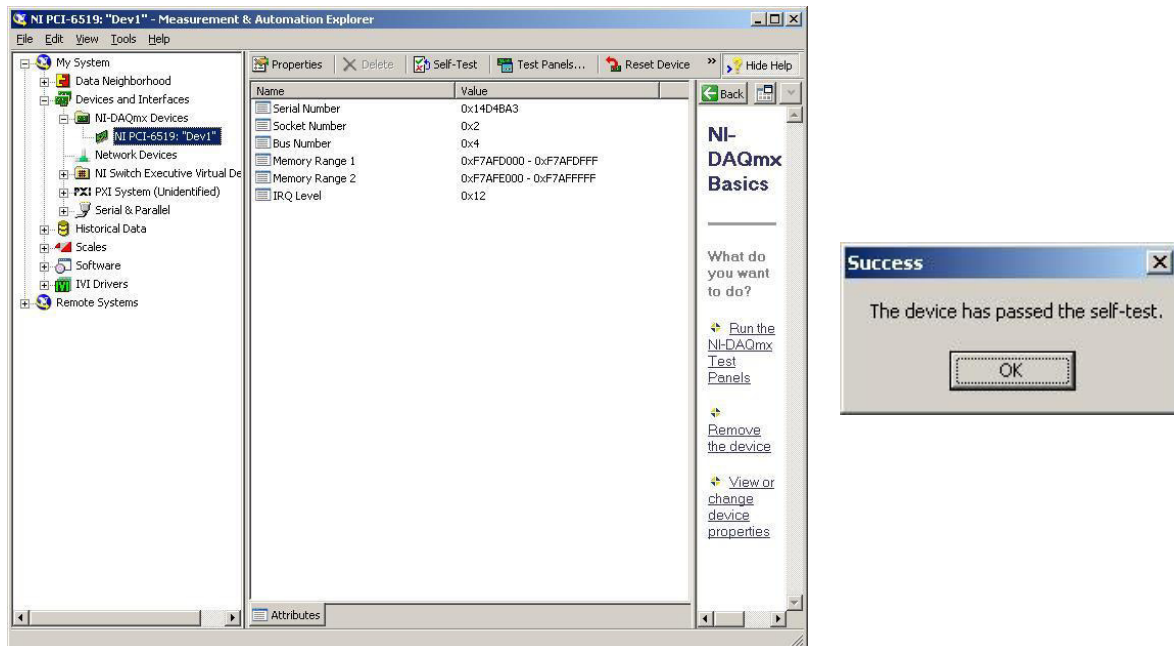


Abbildung 6: li.: Auflistung im MAX, re.: Statusmeldung des Selbsttests im MAX

1.2. Installation der LabVIEW Runtime Version

Der nachfolgende Dialog zeigt die minimal notwendigen Installationsroutinen, die lokal ablaufen müssen, um die Funktionalität zur Verfügung zu stellen. Der Anwender wird sowohl durch die Installationsroutine, als auch durch verschiedene Abfrageroutinen geführt und hat am Ende eine vollwertige Installation der Synthesizerbedienoberfläche im Zugriff.

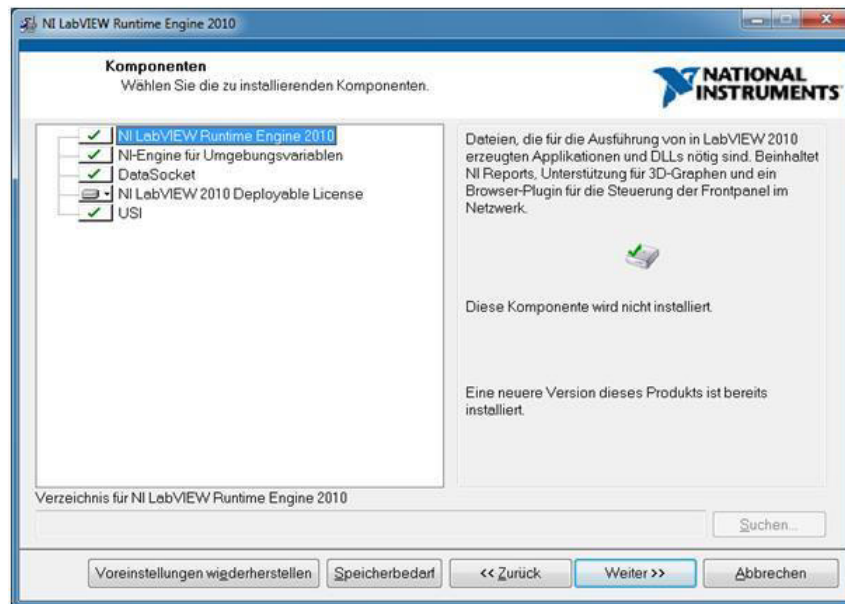


Abbildung 7: Runtime- Installation 1

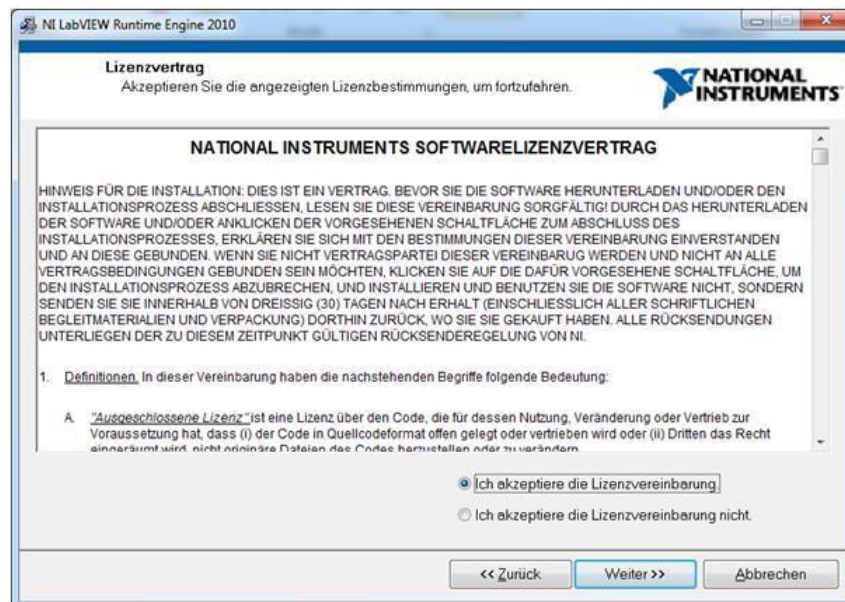


Abbildung 8: Runtime- Installation 2

Die Verknüpfung wird mit der Standardoption, durch ein entsprechendes Desktop-Icon auf die Synthesizeroberfläche abgebildet. Von dort aus verzweigt es in das bereit gestellte Unterverzeichnis des Synthesizers mit seinen verschiedenen Verzeichnisebenen. Es können weiterhin darüber hinaus direkt individuelle Desktop-Verknüpfungen realisiert werden, die

einen durch verschiedene Parameter vorgegebenes Abarbeitungsszenario am Synthesizer auslösen.

Hierdurch lassen sich in seiner maximalen Ausdehnung sogar komplette Sequenzen über die Microsoft-Standardschnittstelle am System abarbeiten.

1.3. DIO Box

Der Eigenbau liefert 24 V und 100 mA an 16 getrennt galvanischen Ausgängen zur Schaltung der Spulenventile von BIO-CHEM. Zusätzlich werden 5 V Spannung an den Bananenstecker-Ausgängen zur Verfügung gestellt.



Abbildung 9: DIO – Box, links Draufsicht vorne , rechts: Draufsicht hinten – erste DIO Verstärkerbox verwendet die Spezialstecker von J-KEM aus den Alt-Synthesizern

Bei der ersten DIO Box kommen die Spezialstecker* für die Niedervoltverbindung zur Kontaktierung der Verstärkerbox vom umgebauten Altgerät aus Kompatibilitätsgründen zur Verwendung. Aufgrund der Recherche zu den Steckertypen und der dokumentierten Anwendungsfälle, resultiert, dass diese Stecker für den Laboranwendungsfall nicht freigegeben sind. Sowohl die chemische Resistenz, als auch die elektrische Sicherheit der Position im Gerät, entsprechen keiner angestrebten Laborsicherheit!

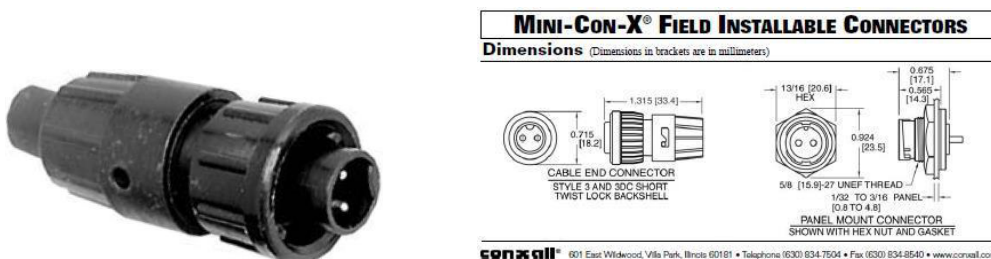


Abbildung 10: Niedervoltstecker der JKEM-Box, bisherige Steckverbinder an den alten Synthesizern

In den neuen DIO Geräten für den Arbeitssynthesizer III und den Entwicklungssynthesizer sind diese durch Standard – Niedervoltstecker mit chemischem Anwendungsbereich ersetzt.

1.4. Kommunikation über RS232 / RS485 - Schnittstellen

Die Komponenten (Spritzenpumpen, Rotationsventile, Fraktionssammler und Temperierungsmodul) arbeiten aufgrund von Steuerungsinformationen über ein serielles Bussystem. Hierbei kommt sowohl die RS232 – Schnittstelle, als auch die RS485 zum Einsatz.

1	2	3	4	5	6	7	8	9
rot/braun	gelb	grün	rosa	blau/grau	lila	schwarz	schwarz	weiß

Abbildung 11: Nullmodemkabel Pinbelegung Buchse

1	2	3	4	5	6	7	8	9
schwarz	grün	gelb	lila	blau/grau	rosa	rot	braun	weiß

Abbildung 12: Nullmodemkabel Pinbelegung Stecker

Da in den standardmäßig verwendeten PC-Varianten heute nicht mehr diese Schnittstellen auf dem Motherboard zur Verfügung gestellt werden, wird die Anzahl der benötigten Busse über die Bereitstellung mit Hilfe von USB – Seriell – Port Adaptern realisiert, deren Funktionalität über den Gerätemanager windowsseitig in Erfahrung gebracht werden kann.

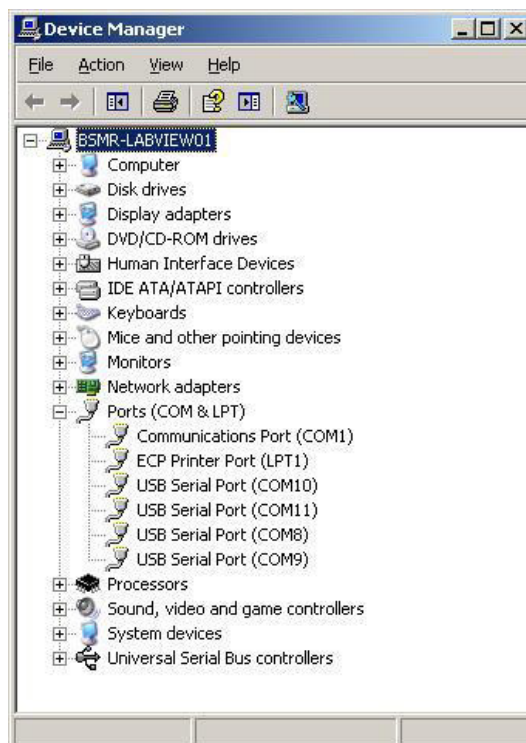


Abbildung 13: Auflistung der seriellen Schnittstellen

Hier hat auch der Anwender die Möglichkeit die entsprechenden Zuweisungen zu den verwendeten COM Ports softwareseitig vorzugeben.

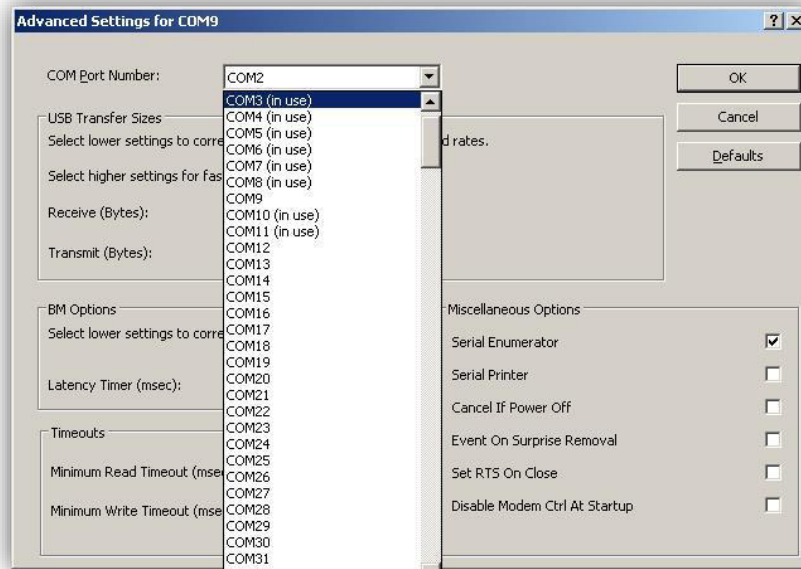


Abbildung 14: Umkonfigurierung der Schnittstellenzuordnung

Die unterschiedlichen Belegungen der jeweiligen Steckervarianten der Endgeräte mit den Informationen zu eventuell genutzten Handshakes sind nachfolgend aufgelistet.

PCCOM1 9Pol männl	PCCOM2 25Pol männl	PC RS232	In/Out	Bezeichnung	Anmerkung
1	8	DCD	in	Data Carrier Detect	Signalträger erkannt
2	3	RxD	in		Empfangsdaten
3	2	TxD	out		Sendedaten
4	20	DTR	out	Data Terminal Ready	
5	7	GND		Ground	Bezugspotential
6	6	DSR	in	Data Set Ready	Endgerät bereit
7	4	RTS	out	Request to send	Sende Anfrage
8	5	CTS	in	Clear to send	Fertig zum Senden
9	22	RI	in	Ring Indicator	Klingel

Abbildung 15: Steckerbelegung der RS232-Stecker

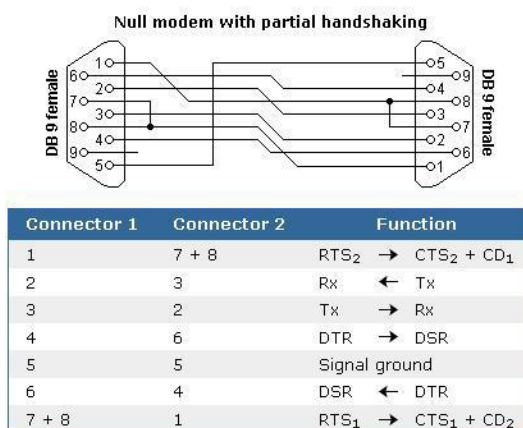


Abbildung 16: Schnittstellenbelegung der Kloehn-Kommunikation

Schnittstellen-Verbindungskabel RS232:

Thermostat (9polig)	PC (9polig)
Pin 2 RxD	Pin 3 TxD
Pin 3 TxD	Pin 2 RxD
Pin 5 GND	Pin 5 GND
Pin 7 RTS	Pin 8 CTS
Pin 8 CTS	Pin 7 RTS

Abbildung 17: Schnittstellenbelegung der Julabo-Kommunikation

Bezeichnung Stecker 1	DCD, DSR	RxD	TxD	DTR	GND	RTS	CTS	RI
Pin Stecker 1	1 und 6	2	3	4	5	7	8	9
Pin Stecker 2	4	3	2	1 und 6	5	8	7	9
Bezeichnung Stecker 2	DTR	TxD	RxD	DCD, DSR	GND	CTS	RTS	RI

Abbildung 18: Komplettbelegung

Da die Adapter relativ tief in die Systeminfrastruktur des Windows eingreifen, müssen für die unterschiedlichen Windowsvarianten die jeweils geeigneten Treiber installiert werden. Dies ist nachfolgend exemplarisch aufgezeigt.

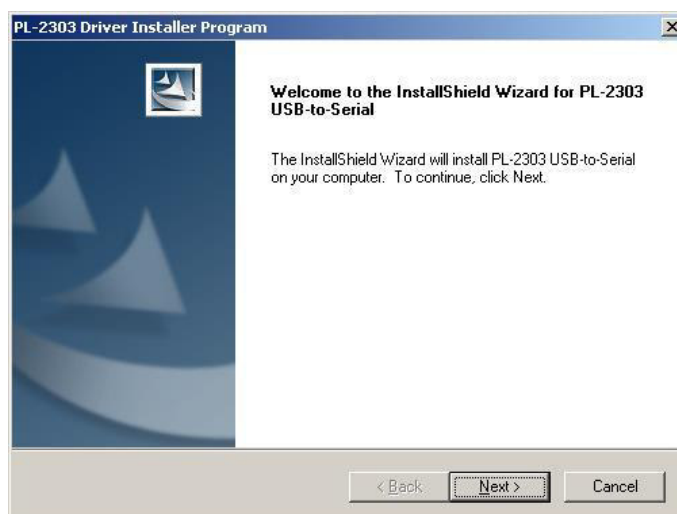


Abbildung 19: Installation der Treibersoftware

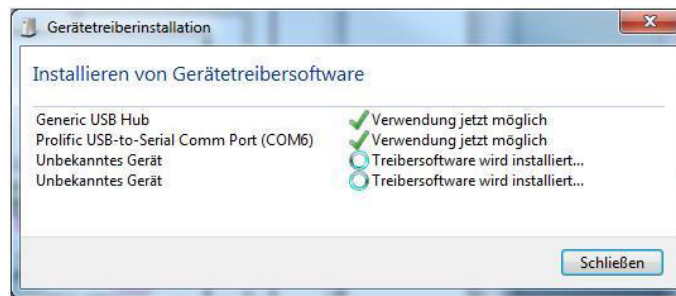


Abbildung 20: Gerätetreiberinstallation

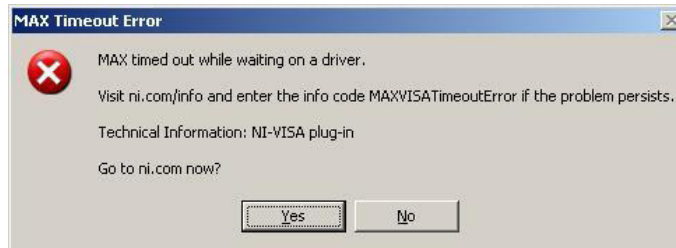


Abbildung 21: Zugriffsprobleme

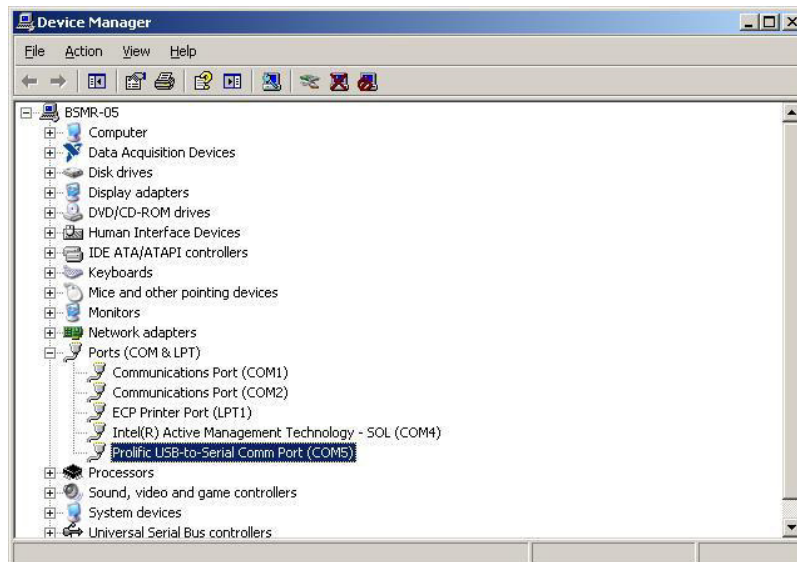


Abbildung 22: Kontrolle der Hardware



Abbildung 23: Funktionstest der Schnittstelle

Nachdem die hardwaretechnischen Voraussetzungen auf der jeweiligen PC-Seite (3 freie COM Schnittstellen) realisiert ist, kann die geräteindividuelle Konfiguration erfolgen:

Kloehngeräte COM1

Julabo COM2

Fraktionssammlers COM5

Die Schnittstellen COM3 und COM4 wurden jeweils zur Protokollierung der Buskommunikation verwendet und haben für die eigentliche Funktion des Systems keine Bedeutung.

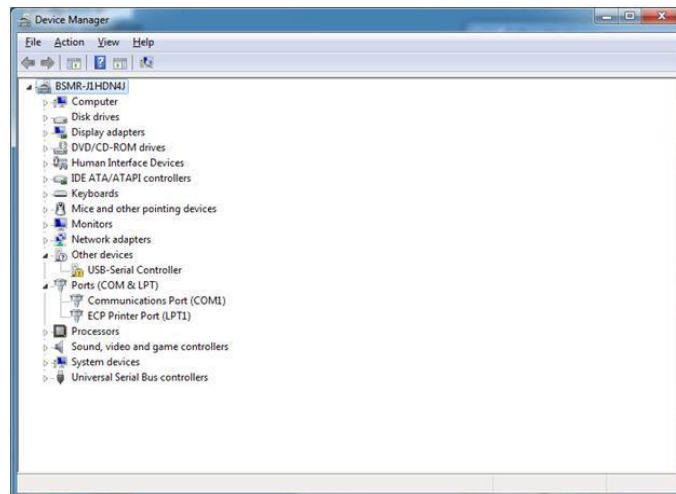


Abbildung 24: Probleme des USB Controllers

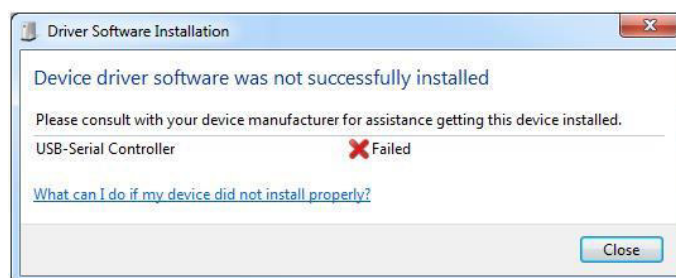


Abbildung 25: Keine vollständige Installation

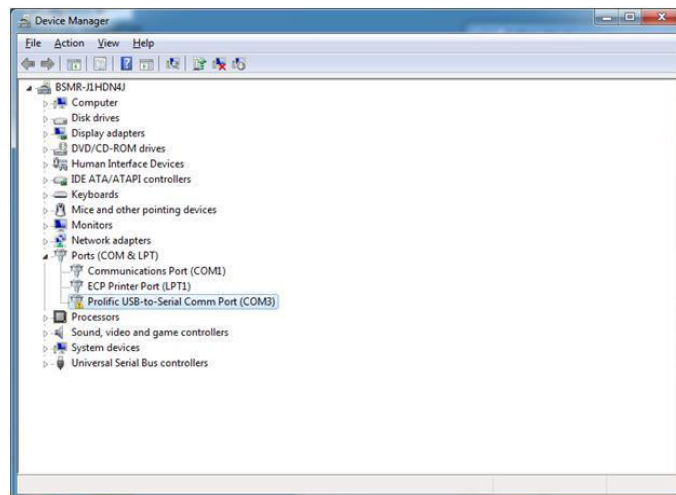


Abbildung 26: Zwar erkannt, jedoch ohne Funktion

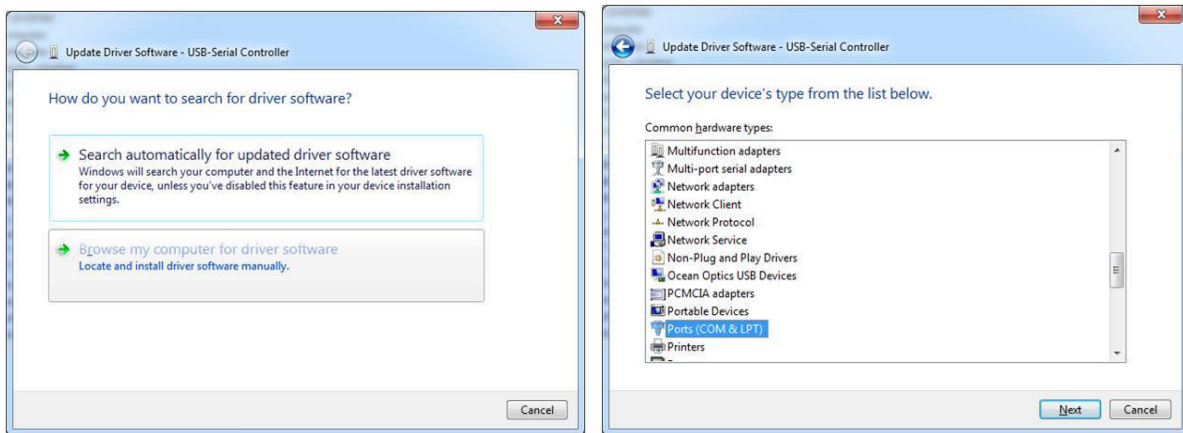


Abbildung 27: Manuelle Auswahl der Version

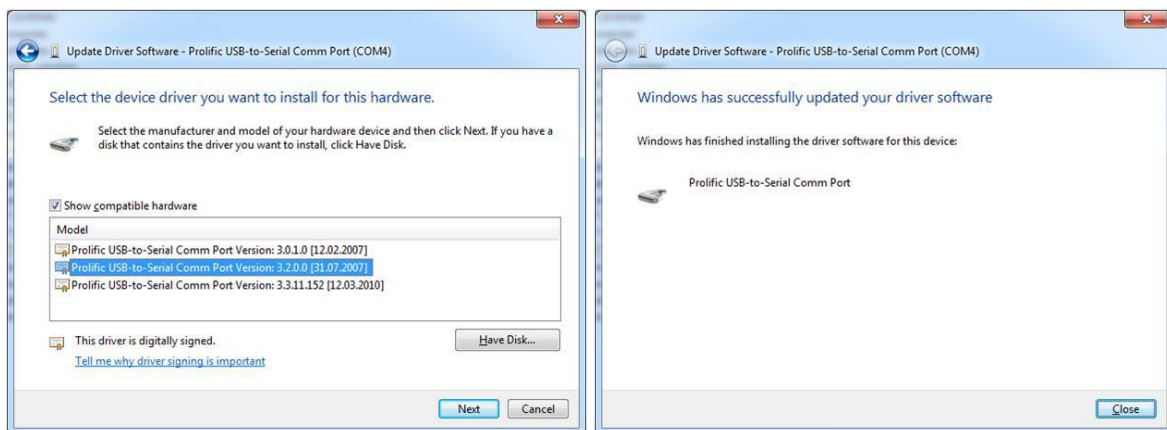


Abbildung 28: Zugriff auf die ältere Version -> Komplette Funktionalität

1.5. Verbindung der Kloehn-Komponenten

Nachdem die Spannungsversorgung der Kloehn-Komponenten über 24 V Gleichspannungsnetzteile universell bereit gestellt wird und die serielle Kommunikation hardwareseitig über eine entsprechendes serielles Kabel mit dem PC verbunden wurde, ist die Parametrierung auf die Standardschnittstellenparameter zur Inbetriebnahme notwendig.

Standardparameter 9600 Baud
8 data bits
1 stop bit
No parity
No flow control

Abbildung 29: Standardparameter der Kloehn Komponenten

Nachdem diese Standardeinstellung vorgenommen worden sind, kann der grundlegende Gerätetest erfolgen. Hierzu wird mit den seitens Kloehn zur Verfügung gestellten Kloehn-Control die Kontaktierung zu den einzelnen Geräten durchgeführt.

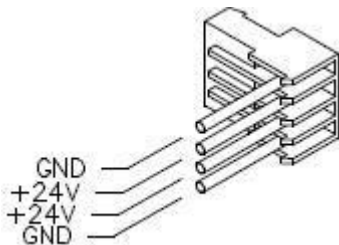


Abbildung 30: Pinbelegung des Versorgungsstecker



Abbildung 31: Intellect V6

Bei den Rotationsventilen ist hierbei die innerhalb des μ -Prozessors hinterlegte Anzahl der Ventilports zu kontrollieren, da durch die Verwendung unterschiedlicher Ventilköpfe von 3-Wege-Ventilen bis 8-Wege-Ventile alle Kombinationen auf der gleichen Grundplattform adaptierbar sind.

Die jeweiligen Geräteadressen werden mit Hilfe eines Adressswitches zwischen 1 und 16 eingestellt.

Bei den Spritzen kommt noch erweiternd hinzu, das die Grundpositionen, auf die das Initialisierungskommando die Spritze verfährt, hardwaretechnisch im Speicher abgelegt werden muss.



Abbildung 32: Funktionstest des Bedienprogrammes

Durch die Verwendung der Autodetektion erhält man bei einem vollständig aufgebauten System das nachfolgende Bild aus dem alle 6 aktuell verwendeten Geräte am Kloehn-Bus hervorgehen.

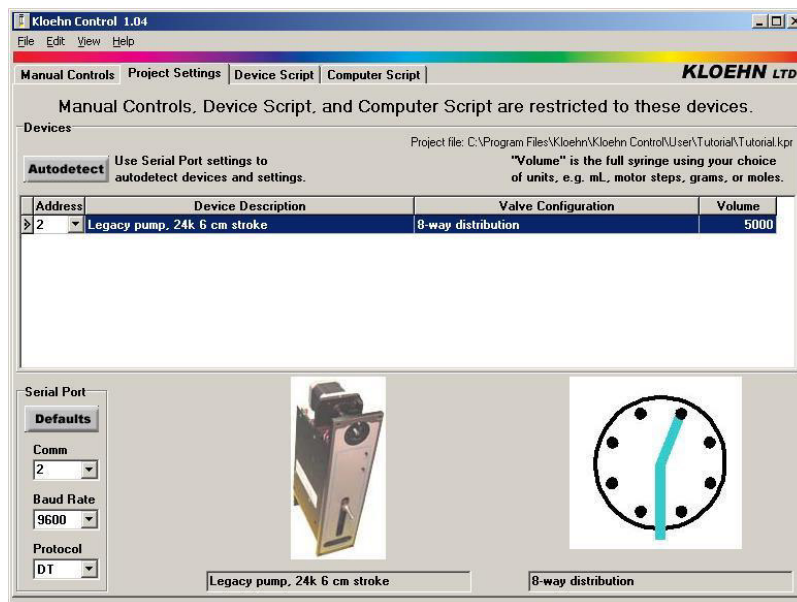


Abbildung 33: Direktzugriff über Kloehn Control

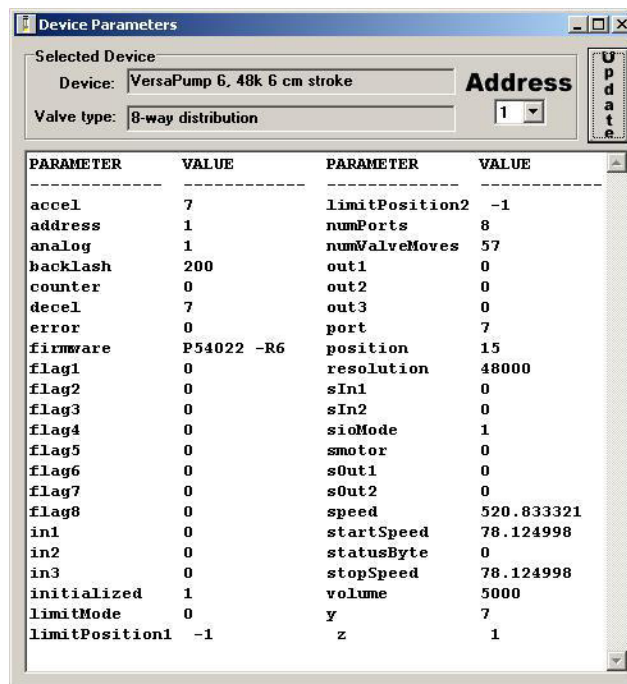


Abbildung 34: Auslesen der aktuellen Geräteparameter



Abbildung 35: Initialisierungsprobleme der Spritzenpumpen

Wie man in Abbildung 36 sieht, wird aufgrund der Autodetektion der gesamte zur Verfügung stehende Adressbereich von 1 – 16 (Hex 1 – F) gescannt und wenn unter der entsprechenden Adresse ein Gerät erkannt wurde, wird dessen Funktionalität in einem zweiten Schritt detaillierter ausgelesen und angezeigt.

Somit befindet sich in diesem Fall an der Adresse 1 und 2 jeweils eine Kloehn-Versa-Pumpe mit 48000 Schritten Auflösung bei einem Kolbenweg von max. 6 cm. Hier kommen aktuell 5 ml Spritzen zum Einsatz, aus der sich dann rein rechnerisch eine Dosiergenauigkeit von ~100nl ergibt.

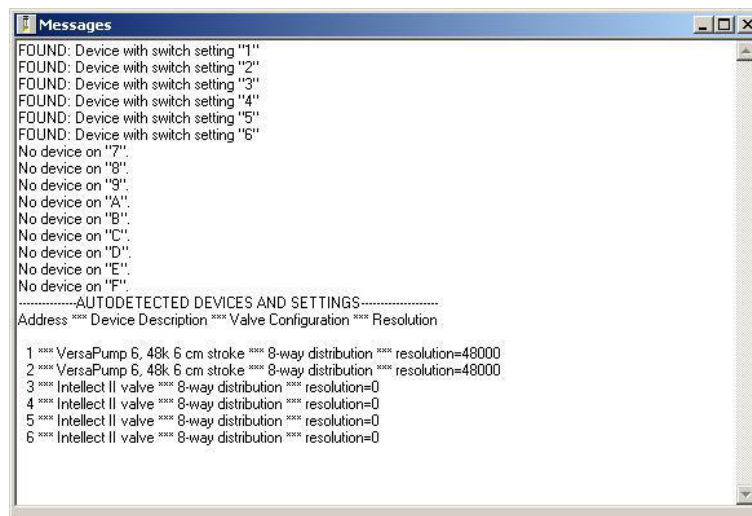


Abbildung 36: Initialisierung und Rückmeldungen der Gerätesuche

Die erweiterte Funktionalität der Spritzen kann mit einem der nachfolgenden Testprogramme geprüft werden. Hierzu ist nach der Initialisierung der Rotationskopf auf dem Zulieferungsport 1 positioniert.

Durch die Definition der Geschwindigkeiten kann zwischen Aufnahme und Abgabe individuell lösungsmittelabhängig die Vorschubgeschwindigkeit der Spritze variiert werden. Nach der Aufnahme der entsprechenden Lösungsmittelmenge in die Spritze wird der Ventilkopf in die Ausgabeposition verdreht und kann die entsprechende Menge dann wiederum dosieren.

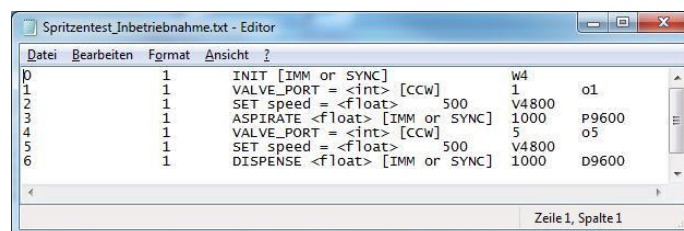
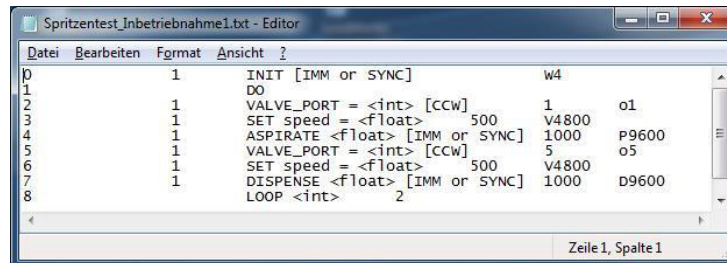


Abbildung 37: Beispielskript eines Testmodules

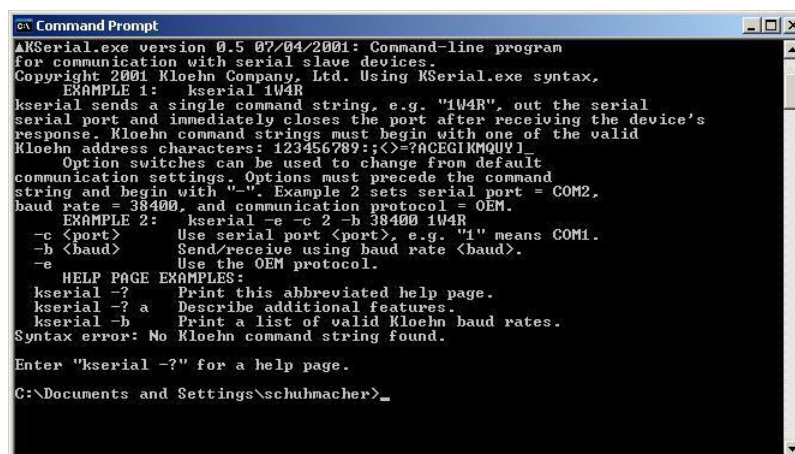
Um die längerfristige Systemstabilität zu testen, was sowohl die Kommunikation, als auch die transportierten Flüssigkeitsmengen angeht, ist der eigentliche Transferbereich der Lösungsmittel auch über eine Schleife mehrfach abarbeitbar.



```
Spritzentest_Inbetriebnahme1.txt - Editor
Datei Bearbeiten Format Ansicht ?
0 1 INIT [IMM or SYNC] w4
1 DO
2 1 VALVE_PORT = <int> [CCw] 1 o1
3 1 SET speed = <float> [CCw] 500 V4800
4 1 ASPIRATE <float> [IMM or SYNC] 1000 P9600
5 1 VALVE_PORT = <int> [CCw] 5 o5
6 1 SET speed = <float> [CCw] 500 V4800
7 1 DISPENSE <float> [IMM or SYNC] 1000 D9600
8 LOOP <int> 2
Zeile 1, Spalte 1
```

Abbildung 38: Schleifenabarbeitung zur Reproduzierbarkeit

Der hardwarenahe Zugriff auf die Kloehn-Devices kann über das Hilfsprogramm K-Seriell.exe als Konsolenprogramm erfolgen. Die auf dieser Schnittstelle bereitgestellten Kommandos können im hinteren Bereich der oberen Ansteuerungsbeispiele entnommen werden.



```
Command Prompt
KSerial.exe version 0.5 07/04/2001: Command-line program
for communication with serial slave devices.
Copyright 2001 Kloehn Company, Ltd. Using KSerial.exe syntax,
EXAMPLE 1: kserial 1W4R
kserial sends a single command string, e.g. "1W4R", out the serial
serial port and immediately closes the port after receiving the device's
response. Kloehn command strings must begin with one of the valid
Kloehn address characters: 123456789:;<>=?@CEILMQUW!
Option switches can be used to change from default
communication settings. Options must precede the command
string and begin with "-". Example 2 sets serial port = COM2,
baud rate = 38400, and communication protocol = OEM.
EXAMPLE 2: kserial -e -c 2 -b 38400 1W4R
-c <port> Use serial port <port>, e.g. "1" means COM1.
-b <baud> Send/receive using baud rate <baud>.
-e Use the OEM protocol.
HELP PAGE EXAMPLES:
kserial -? Print this abbreviated help page.
kserial -? a Describe additional features.
kserial -b Print a list of valid Kloehn baud rates.
Syntax error: No Kloehn command string found.
Enter "kserial -?" for a help page.
C:\Documents and Settings\schumacher>_
```

Abbildung 39: Konsoleninformation zu KSerial

1.6. VBA Zugriff auf die Kloehngeräte

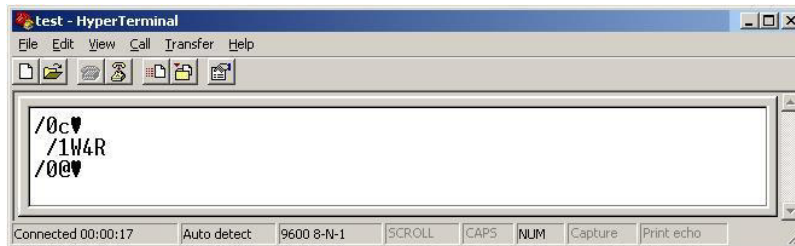


Abbildung 40: Kommunikation über HyperTerminal

Wie in diesem Beispiel gezeigt, ist der Textstring für gültig befunden worden, jedoch das Gerät immer noch in der Abarbeitung des dazugehörigen Befehls. Wohingegen in dem nachfolgenden Fall bei der Abfrage und der Position auch das Gerät sich in einem Zustand der Abarbeitungsbereitschaft befindet.



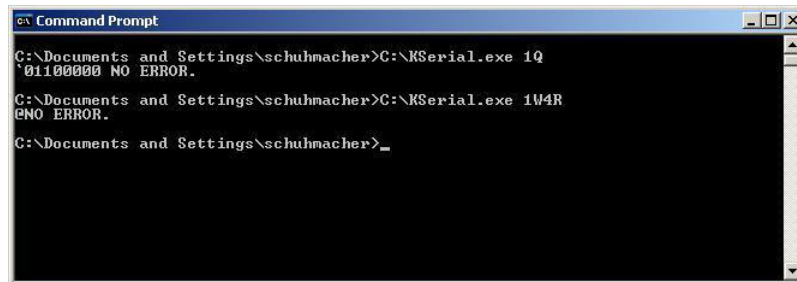
Abbildung 41: Initialisierungsstring



Abbildung 42: Statusabfrage

Die gleichen Anfragen an die jeweiligen Geräte sind auch auf Basis eines DOS Prompts realisierbar und führen zu den gleichen Ausgaben. Mit Hilfe dieser systemnahen Testvarianten kann auch in einem Fehlerfall somit schrittweise die unterschiedlichen Fehlerquellen hierarchisch ermittelt werden. Da die gesamte Programmierung des

Synthesizers auf Basis von LabVIEW realisiert wurde, bietet sich natürlich auch die schrittweise Funktionsüberprüfung mit NI Bordmitteln an.



```
c:\ Command Prompt
C:\Documents and Settings\schuhmacher>C:\NKSerial.exe 1Q
01100000 NO ERROR.
C:\Documents and Settings\schuhmacher>C:\NKSerial.exe 104R
0NO ERROR.
C:\Documents and Settings\schuhmacher>_
```

Abbildung 43: Direkter Konsolenzugriff

1.7. MAX-Zugriff auf die Kloehngeräte

Hierzu können beispielsweise auch über die COM Schnittstellen des MAX die Gerätestrings an die Kloehn-Komponenten versendet werden. Wie man sieht, kommen jedoch in diesem Fall zusätzlich zu der gesamten Textstruktur des Hyperterminals auch noch die Steuerzeichen für die Interpretation eines Zeilenumbruchs mit zur Anwendung. Nachfolgend gezeigt ist der elektrische Anschluss an die Kloehn-Pumpe, bei der direkt auf das Motherboard integriert, durch einen Eurocart-Stecker die Verbindung hergestellt wird. Den jeweils eingestellten Adresswert zeigt der unten eingestellte Adress-Selektor.

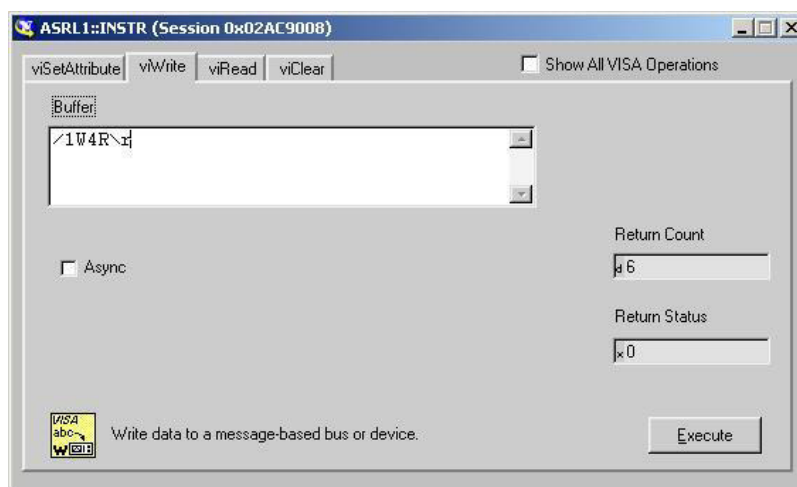


Abbildung 44: direkter Testzugriff aus dem MAX

1.8. Wartung: Testprogramme für Kloehn – Pumpe

Testprogramme für Kloehngeräte	INIT(1) PORT(1,2) WITHDRAW(1,1000) DELAY_SEC(2) DISPENSE(1,1000) FILL(1) DELAY_SEC(2) HOME(1) DELAY_SEC(2)
1ml-Schrittweise Befüllung und danach komplette Entleerung	WITHDRAW(2,1000) DELAY_SEC(2) WITHDRAW(2,1000) DELAY_SEC(2) WITHDRAW(2,1000) DELAY_SEC(2) WITHDRAW(2,1000) DELAY_SEC(2) WITHDRAW(2,1000) DELAY_SEC(2) WITHDRAW(2,1000) DELAY_SEC(2) HOME(2)

Abbildung 45: Testprogramme für Kloehn-Pumpe

Komplette Befüllung danach 1ml- Schrittweise	FILL(2)
Entleerung	DELAY_SEC(2)
	DISPENSE(2,1000)
	DELAY_SEC(2)
	DISPENSE(2,1000)
	DELAY_SEC(2)
	DISPENSE(2,1000)
	DELAY_SEC(2)
	DISPENSE(2,1000)
	DELAY_SEC(2)
	DISPENSE(2,1000)
	DELAY_SEC(2)
	DISPENSE(2,1000)
	DELAY_SEC(2)
Pumpentest an Adresse1	INIT(1)
	PORT(1,2)
	WITHDRAW(1,1000)
	DELAY_SEC(2)
	DISPENSE(1,1000)
	FILL(1)
	DELAY_SEC(2)
	HOME(1)
	DELAY_SEC(2)
Pumpentest an Adresse2	INIT(2)
	PORT(2,2)
	WITHDRAW(2,1000)
	DELAY_SEC(2)
	DISPENSE(2,1000)
	FILL(2)
	DELAY_SEC(2)
	HOME(2)
	DELAY_SEC(2)

Abbildung 45: Testprogramme für Kloehn-Pumpe

1.9. Konfiguration der Rotationsventile

In dem jeweiligen Auslieferungszustand der Rotationsventil beinhaltenden Kloehn-Geräte, ist die Anzahl der möglichen Ventilstellungen durch entsprechende Programmierung des Festspeichers hinterlegbar. Die Abfrage der im Augenblick existierenden Einstellungen kann mit ~V unter Angabe der entsprechenden Adresse abgefragt werden. Wohingegen umgekehrt mit ~V10 ein 8-Wege-Ventil dem Pumpenkopf als zukünftig zu verwendende Information eingeschrieben wird.

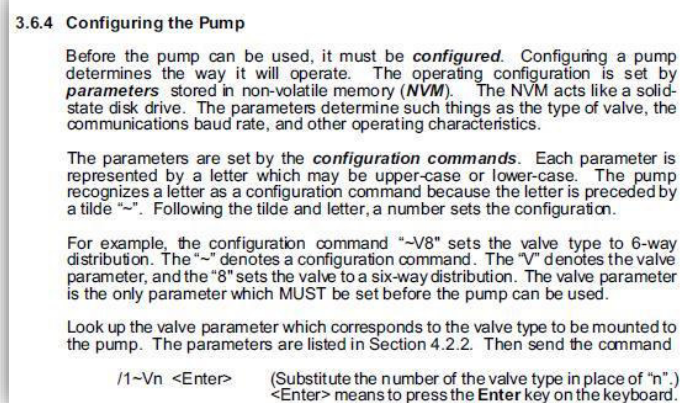


Abbildung 46: Pumpenwerte setzen bei Konfiguration

Da der Festspeicher jedoch von der Anzahl der Schreibzyklen begrenzt ist, sollte diese Aktualisierung nur in begründeten Umbausituationen zur Anwendung kommen. Aus diesem Grund sind solche Funktionalitäten nur über das erweiterte Konfigurations- / Administrationsterminal zugänglich.

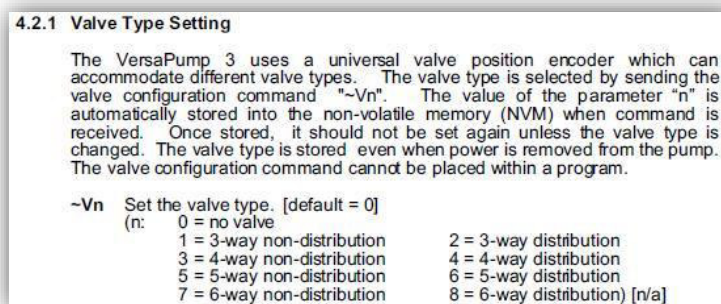


Abbildung 47: Ventiltyp setzen bei Konfiguration

Überprüfung der Geräteparameter	Mit ~V kann die aktuelle Ventildefinition abgefragt werden. ~V10 setzt das 8-Wegeventil 1~Y7 ohne abschließendes Return setzt das Initialisierung Valve auf 7 Mit 1~Y abfragbar Dann mit Y4 anstatt W4 die Initialisierung durchführen
--	---

Abbildung 48: Verschiedene Geräteparameter

1.10. Austausch des Pumpenkopfes

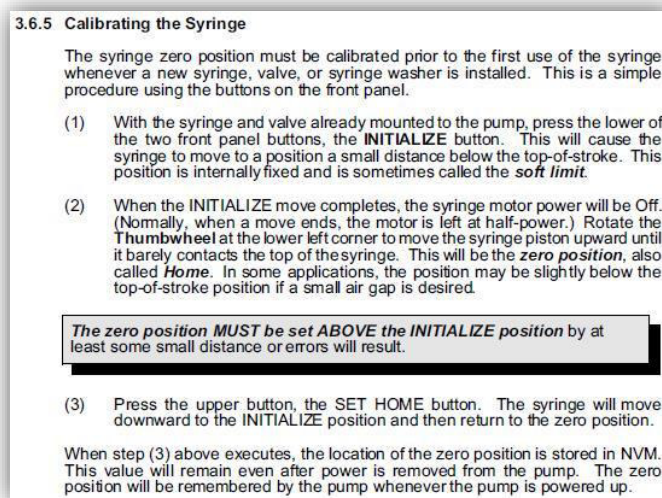


Abbildung 49: Kalibrierung der Spritzen bei Erstbenutzung

Der Austausch des Pumpenkopfes ist im Regelfall nur bei kompletter Verstopfung notwendig. Hier hat sich bisher gezeigt, dass der Ausbau mit nachfolgenden Spülungen durch eine Abfolge von DCM und Methanol sowie entmineralisiertem Wasser die Funktionsbereitschaft wiederherstellen kann. Natürlich sind in einem solchen Fall alle Positionen des Ventilkopfes zu testen, welches am einfachsten durch einen kleinen automatisierten Versuchsaufbau ermöglicht wird. Nach dem abschließenden Dichtigkeitstest, mit der Verwendung von neuen Teflon-Unterlegscheiben an den einzelnen Ventilausgängen kann dann wiederum der Rückbau in den Synthesizer erfolgen.

Beim Ausbau des Pumpenkopfes ist jedoch eine gewisse Reihenfolge in der Prozedurführung zu beachten. Auf der Vorderseite der Kloehn-Geräte sind zwei Taster.

Soll der Pumpenkopf oder die Spritze gewechselt werden, ist der untere der beiden Knöpfe, der Initialisierungsknopf zu betätigen. Hierdurch wird die Spritze an ihre oberste Stempelposition bewegt. Nach Erreichen dieser Position, wird die Versorgungsspannung des Schrittmotors reduziert und somit kann per Hand über das Rändelrad im unteren Bereich der Kloehn-Spritze die mechanische Position des Stempels näher an den Ausgang positioniert werden. Die dadurch abgebildete Position kann durch den oberen Taster (Set Home) nun als zukünftige Nullposition in den Gerätespeicher übernommen werden. Diese Position wird auch dann angefahren wenn die Pumpe nach einem Neustart erstmalig spannungsversorgt wird.

1.11. Tests über Kloehn Control

Zum Testen der zwischen LabVIEW und der Kloehn-Geräte entsprechenden Skripten, bietet es sich an zwischenzeitlich die jeweilige Kloehn-Control-Umgebung als Werkzeug zu verwenden. Wie in dem nachfolgenden Bild gezeigt, sind nun alle Ventilköpfe auf 8-Wege-Distribution angepasst. Auch die Testsequenzen ergeben gleiche Parametrierungen wie man im Datenbereich der Sequenzstrukturen nachvollziehen kann.

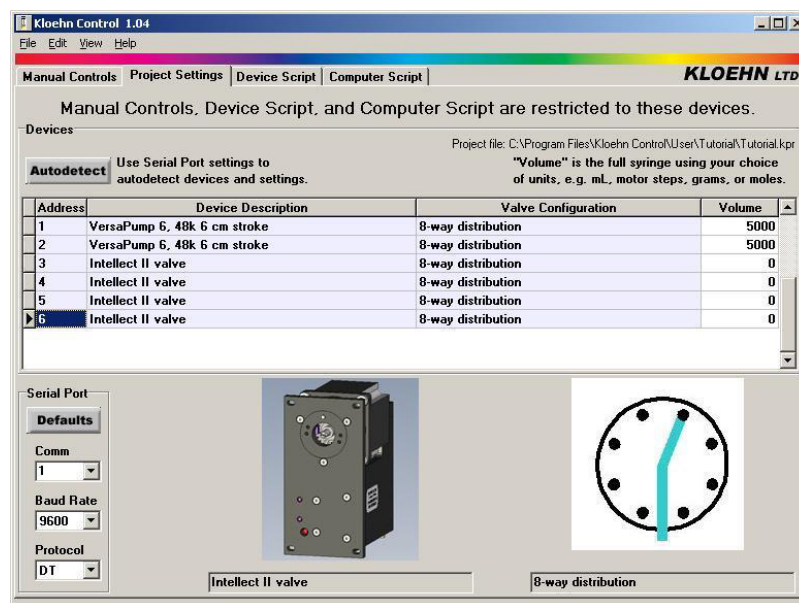


Abbildung 50: Projektparameter der Kloehngeräte

Die jeweiligen Zugriffe auf die unterschiedlichen Geräte, sowie den individuellen Gerätecode sieht man im rechten Bereich des Kloehn-Control-Device Skriptes (siehe Abbildung 51 u. f.). Somit ist ein eindeutiger Abgleich zwischen den aus den LabVIEW Programmen resultierenden Strings mit dem hier vom Hersteller ermittelten Befehlssequenzen durchführbar. Dadurch kann gewährleistet werden, dass keine ungültigen Befehle an die Endgeräte versendet werden.

Nichtsdestotrotz kann es seitens des Zielgerätes zu Fehlerfällen kommen, welches exemplarisch an den Beispielen Fehlerbeseitigung (siehe Seite 96) gezeigt wird.

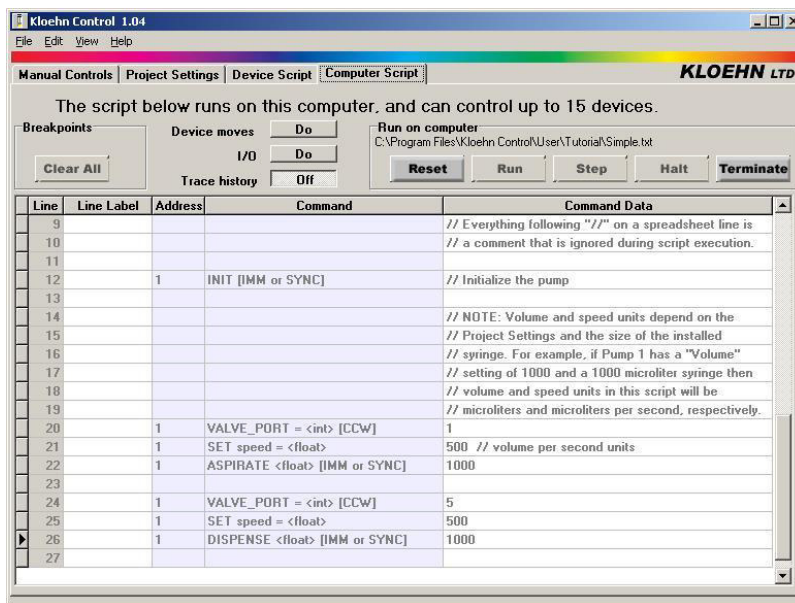


Abbildung 51: Skriptoberfläche der Kloehnhardware

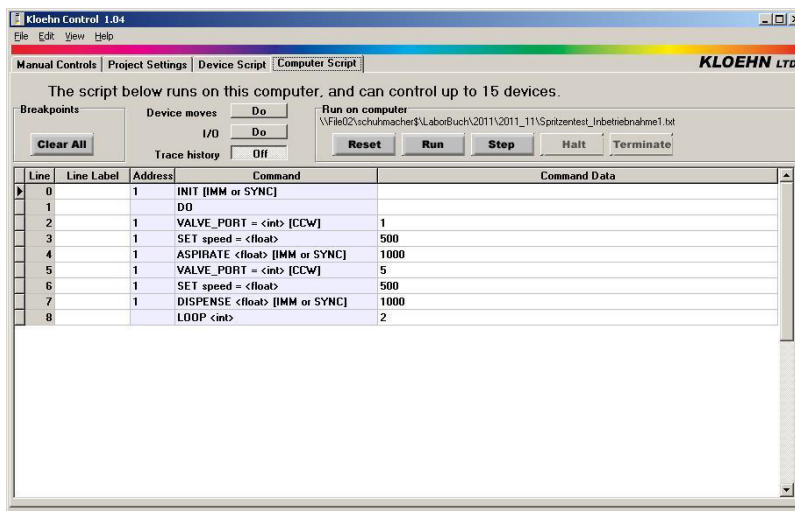


Abbildung 52: Testskript zur Reproduzierbarkeit

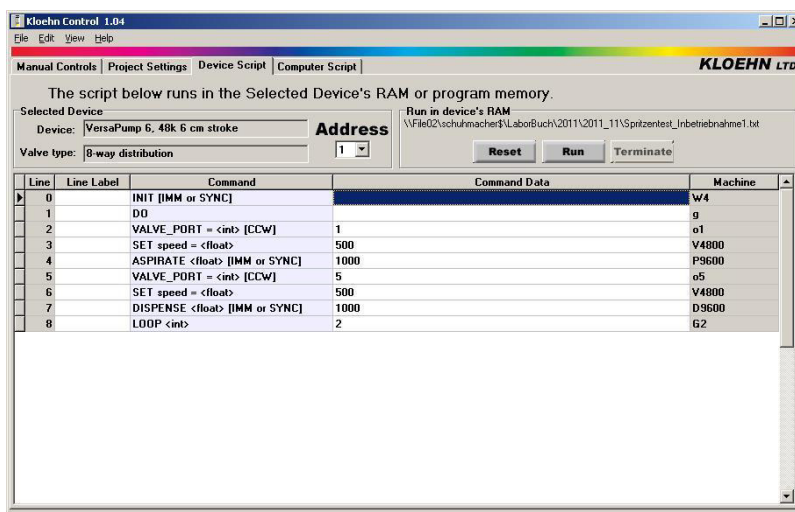


Abbildung 53: Textcode zur Gerätesteuerung

Um die jeweiligen Textstrings an die Geräte über die serielle Schnittstelle verschicken zu können, bietet sich die Verwendung der windowseigenen Bordmittel an. Das von Microsoft bereitgestellte Tool „Hyperterminal“ muss jedoch eigenständig installiert werden, da es nicht mehr Bestandteil der Standardinstallation des Betriebssystems ist.

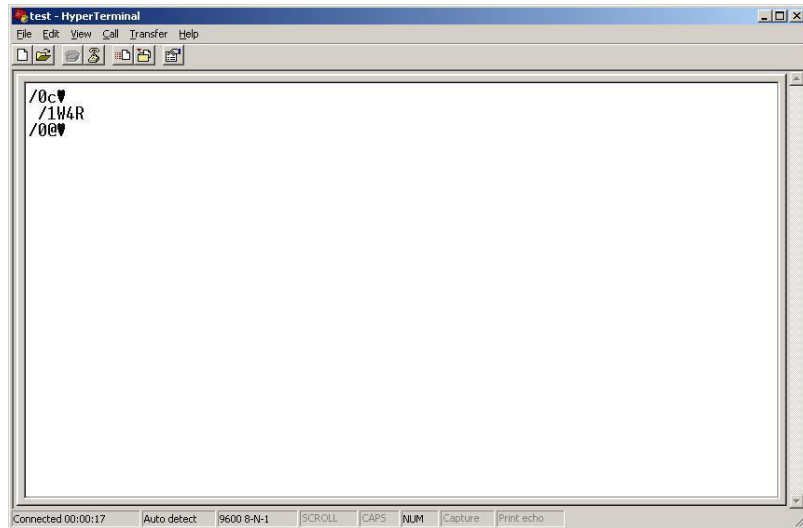


Abbildung 54: Verwenden von HyperTerminal zum Umkonfigurieren

Als Alternative hierzu existiert ein Visual Basic (VB) Programm namens „Pump Commp“, mit dem die gleichen Textstrings ohne vorangestellten Schrägstrich und Adresse an die Geräte versendet werden können. Die Geräteadresse wird hierbei über ein Selektionsfeld zur Auswahl gebracht und die entsprechenden Schnittstellenparameter können über das Setup individuell vorgegeben werden. Die rückgemeldeten Steuerzeichen, in diesem Beispiel der @ Klammeraffe wird direkt interpretiert und vom Status her visualisiert.

Die Kloehngeräte werden mit folgendem Befehlsstring konfiguriert: 1~V10

3.1.1 Valve Type Setting

The Intellect II uses a universal valve position encoder which can accommodate different valve types. The valve type is selected by sending the valve configuration command "~Vn". The value of the parameter "n" is automatically stored into the non-volatile memory (NVM) when command is received. Once stored, it should not be set again unless the valve type is changed. The valve type is stored even when power is removed from the pump. The valve configuration command cannot be placed within a program.

~Vn Set the valve type. [default = 0]
 (n: 0 = no valve
 1 = 3-way non-distribution 2 = 3-way distribution
 3 = 4-way non-distribution 4 = 4-way distribution
 5 = 5-way non-distribution 6 = 5-way distribution
 7 = 6-way non-distribution 8 = 6-way distribution
 9 = 8-way non-distribution 10 = 8-way distribution
 11 = 12-way distribution

Abbildung 55: Konfigurationscode zur Typendefinition

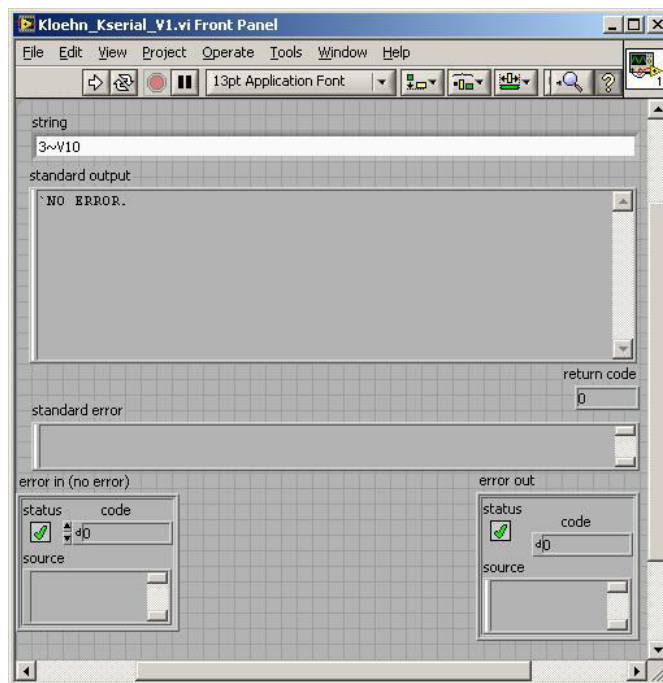


Abbildung 56: Konfigurationstest der Komponenten



Abbildung 57: Aktualisierung der Geräteparameter

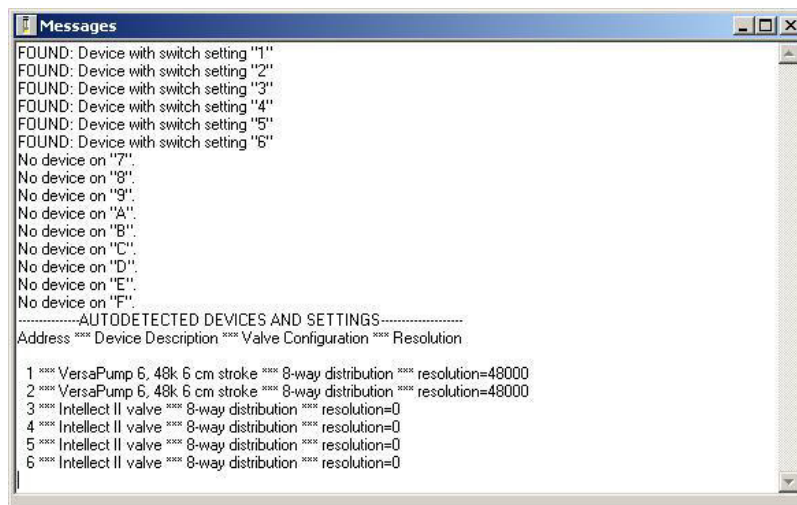


Abbildung 58: Gesamtliste der aktuell verwendeten Geräte

1.12. Fraktionssammler - Schnittstellenparameter

Die Schnittstellenparameter des Fraktionssammlers wurden mit Hilfe eigens gebauter Sniffer ermittelt, da keine Dokumentation der Altgeräte mehr vorhanden war. Dazu wurde die Kommunikation zwischen dem vorliegenden Bedientool und der existierenden Hardware mit protokolliert und aus der Protokollsequenz die Befehle zur Ansteuerung und das erwartete Feedback ermittelt.

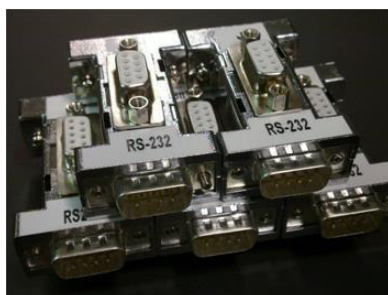


Abbildung 59: Snifferhardware, Eigenbau zur Parameterprotokollierung

Die Parameterdaten der seriellen Schnittstelle lassen sich direkt aus dem Parametrierungsdialog der Bediensoftware bei funktionierender Kommunikation ermitteln. Wobei eine ganze Reihe der im Hintergrund normalerweise auch zu definierenden Parameter nicht offen gelegt sind, jedoch mit einigen Fehlversuchen ermittelt werden konnten.

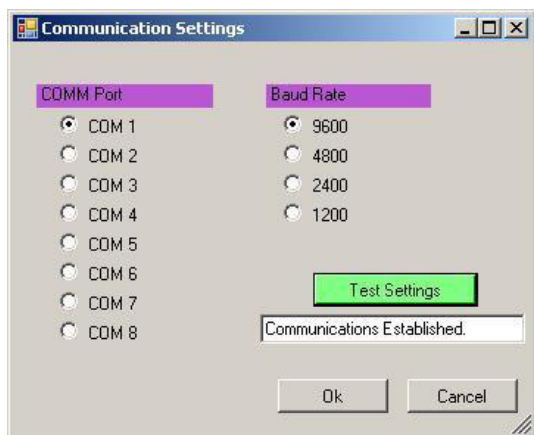


Abbildung 60: Parametrierungsdialog der Bediensoftware für den Fraktionssammler

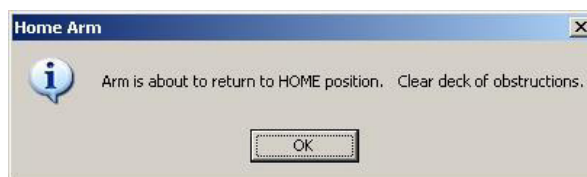


Abbildung 61: Rückmeldung auf die Initialisierung

Da einzelne Kommandos seitens des PC in mehrere Anweisungen für den Fraktionssammler umgesetzt werden, ist an der lokalen Anzeige des Fraktionssammlers eine Möglichkeit gegeben, die Plausibilität am Zielsystem zu kontrollieren.

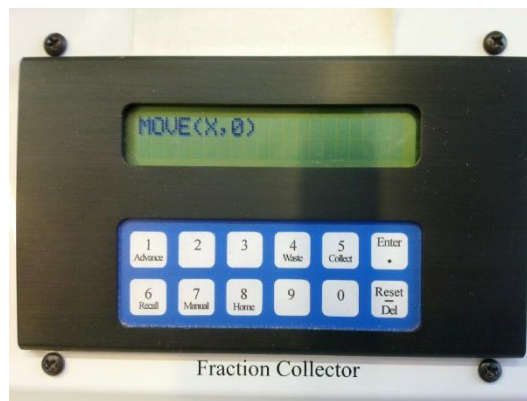


Abbildung 62: Lokale Anzeigen der übergebenen Kommandos

Schnittstellenparameter für COM5, Fraktionssammler:

VI_ATTR_TMO_VALUE = 2000	VI_ATTR_ASRL_DTR_STATE = 1
VI_ATTR_MAX_QUEUE_LENGTH = 50	VI_ATTR_ASRL_RI_STATE = 0
VI_ATTR_SEND_END_EN = VI_TRUE	VI_ATTR_ASRL_RTS_STATE = 1
VI_ATTR_TERMCHAR = 0x0A	VI_ATTR_ASRL_XON_CHAR = 0x11
VI_ATTR_TERMCHAR_EN = VI_FALSE	VI_ATTR_ASRL_XOFF_CHAR = 0x13
VI_ATTR_IO_PROT = 1	VI_ATTR_ASRL_REPLACE_CHAR = 0x00
VI_ATTR_SUPPRESS_END_EN = VI_FALSE	VI_ATTR_DMA_ALLOW_EN = VI_FALSE
VI_ATTR_ASRL_BAUD = 9600	VI_ATTR_FILE_APPEND_EN = VI_FALSE
VI_ATTR_ASRL_DATA_BITS = 8	VI_ATTR_ASRL_DISCARD_NULL = VI_FALSE
VI_ATTR_ASRL_PARITY = 0	VI_ATTR_ASRL_BREAK_STATE = 0
VI_ATTR_ASRL_STOP_BITS = 10	VI_ATTR_ASRL_BREAK_LEN = 250
VI_ATTR_ASRL_FLOW_CNTRL = 0	VI_ATTR_ASRL_ALLOW_TRANSMIT =
VI_ATTR_ASRL_END_IN = 2	VI_TRUE
VI_ATTR_ASRL_END_OUT = 0	VI_ATTR_ASRL_WIRE_MODE = 128
VI_ATTR_ASRL_DCD_STATE = 0	

Mit den so ermittelten Parametern, ließ sich nun eine direkte Ansteuerung aus dem MAX heraus testen. Wie im nachfolgenden Terminal gezeigt, erfolgt das Initialisieren auf die Home Position durch zwei eigenständige Strings, einmal in x-Richtung und einmal in y-Richtung.

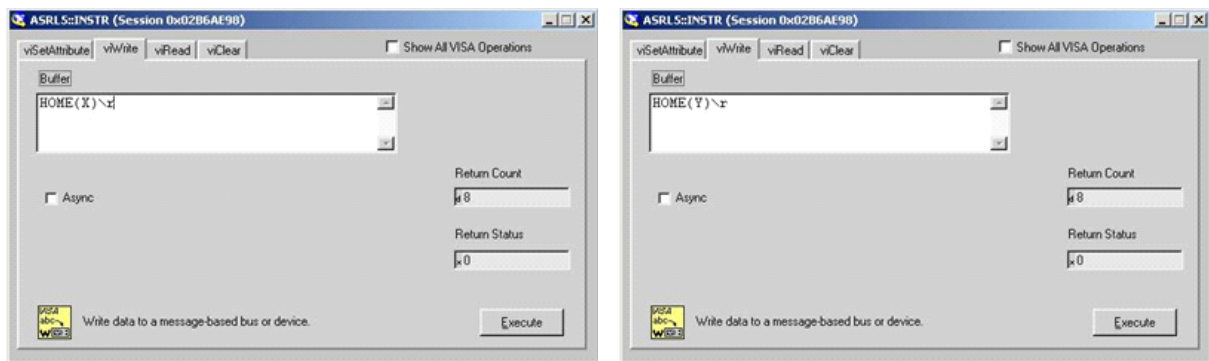


Abbildung 63: Kommunikation über MAX

1.13. Testprogramme Fraktionssammler

FRACTINIT() DELAY_SEC(2) FRACTHOME() DELAY_SEC(2) ADVANCE() DELAY_SEC(2) ADVANCE() DELAY_SEC(2) FRACTSTOP()	FRACTINIT() LOOP_START(20) ADVANCE() DELAY_SEC(3) LOOP_END()
FRACTINIT() LOOP_START(20) ADVANCE() DELAY_SEC(3) LOOP_END()	FRACTINIT() DELAY_SEC(2) FRACTHOME() DELAY_SEC(2) ADVANCE() DELAY_SEC(2) ADVANCE() DELAY_SEC(2) FRACTSTOP()
FRACTINIT() LOOP_START(3) DELAY_SEC(5) ADVANCE() LOOP_END() FRACTSTOP()	FRACTINIT() LOOP_START(11) DELAY_SEC(2) ADVANCE() LOOP_END() FRACTSTOP()

Abbildung 64: Verschiedene durchgeführte Testprogramme für den Fraktionssammler, Anweisungsliste

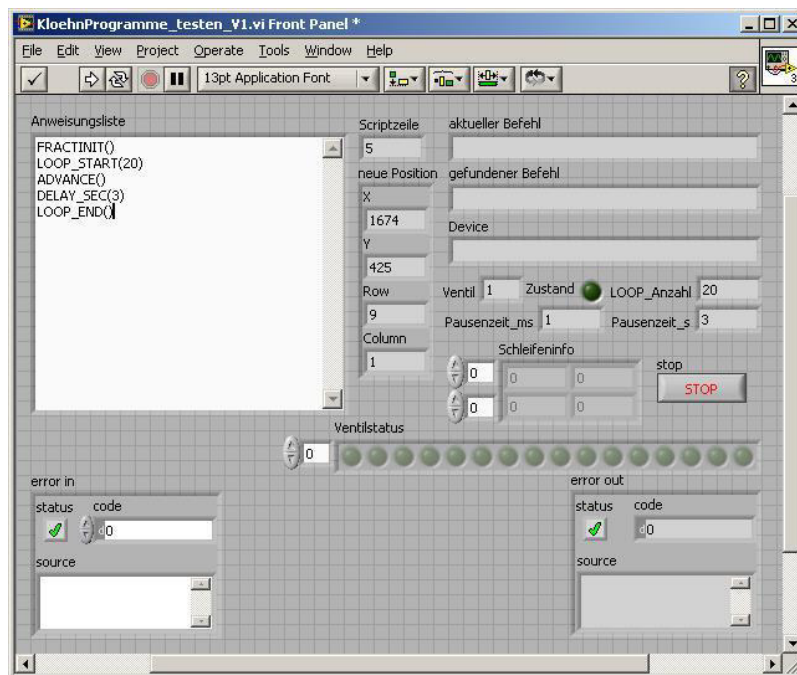


Abbildung 65: Fraktionssammler Schleifentest

Alle durchgeführten Tests und Positionskontrollen, sowohl bei kompletter Neuinitialisierung, als auch bei ausführlichen Verfahrensequenzen, zeigten eine 100%ige Funktionalität.

1.14. Temperiereinheit – Julabo – Konfiguration der RS232 Schnittstelle

Individuelle Parametrierungseigenschaften seitens der Schnittstelle gerade bei Wartungsmaßnahmen, die ein herstellerseitiges Eingreifen notwendig machen, werden nicht zurückgesetzt, deshalb werden nachfolgend die Werkseinstellungen der seriellen Schnittstelle als empfohlene Parameter verwendet. Erst wenn mit zunehmender Komplexität in den gewünschten Temperaturprofilen diese Werte der Optimierung bedürfen, können - über das Konfigurationsmenü -die entsprechenden Einstellungen getätigt werden. Somit wird auf den nachfolgend gezeigten Parametrierungsdialogen immer von den Werkseinstellungen ausgegangen.

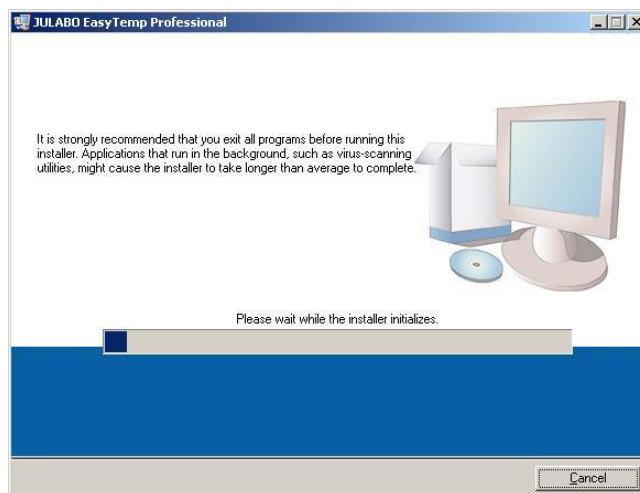



Abbildung 66: Test Software Julabo

11. Fernsteuerbetrieb, Laborautomatisierung

11.1. Vorbereitung zur Fernsteuerung



Die Schnittstelle im Untermenü „Konfiguration“ unter Menüpunkt „Sollwert“ einstellen (siehe Seite 43).

Die meist einmalige Einstellung der Schnittstellenparameter erfolgt am Gerät im Untermenü „Schnittstelle“.
(Menüpunkte anwählen und einstellen, siehe Seite 56)

Werkseinstellungen der seriellen Schnittstelle:

<u>RS232</u>	
BAUDRATE	4800 Baud
PARITY	even parity
HANDSHAKE	Protokoll RTS/CTS (Hardwarehandshake)
Datenbits	7
Stopbit	1

Abbildung 67: Konfiguration der Hardware im Fernsteuerbetrieb

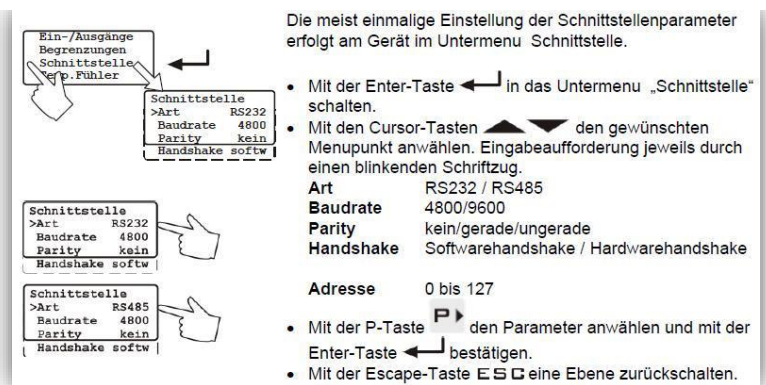


Abbildung 68: Umkonfiguration der Hardware

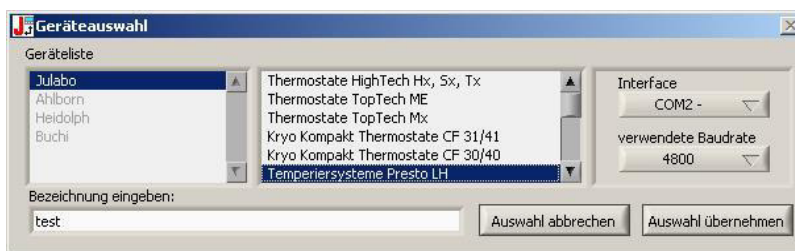


Abbildung 69: Geräteauswahl

Nach der grundlegenden Definition des Temperierungssystems und den wichtigen Interface – Parametern wird das Gerät initialisiert und steht mit seinen Statusinformationen über die serielle Schnittstelle im Zugriff.

Die jeweiligen Temperaturvorgaben können über das Kurvenfenstermenü in Korrelation zur Heiz- / Kühlleistung direkt visualisiert werden. Hieraus lassen sich auch für die späteren Synthesen die jeweiligen Temperierungszeiten der gesamten Temperierungsanlage visuell ermitteln.

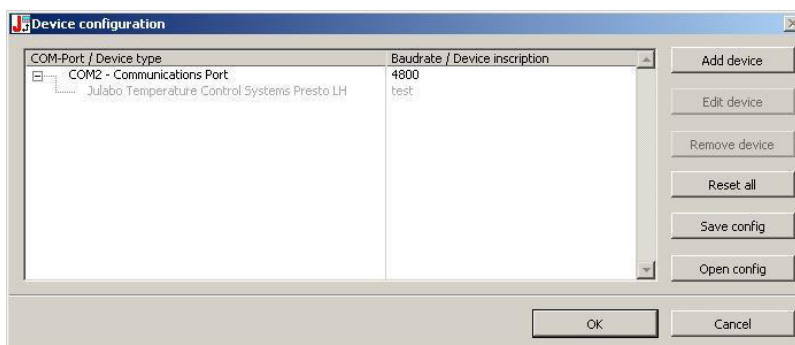


Abbildung 70: Gerätekonfiguration

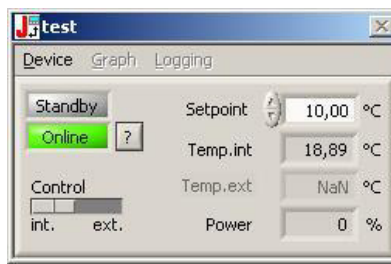


Abbildung 71: Monitoring der Statusinformationen

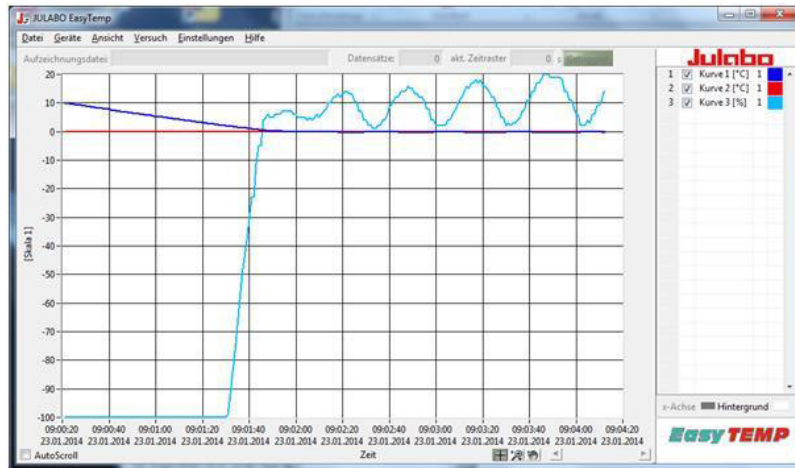


Abbildung 72: Funktionstest

1.15. Test Julabo Hyperterminal Programm

Nachdem die grundlegenden Funktionstests vollständig und erfolgreich abgeschlossen werden konnten, erfolgt auch hier der beispielhafte Zugriff über das Hyperterminal Programm. Die nachfolgende Zusammenstellung, der hierfür notwendigen Parametrierung, zeigt wie eine Funktionalitätsprüfung erfolgen kann und ist als Hilfsinformation zum strukturierten Testen im Fehlerfall heranzuziehen. Der Test im Hyperterminal hat den Vorteil, dass ausschließlich Windows Bordmittel zur Anwendung kommen und somit den rudimentärsten Zugriff auf die serielle Schnittstelle ermöglichen.



Abbildung 73: Hyperterminal und Schnittstellenzuordnung

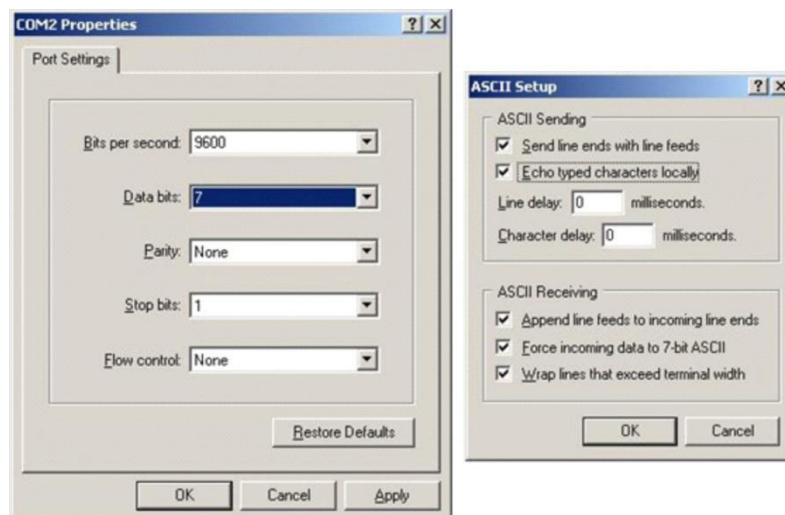


Abbildung 74: Schnittstellenparameter und Zusatzparametrierung

1.16. Test Julabo im MAX

Die nächste Testebene ist der Zugriff auf den MAX in LabVIEW - nach entsprechender Parametrierung - unter Kontrolle der übergebenen Schnittstellenparameter. Diese Beispiele sind für alle zukünftigen Benutzer der Systeme gedacht, um hier einen reproduzierbaren Sachverhalt bereitstellen zu können.

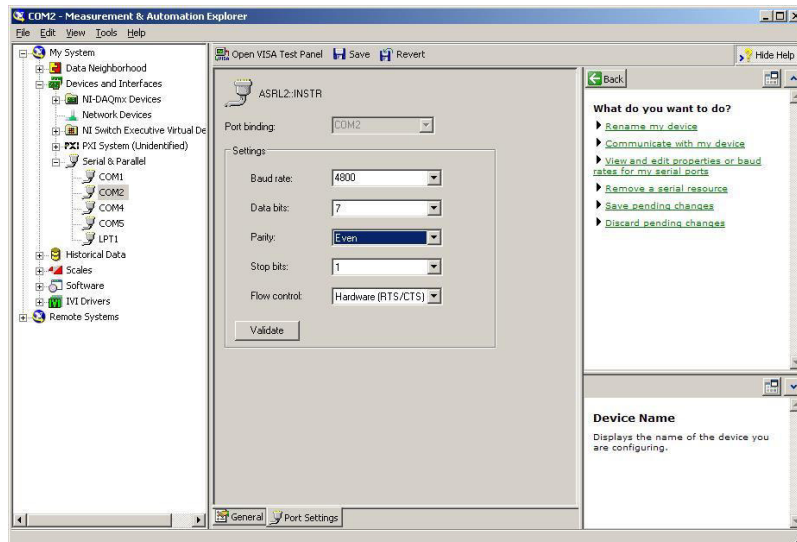


Abbildung 75: Parametrierung der Schnittstelle im MAX



Abbildung 76: Funktionstest

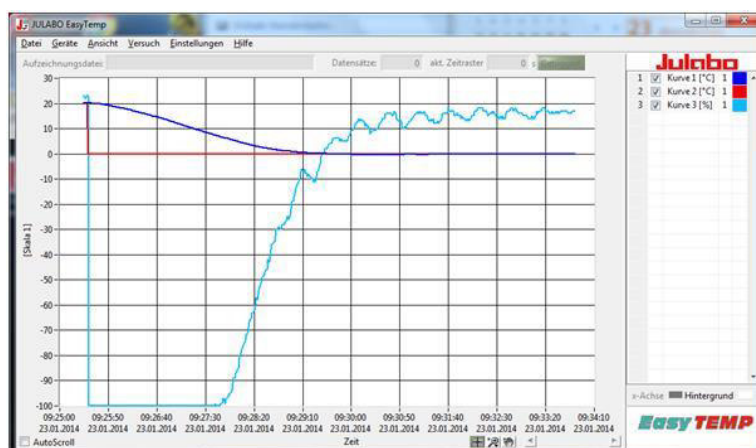


Abbildung 77: Funktionstest 20°C -> 0°C

Nachfolgend gezeigt, ist die Abfrage des angeschlossenen Julabos aus dem Direktterminal des MAX. Über die Registerkarte Write wird hierbei nach der aktuellen Version fragend, der String auf die Schnittstelle geschrieben.

Auf der Registerkarte Read erhält man dann, mit dem daraus resultierenden Antwortstring, die entsprechende Information.

Auch in diesem Beispiel werden die notwendigen Terminatoren zum Wagenrücklauf *Carriage Return* /r oder Zeilenvorschub *New Line* /n/r verwendet. Da diese maskierte Zeichen darstellen, können sie über die entsprechende Escape Sequenz abgebildet werden. Diese Zeichen sollten somit nicht in den übergebenen Strings vorkommen oder müssen gesondert, durch die Deklaration als Text, in ihrer Funktionalität unterbunden werden.

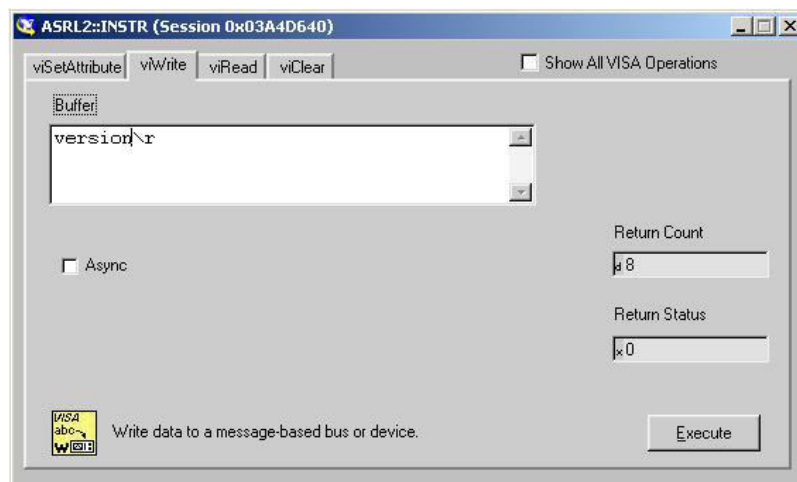


Abbildung 78: Versionsabfrage mit MAX

Neben den rein informativen Abfragen existiert eine Vielzahl von Steuerungsanweisungen mit denen das Temperiergerät in unterschiedliche Modi versetzt werden kann. Diese wurden testweise alle in ein VI funktional umgesetzt, bei denen in Abhängigkeit von dem Sendebefehl die unterschiedlichen Parameterinformationen mit bereitgestellt werden.

Wie nachfolgend dargestellt, stehen die seitens des Gerätes zurückgemeldeten Antwortstrings zur Auswertung bereit.

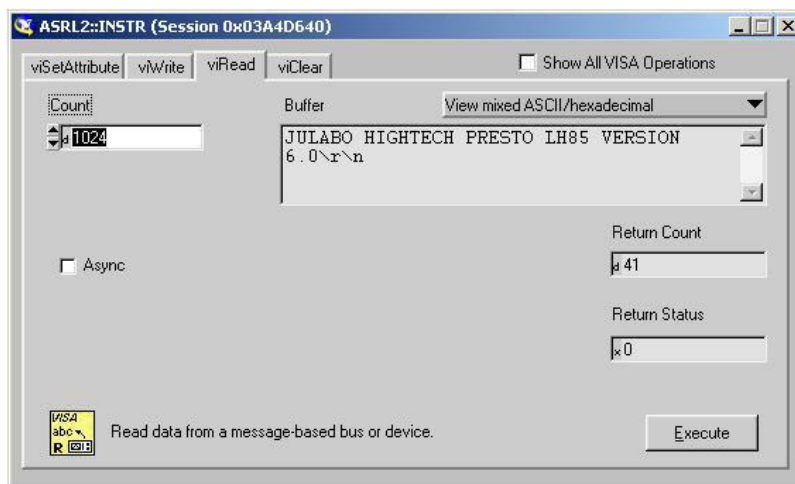


Abbildung 79: Antwort auf Versionsabfrage

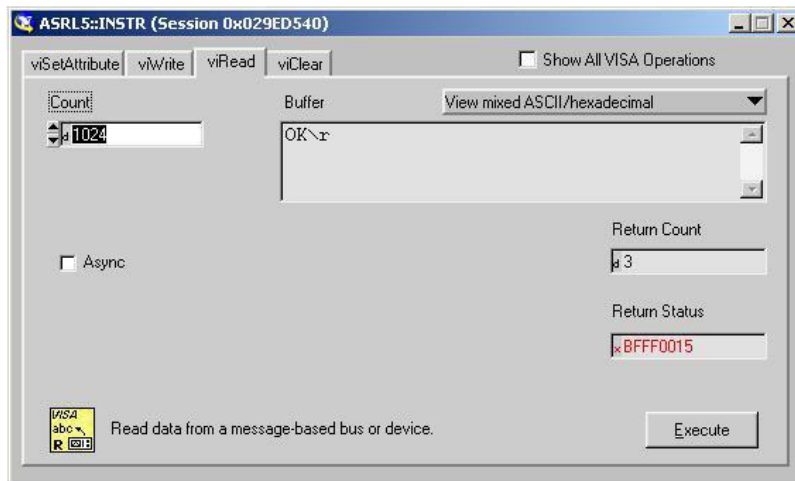


Abbildung 80: Fehlereffekte

1.17. Schnittstellenparameter Temperiereinheit Julabo

Da sich aufgrund der fortschreitenden Versionsentwicklung die Betriebssystemsoftware von Windows XP auf Windows 7 und auch die zur Anwendung kommende LabVIEW Version geändert hat, bilden sich die Schnittstellenparameter wie nachfolgend gezeigt in den neuen Dialogen ab. Für die vorgestellten VIs hat diese Versionsänderung keinerlei Auswirkungen.

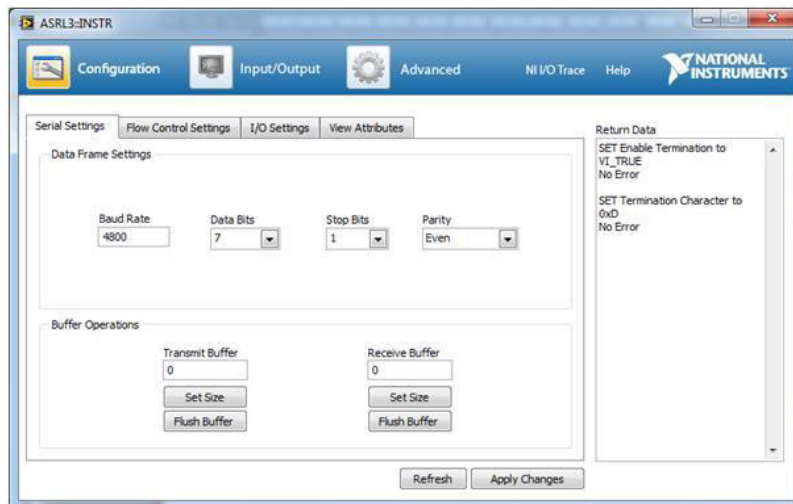


Abbildung 81: Configuration – Serial Settings

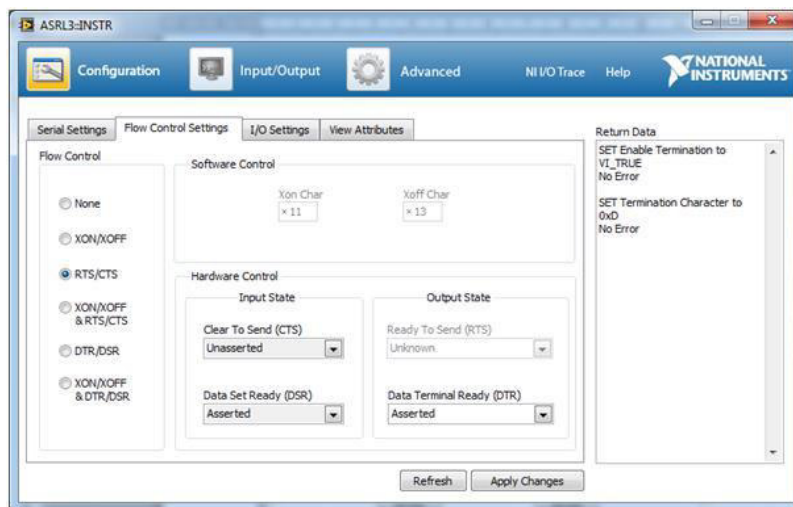


Abbildung 82: Configuration – Flow Control Settings

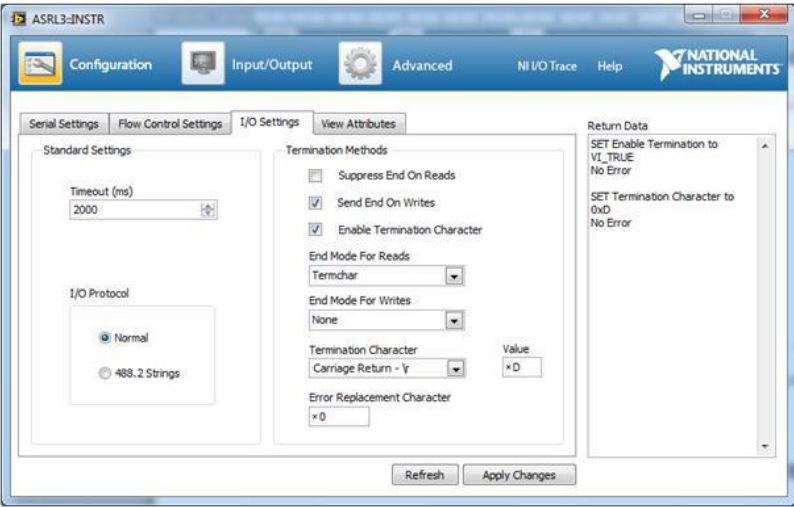


Abbildung 83: Configuration 3 – I/ O Settings

1.18. Testprogramme Temperiereinheit Julabo

Diese Listings können 1:1 innerhalb der Anweisungsliste zur Ausführung gebracht werden.

Steuerung des Chillers	CHILLER(SETPOINT,-15) PTEMPLOW(-14)
Test der Funktionalität von PTEMPHIGH()	CHILLER(SETPOINT,0) PTEMPHIGH(-1)
Chillertest	CHILLERINIT() DELAY_SEC(15) CHILLER(SETPOINT,25) DELAY_SEC(2) PTEMPHIGH(24) PAUSE() CHILLER(SETPOINT,20) PTEMPLOW(22) PAUSE() DELAY_SEC(2) CHILLERSTOP()

Tabelle 1: verschiedene Testprogramme

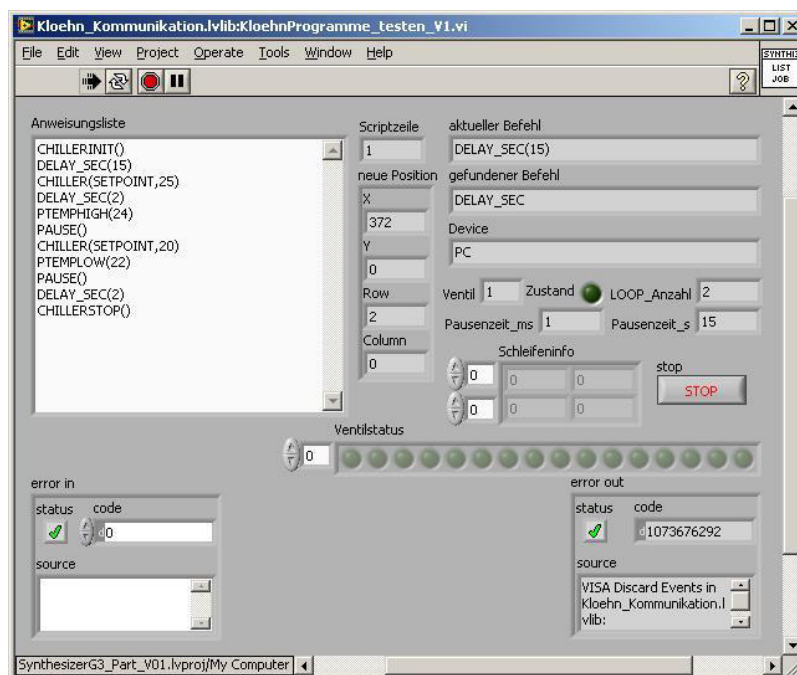


Abbildung 84: Chillertest im Befehlsinterpretier-Test-VI

1.19. Testprogramme Ventilschaltungen

Ventilschaltungen

VALVE(1,ON)	VALVE(9,ON)
VALVE(2,ON)	VALVE(10,ON)
VALVE(3,ON)	VALVE(11,ON)
VALVE(4,ON)	VALVE(12,ON)
DELAY_MS(1000)	DELAY_MS(1000)
VALVE(4,OFF)	VALVE(12,OFF)
VALVE(3,OFF)	VALVE(11,OFF)
VALVE(2,OFF)	VALVE(10,OFF)
VALVE(1,OFF)	VALVE(9,OFF)
DELAY_SEC(1)	DELAY_SEC(1)

VALVE(5,ON)	VALVE(13,ON)
VALVE(6,ON)	VALVE(14,ON)
VALVE(7,ON)	VALVE(15,ON)
VALVE(8,ON)	VALVE(16,ON)
DELAY_MS(1000)	DELAY_MS(1000)
VALVE(8,OFF)	VALVE(16,OFF)
VALVE(7,OFF)	VALVE(15,OFF)
VALVE(6,OFF)	VALVE(14,OFF)
VALVE(5,OFF)	VALVE(13,OFF)
DELAY_SEC(1)	DELAY_SEC(1)

Testprogramm: Spülprozedur:

LOOP_START(6)	VALVE(14,OFF)
VALVE(1,ON)	VALVE(12,ON)
VALVE(16,ON)	VALVE(16,ON)
VALVE(2,ON)	VALVE(11,ON)
DELAY_MS(1000)	DELAY_SEC(5)
VALVE(2,OFF)	VALVE(11,OFF)
VALVE(16,OFF)	VALVE(12,OFF)
VALVE(14,ON)	VALVE(16,OFF)

DELAY_SEC(25)

LOOP_END()

VALVE(1,OFF)

DELAY_MS(100)

1.20. Testprogramme Synthesizer

Reaktionsgefäß leeren (Removing solvent from RV)

VALVE(12,ON)
 VALVE(16,ON)
 VALVE(11,ON)
 DELAY_SEC(5)
 VALVE(11,OFF)
 VALVE(12,OFF)
 VALVE(16,OFF)

Buildingblock Zulieferungstests BB1

PORT(2,1)	BB3
RATE(2,10)	PORT(2,1)
WITHDRAW(2,2500)	RATE(2,10)
PORT(2,3)	WITHDRAW(2,2500)
PORT(3,2)	PORT(2,3)
HOME(2)	PORT(3,4)
	HOME(2)

BB2

PORT(2,1)	BB4
RATE(2,10)	PORT(2,1)
WITHDRAW(2,2500)	RATE(2,10)
PORT(2,3)	WITHDRAW(2,2500)
PORT(3,3)	PORT(2,3)
HOME(2)	PORT(3,5)
	HOME(2)

Waschvorgang als Beispiel

LOOP_START(1)	VALVE(14,ON)	DELAY_SEC(5)
VALVE(1,ON)	DELAY_SEC(25)	VALVE(11,OFF)
VALVE(16,ON)	VALVE(1,OFF)	VALVE(12,OFF)
VALVE(2,ON)	VALVE(14,OFF)	VALVE(16,OFF)
DELAY_MS(1000)	VALVE(12,ON)	LOOP_END()
VALVE(2,OFF)	VALVE(16,ON)	DELAY_MS(100)
VALVE(16,OFF)	VALVE(11,ON)	

Initialisierungs- Modul

FRACTINIT()	INIT(2)
DELAY_SEC(2)	DELAY_SEC(2)
FRACTHOME()	INIT(4)
DELAY_SEC(2)	DELAY_SEC(2)
CHILLERINIT()	INIT(5)
DELAY_SEC(2)	DELAY_SEC(2)
INIT(1)	INIT(6)
DELAY_SEC(2)	DELAY_SEC(2)

Definierte Volumenzulieferung an das Reaktionsgefäß

Test der Zulieferung an das Reaktionsgefäß

PORT(2,1)
FILL(2)
PORT(2,4)
DELAY_SEC(5)
PORT(4,6)
DELAY_SEC(5)
HOME(2)

1.21. Verschaltung des druckgeregelten Massendurchflussmessers (ES)

Die Verschaltung des Bussystems ist nachfolgend gezeigt und die jeweilige Pinbelegung des 9-poligen DSub – Steckers geht daraus hervor.



Abbildung 85: Pin- Belegung

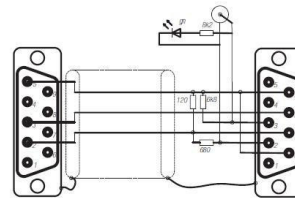


Abbildung 86: Kabelführung und Steckerbelegung

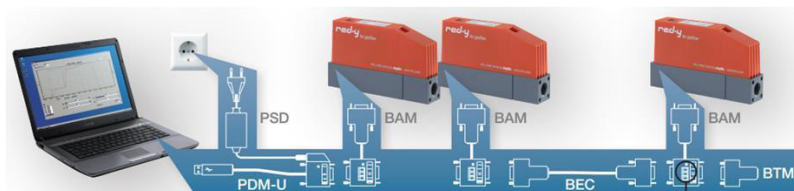


Abbildung 87: Schematische Darstellung der Kommunikation

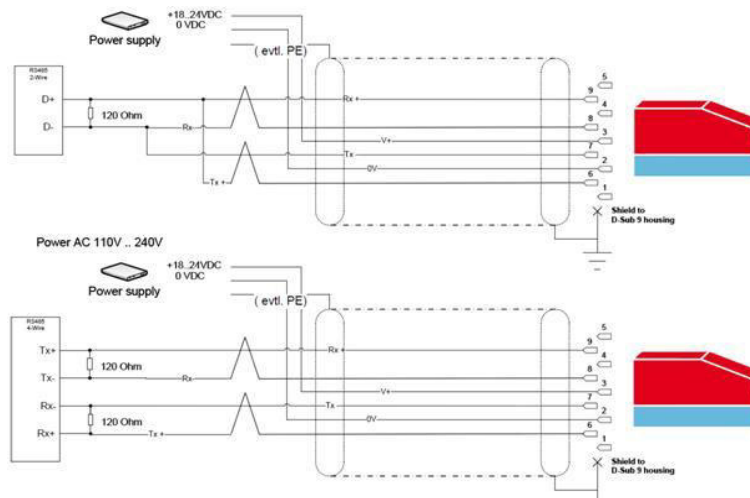


Abbildung 88: Schematische Verkabelung

Seitens des Herstellers steht eine rudimentäre Bediensoftware im Lieferumfang zur Verfügung, mit der die grundlegenden Funktionen der Druckregelung überprüft werden können.

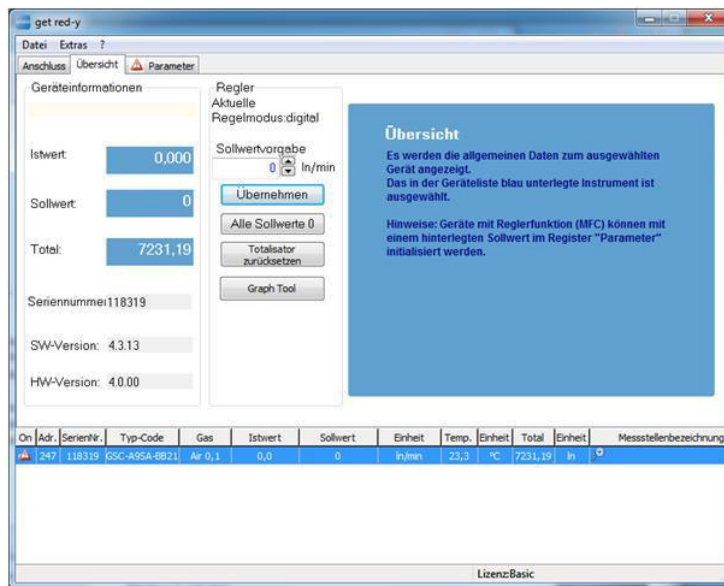


Abbildung 89: get red-y Kommunikation



Abbildung 90: Schnittstelle für red-y Adapter

Nach Parametrierung der Schnittstelleneigenschaften kann über einen Busscanner nach den existierenden Teilnehmern gesucht werden und - wie nachfolgend gezeigt - sind somit acht Massendurchflussregler im Zugriff, die im Geräteverbund das gesamte Schutzgasmanifold abbilden können.

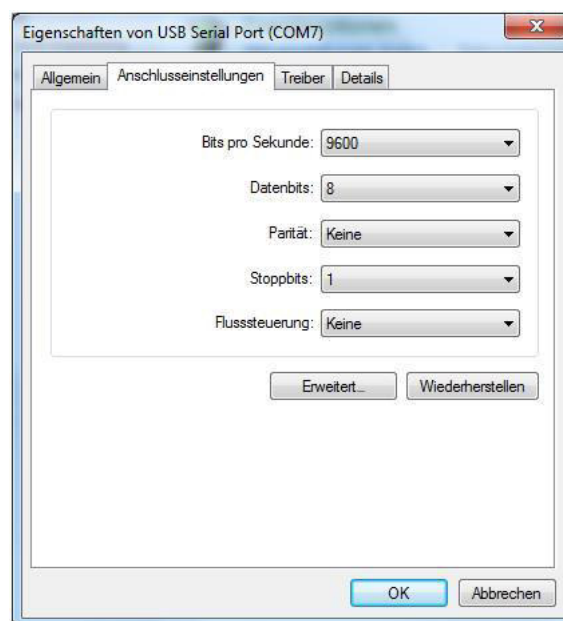


Abbildung 91: Schnittstellenparametrierung

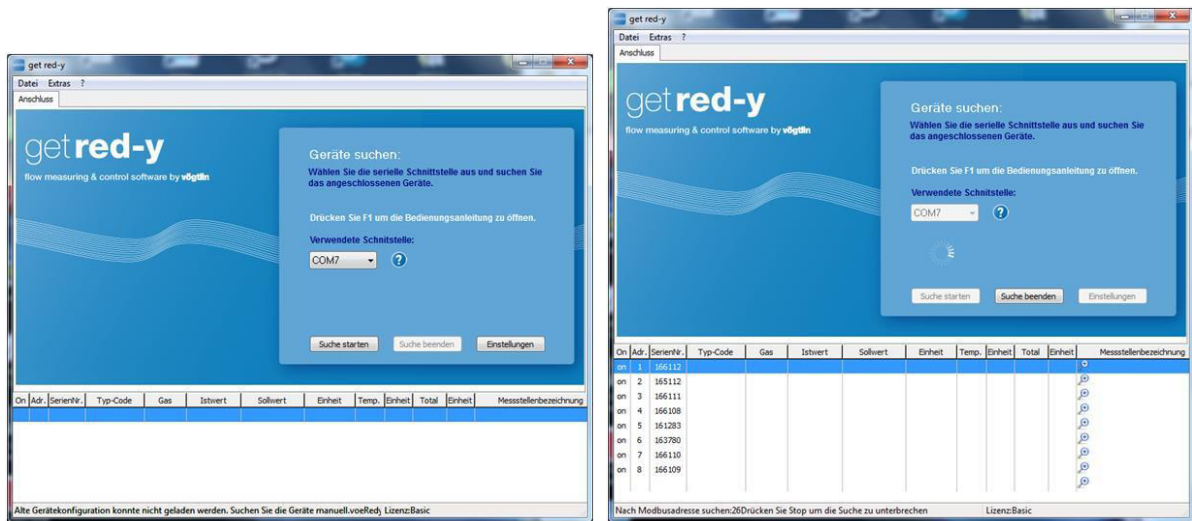


Abbildung 92: Funktionstest elektrisches Manifold und Kommunikationsaufbau

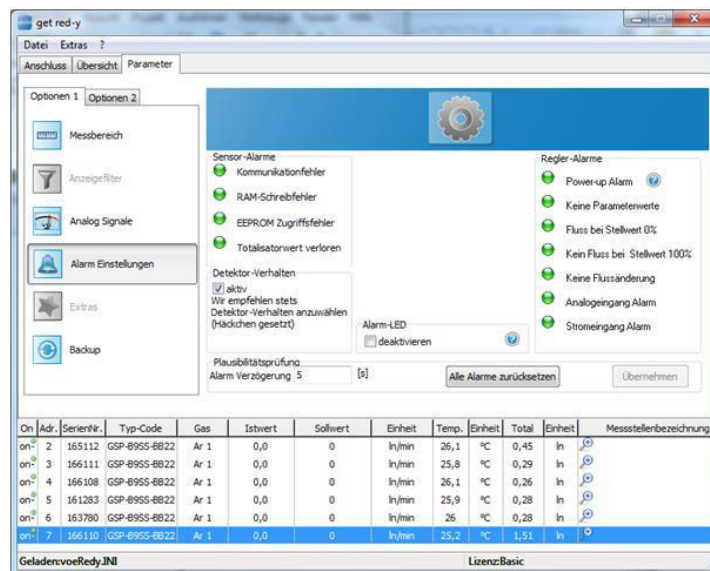


Abbildung 93: Gesamtfunktionalität

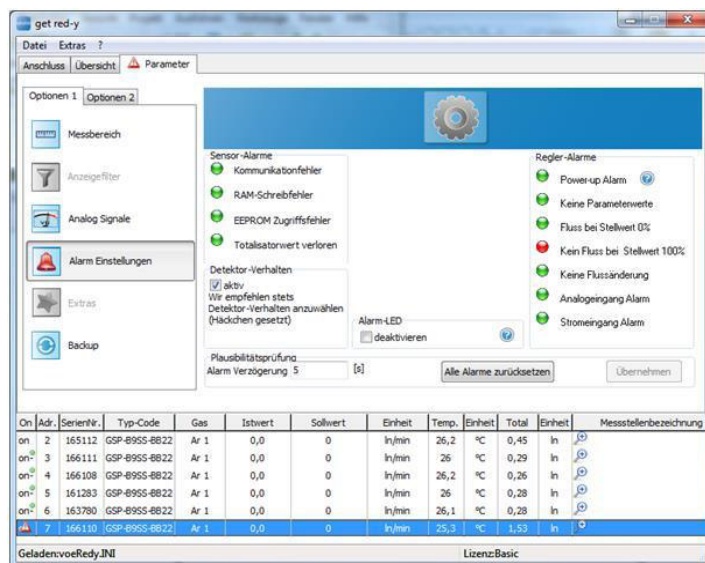


Abbildung 94: Geräterückmeldung

2. LabVIEW VIs

2.1. Befehlsinterpretier: Liste_interpretieren VI

Im folgenden Sub-VI wird die Zeilenanzahl, der in die Anweisungsliste kopierten, historischen Ansteuerungscode ermittelt. Diese Information dient als Abbruchkriterium für die Übersetzung der Textzeilen in funktionalen Steuercode für den neuen Synthesizer.

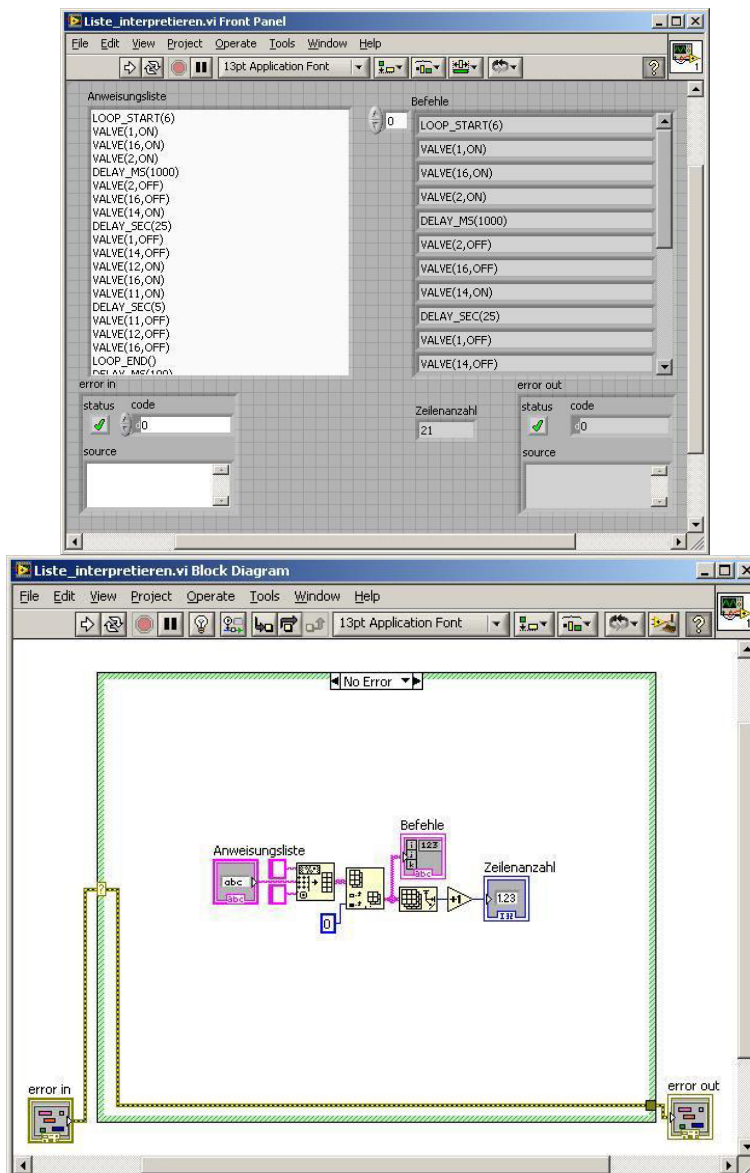


Abbildung 95: Sub-VI zur Ermittlung der Zeilenanzahl, des in die Anweisungsliste kopierten Textes des „alten“ Synthesizerskripts

2.2. Befehlsinterpreter: Gerätezuordnung VI

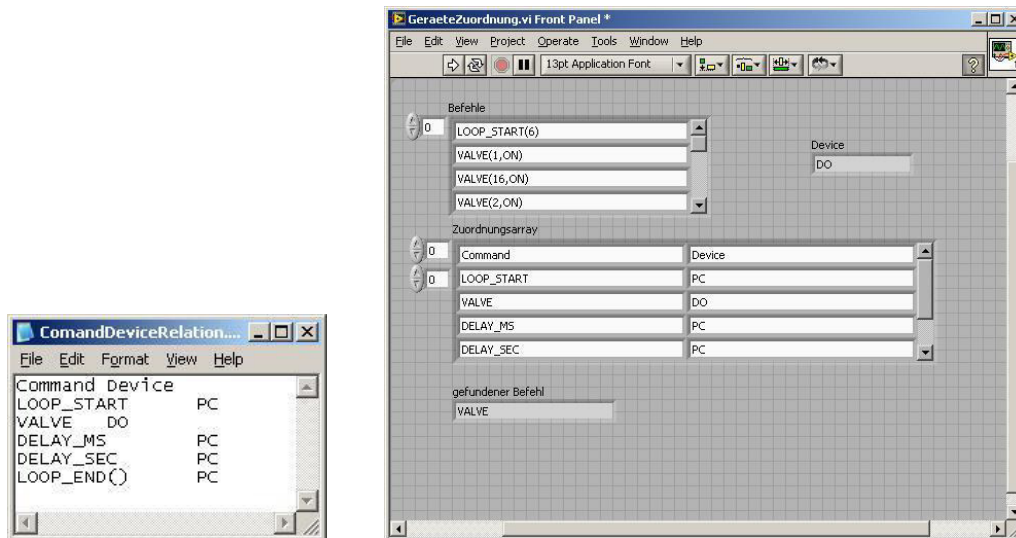


Abbildung 96: links: Zuordnungstabelle zwischen Kommando und Gerät als *.ini Datei, die leicht – ohne Programmierkenntnis - gepflegt werden kann. Diese Liste liefert die Datenbasis, um die Befehle der „Alt-Skripte“ zu den entsprechenden Gerätetreibern / Gerätekomponenten des neuen Systems zuzuweisen. Rechts: Frontpanel: Gerätezuordnung für den Befehlsinterpreter, der aus Kompatibilitätsgründen zum „Altsystem“ integriert ist.

Bei der grundlegenden Auswertung der Zuordnungsliste zwischen den Befehlen und den jeweils ausführenden Geräten wird exemplarisch aufgrund einer Befehlsliste über eine bereitgestellte Zuordnungsdatei die Entscheidung getroffen, in welche Ereignisstruktur der abzuarbeitende Befehl eingekoppelt wird.

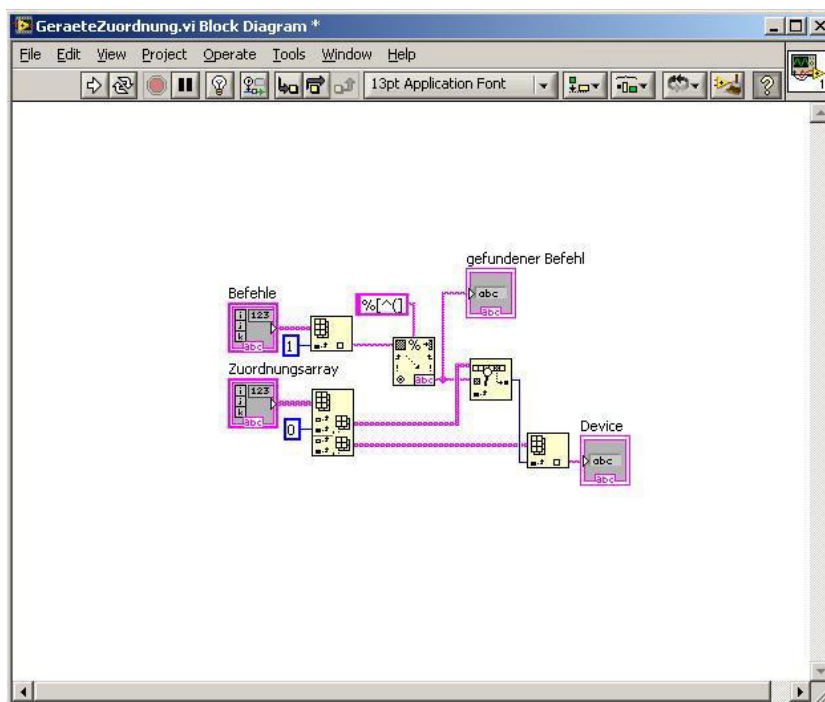


Abbildung 97: Blockdiagramm: Gerätezuordnung

2.3. Befehlsinterpreter: INI () – Spritzenpumpe / Rotationsventil

Dieses VI übernimmt die Initialisierungsprozedur für die Spritzenpumpen (Kloehn). Nach der Adresse und einer Buchstabenkombination, werden die Parameter bereitgestellt, die in den meisten Fällen mit einem Anschlusszeichen versehen an das Zielgerät versendet werden. Dieses wiederum bestätigt die Erkennung des Textstrings oder gibt über die Rückgabe seinen aktuellen Status bekannt. Neben den Funktionstests für die Initialisierung der Geräte mit 1W4R werden Test- VI's für den direkten Zugriff auf die absolute Position der Spritzenpumpe realisiert und getestet.

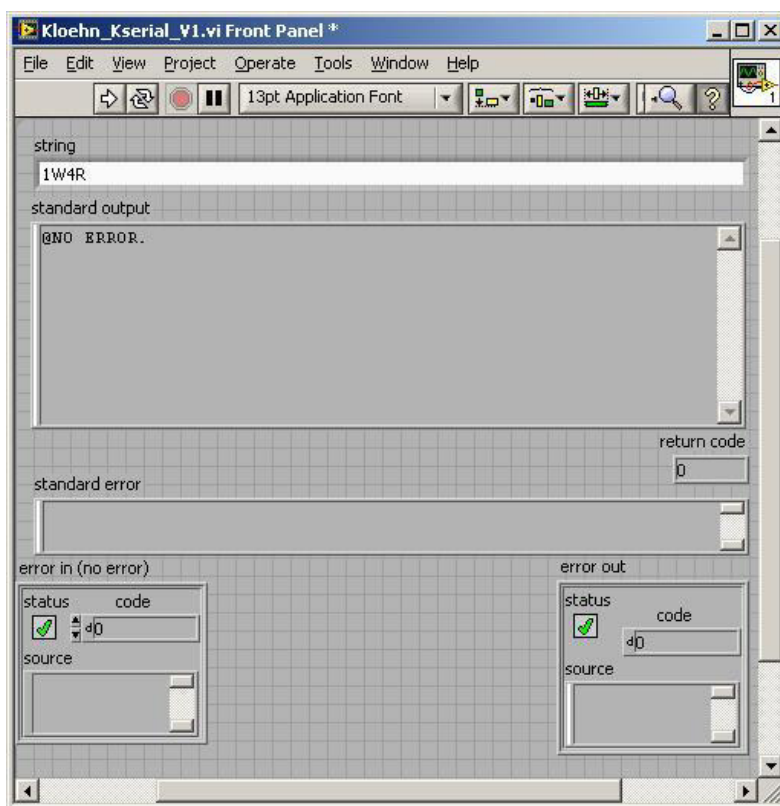


Abbildung 99: Initialisierungsbeispiel einer Pumpe an Adresse 1

Wie man in der Blockdiagrammdarstellung erkennt, sind im nächsten Schritt die Separierung der jeweiligen Pumpenadresse aus dem übergebenen Befehl und die daraus resultierende Umformatierung in den notwendigen Gerätebefehl exemplarisch abgebildet. Diese Gerätebefehle werden dann testweise an das VI Klöhn_Kserial übergeben.

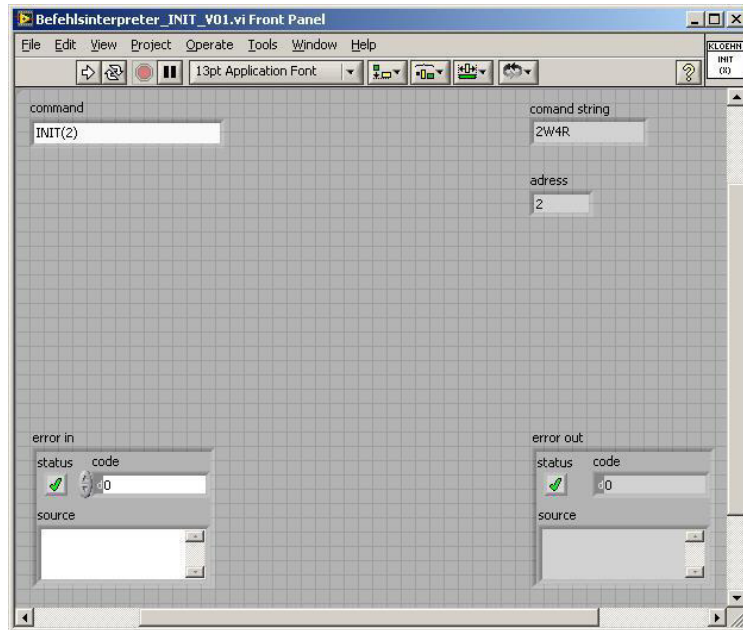


Abbildung 100: Frontpanel: Init - Kommando

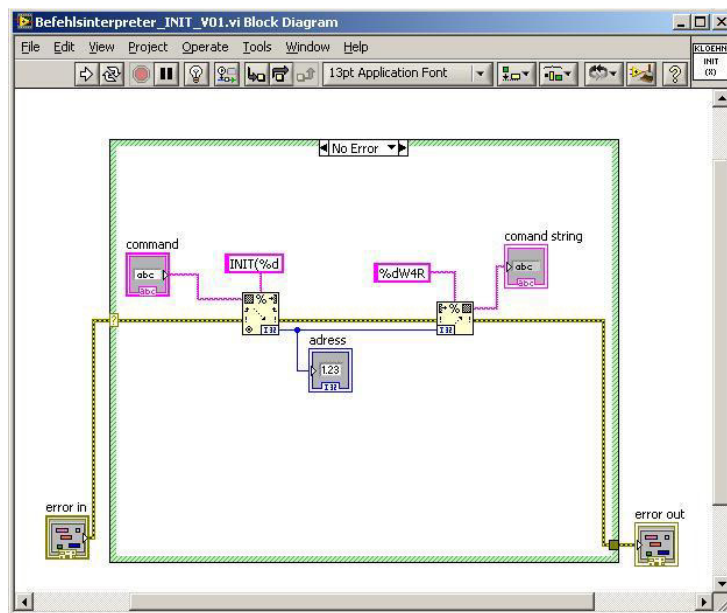


Abbildung 101: Blockdiagramm Init-Kommando

Damit es bei den Installationen auf unterschiedlichen Rechnern zu keinen Problemen in der Fahrtauflösung kommt, wird der Zugriff auf die Kserial.exe über eine relative Pfadangabe in Verbindung mit der Interpretation des aktuellen Arbeitsverzeichnisses des aufgestarteten VIs flexibilisiert.

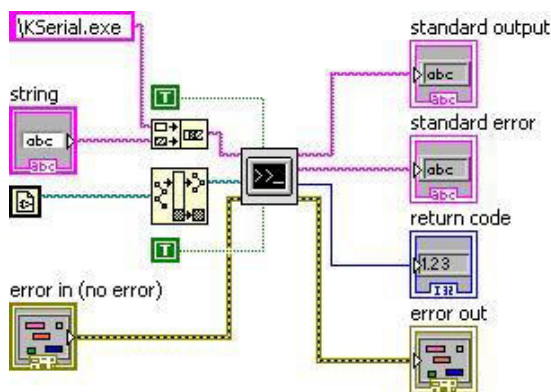


Abbildung 102: LabVIEW Code für die Verwendung von KSerial

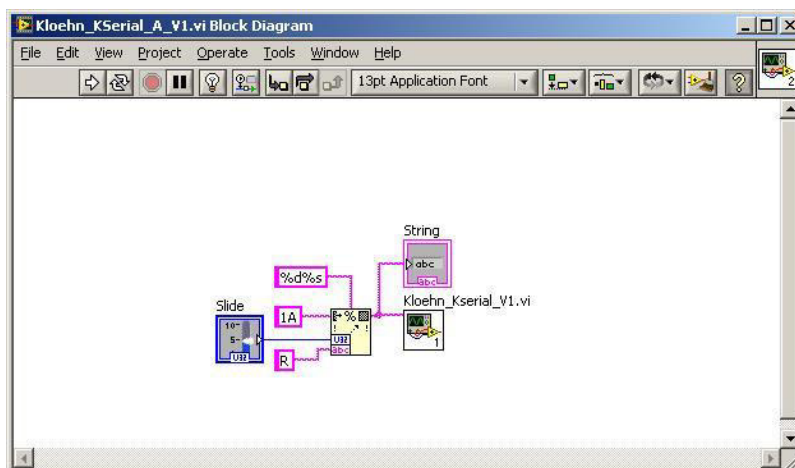


Abbildung 103: Blockdiagramm Kloehn KSerial A

RS485 Communications I/O

The RS485 I/O is available on pins 11 through 14. There are two signals, RS485 "A" and RS485 "B". The "A" line is the "positive" line, and the "B" line, is the "negative" line under idle bias conditions. To prevent common-mode voltage errors, the communications should also use the com ground on pin 10 for an RS485 communication ground in addition to the "A" and "B" lines.

The "A" signal is duplicated on pins 13 and 14, while the "B" signal is duplicated on pins 11 and 12. This duplication permits a straight wire to pass straight through each pair of the "A" and "B" lines to interconnect a series of devices.

The RS485 bus requires a proper bias and termination network for reliable operation. The necessary network is included in the pump and is applied via the RS485 Bias toggles "3" and "4" on the Com Setup Switch shown in Figure 2-3. The first and last devices on an RS485 bus should have the network switched "on". All other devices between the first and last devices should have the network switched "off". A toggle is "on" when the button is positioned to the center of the switch housing; a toggle is "off" when the button is positioned nearest the edge of the switch housing.

Abbildung 104: Verschaltung der Kloehnkomponenten

2.4. Befehlsinterpreter: RATE () – Spritzenpumpe

Der nachfolgende Befehlsinterpreter zeigt die Aufschlüsselung in die Geräteadresse, sowie den durch die Umrechnung der Skalierung sich ergebenden Befehlsstring für die Ansteuerung der Spritzengeschwindigkeit.

Syringe Size	25µl	50µl	100µl	250µl	500µl	1ml	1.25ml	2.5ml	5ml	10ml	25ml	50ml
Default Speed (vol/min)	156µl	312µl	624µl	1.56ml	3.12ml	6.24ml	7.8ml	15.6ml	31.25ml	62.5ml	118.8ml	125ml
Minimum Speed (vol/min)	1.25µl	2.5µl	5µl	12.5µl	25µl	50µl	62.5µl	125µl	250µl	500µl	1.25ml	2.5ml
Maximum Speed** (vol/min)	312.5µl	625µl	1.25ml	3.125ml	6.25ml	12.5ml	15.625ml	31.25ml	62.5ml	125ml	312.5ml	625ml

Abbildung 105: Verfahrensgeschwindigkeiten der Pumpensteuerung

Spritzengröße	Standard Verfahrensgeschwindigkeit
10 µl – 10 ml	10000
25 ml	4000
50 ml	2000

Abbildung 106: Standardwerte für die Verfahrensgeschwindigkeit

Sn Set a predefined *syringe speed*. The speed in the table below are in steps per seconds (sps).
(n: 0...33, @n)[n/a]

Sn	sps	Sn	sps	Sn	sps	Sn	sps
0	6400	9	1800	18	190	27	100
1	5600	10	1600	19	180	28	90
2	5000	11	1400	20	170	29	80
3	4400	12	1200	21	160	30	70
4	3800	13	1000	22	150	31	60
5	3200	14	800	23	140	32	50
6	2600	15	600	24	130	33	40
7	2200	16	400	25	120		
8	2000	17	200	26	110		

Abbildung 107: Parametrierungswerte zum Setzen der Verfahrensgeschwindigkeit

Da die Skalierung des Befehls in steps per second (sps) erfolgt und diese Angabe nicht sehr benutzerfreundlich ist, muss die Umrechnung innerhalb des Befehlsinterpreters aus einer Skalierung von 1 – 10 in die dazu gehörigen Schritte pro Sekunden erfolgen.

$$5 \text{ ml} = 48000 \text{ steps}$$

$$5\text{ml}/48000\text{steps}=0,1042\mu\text{l}/\text{step}$$

$$250\mu\text{l}/0,1042\mu\text{l}/\text{step}=2400\text{step}$$

$$2400/60\text{sec}/\text{min}=40\text{step}/\text{s}=>33$$

Formel 1: Beispielkalkulation der Verfahrensgeschwindigkeit

RATE(address, rate) Address is the pump address. Rate is the rate of fluid delivery. Units are in ml/minute Do not use global addressing	RATE(2,1.75)r RATE(1,DEFAULT)r	1.75r DEFAULTr	Sets the syringe speed to the specified rate. Rate commands apply to both dispenses and withdrawals. Specifying the rate as "DEFAULT" sends the syringe rate to its default valve. The table below shows the default, minimum and maximum syringe speeds for the various syringe sizes.
--	---------------------------------------	-----------------------	--

Abbildung 108: Parametrierung der Verfahrensgeschwindigkeit

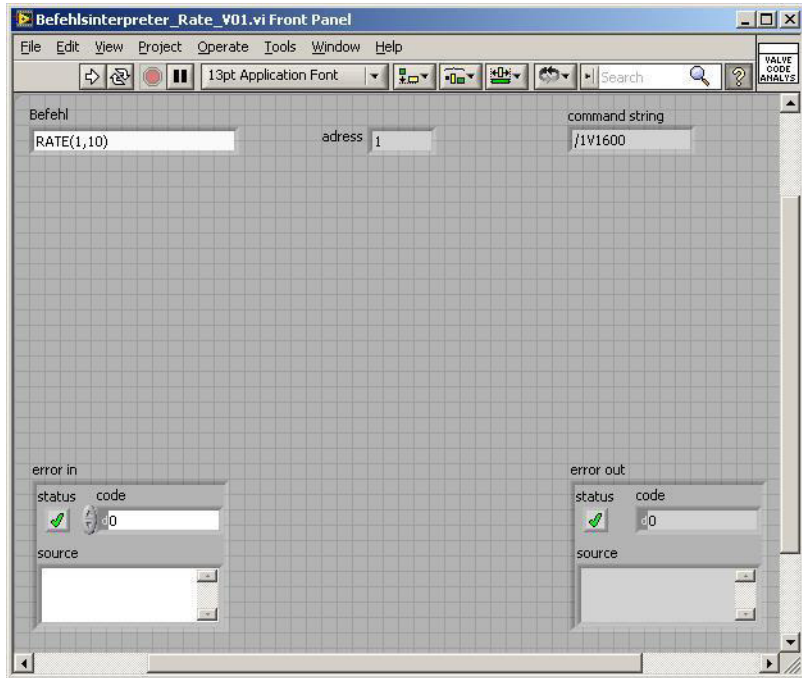


Abbildung 109: Befehlskonstruktion für die Geschwindigkeit

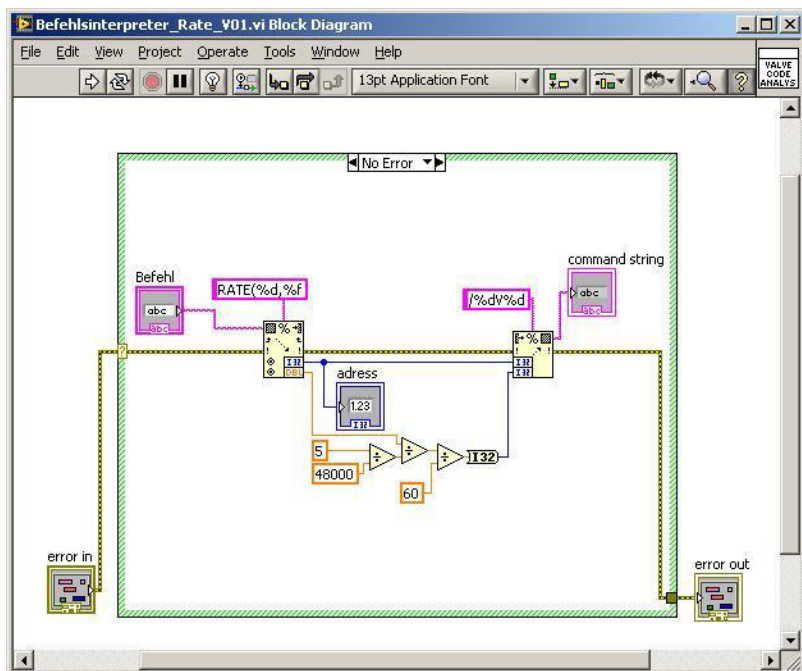


Abbildung 110: Blockdiagramm Befehlskonstruktion für die Geschwindigkeit Umrechnung von steps per second

Umrechnung von Volumina zu Schritten

Die jeweiligen zusammengestellten Befehlsstrings können durch vergleichbare Test VIs wie in dem nachfolgenden Funktionstest in ihrem Funktionsumfang geprüft werden. Um dem Anwender die Umrechnung der Schritte auf die jeweiligen zu dosierenden Volumina abzunehmen, kalkuliert das VI aus übergebenen Volumina die anzusteuernden Steps.

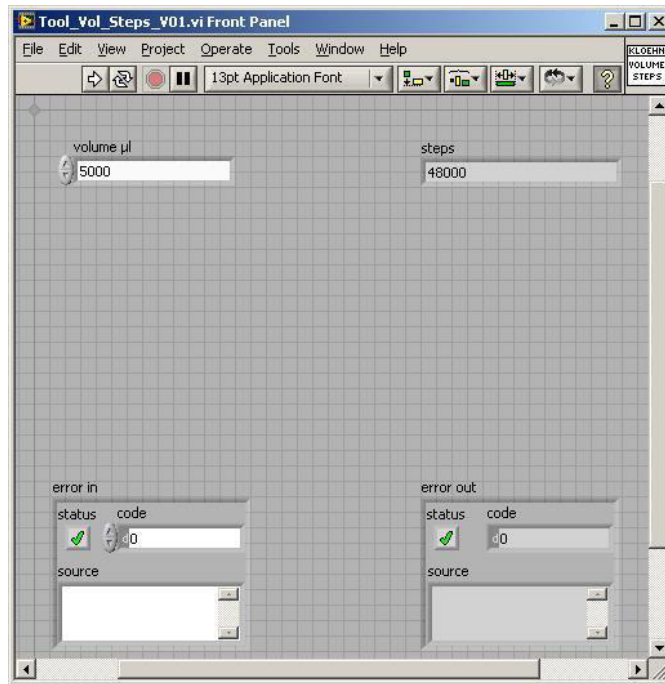


Abbildung 111: Umrechnung von Volumina zu Schritten

In diesem Beispiel VI sind sowohl die Steps des Gesamthubes, als auch das Spritzenvolumen über Konstanten innerhalb des Codes definiert, welche bei der weiterführenden Umsetzung jedoch aus extern bereitgestellten ini-Dateien zur einfacheren Wartung und flexiblen Anpassung auf die jeweiligen Hardwareigenschaften Verwendung finden.

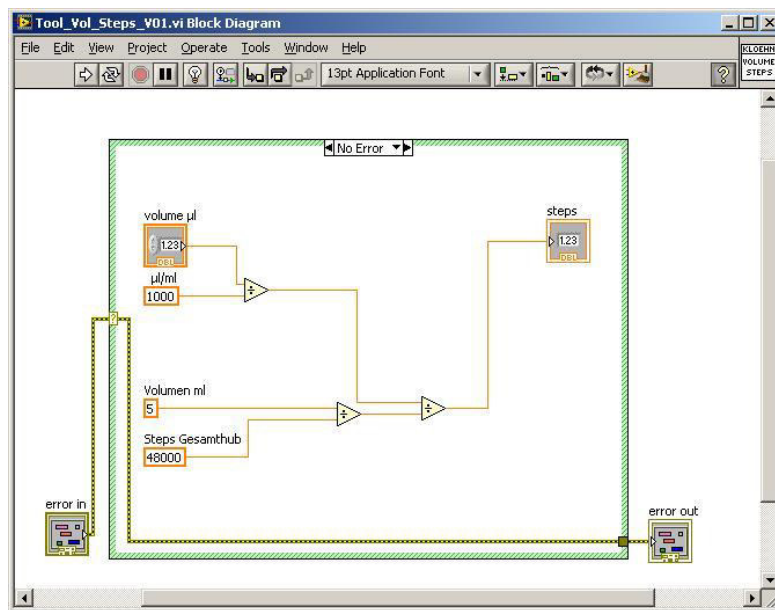


Abbildung 112: Blockdiagramm Umrechnung von Volumina zu Schritten

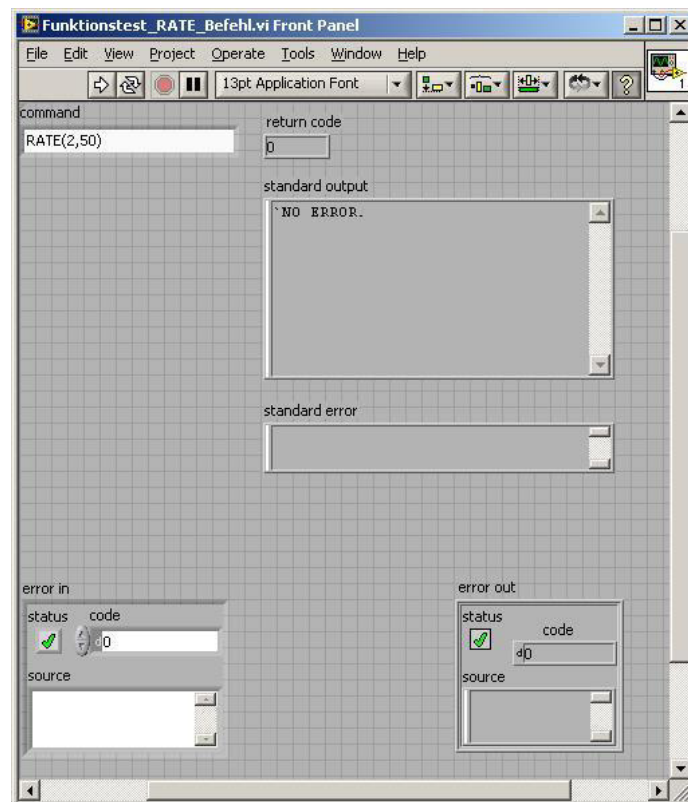


Abbildung 113: Beispiel zum Setzen der Rate

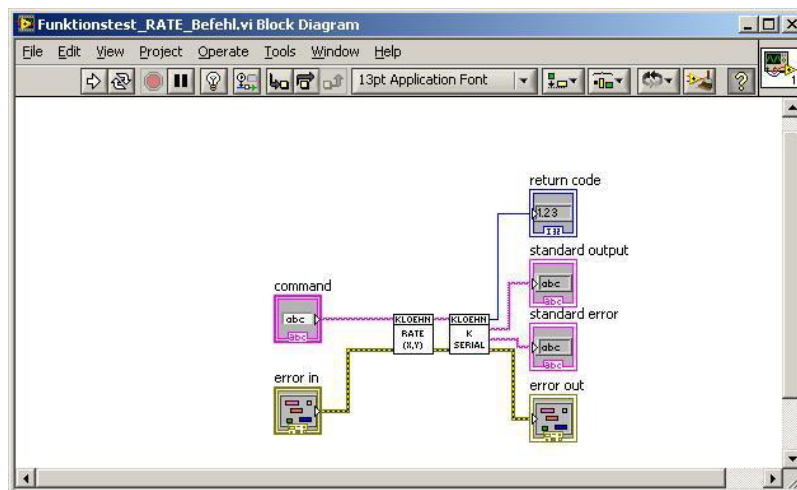


Abbildung 114: Prinzipieller Funktionsaufbau

2.5. Befehlsinterpreter: WITHDRAW() – Spritzenpumpe

Die Verwendung dieser Umrechnung zeigt das nachfolgende Beispiel, bei dem mit dem Befehl *Withdraw (2,1000)* ein Aufziehen der Spritze an Adresse 2 auf 1 ml seitens des Benutzers vorgegeben wird, intern jedoch der Ansteuerungsbefehl 2p9600r zusammengestellt wird.

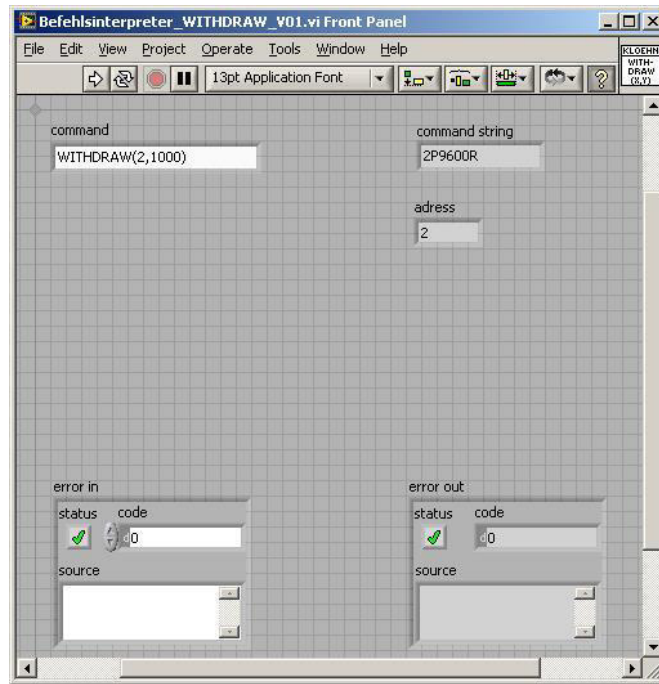


Abbildung 115: Frontpanel Withdraw-Commando

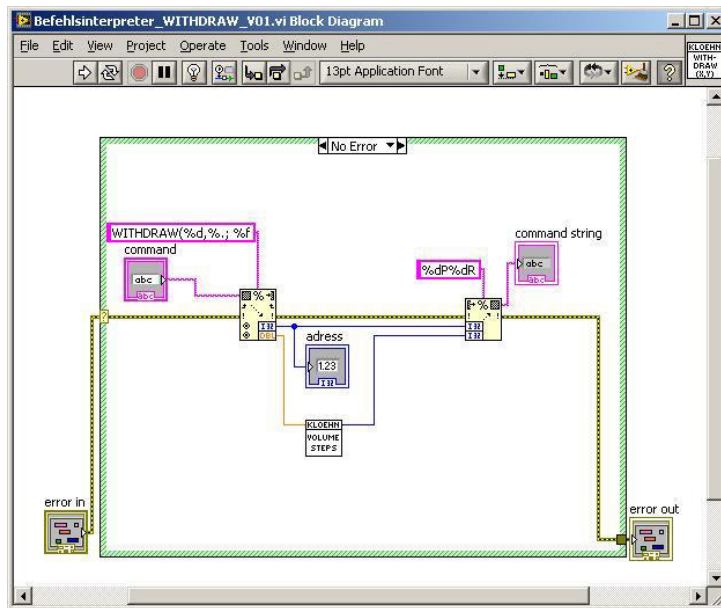


Abbildung 116: Blockdiagramm Withdraw-Kommando

2.6. Befehlsinterpreter: DISPENSE () – Spritzenpumpe

Wie das nachfolgende Blockdiagramm zeigt, kommt bei der Interpretation der Volumina des Skalierungs-VI zum Einsatz. Da sich an der Skalierung nichts ändert, ob die Spritze aufgezogen wird oder Lösungsmittel nach extern dosiert wird, verhält sich die Ansteuerung bei dem Dispense Befehl bis auf den Buchstabencode identisch.

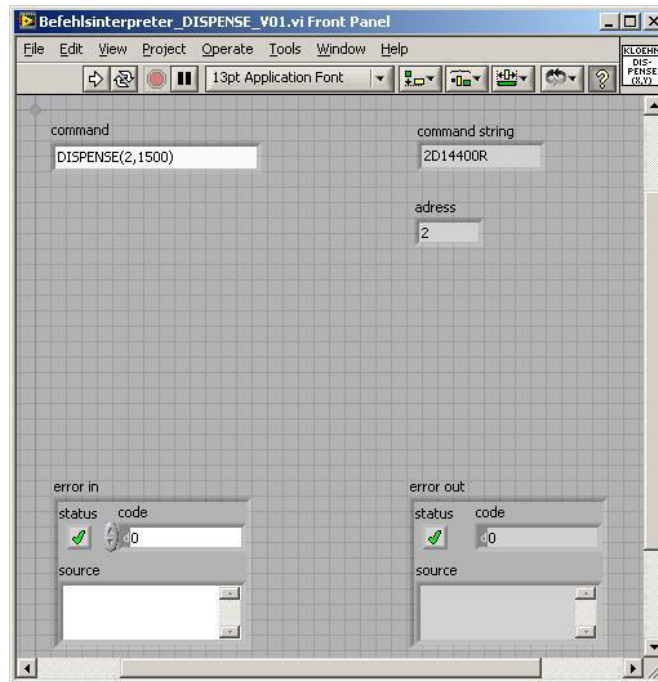


Abbildung 117: Frontpanel Dispense-Commando

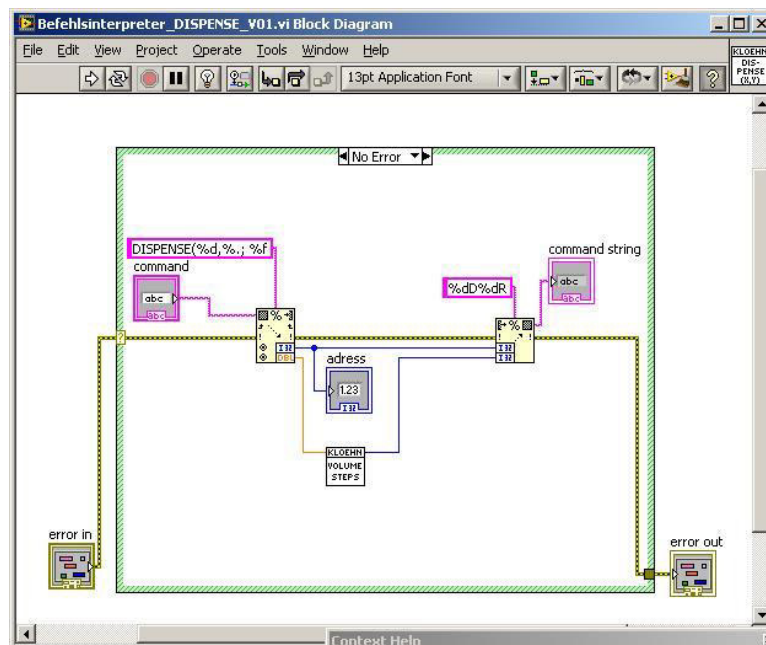


Abbildung 118: Blockdiagramm Dispense - Commando

2.7. Auswertprogramm Druckmessung am Druckverteiler (Manifold)

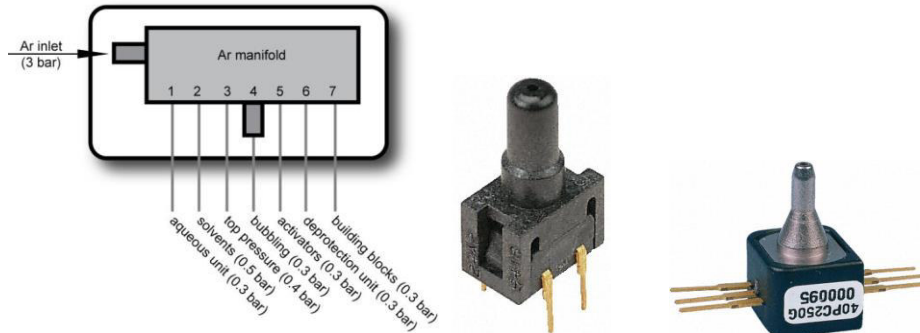


Abbildung 119: Druckverteiler und StandardEinstellung Drücke an den verschiedenen Eingängen, sowie Gasdrucksensoren zur Druckmessung

Zum Test der Druckstabilität wurde eine Druckvergleichsmessung am Honeywell_24PCC, Honeywell 40PC1506 mit dem Gasdrucksensor 24PCCFA2G durchgeführt. Die Daten Drucküberwachung mit den Gasdrucksensoren wird durch folgendes Blockdiagramm visualisiert:

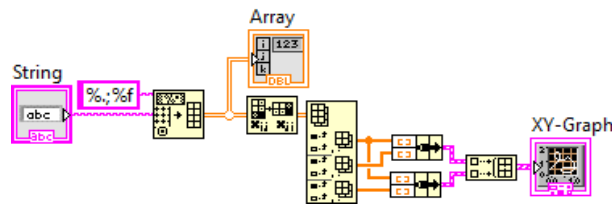


Abbildung 120: Blockdiagramm zur Darstellung der Messwerte

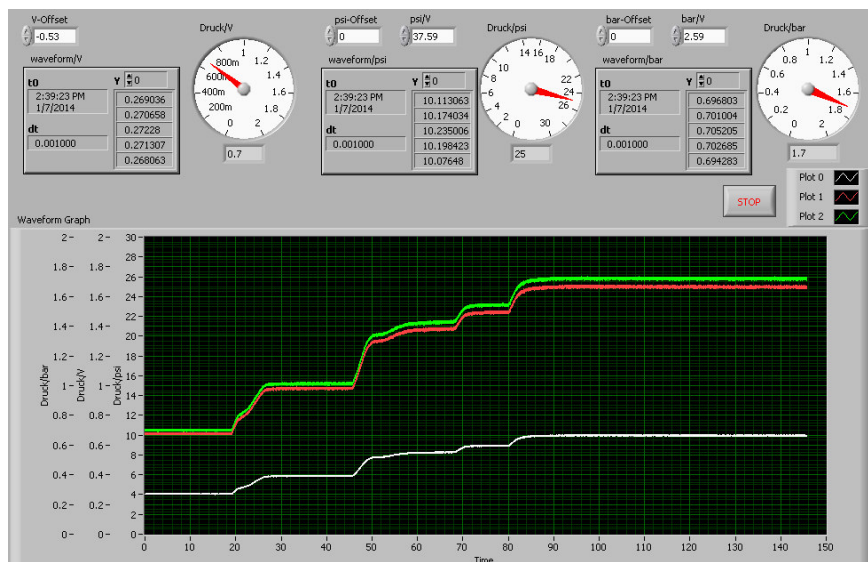


Abbildung 121: Für das Auswertprogramm Druckmessung Manifold existieren verschiedene Benutzeroberflächen mit verschieden detaillierter Darstellung der Messergebnisse

2.8. Fraktionssammler: HOME Interpretation

Die Ansteuerungssequenz innerhalb der LabVIEW Programmierung, in der die beiden Befehle HOME(x) und HOME(Y) lt. Schnittstellenparameter (vgl. *Anhang 1.12.*) direkt als Home Funktionalität ausprogrammiert ist, zeigt die nachfolgende Abbildung.

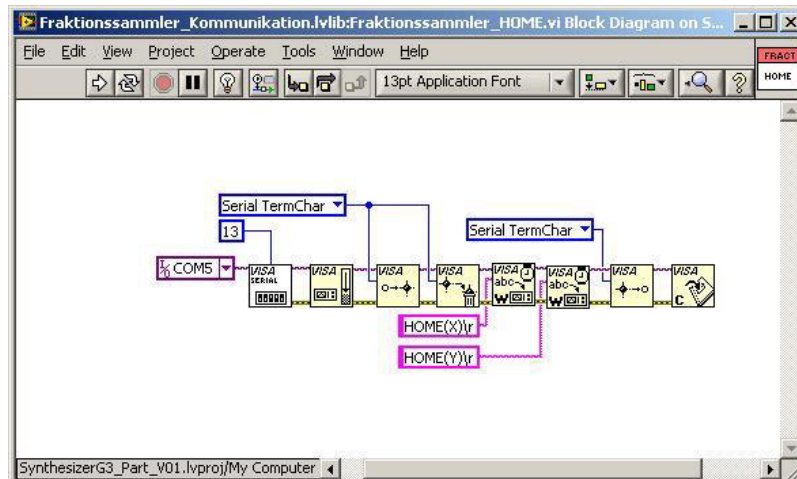


Abbildung 122: LabVIEW Programm HOME - Interpretation

2.9. Fraktionssammler: MOVE - Befehl

Die jeweilige Ansteuerung einer Individualposition resultiert auch achsenbasiert durch den Move Befehl, welcher nach vergleichbarer Vorgehensweise, wie zuvor beschrieben, ermittelt und getestet wurde, aus den als Zwischenschritt das nachfolgende VI entstand.

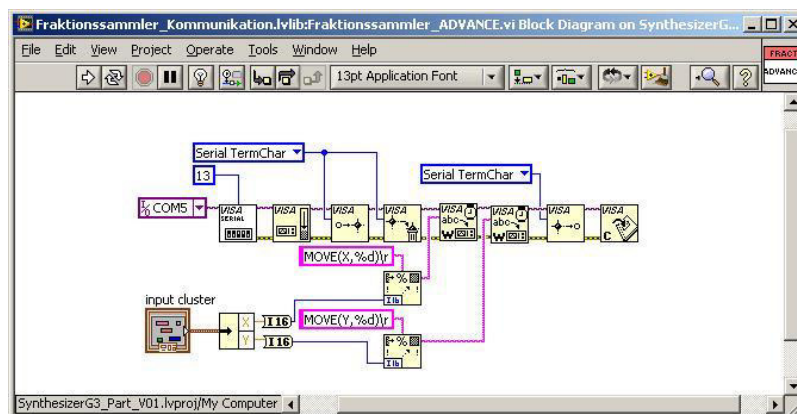


Abbildung 123: Blockdiagramm Fraktionssammlertest MOVE Befehl

2.10. Fraktionssammler: Testprogramm

Mit dieser Art der Vorarbeit konnte dann ein individueller Gerätetreiber für den Fraktionssammler, der übergreifend alle bereit gestellten Funktionalitäten programmiertechnisch zur Verfügung stellt, realisiert werden.

Mit Hilfe dieses Terminals sind somit auch zwischenzeitlich Funktionstest nach z. B. Wartungsarbeiten oder Kommunikationsproblemen durchführbar.

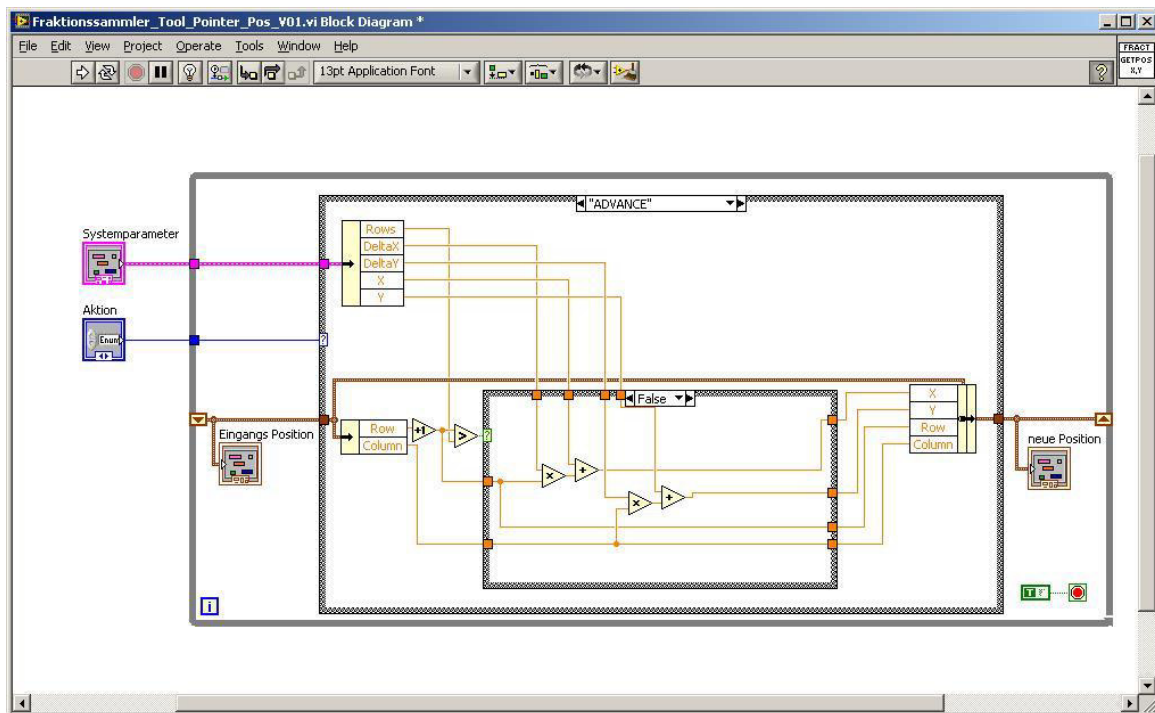


Abbildung 124: Blockdiagramm Fraktionssammler Tool

2.11. Auswertprogramm Protokolldatei Julabo

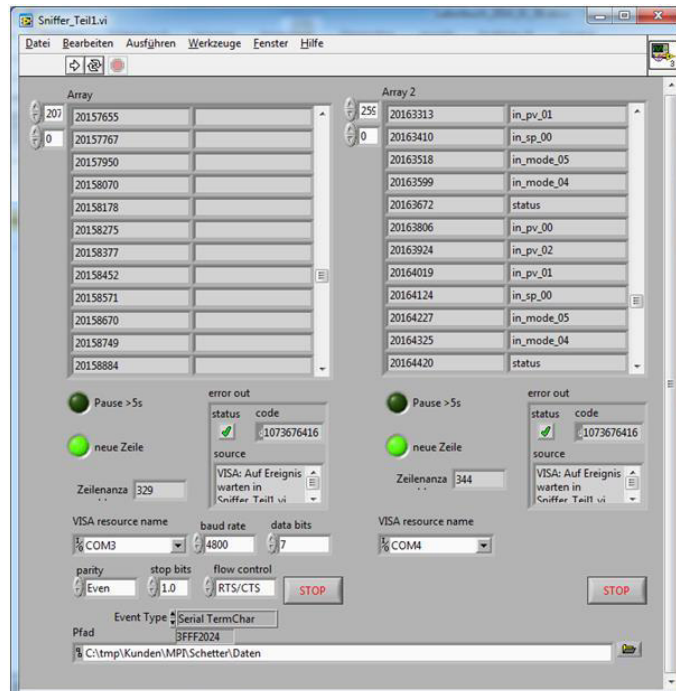


Abbildung 125: Sniffer-Informationen

Nach dem Import der Textdatei kann durch die \Codedarstellung kontrolliert werden ob sich die Datenstruktur zur Auswertung über einen Tabellenstring eignet.



Datum	Zeit	relative Zeit	Sollwert	Temp.int	Tem.ext
16.05.2012	11:26:42	00:00:00	50,000000	42,830000	NaN
16.05.2012	11:26:45	00:00:03	50,000000	42,860000	NaN
16.05.2012	11:26:48	00:00:06	50,000000	42,890000	NaN
16.05.2012	11:26:51	00:00:09	50,000000	42,920000	NaN
16.05.2012	11:26:54	00:00:12	50,000000	42,950000	NaN
16.05.2012	11:26:57	00:00:15	50,000000	42,990000	NaN
16.05.2012	11:27:00	00:00:18	50,000000	43,020000	NaN
16.05.2012	11:27:03	00:00:21	50,000000	43,050000	NaN
16.05.2012	11:27:06	00:00:24	50,000000	43,080000	NaN
16.05.2012	11:27:09	00:00:27	50,000000	43,110000	NaN
16.05.2012	11:27:12	00:00:30	50,000000	43,140000	NaN
16.05.2012	11:27:15	00:00:33	50,000000	43,190000	NaN
16.05.2012	11:27:18	00:00:36	50,000000	43,220000	NaN
16.05.2012	11:27:21	00:00:39	50,000000	43,260000	NaN
16.05.2012	11:27:24	00:00:42	50,000000	43,280000	NaN
16.05.2012	11:27:27	00:00:45	50,000000	43,310000	NaN
16.05.2012	11:27:30	00:00:48	50,000000	43,350000	NaN

Abbildung 127: Formatierte Protokolldatei

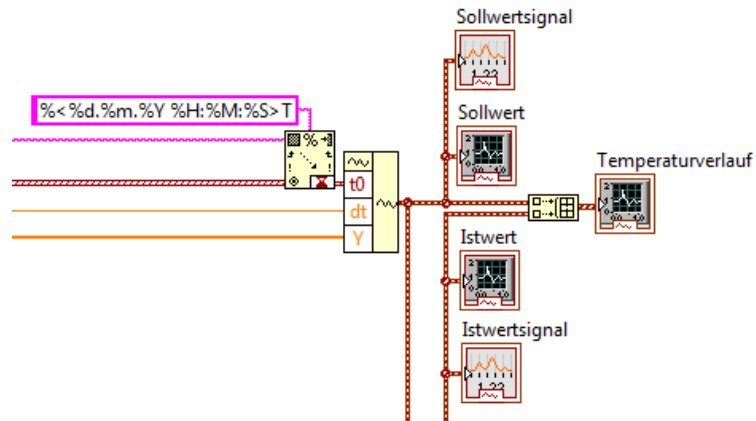


Abbildung 128: LabVIEW-Code des Auswertprogramms

2.12. Testversion Benutzerschnittstelle Arbeitssynthesizer I

Das nachfolgend gezeigte Blockdiagramm zeigt hierbei die vollständige Abarbeitungsprozedur für die Anweisungsliste. Mit Hilfe der jeweils realisierten Befehlsinterpreter und den individuellen Funktionstests ergibt sich die komplette Auflistung der VIs zum Zeitpunkt des Tests der funktionalen Einheiten der alten Syntheseprotokolle.

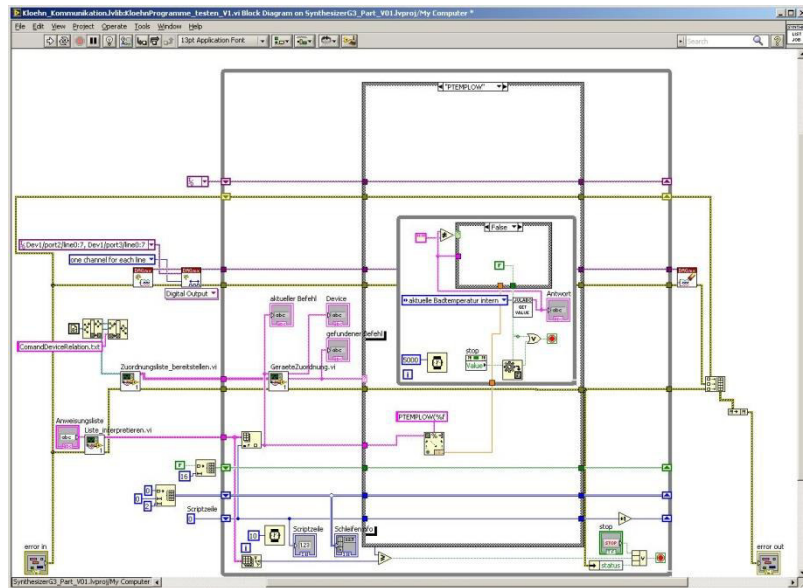


Abbildung 129: Blockdiagramm Testversion zum Arbeitssynthesizer zum Systemtest der chemischen Funktionseinheiten aufgrund der alten Syntheseskripte

2.13. Unterscheidung der Applikationsumgebung

Um die jeweilige Synthese auf eindeutige Versionen rückführbar zu machen, bietet es sich an, jeweils vollwertige ausführbare Dateien zu realisieren und da sich hierdurch auch die Installationen auf eine Runtime Version von LabVIEW reduziert, wesentlich leichter auf die verschiedenen PCs exportieren lässt.

Jedoch unterscheidet sich dann der jeweils resultierende Pfadaufbau zu den Applikationen. Dieser Sachverhalt ist durch die nachfolgende CASE Struktur exemplarisch administriert.

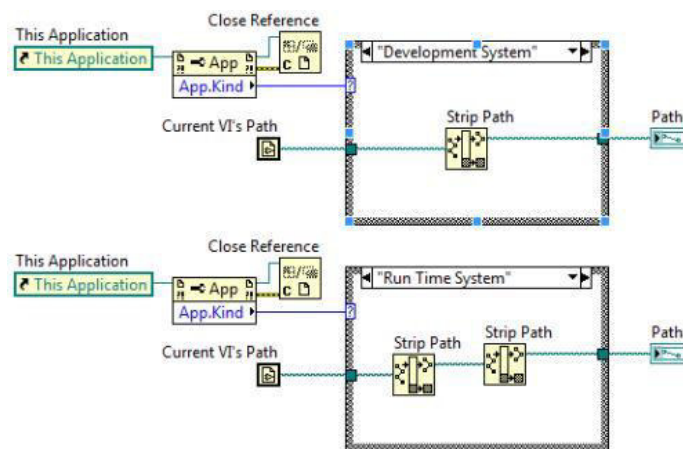


Abbildung 130: Unterscheidung der Applikationsumgebung

2.14. Drag & Drop Bedienung

Die bisher zur Anwendung gekommene textbasierte Interaktionsmöglichkeit mit dem Synthesizer hat große Vorteile in der Flexibilität und in der zeitnahen Umsetzung benutzerindividueller Anpassungen. Auch sehr weitreichende und tief in die Programmierstruktur hineingreifende Anpassungen sind mit adäquatem Aufwand umsetzbar. Bei den Endanwendern ist in vielen Fällen jedoch die visuelle Aufbereitung intuitiver erlernbar und von höherer Wertschätzung. Somit ist das Ziel eine gerätenahe Drag & Drop Bedienphilosophie zur Verfügung zu stellen. Zwar wird im Vordergrund, durch Interaktion mit grafischen Bedienelementen, die Benutzerschnittstelle realisiert, jedoch im Hintergrund weiterhin auf die bereits aufgezeigte textbasierte geräteseitige Anweisungslistenenerstellung zurück gegriffen.

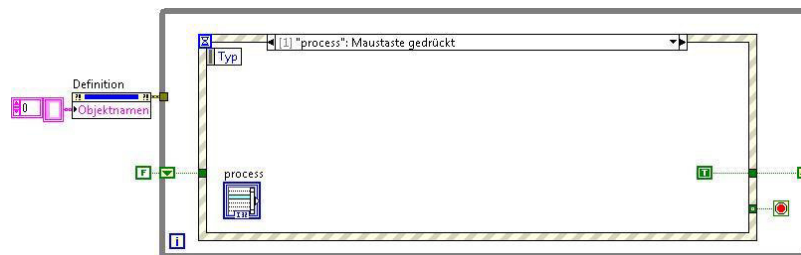


Abbildung 131: Start einer Drag & Drop Aktion durch das Drücken der linken Maustaste im Quellobjekt

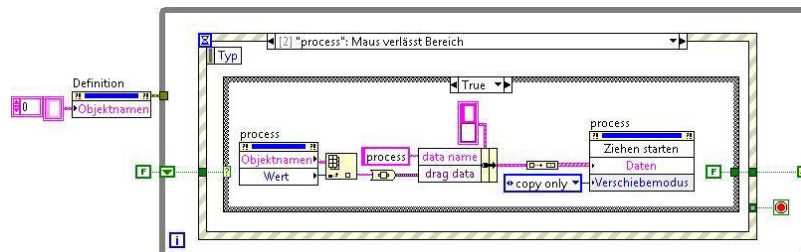


Abbildung 132: Befüllen des Drag & Drop Kontext mit den bereitgestellten Daten

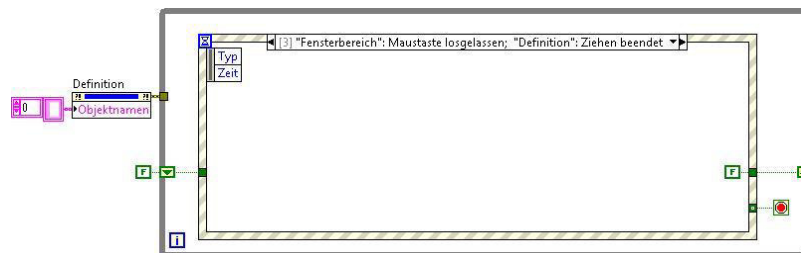


Abbildung 133: Abbruchmöglichkeit der Drag & Drop Aktion

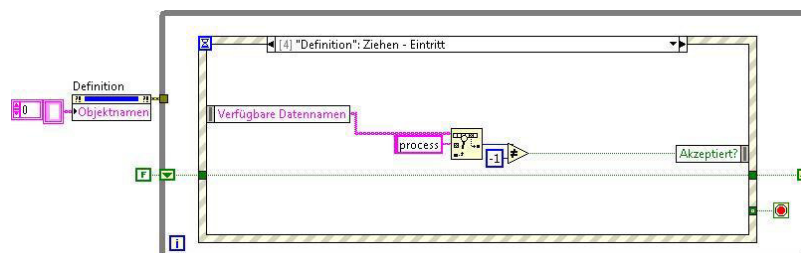


Abbildung 134: Maus-Icon-Anpassung durch Kontrolle der bereitgestellten Werte

Da nicht zu jedem beliebigen Zeitpunkt eine Unterbrechung der Abarbeitung der Befehlsliste sinnvoll ist, wird nach der Beendigung modularer Einheiten auf die Benutzeranforderung detektiert. Ist das Flag gesetzt, wird die Abarbeitung angehalten. Die jeweiligen Prozessinformationen bleiben erhalten und ermöglichen somit eine Fortführung des Programms.



Abbildung 135: Ereignisorientierte Auswertung der Stopp-Schaltfläche

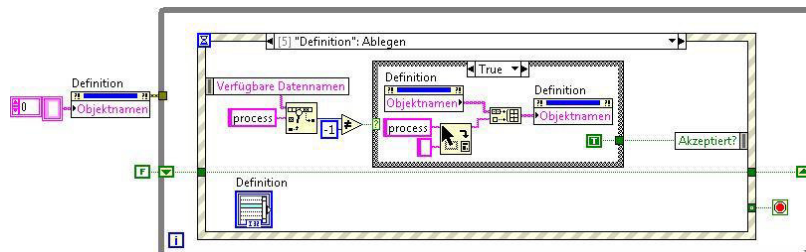


Abbildung 136: Einfügen der übergebenen Daten in die jeweiligen Objekte

2.15. Aktualisierung bei Wechsel der Registerkarte

Beim Wechsel der Registerkarte erfolgt sofortige Aktualisierung der Listen. Die hierzu verwendete Programmierung ist nachfolgend gezeigt. Das Ereignis der Wertänderung führt die entsprechende Codesequenz zur Aktualisierung der Listen aus.

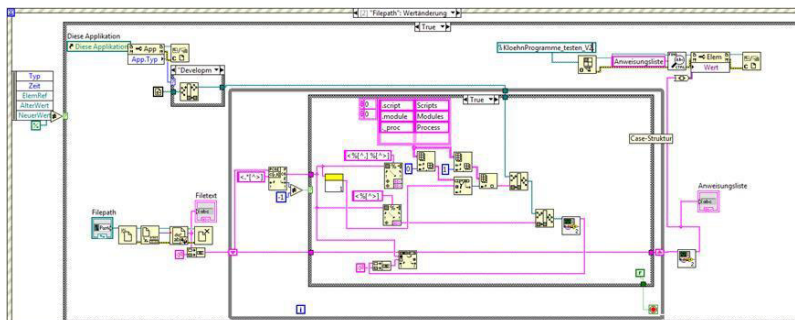


Abbildung 137: Interpretation bei der aktiven Abarbeitung

Durch die Interpretation wird der zu Laufzeit abzuarbeitende Code generiert und in der Anweisungsliste zur Verfügung gestellt. Mit Betätigung der START- Schaltfläche wird dann die Synthese abgearbeitet.

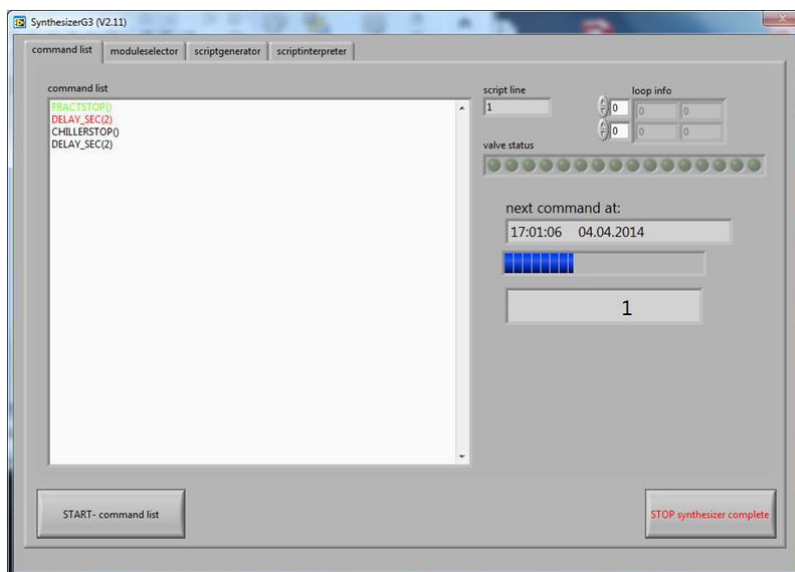


Abbildung 138: Gerätstopp

Soll der Synthesizer ausgeschaltet werden, so wird nach der Benutzung der Schaltfläche STOP die jeweils sinnvollen Reset- Zustände hergestellt und die Kommunikation zu den jeweiligen Geräten terminiert.

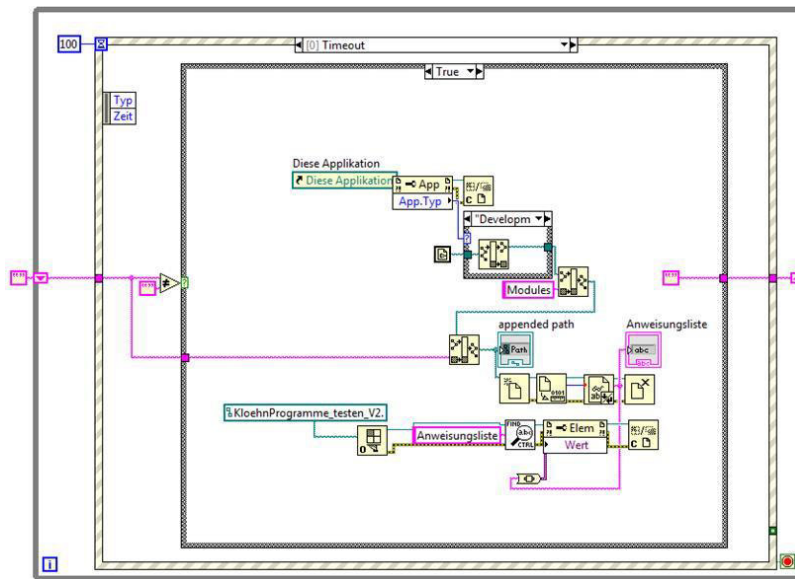


Abbildung 139: Listenübergabe an den Abarbeitungsprozess

2.16. Programmbeispiel-VI: Benutzerkommunikation mit dem elektrischen Manifold

In dem nachfolgenden Programmcode- Beispiel ist gezeigt wie die eine While- Schleife die Benutzeroberflächenereignisse abarbeitet und die hieraus resultierenden Anpassungen in der zweiten unabhängigen Schleife signalisiert.

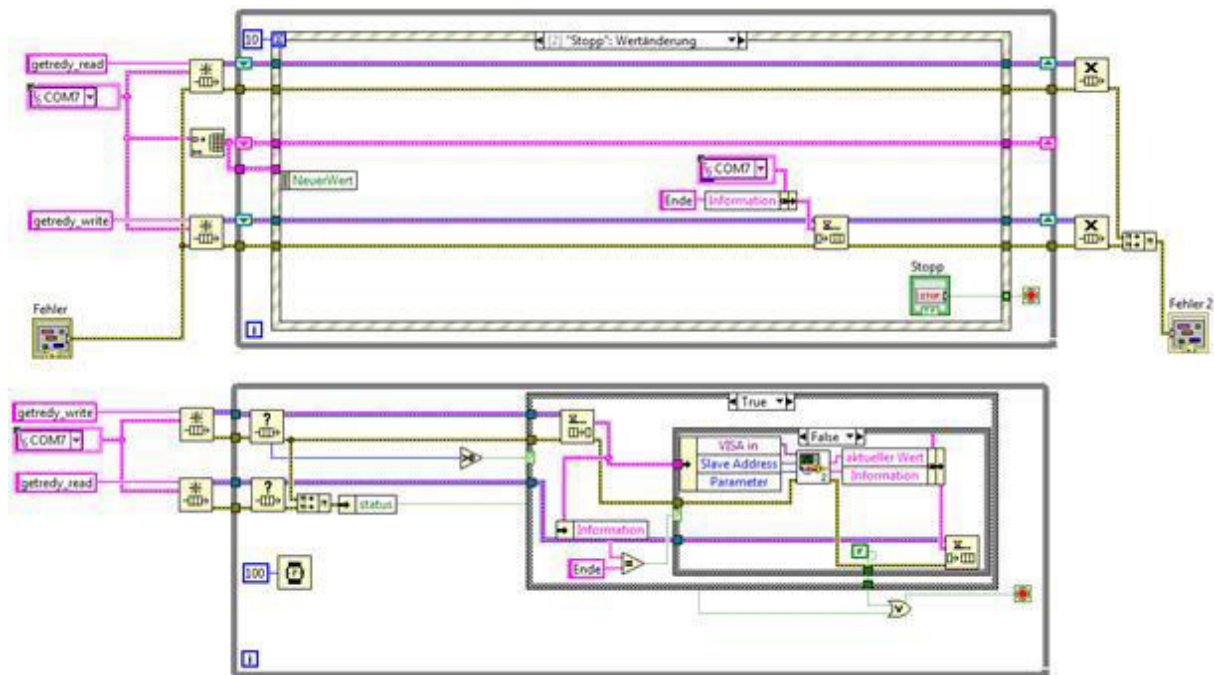


Abbildung 140: Kommunikation über LabVIEW

2.17. Einbindung von ChemDraw Funktionalitäten

In den Benutzeroberflächen des Entwicklungssynthesizers wurden die Funktionalitäten der Software ChemDraw dem Anwender vollumfänglich eingebunden und zur Generierung der Synthesesteuerung genutzt.

Durch die offengelegte API Schnittstelle und der Zusammenfassung in Form des Software-Development-Kit (SDK), ergibt sich eine flexible Umsetzung dieser Interaktion.



Abbildung 141: SDK – ChemOffice



Abbildung 142: Detailansicht Objektmodell Chemdraw

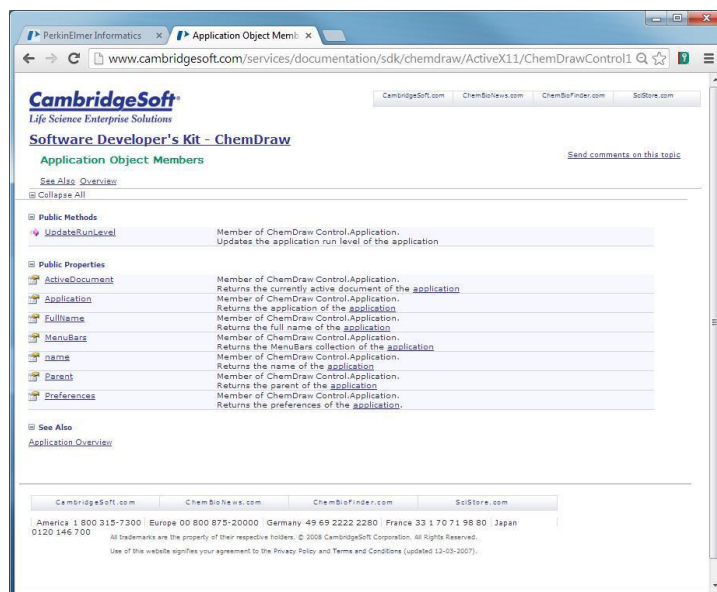


Abbildung 143: Informationen aus dem SDK - Online

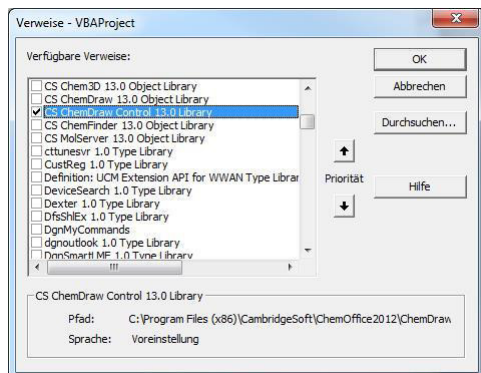


Abbildung 144: Anmeldung von ChemDraw-Objekten in VBA

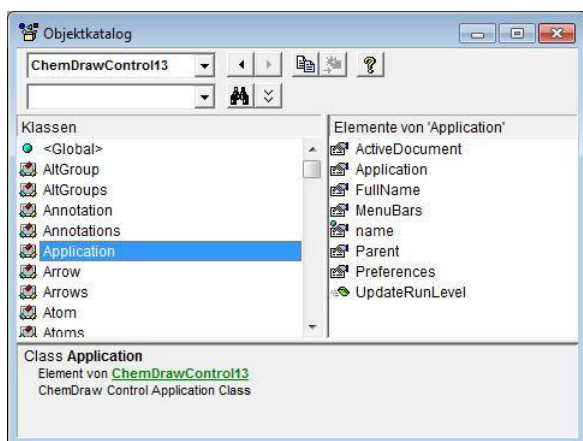


Abbildung 145: Objektkatalog als lokale Informationsquelle

Bevor die entsprechende Programmierung in LabVIEW erfolgte, sind die funktionalen Tests innerhalb eines VBA Projektes erfolgt.

Nach der Anmeldung der Verweise der ChemDraw – Control - Library stehen die damit bereit gestellten Eigenschaften und Methoden zur Verfügung und können z. B. wie in der nachfolgenden Subroutine gezeigt, verwendet werden, um aus einem aktiven Control Informationen in die Programmierumgebung auslesbar zu machen.

'Declares a ChemDrawControl object and returns number of objects in the application

```
Private Sub cmdShowNumberOfAtoms_Click()
```

```
    Dim myAtom1 as ChemDrawControl10Ctl.Atom
```

```
    Dim myAtom2 as ChemDrawControl10Ctl.Atom
```

```
'Checks if at least two Atoms have been drawn
```

```
If ChemDrawControlCtl.Objects.Atoms.Count > 2 Then
```

```
    Set MyAtom1 = ChemDrawControlCtl.Objects.Atoms.Item(1)
```

```
    Set MyAtom2 = ChemDrawControlCtl.Objects.Atoms.Item(2)
```

```
'Returns the number of Atoms within the ChemDraw document
```

```
    Text2.text = "There are: " & format(ChemDrawControlCtl.Objects.Atoms.Count,  
"##") & " Atom/s"
```

```
End If
```

```
End Sub
```

<http://www.cambridgesoft.com/services/documentation/sdk/chemdraw/ActiveX11/ChemDrawControl10~ChemDrawCtl.html>

Tabelle 2: Beispiel ChemDraw Subroutine

Neben den Möglichkeiten individualisierte Grafik dann funktionale Bedeutung zuzuweisen, d.h. Textobjekte entsprechend ihrer Position als zu interpretierenden Steuerungscode auszuwerten, bei denen die im Verbund abgebildeten Strukturdarstellungen keinerlei Bedeutungen haben, jedoch für die Informationen benutzerseitig sehr hilfreich sind, entstand die Idee der aktiven Interpretation von Reaktionsgleichungen.

Hierzu bedarf es der ausführlichen strukturellen Kenntnis und Zuordnung der Einzelobjekte innerhalb des komplexen ChemDraw – Controls, welches nachfolgend gezeigt ist, dessen Struktur sich auch über die Verwendung des Objektkatalogs visualisieren und erschließen lässt.

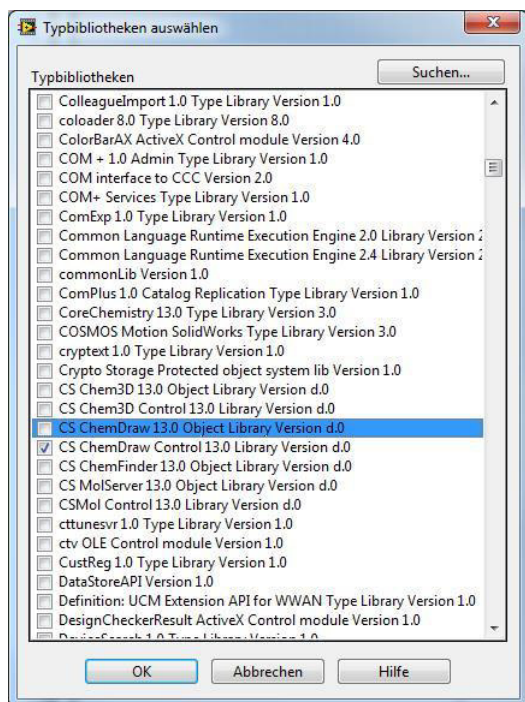


Abbildung 146: Anmelden des ChemDrwaCtl

Neben der webbasierten SDK – Umgebung, können auch die erweiterten technischen Supportinformationen seitens CambridgeSoft ausgewertet werden. Nach Anmeldung der ChemDraw – Control – Library über die Typenbibliothekauswahl von LabVIEW, ist dann die Umsetzung, der in VBA gewonnenen Erkenntnisse, direkt möglich und der Zugriff auf die Eigenschaftsknoten der ActiveX – Palette, ergibt sich dann als Beispiel der nachfolgende Code als Zugriff auf die individuell selektierten Objekte und deren Manipulation über die Methodenknoten.

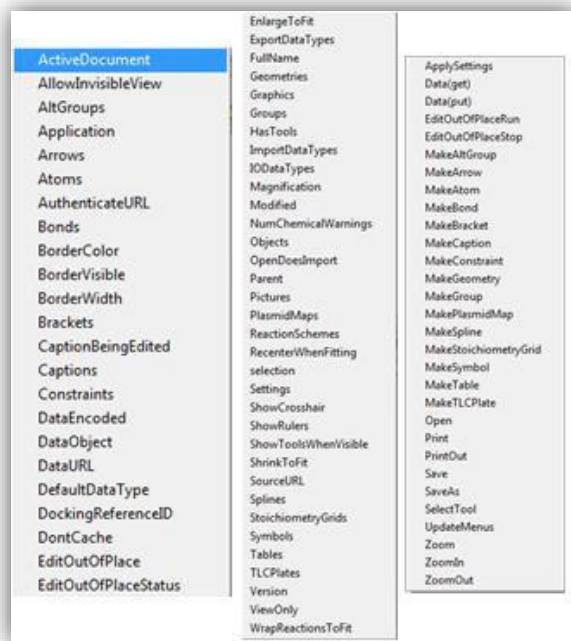


Abbildung 147: Eigenschaftsknoten in LabVIEW

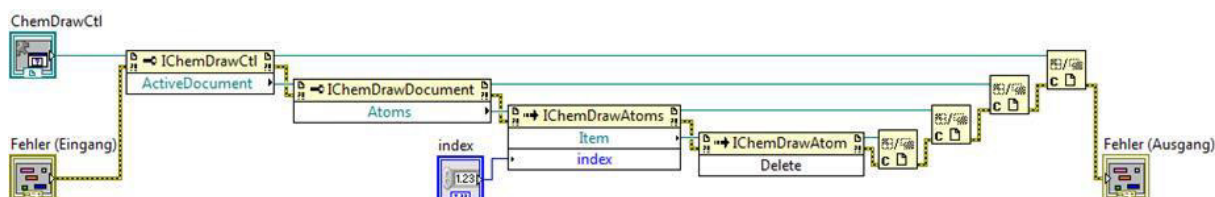


Abbildung 148: Zugriffsbeispiel in ChemDraw

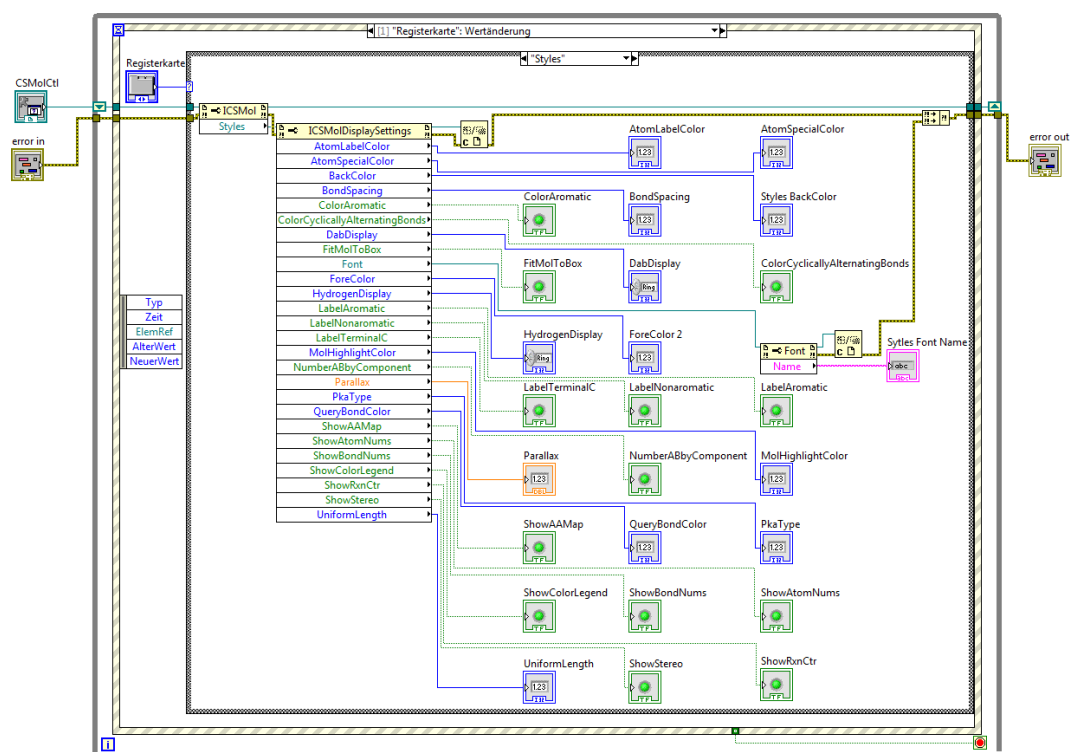


Abbildung 149: Blockdiagramm eines Zugriffsbeispiels auf die Eigenschaften des Chem Draw ActivX-Contol

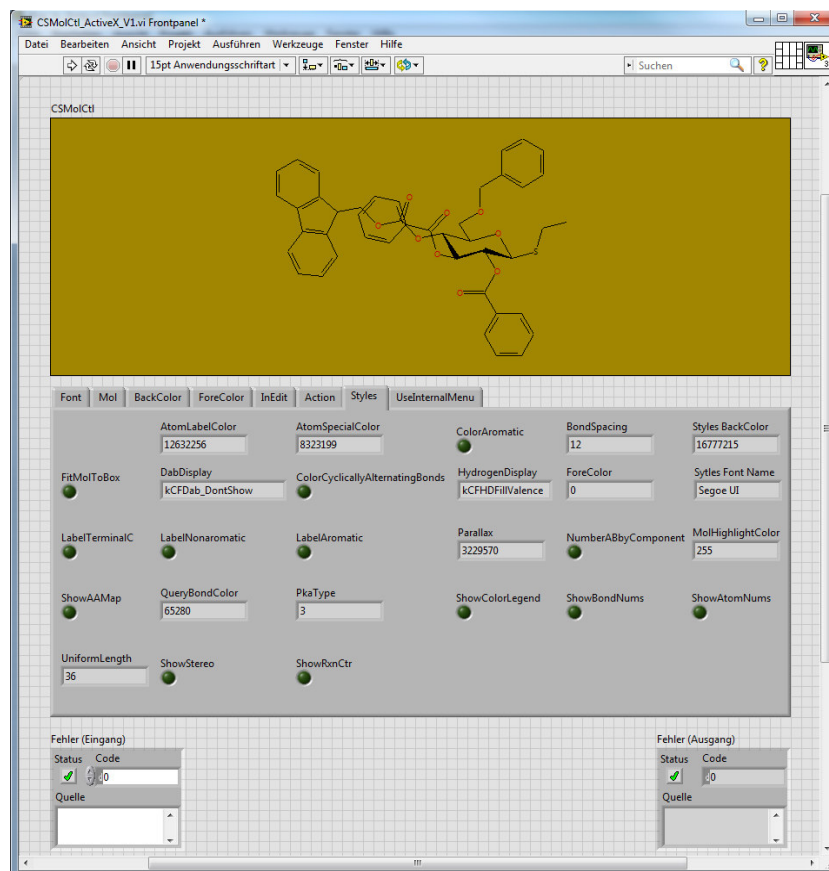


Abbildung 150: Frontpanel eines Zugriffsbeispiels auf die Eigenschaften des Chem Draw ActivX- Contol

Hiermit können zur Laufzeit auf die gleichen Informationen zugegriffen werden wie über den LabVIEW – Eigenschaftsbrowser zur Entwicklungszeit.

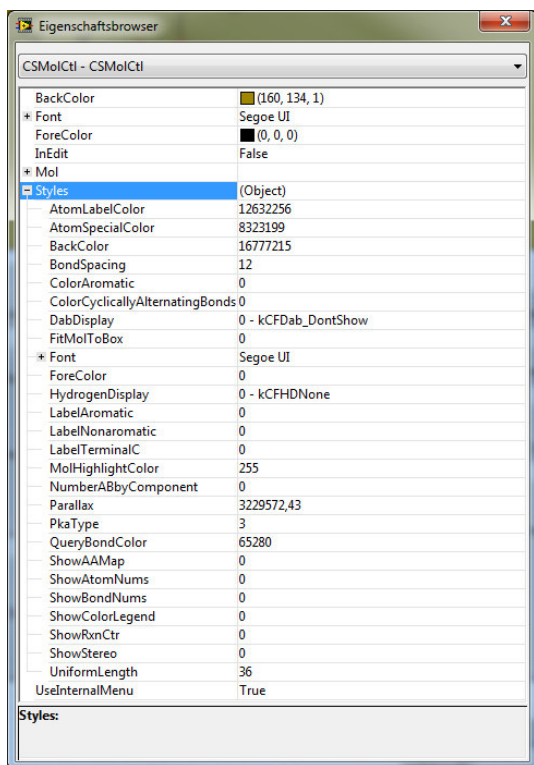


Abbildung 151: Eigenschaftsbrowser

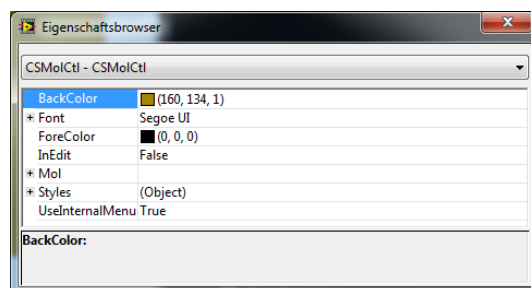


Abbildung 152: Registerkarte Eigenschaftsbrowser

Die Objekte der höchsten Ebene sind hierbei auf eigenen Registerkarten aufgeführt.

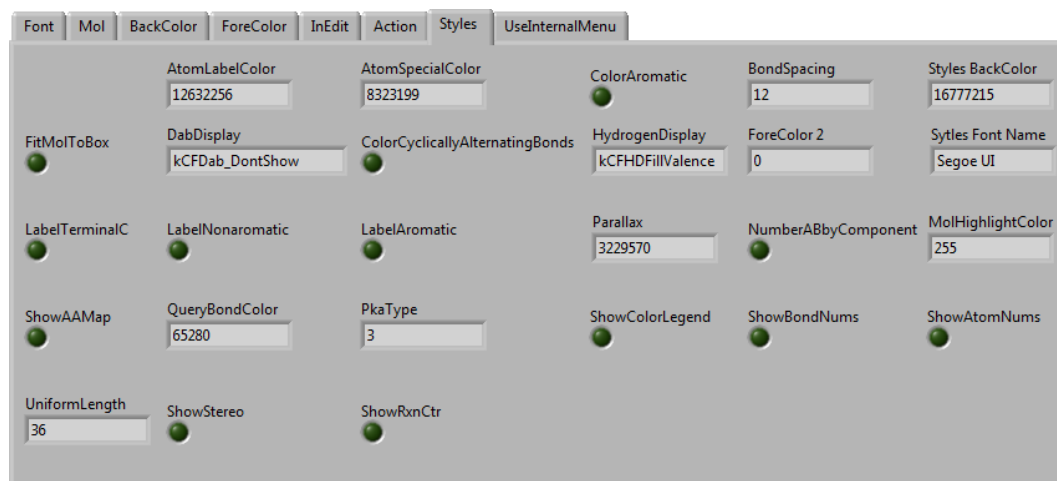


Abbildung 153: Registerkarte auf Objektebene

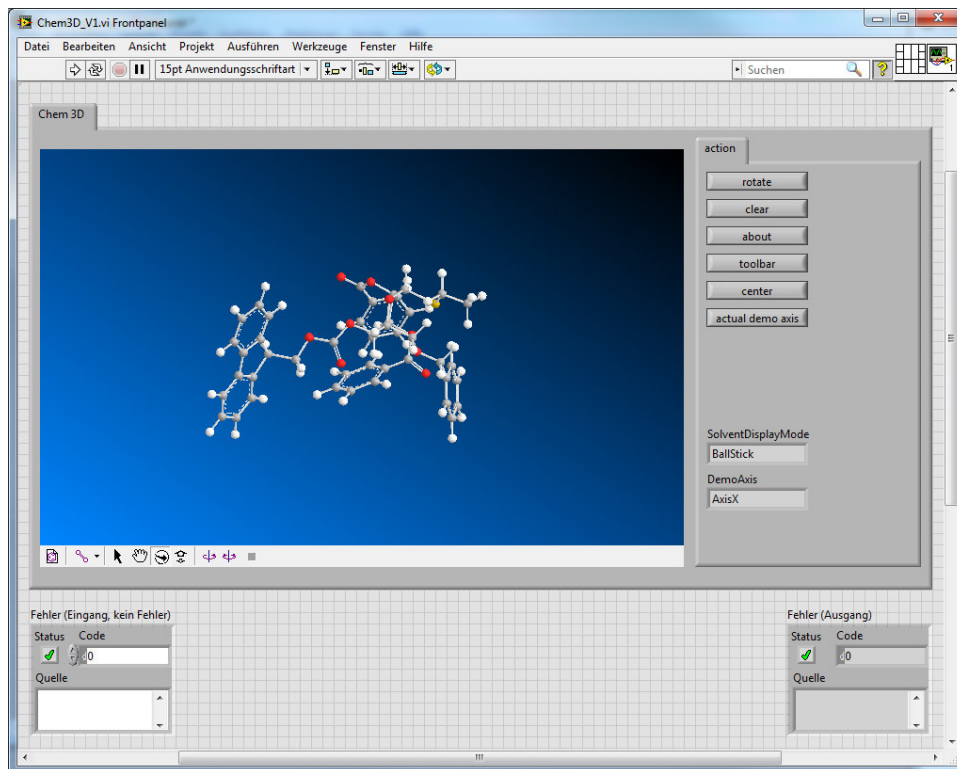


Abbildung 154: Visualisierungsmöglichkeit der chemischen Komponenten innerhalb der Synthesizer Umgebung

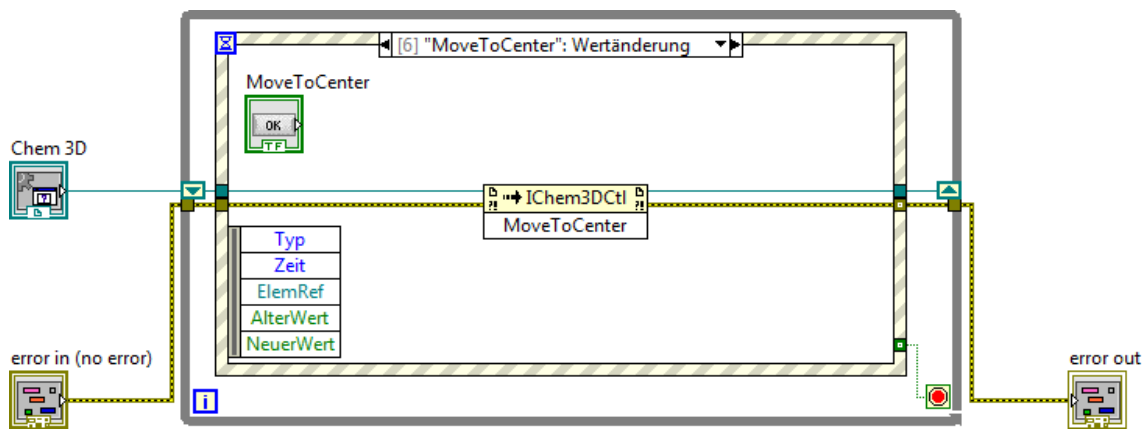


Abbildung 155: LabVIEW Beispielcode hinter der Schaltfläche „center“

Bei der Verwendung von Reaktionspfeilen in ChemDraw, kann nun durch die Zuordnung der Bestandteile einer Reaktionsgleichung zu einzelnen Objekten, die Synthesizer-Steuerung erfolgen.

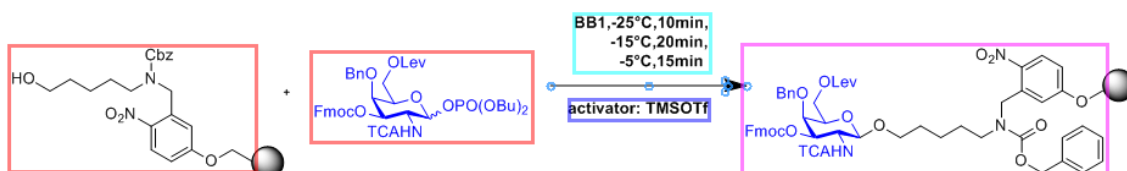


Abbildung 156: Zuordnung der einzelnen chemischen Komponenten als „Funktionssteuerung“

Hierzu stehen über die Objekteigenschaften diese Informationen zur Verfügung.

ICChemDrawReactionScheme	
Arrows	▶
Intermediates	▶
Plususes	▶
Products	▶
Reactants	▶
ReactionSchemeType	▶
ReactionSteps	▶

Abbildung 157: Objekteigenschaften der Komponenten

2.18. Zugriff auf ChemFinder aus LabVIEW

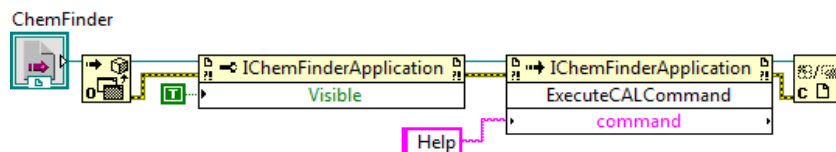


Abbildung 158: Zugriff auf ChemFinder aus LabVIEW

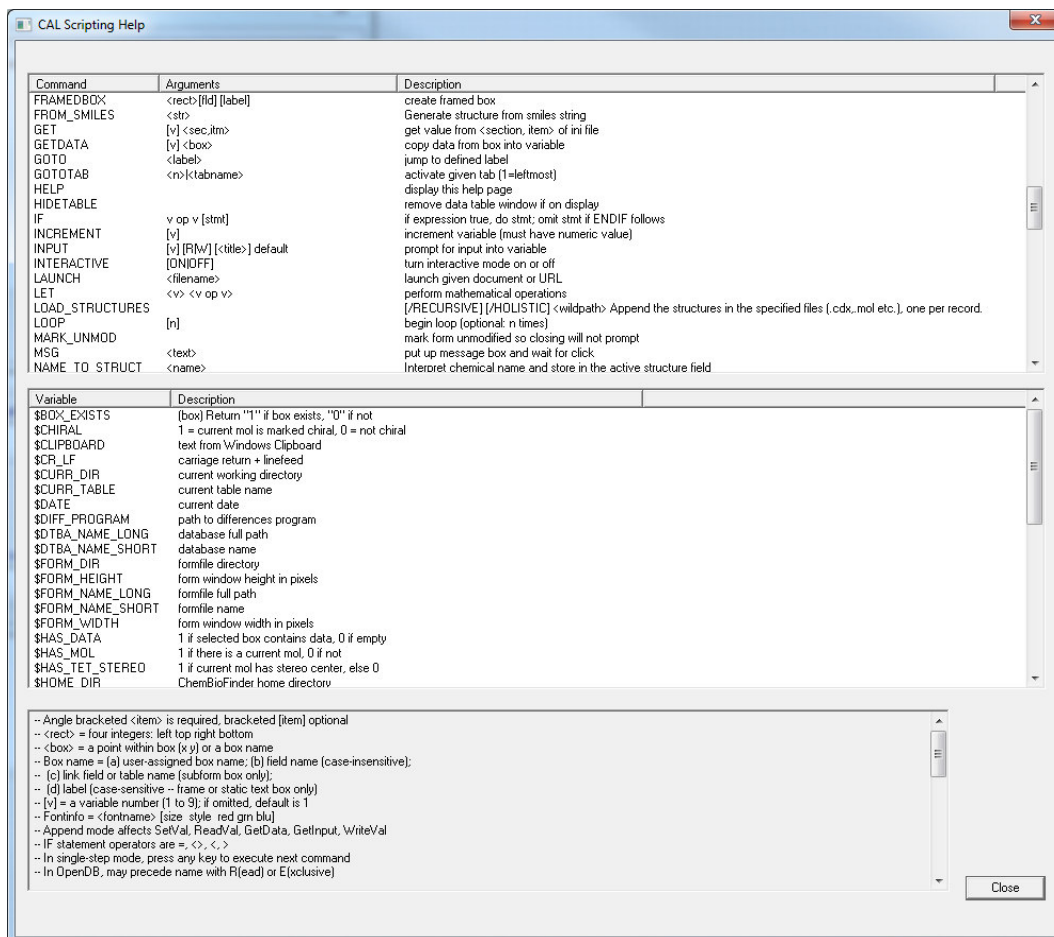


Abbildung 159: Zugriff auf ChemFinder Automation Language (CAL)

3. Fehlerbilder - Ursachen

3.1. Rückschlagventile, Spritzenpumpe

Ein Fehlerbild bei der Flüssigkeitsführung zeigt sich im nachfolgenden Bild:



Abbildung 160: Verstopftes Rückschlagventil im geöffneten Zustand (li.) und re. komplett demontiert

Durch die Volumenkalibrierung zum Anfang einer Synthese, können auch schleichende Veränderungen in der Systemstruktur registriert werden. Obwohl die Rückschlagventile immer mit einem Versorgungsdruck seitens des Schutzgases beaufschlagt sind, kommt es im Zeitrahmen von ca. 6-8 Monaten zu Reaktionen im Ventil, welche die Funktion beeinträchtigen können. Dieser Effekt wirkt sich jedoch zuvor in einer Verringerung der Volumenzulieferung bei gleichem Druck aus. Um hier ein Versagen dieser Schutzeinrichtung zu vermeiden, sollte in einem solchem Fall das jeweilige Ventil ausgebaut und gereinigt werden.

Dies ist unter Verwendung von Ultraschall im Wasserbad, sowie Spülen mit Aceton und abschließendem Trocken möglich. Danach sollte eine Funktionskontrolle durch Direktanschluss am Schutzgas beidseitig erfolgen.

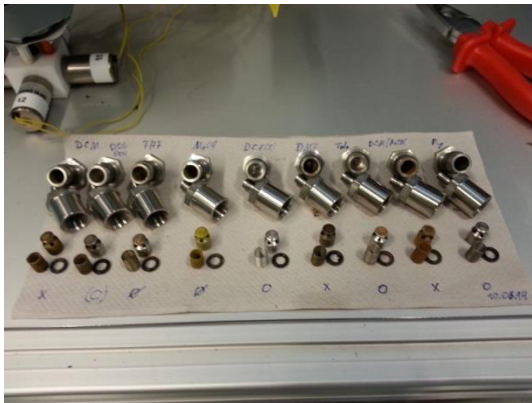


Abbildung 161: Wartungskontrolle der Rückschlagventile

- O vollwertig funktional
- (O) leichte Spuren
- Ø stärkere Kontaminierungen
- X eingeschränkte Funktionalität

Bei ungeeigneter Prozessführung, kann es zum Zurückdrücken in die Reservoirs kommen. Solcher Fehlerbilder sollen aufgrund des Simulationslaufes zukünftig ausgeschlossen sein.



Abbildung 162: Kontamination der Deprojektorenlösung



Abbildung 163: Verunreinigungen in der Transportspritze

Obwohl durch die Verwendung der Pufferloops das Eintragen von Reaktanten vermieden wurde, kommt es selten zu Kontamination der Spritze. Über die Parameter bei dem *RATE()*- Befehl kann für die einzelnen Pumpen jeweils individuelle Dosier- und Befüllungsflüsse definiert werden. Welche Auswirkung eine

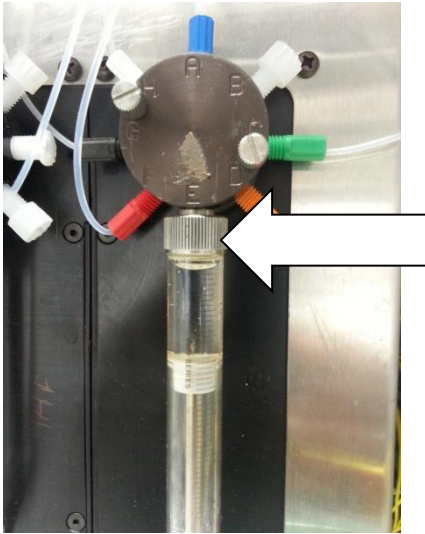


Abbildung 164: Gasbildung bei zu schneller Befüllung

zu hohe Befüllungsrate bei einer erhöhten Raumtemperatur haben kann zeigt das nebenstehende Bild.

Bei allen diesen aufgezeigten Fehlerbildern zeigt sich wie wichtig eine visuelle oder elektrische Kontrollfunktion ist, denn jeder Effekt verschlechtert das Gesamtergebnis der Synthese.

3.2. Fehlerbeseitigung - Kloehngeräte

Hierbei konnte die Endposition nicht ordnungsgemäß angefahren werden und erst nach Entnahme und Reinigungswartung wieder volle Funktionsbereitschaft hergestellt werden.

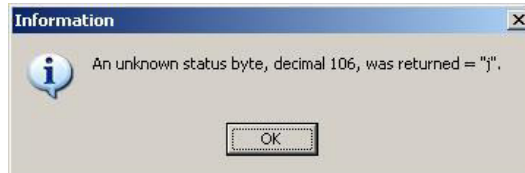


Abbildung 165: Fehlerrückmeldung der Spritze 1

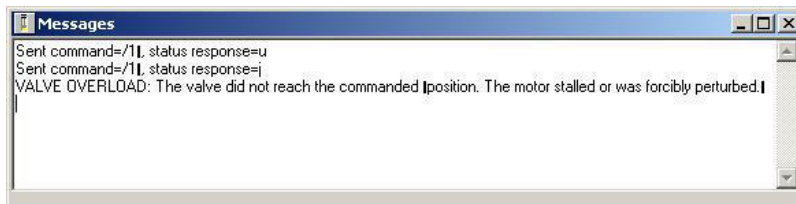


Abbildung 166: Textbasierte Informationsbereitstellung

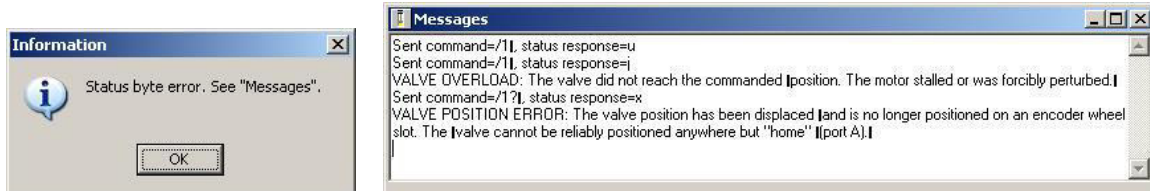


Abbildung 167: Fehlermeldung 2 und textbasierte Rückmeldung

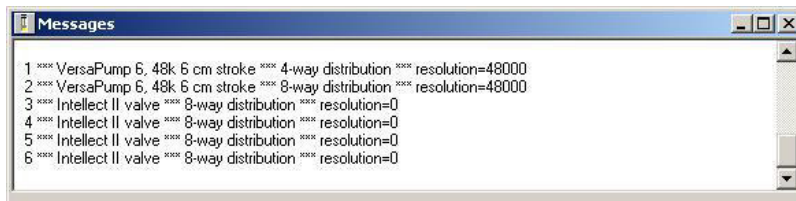


Abbildung 168: Falsche Pumpe an Adresse 1 erkannt

Als weiteres Beispiel einer Fehlerbeseitigung (Abbildung 169) sieht man im nachfolgenden Nachrichtendialog, dass das auf Adresse 3 angeschlossene Gerät bei der Initialisierung nicht erkannt wurde und somit nicht für die Ansteuerung zur Verfügung steht.

Die Kontrolle der Kommunikationsschnittstelle zeigte, dass der Verbindungsstecker nicht mehr aufgesteckt war.

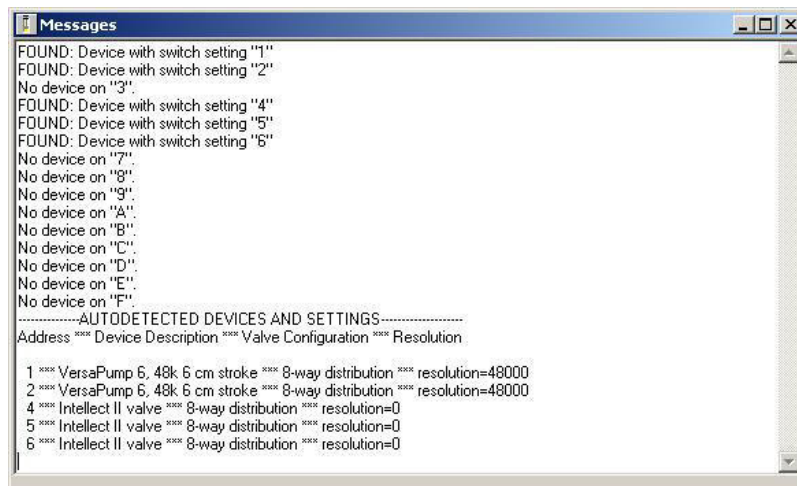


Abbildung 169: Fehler an Adresse 3

Wenn wie im nächsten Beispiel gezeigt gleich zwei Geräte bei der Erkennungsprozedur nicht aufgelistet werden, ist die Spannungsversorgung zu kontrollieren. Denn über die aktuell verwendeten Netzteile werden gleichzeitig zwei Geräte versorgt. Ein weiteres Indiz dieser Ursache lässt sich aus der LED- Anzeige der Geräte ableiten.



Abbildung 170: Fehlerbild

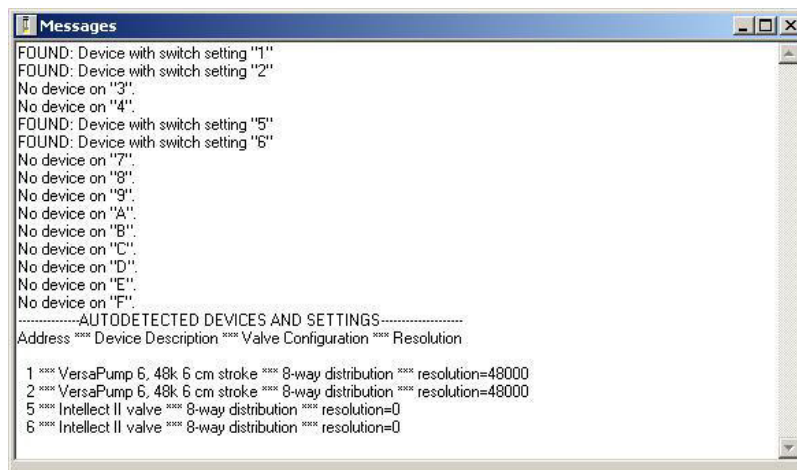


Abbildung 171: Gerätescan

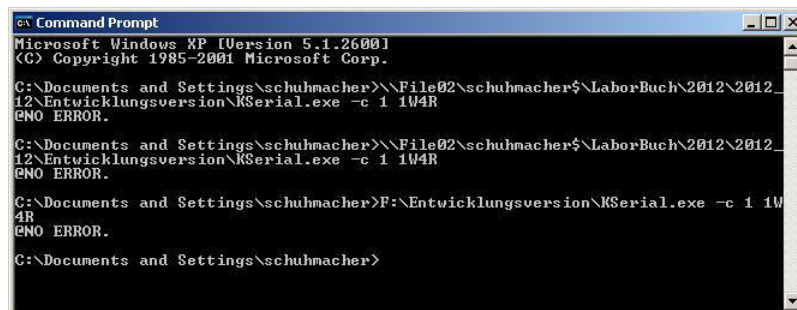


Abbildung 172: direkter Zugriff mit Command Prompt

Wie in diesem Beispiel gezeigt, ist der Textstring für gültig befunden worden, jedoch das Gerät immer noch in der Abarbeitung des dazugehörigen Befehls. Wohingegen in dem nachfolgenden Fall bei der Abfrage und der Position auch das Gerät sich in einem Zustand der Abarbeitungsbereitschaft befindet.

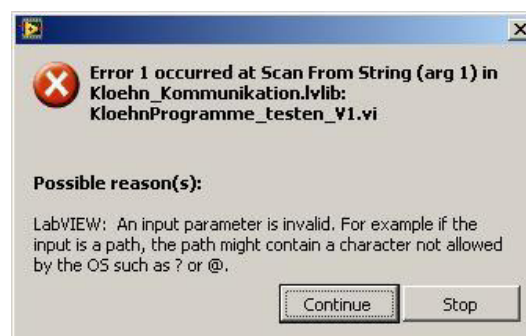


Abbildung 173: Fehlerbild 6

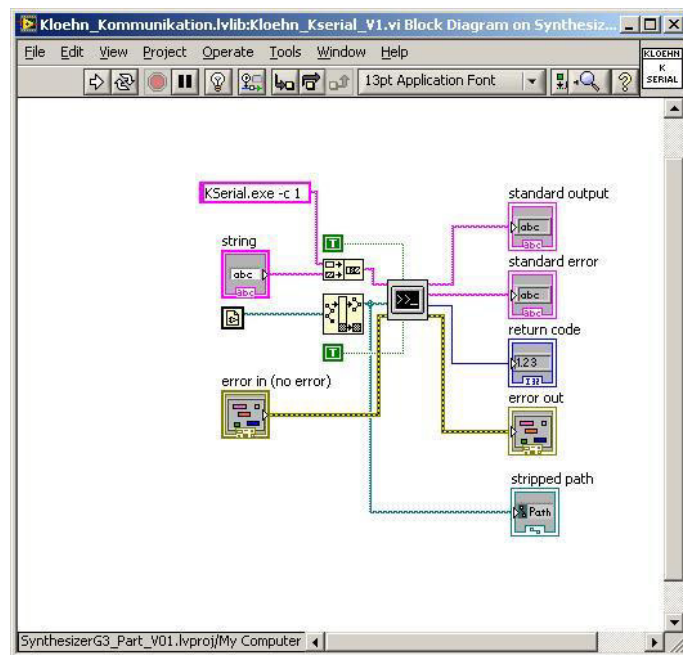


Abbildung 174: Parametererweiterung – c1

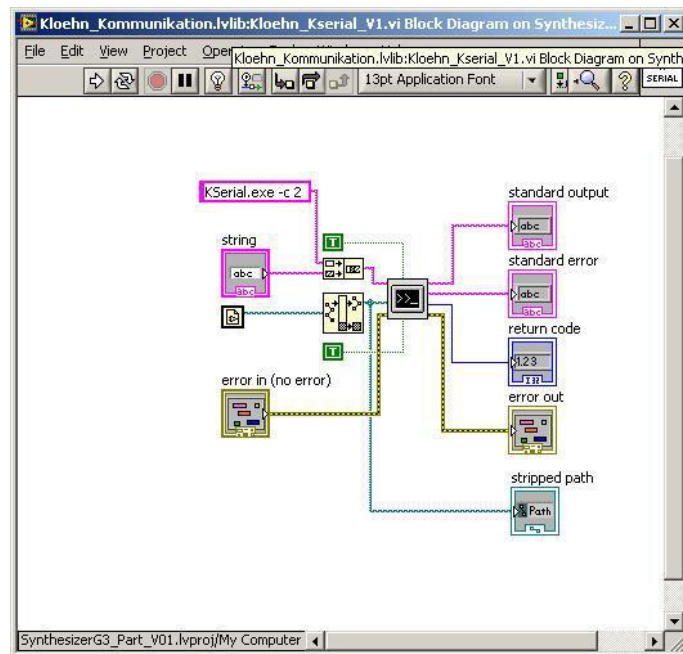


Abbildung 175: Parametererweiterung –c2

3.3. Fehler Fraktionssammler

Bei der Adaption des Gerätetreibers auf ein Altgerät (J-KEM-Synthesizer), ergab sich nach einigen Testsequenzen nachfolgendes Fehlerverhalten, welches sich durch einen Fehlerüberlauf innerhalb der Kommunikationspuffer reproduzieren ließ.

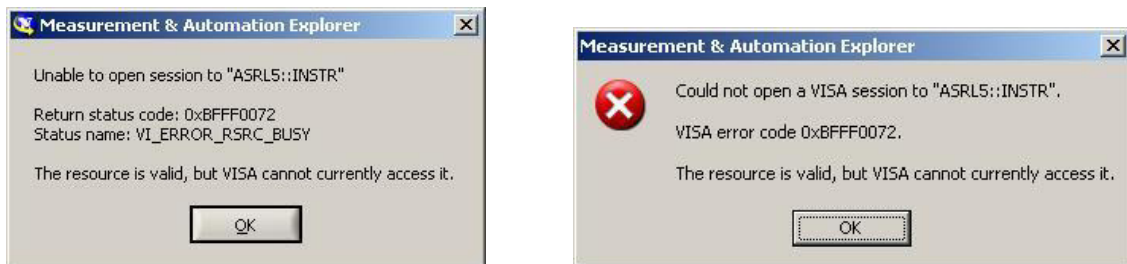


Abbildung 176: Fehlermeldungen bei Adaption auf weiteres Altgerät

Dieses Fehlerbild konnte, entsprechend auf eine veränderte Zählweise der Ansteuerungsschritte innerhalb des steuernden μ Prozessors, zurückgeführt werden. Nach der Anpassung der Programmierung auf die Bereitstellung über Initialisierungsdateien, in denen solche geräteindividuellen Verhaltensmuster hinterlegt sind, ist auch hier die Funktionalität vollständig hergestellt worden.

Positionsbestimmung Fraktionssammler

Y=50 Schrittweite:250

X=0 (Bei verschobenem Anschlag) Schrittweite:115

Tabelle 3: Initialisierungsdatei für gerätetypische Eigenschaften

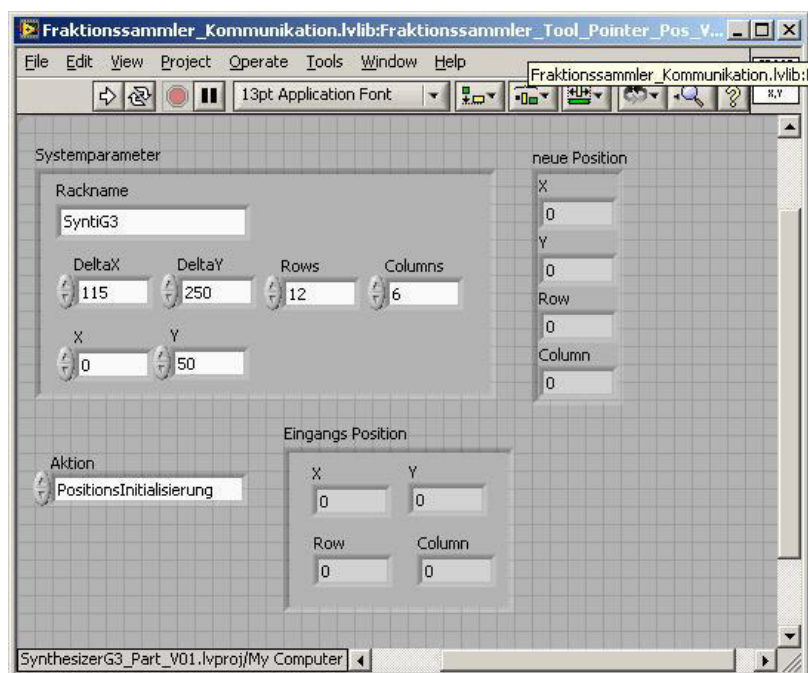


Abbildung 177: Ausgewertete Initialisierungsdatei

3.4. Fehler / Wartung – Temperiereinheit Julabo

Über die vergangenen drei Jahre, in denen drei vergleichbare Systeme vollwertig im Syntheseeinsatz verwendet wurden, haben sich auch hier verschiedene Fehlerbilder gezeigt, die nachfolgend kurz erörtert werden.

Der Julabo ist mit dem Fehlercode -41 in eine Fehlerschleife gegangen. Dieser Fehler signalisiert ein Überschreiten der internen Maximalbefüllung. Hierdurch ist keine weitere Temperaturführung mehr möglich. Es ist ein manueller Benutzereingriff notwendig, um hier Kühlmittel abzulassen. Leider gibt es in der aktuellen Version keine Möglichkeit den Füllsensor auszulesen, um hier eine Vorwarnung zu geben. Das Level muss manuell korrigiert werden und dann, mit dem zuletzt gesendeten Temperaturbefehl initialisiert, der Programmablauf weitergeführt werden.

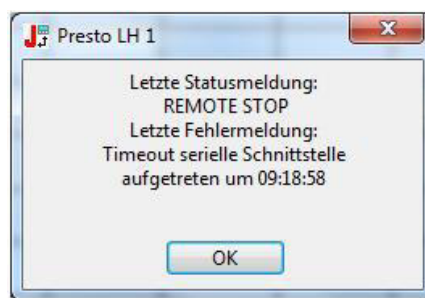


Abbildung 178: Fehlerbeseitigung

Wie kann es zu diesem Fehlerfall kommen?

Die durchschnittliche Normalbefüllung wird auf dem internen Display mit 2 Balken beim Füllstandslevel angezeigt. Bei extremen Temperaturvorgaben $< -40^{\circ}\text{C}$ kann der interne Minimalwert unterschritten werden und es somit notwendig sein, Kühlflüssigkeit während des Betriebs nachzufüllen.

Hierbei ist jedoch darauf zu achten, dass es bei erneuter Erhöhung der Temperatur oberhalb der Raumtemperatur eventuell zum angesprochenen Fehlerfall führen kann.

Nach mehrfacher Fehlbedienung aufgrund dieses physikalischen Sachverhaltes, wurden die Erweiterungsoptionen eines Vorrats- / Überlaufgefäßes an den Systemen implementiert.

Damit der Julabo effizient die Temperierungen im negativen Temperaturbereich durchführen kann, ist neben der gerätetechnischen Wartung, siehe Synthesizerwartung (Kühlrippenreinigung), die Wasserkühlung zur Unterstützung der Kühlleistung angeschlossen worden.



Abbildung 179: Verschmutzte Kühlrippen reduzieren die Kühlleistung erheblich

Hierdurch konnten die Temperierungszeiten im negativen Temperaturbereich erheblich reduziert (halbiert) werden. So ist mit dem aktuellen Aufbau das Erreichen von -70°C innerhalb von 45min von Raumtemperatur aus realisierbar.



Abbildung 180: Julabo erreicht -70°C

Da aufgrund des geplanten Optimizers zur Temperierung ein Huber-Kyrostat zur Verfügung stand, wurden die gleichen Tests und Ansteuerungsüberprüfungen auch hiermit vorgenommen.

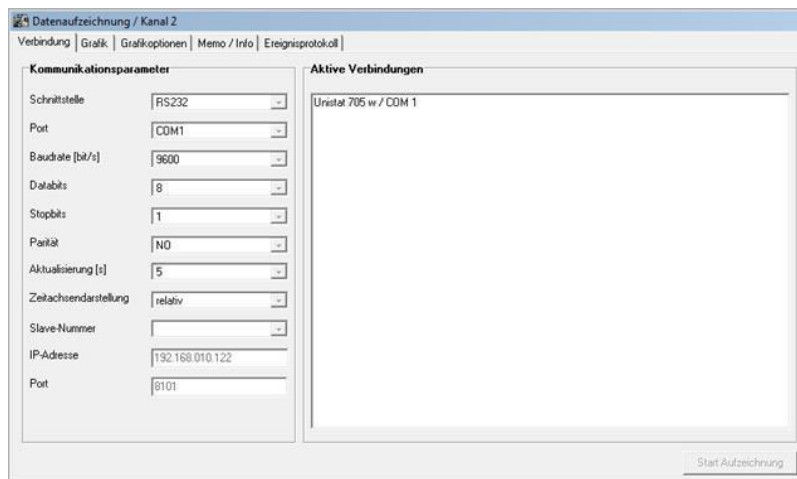


Abbildung 181: Geräteparametrierung Unistat

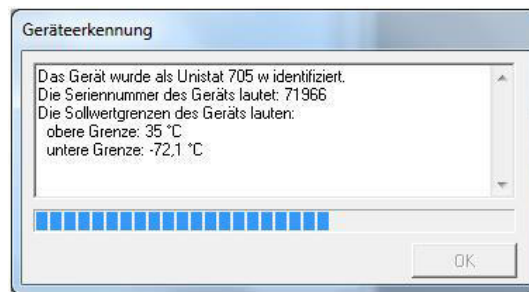


Abbildung 182: Unistat - Initialisierung



Abbildung 183: Erfolgreiche Gerätekommunikation

4. Lösungsmittel Beständigkeitstest für den kommerziellen Synthesizer

Lösungsmittel- Beständigkeitstest: Für die zukünftige mechanische Gestaltung des Geräte Chassis kommen verschiedene Oberflächen (Material, Beschichtung) in Frage. Diese wurden alle den verschiedenen zur Anwendung kommenden Lösungsmittel lokal beträufelt. Um hierbei nicht überproportional Abzugsfläche zu belegen, ist der gezeigte Aufbau zur gestapelten Unterbringung der 12 verschiedenen Platten realisiert.



Abbildung 184: Lösungsmittel – Beständigkeitstests

Die 12 verschiedenen Lösungsmittel wurden an zwei Tagen zyklisch im 1h Intervall 8 malig aufgebracht. Dieser Vorgang wurde nach 14 Tagen nochmalig wiederholt.



Abbildung 185: Lösungsmittel für den Test

Um die Lösungsmitteltropfen lokal begrenzen zu können, kamen V4A-Unterlagescheiben zum Einsatz.

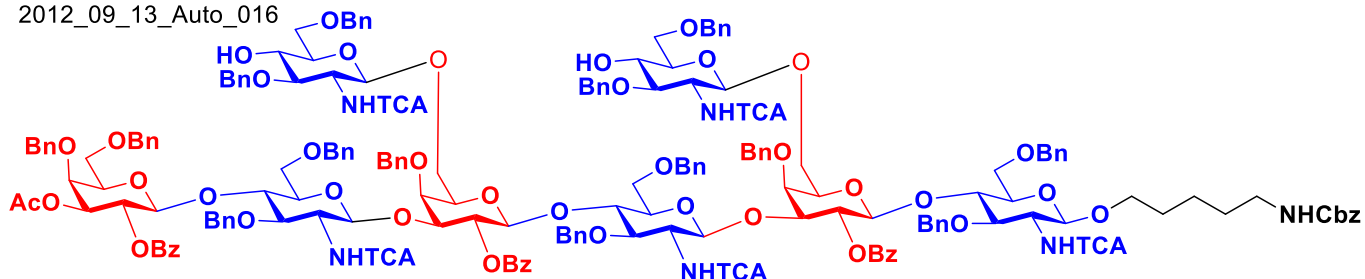


Abbildung 186: lokale Begrenzung der Lösungsmittel auf Testplatte

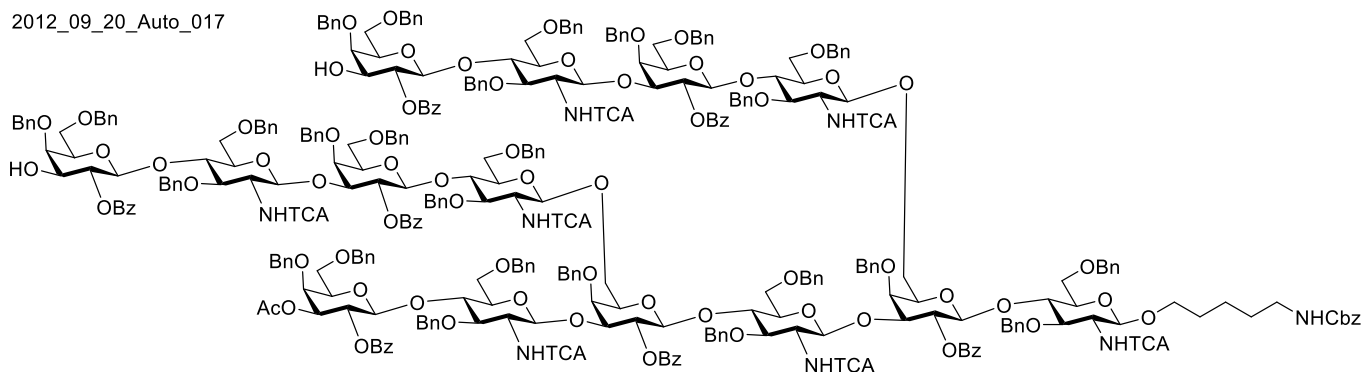
5. Liste der Syntheseprodukte aus den Arbeitssynthesizern

Nachfolgend eine temporäre Auswahl der Produkte aus den erfolgreich durchgeführten Synthesen der drei Arbeitssynthesizer.

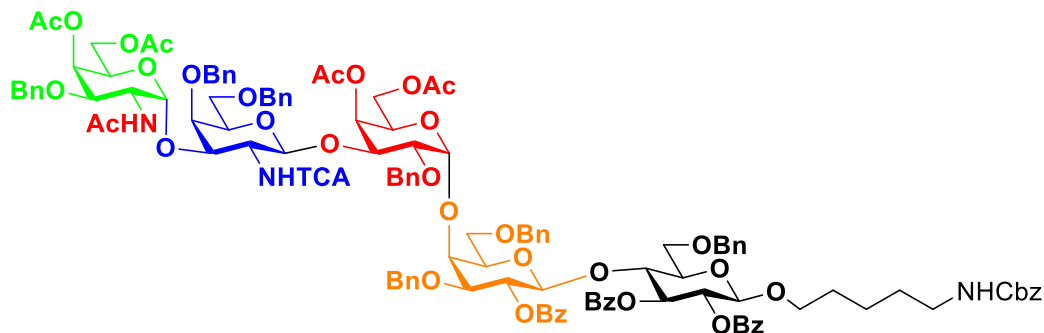
2012_09_13_Auto_016



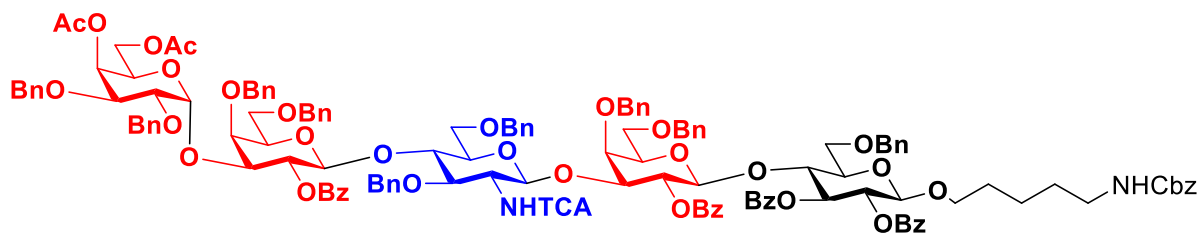
2012_09_20_Auto_017



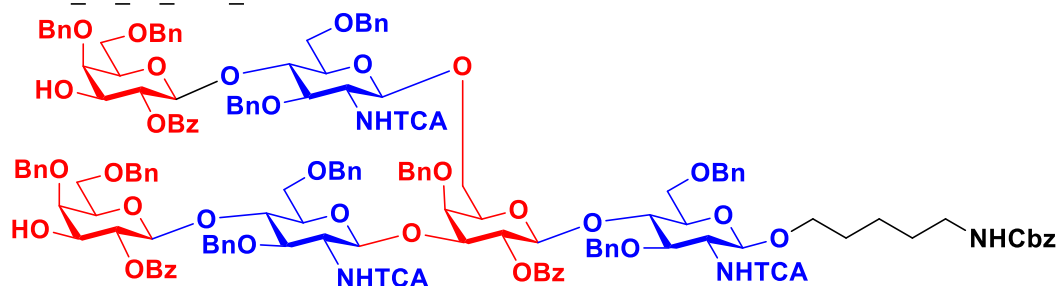
2012_12_11_Auto_Alpha_Gla



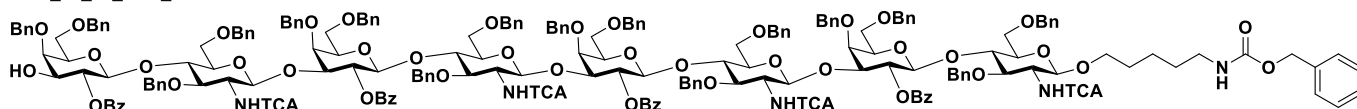
2012_12_11_Alpha_Gal_1



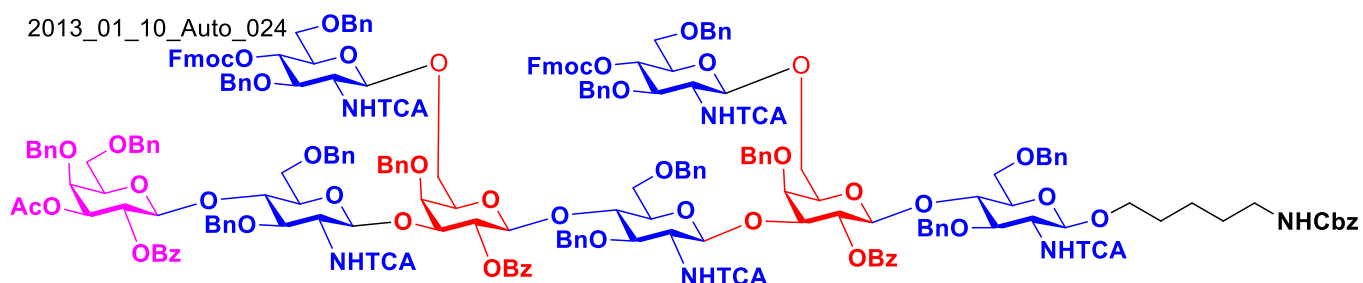
2013_01_08_Auto_023



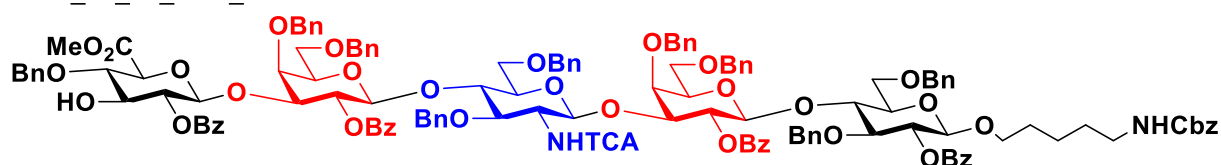
2013_01_09_Auto_022



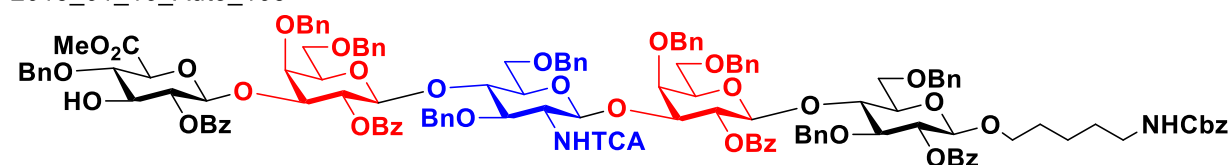
2013_01_10_Auto_024



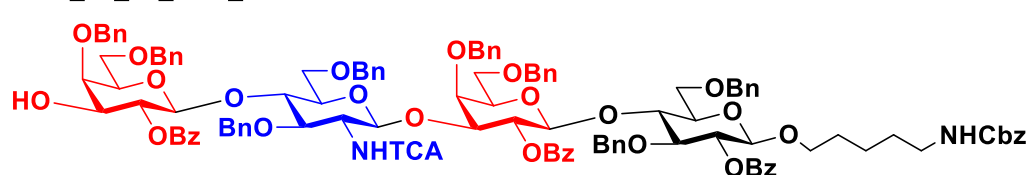
2013_01_16_Auto_100



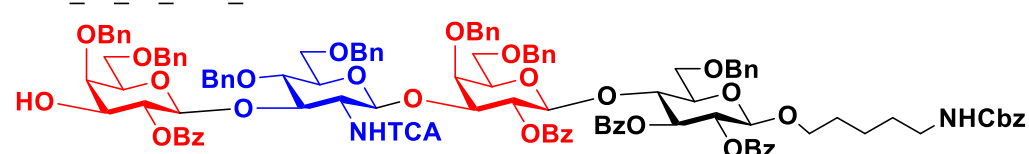
2013_01_16_Auto_103



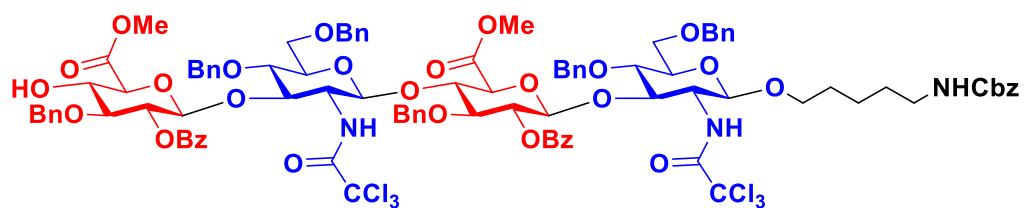
2013_01_17_Auto_101



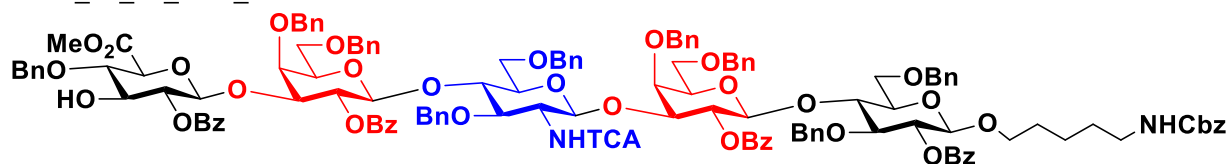
2013_01_18_Auto_102



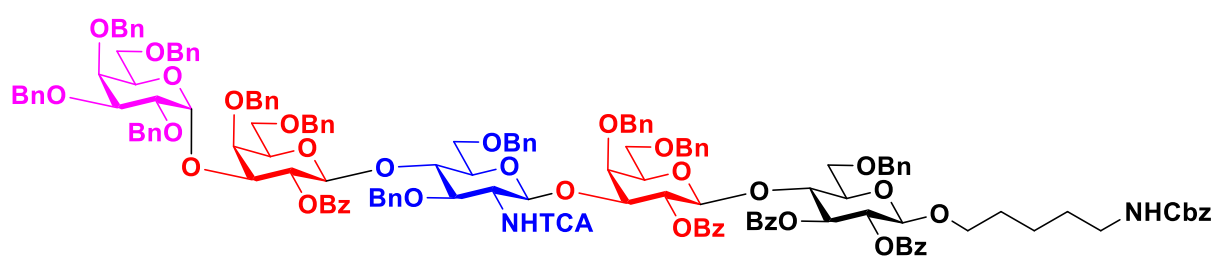
2013_01_31_Auto_051



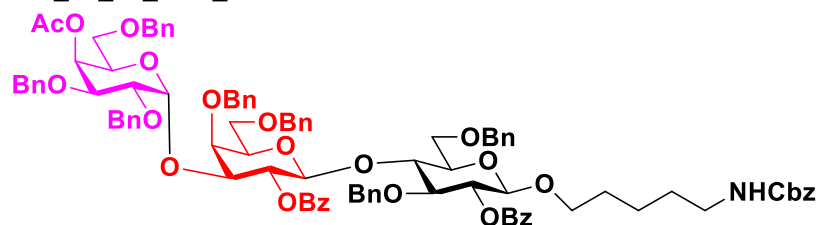
2013_02_12_Auto_104



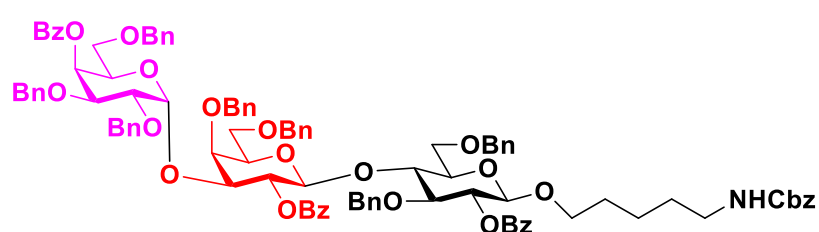
2013_02_13_Auto_120



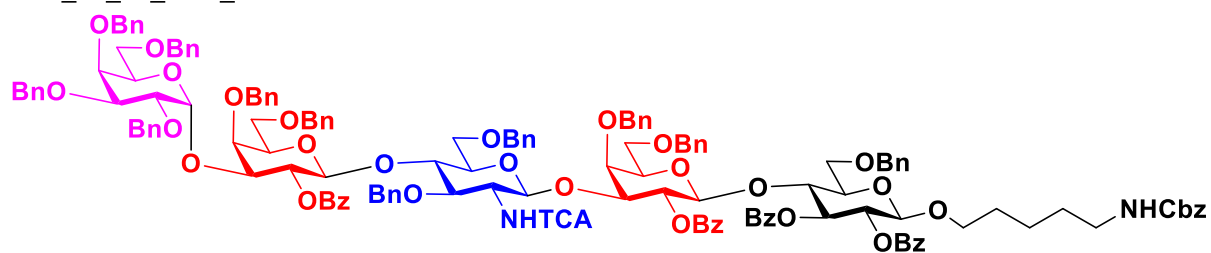
2013_02_14_Auto_111



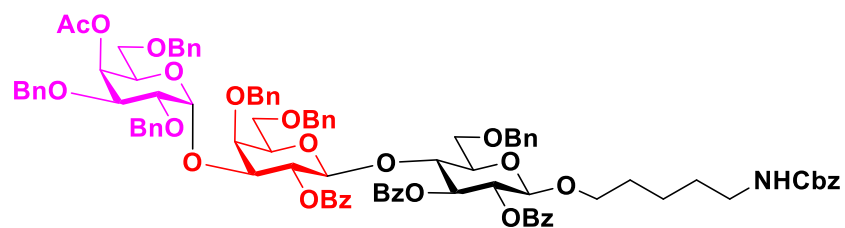
2013_02_14_Auto_112



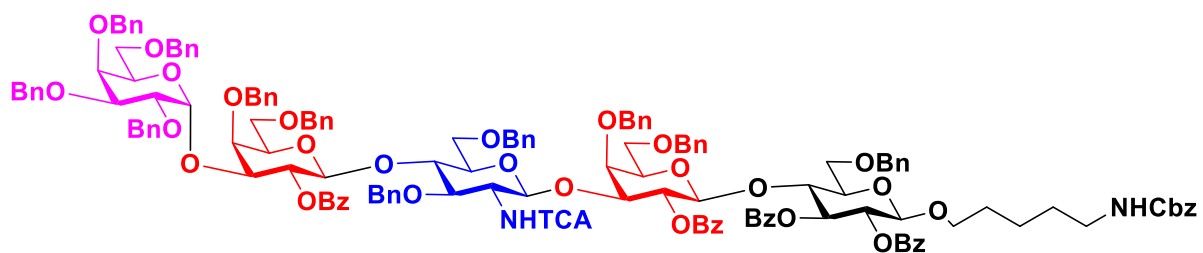
2013_02_19_Auto_121



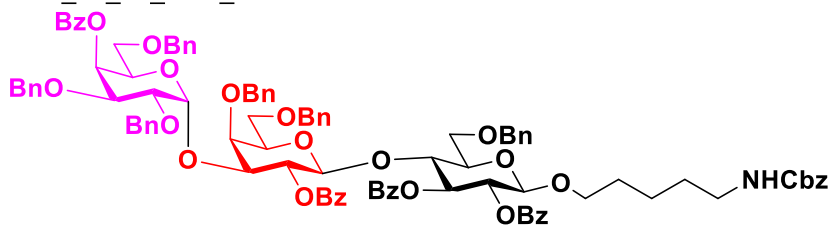
2013_02_20_Auto_113



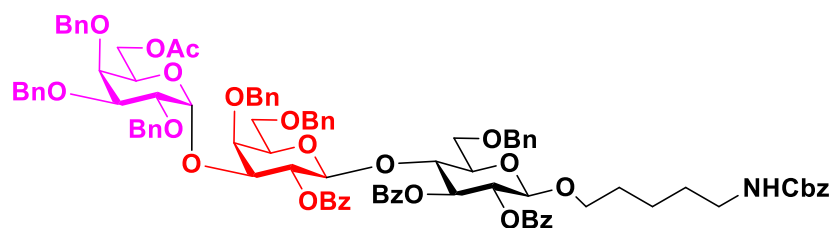
2013_02_20_Auto_122



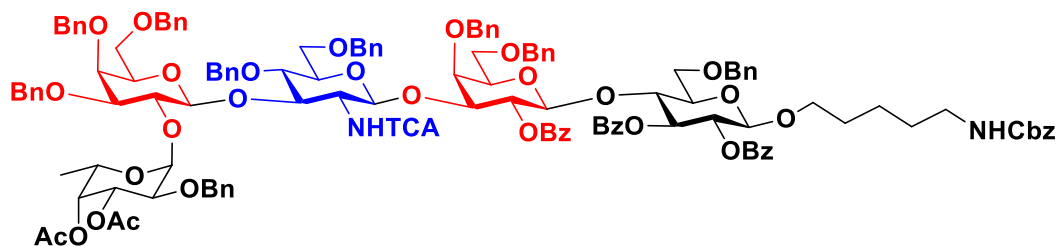
2013_02_20_Auto_114



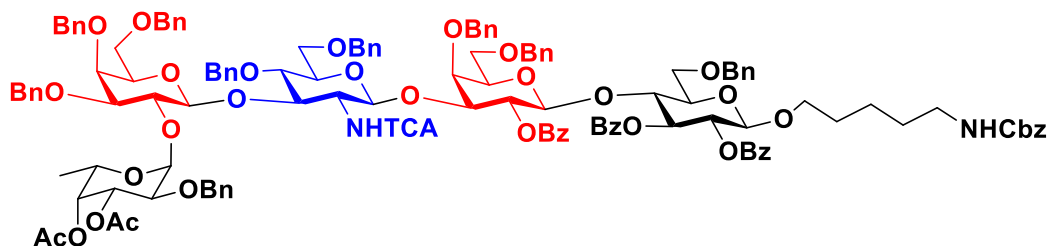
2013_02_21_Auto_116



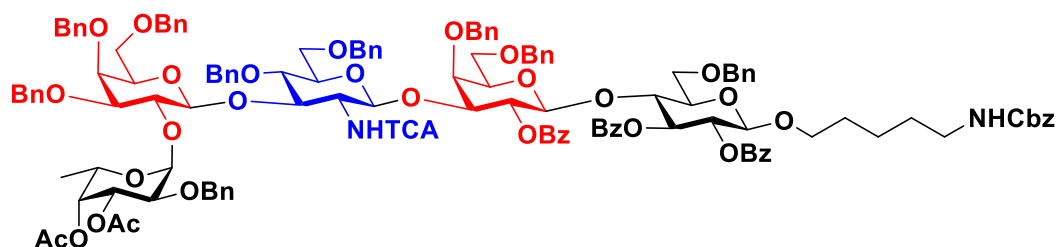
2013_03_05_Auto_124



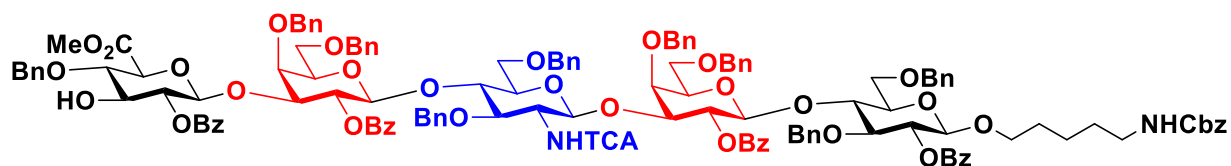
2013_03_06_Auto_125



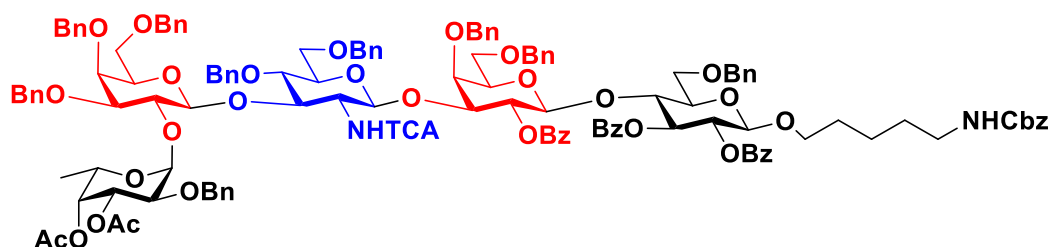
2013_03_06_Auto_126



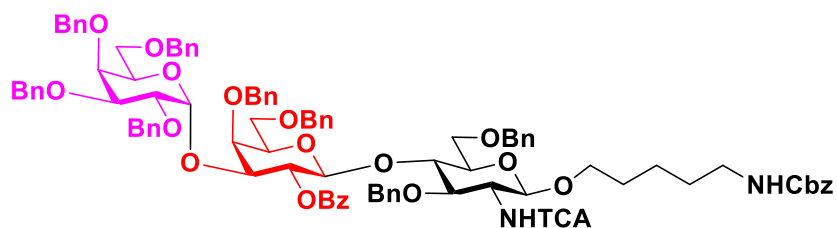
2013_03_07_Auto_105



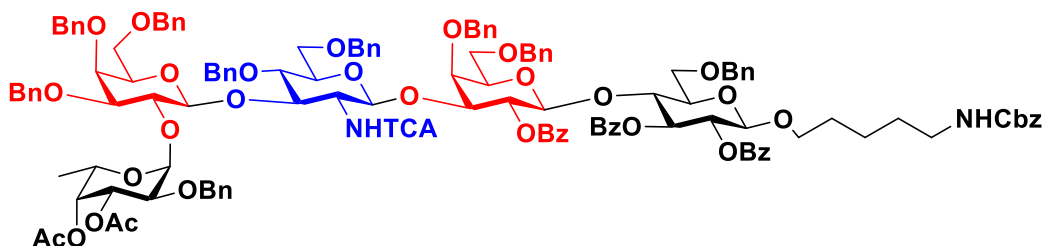
2013_03_15_Auto_127



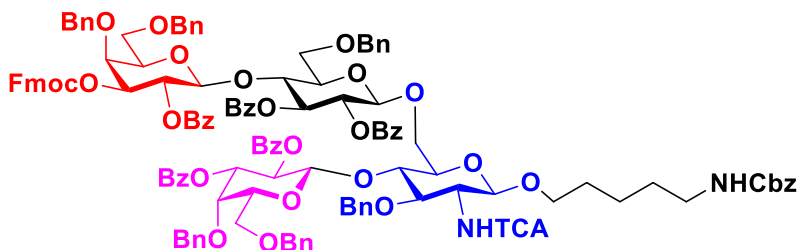
2013_03_18_Auto_115b



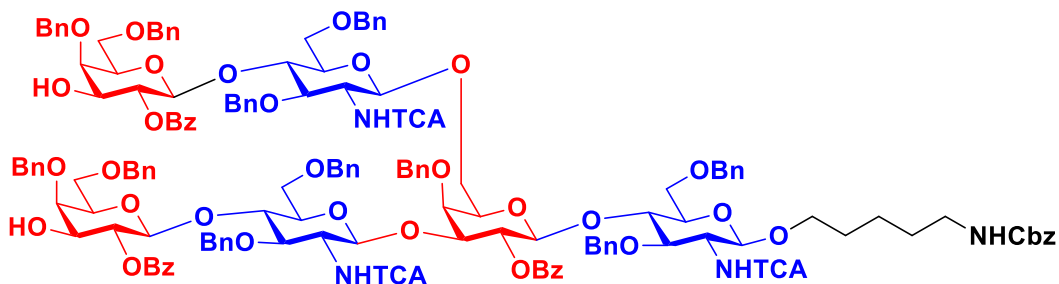
2013_03_27_Auto_128



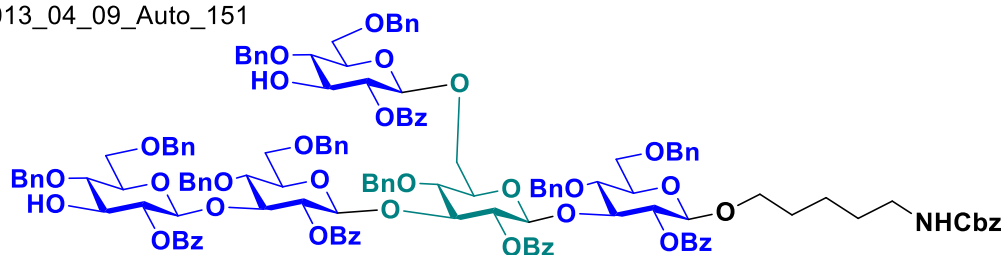
2013_03_28_Auto_201



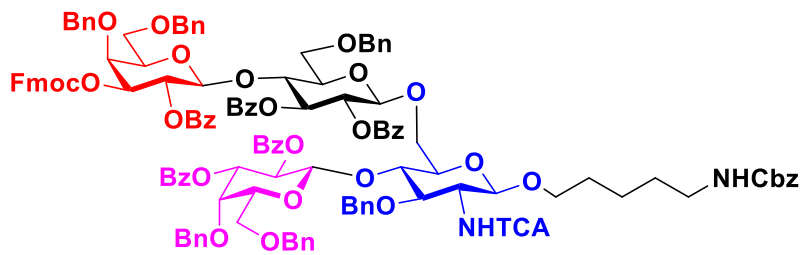
2013_04_08_Auto_027



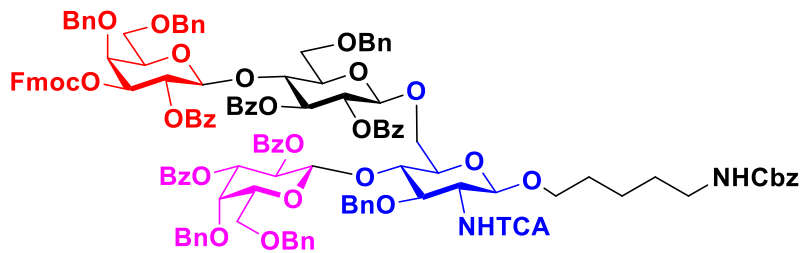
2013_04_09_Auto_151



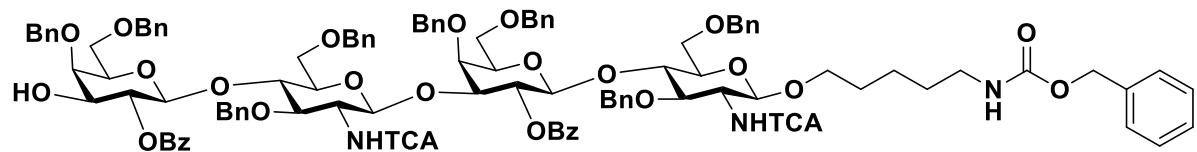
2013_07_01_Auto_202



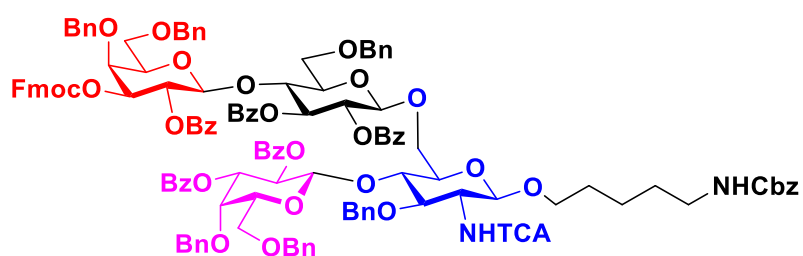
2013_07_01_Auto_202_I



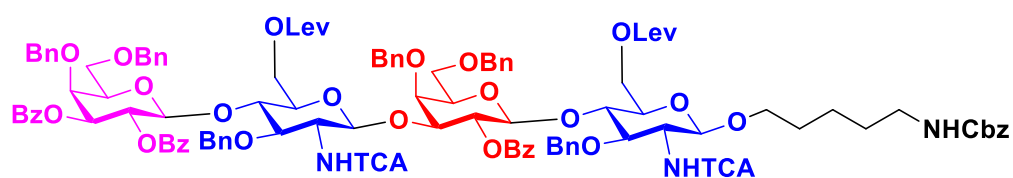
2013_07_02_Auto_028



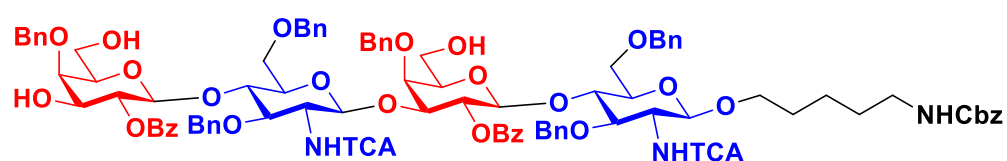
2013_07_22_Auto_204



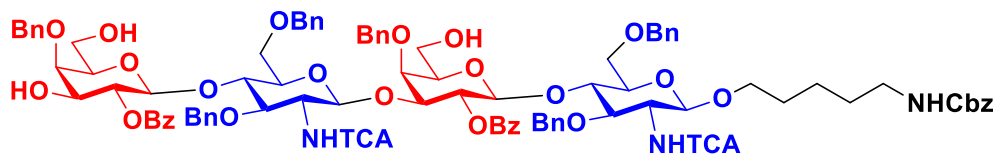
2013_07_23_Auto_040



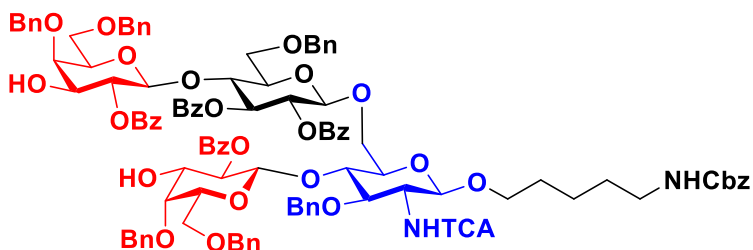
2013_07_24_Auto_041



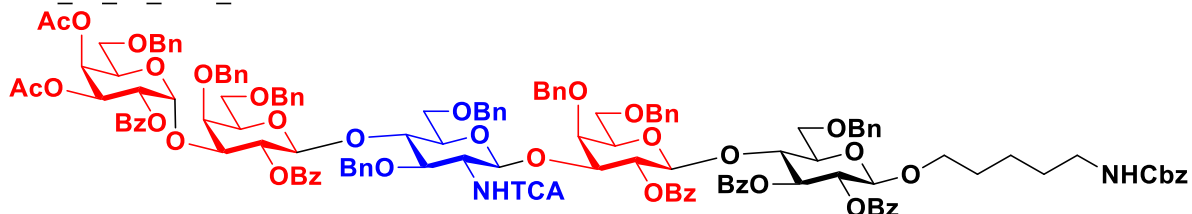
2013_07_24_Auto_042



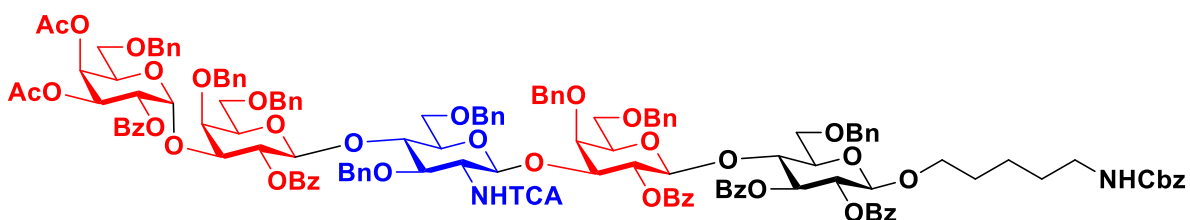
2013_07_25_Auto_205



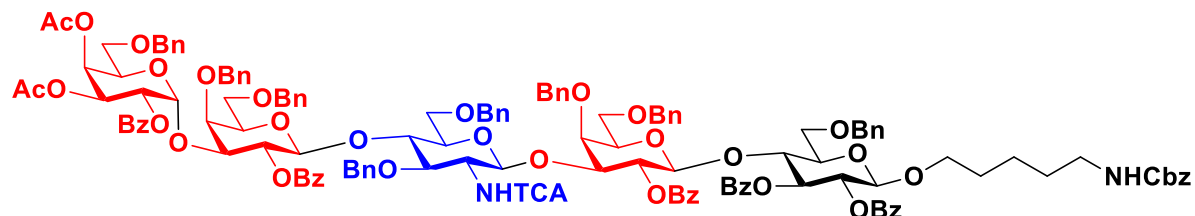
2013_07_26_Auto_129



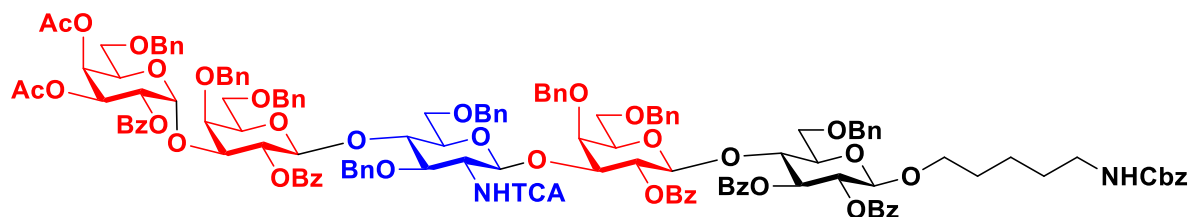
2013_07_27_Auto_105_2



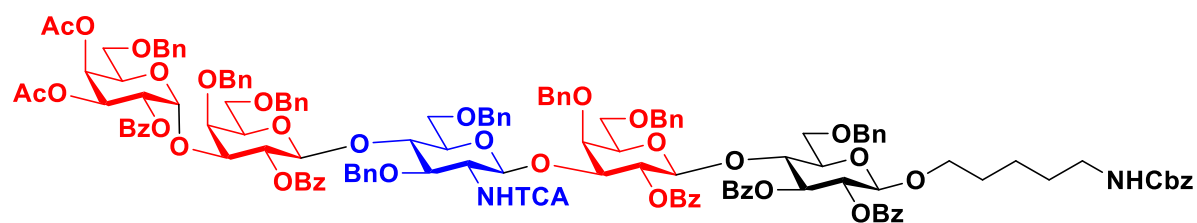
2013_08_01_Auto_106



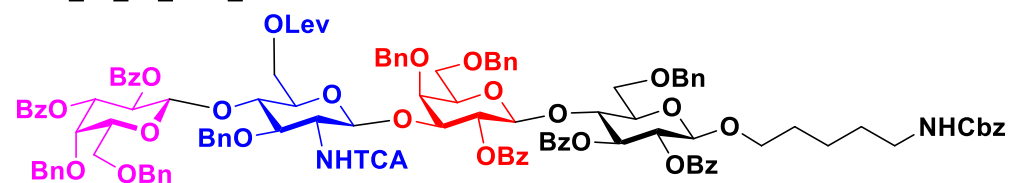
2013_08_11_Auto_107



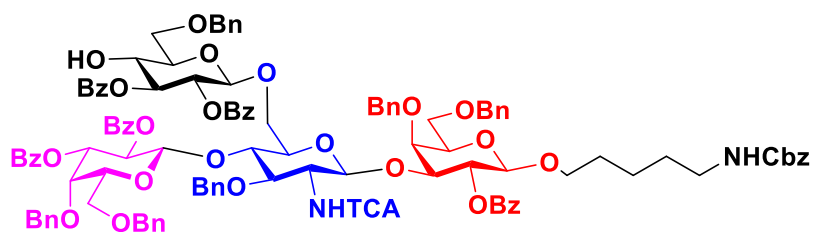
2013_08_12_Auto_108



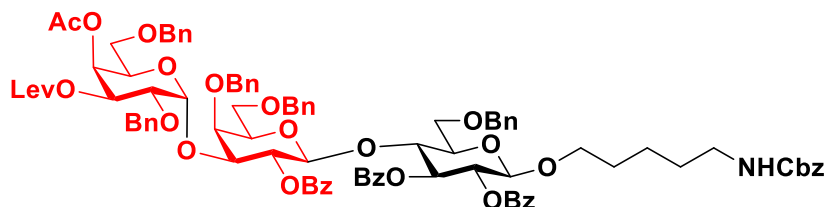
2013_08_13_Auto_207



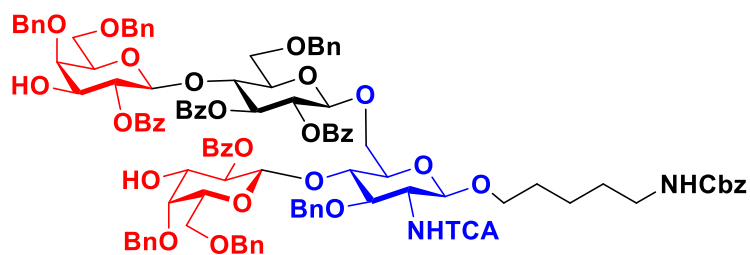
2013_08_13_Auto_206



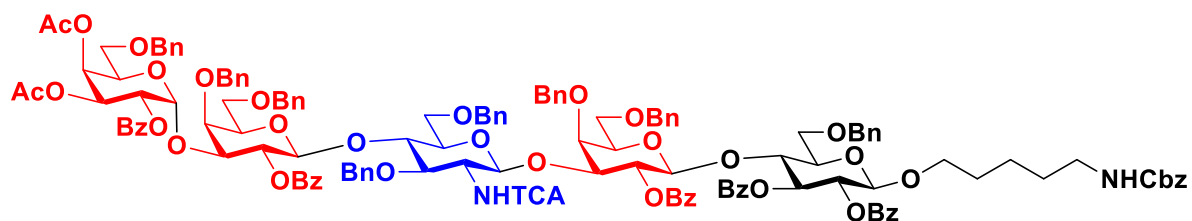
2013_08_15_Auto_131



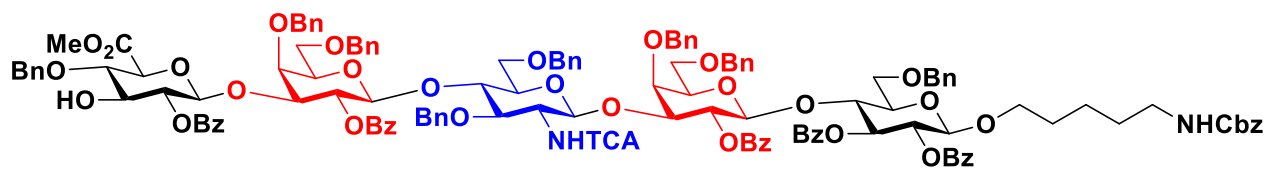
2013_08_21_Auto_208



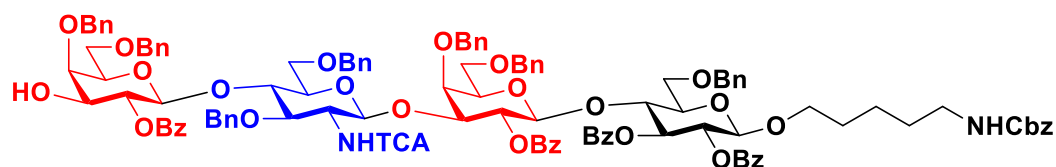
2013_08_22_Auto_129_a



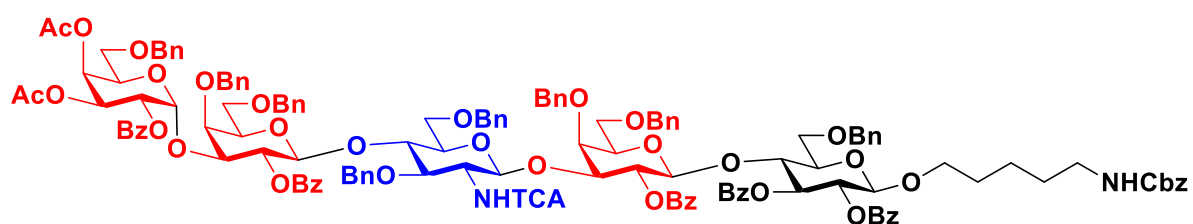
2013_09_05_Auto_098



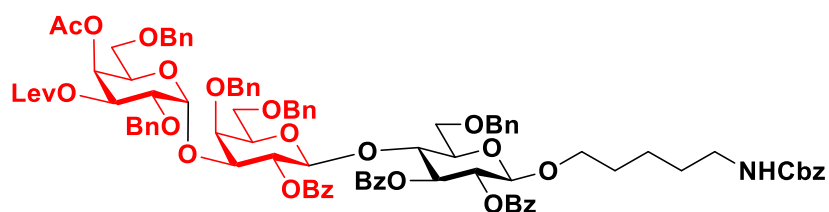
2013_09_06_Auto_099



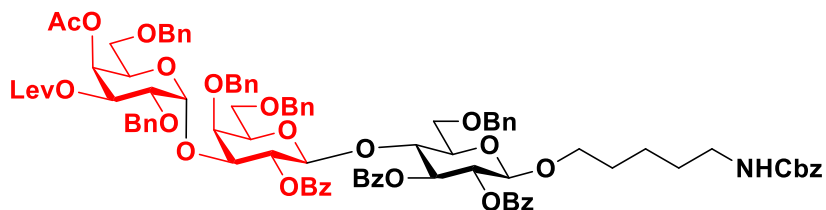
2013_09_07_Auto_129_a2



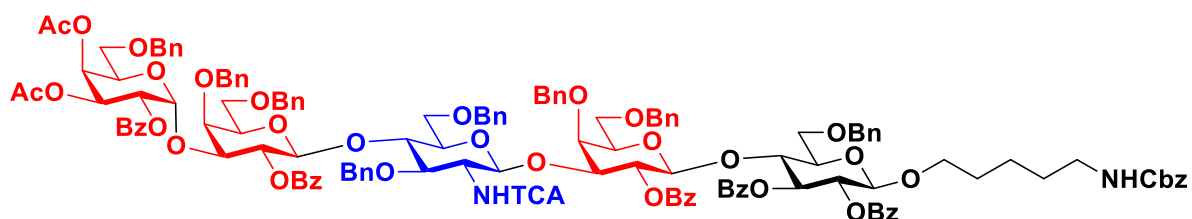
2013_09_09_Auto_133



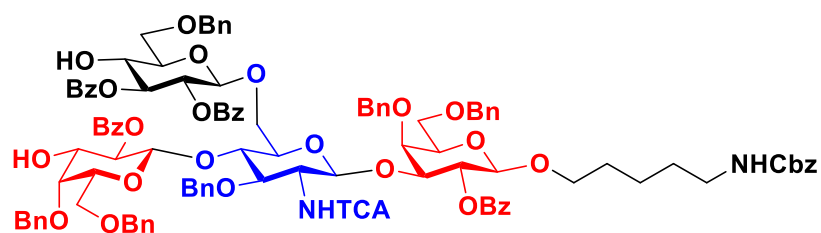
2013_09_10_Auto_132



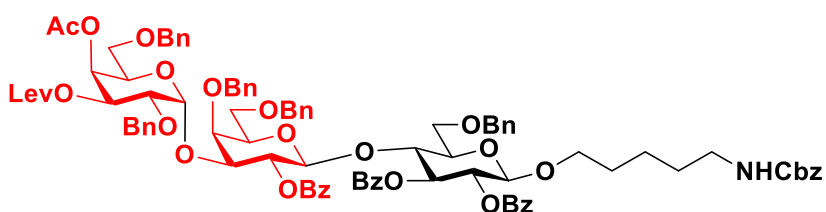
2013_09_11_Auto_109



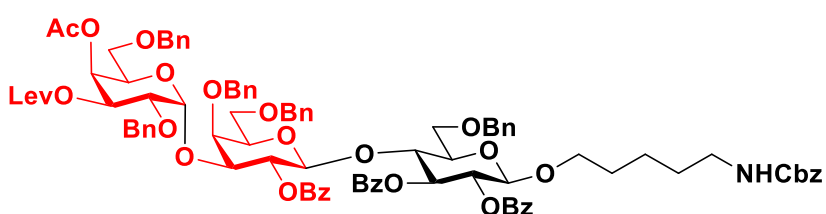
2013_09_13_Auto_209



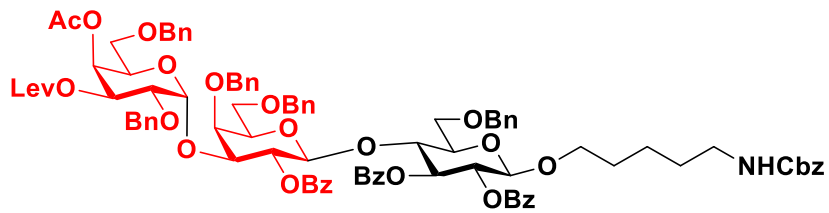
2013_09_20_Auto_161



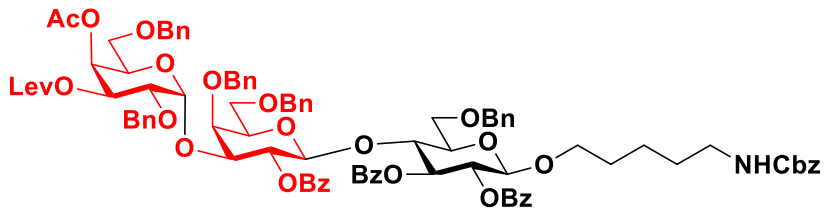
2013_09_20_Auto_164



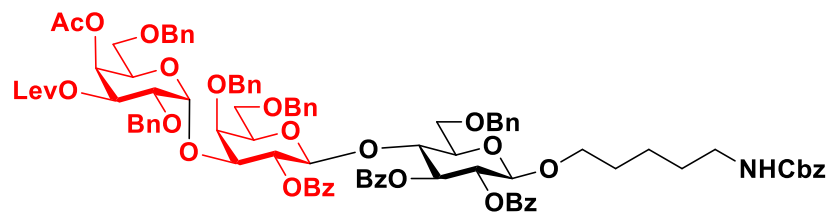
2013_09_21_Auto_162



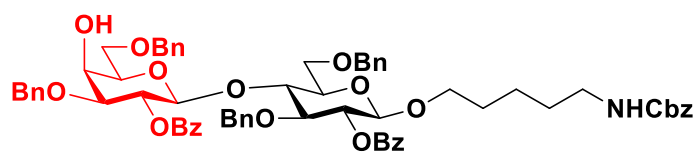
2013_09_21_Auto_163



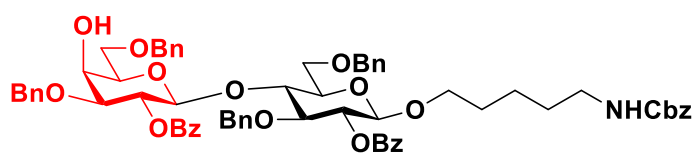
2013_09_21_Auto_165



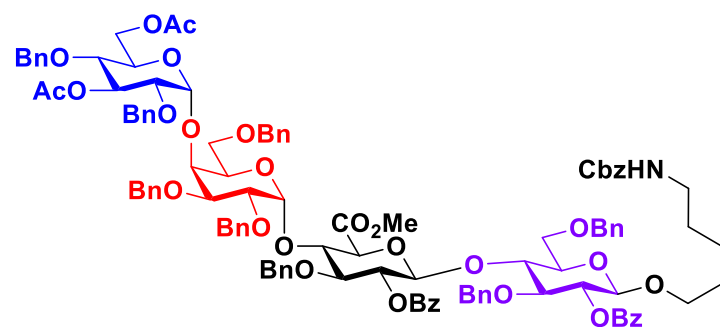
2013_09_23_Auto_141



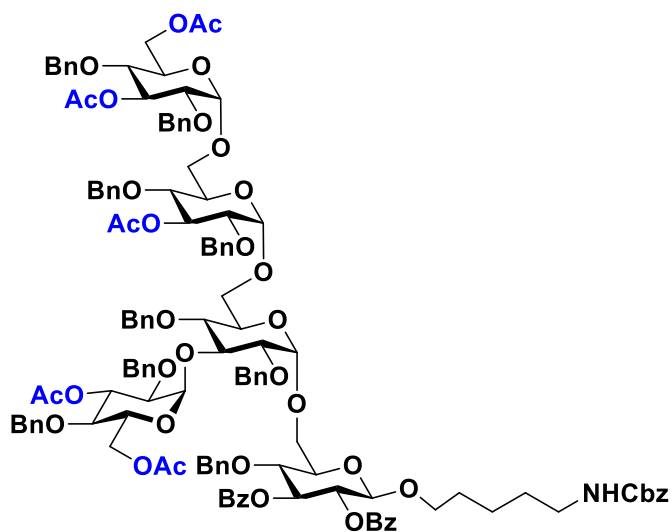
2013_09_24_Auto_142



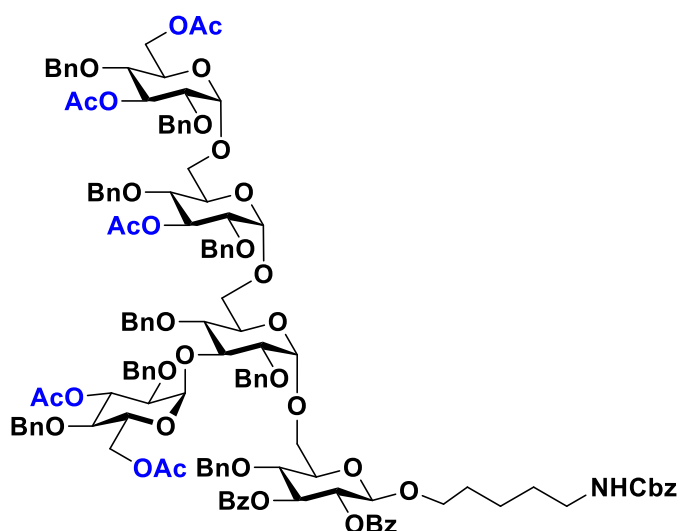
2013_09_30_Auto_210



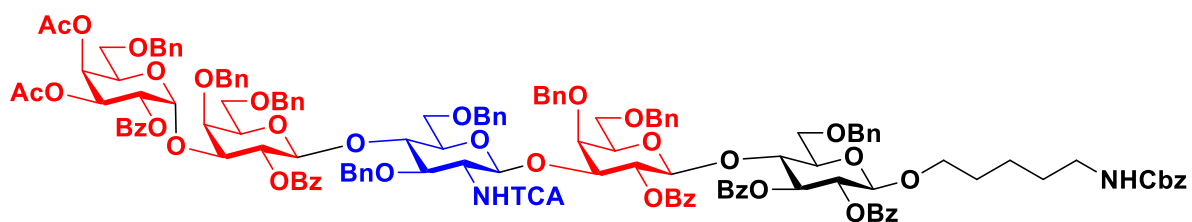
2013_10_01_Auto_174



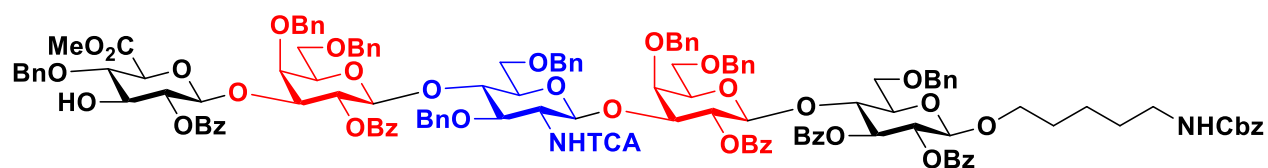
2013_10_02_Auto_175



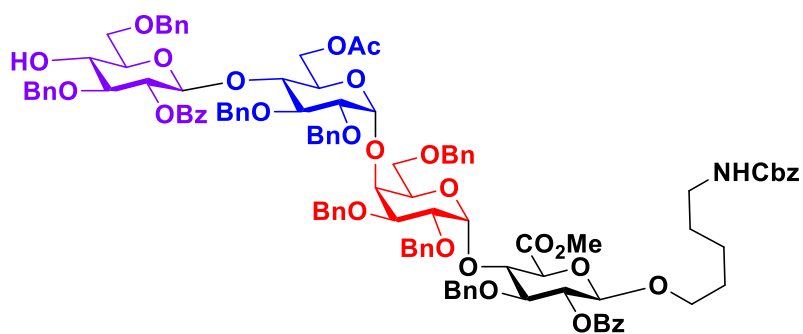
2013_10_07_Auto_129_a3



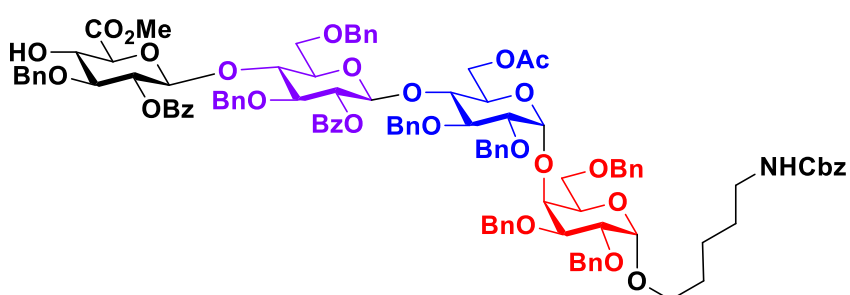
2013_10_08_Auto_109_AG_I



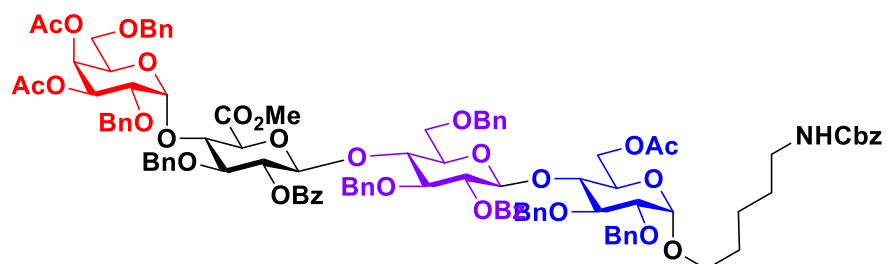
2013_10_09_Auto_211



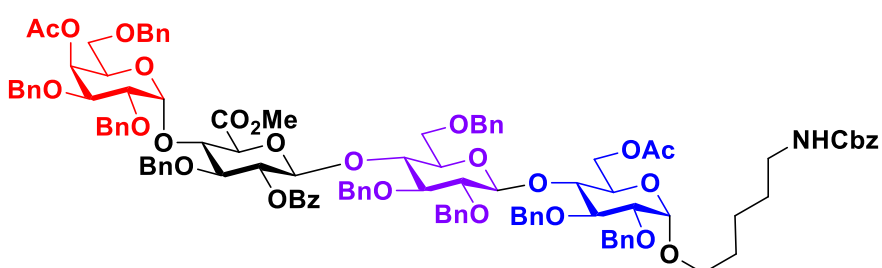
2013_10_10_Auto_212



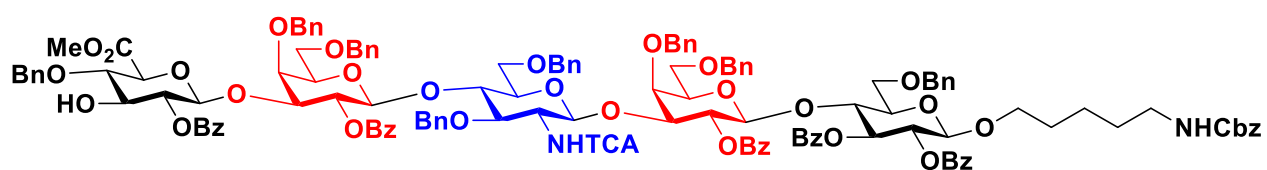
2013_10_10_Auto_213



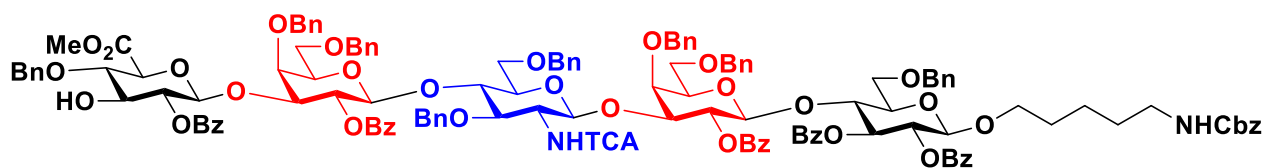
2013_10_11_Auto_214



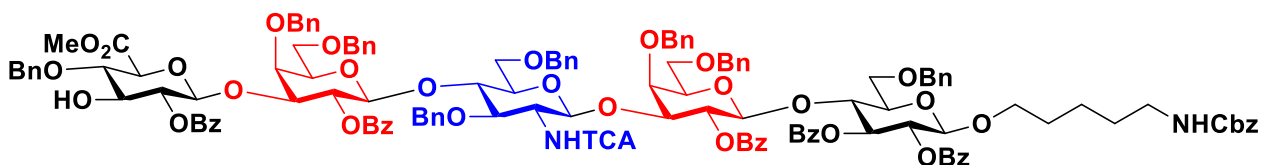
2013_10_23_Auto_109_AG_II



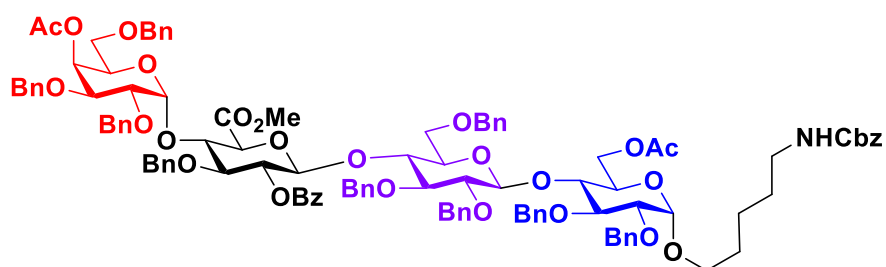
2013_10_25_Auto_109_AG_III



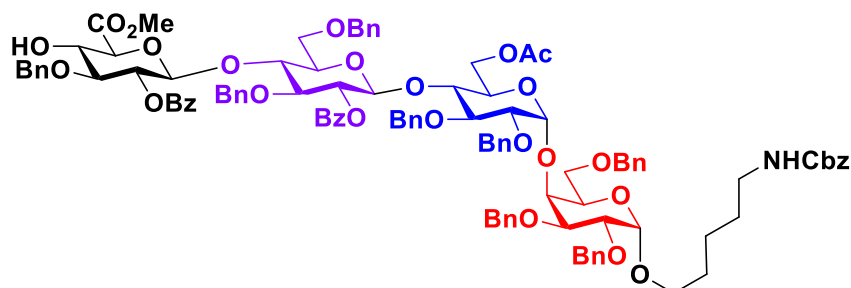
2013_10_26_Auto_109_AG_III



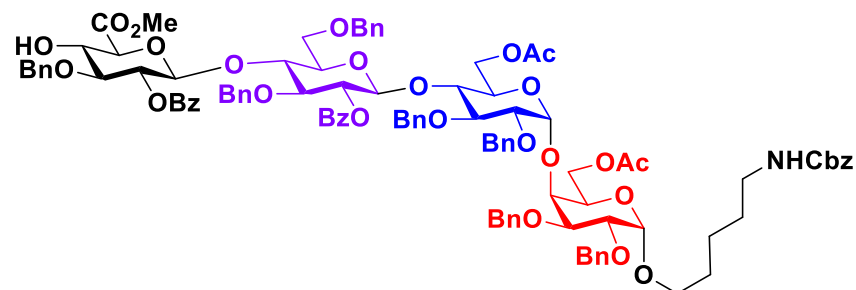
2013_11_18_Auto_215



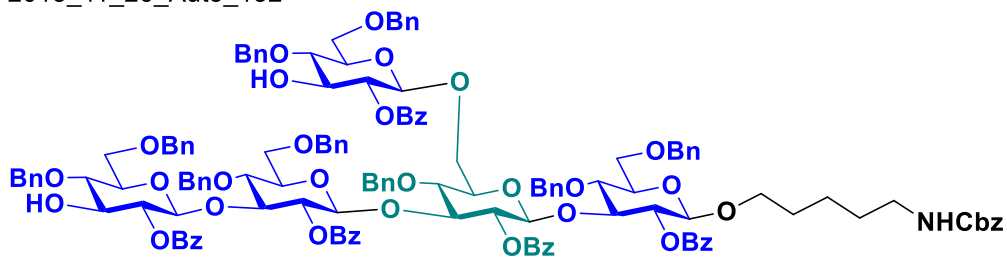
2013_11_19_Auto_216



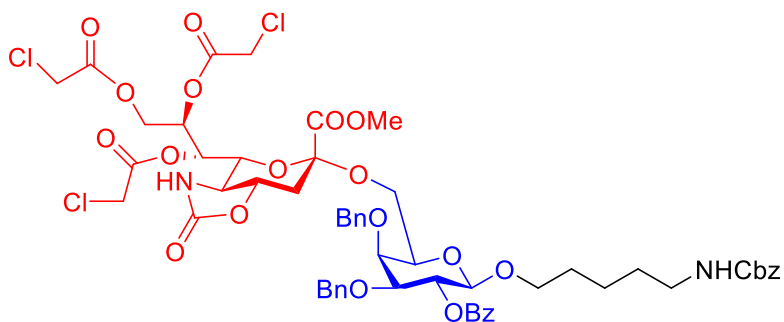
2013_11_25_Auto_217



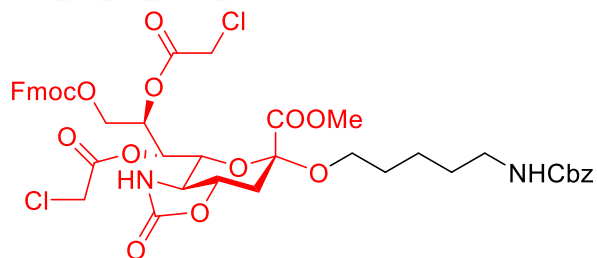
2013_11_26_Auto_152



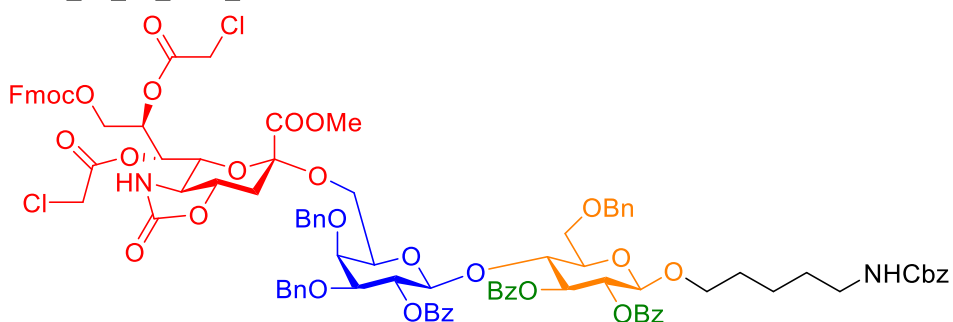
2013_12_03_Auto_031



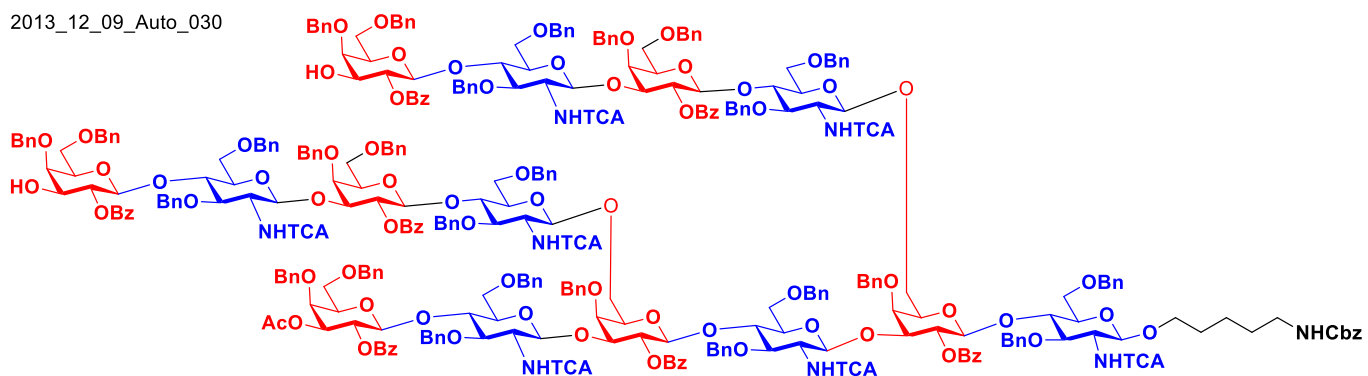
2013_12_03_Auto_033



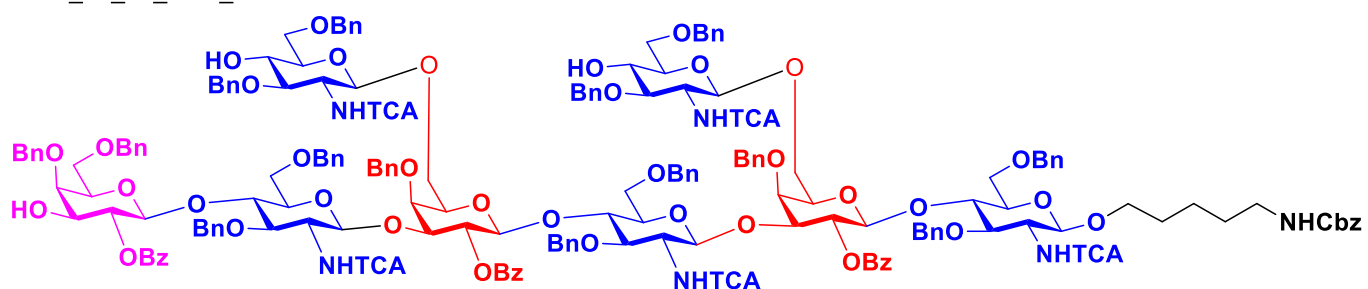
2013_12_05_Auto_030



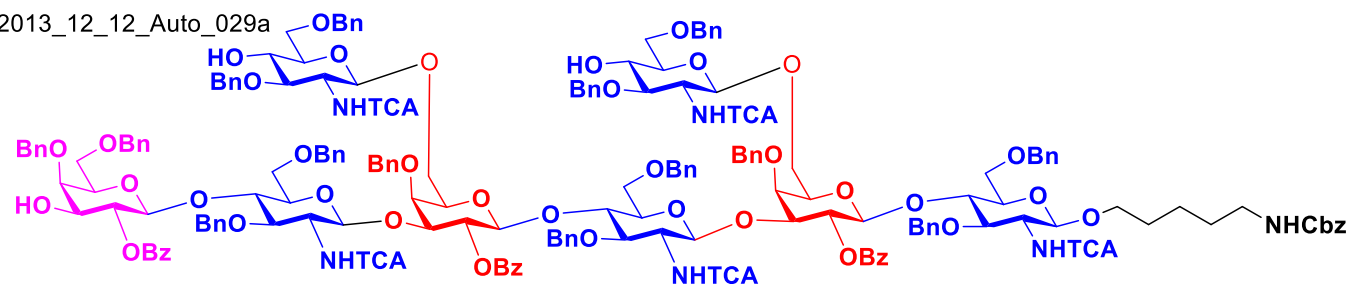
2013_12_09_Auto_030



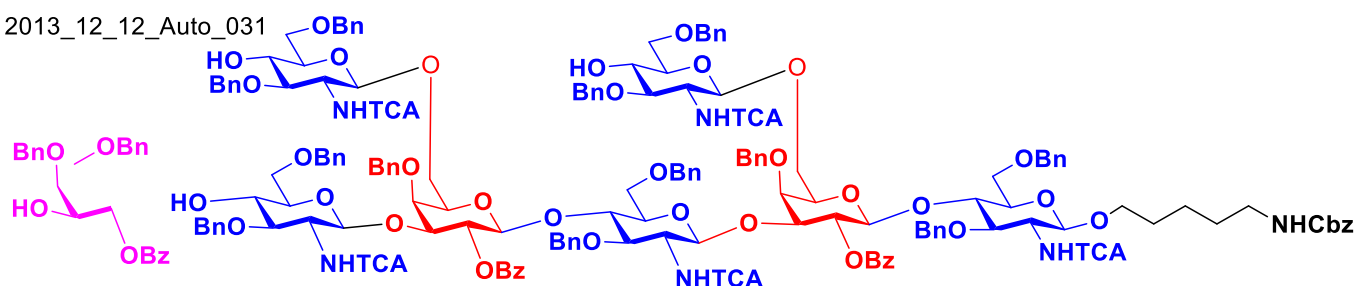
2013_12_12_Auto_029



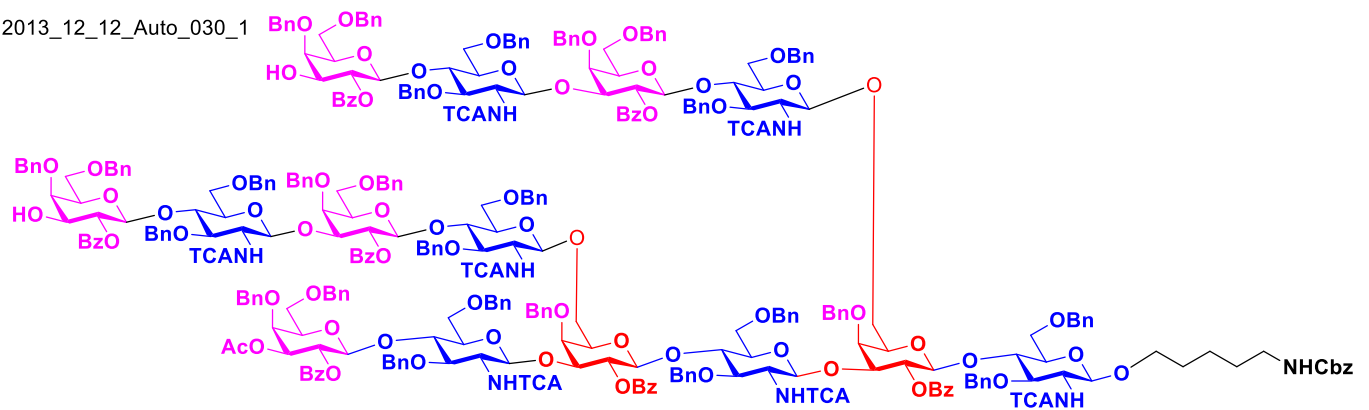
2013_12_12_Auto_029a



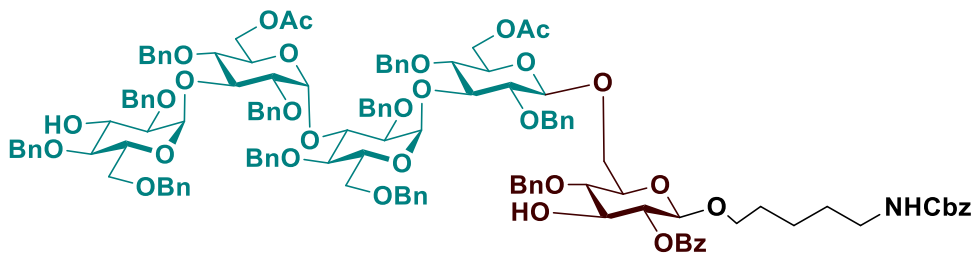
2013_12_12_Auto_031



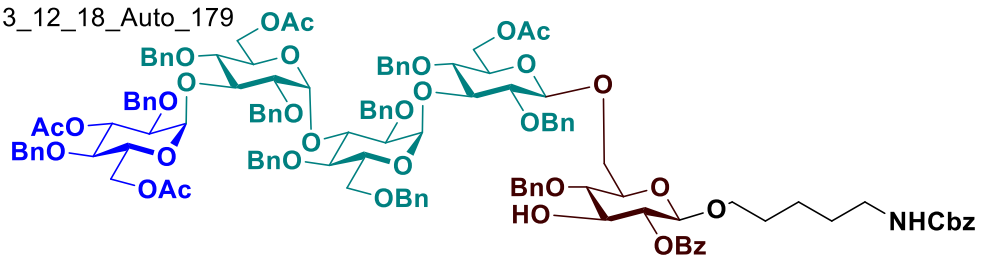
2013_12_12_Auto_030_1



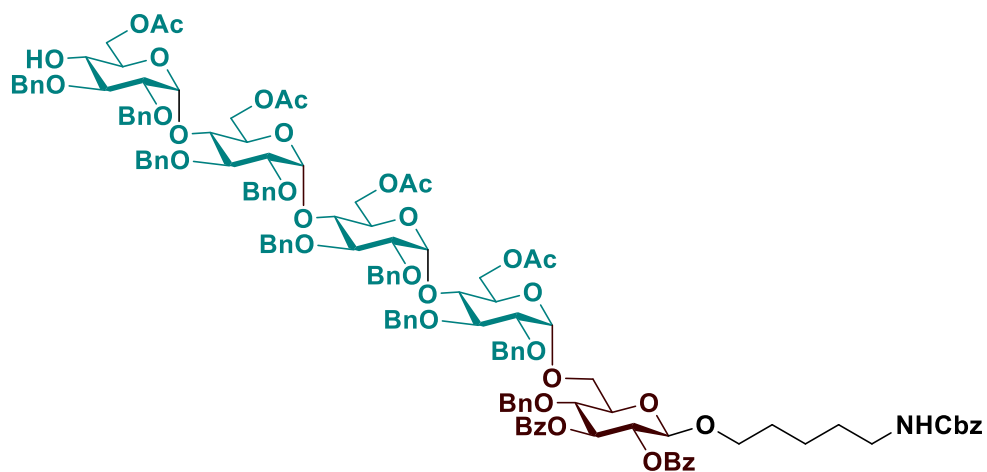
2013_12_17_Auto_178



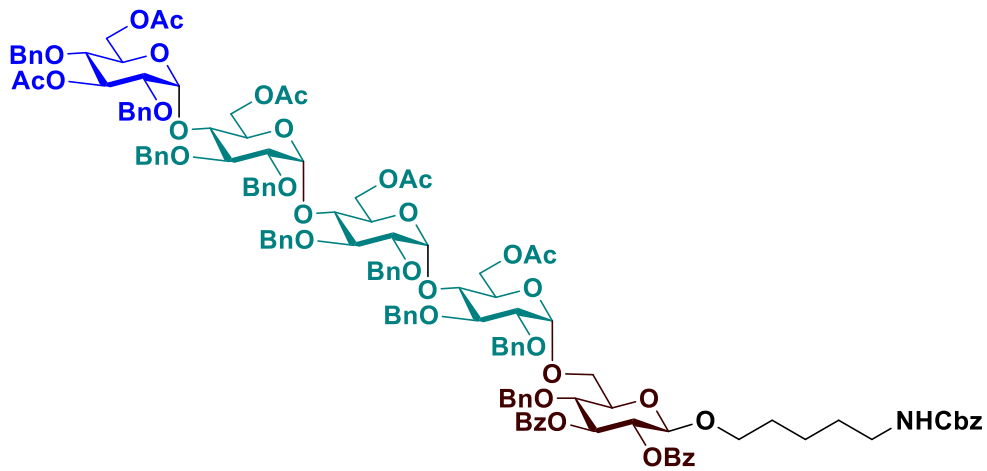
2013_12_18_Auto_179



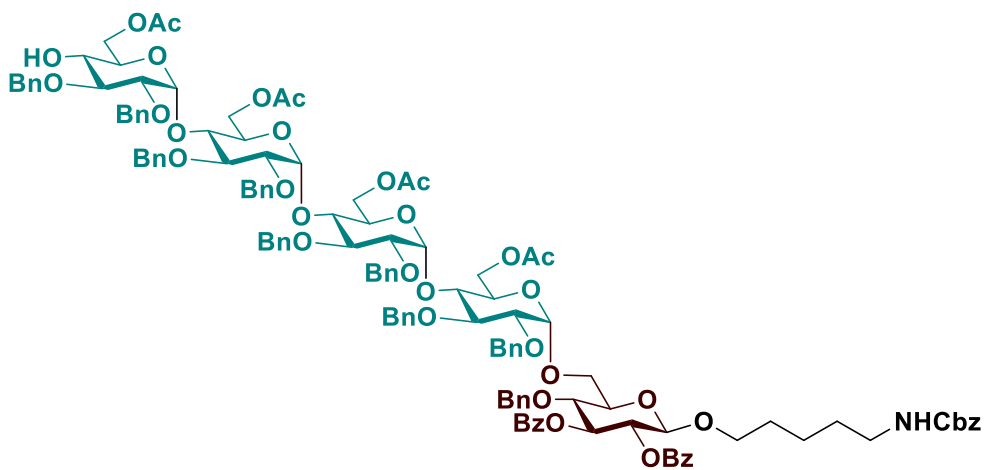
2013_12_18_Auto_180



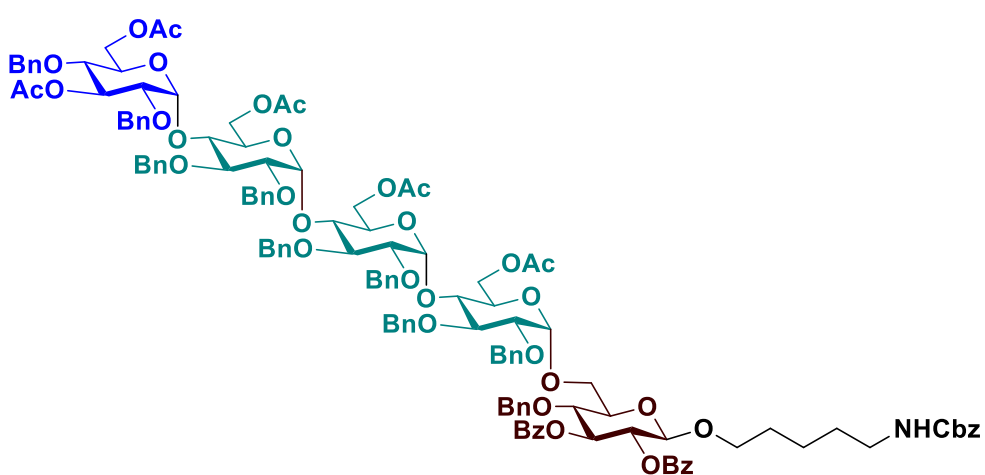
2013_12_18_Auto_181



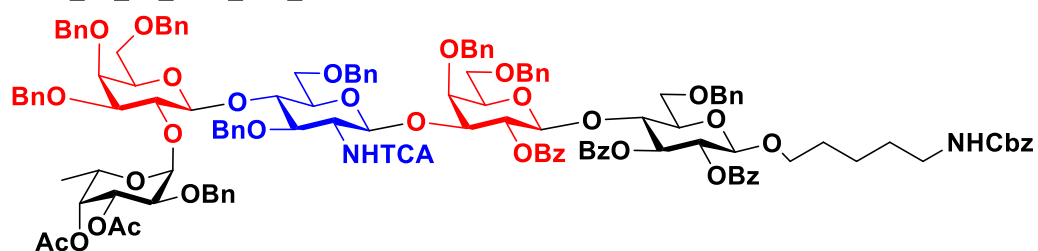
2013_12_19_Auto_180



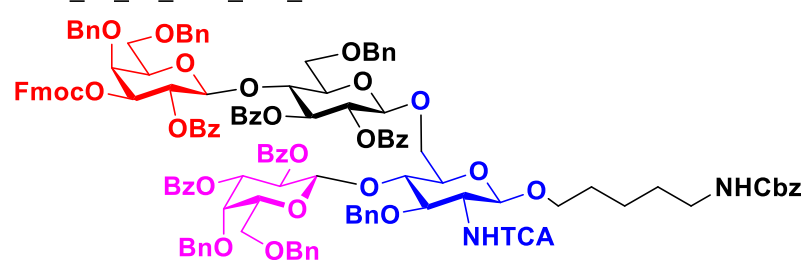
2013_12_20_Auto_181



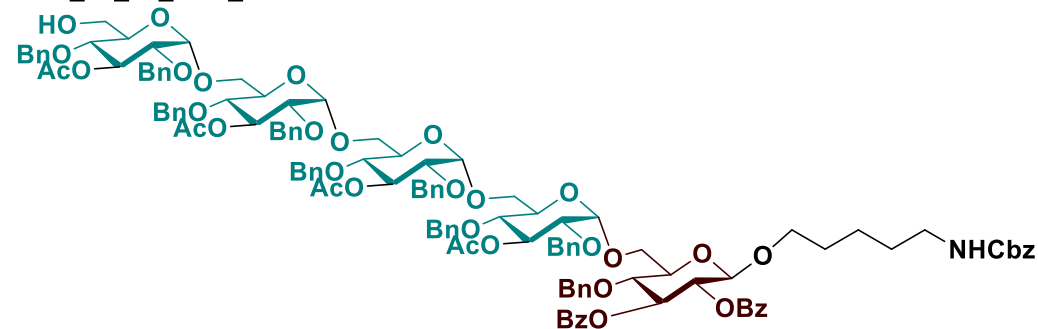
2014_01_03_Auto_127_I



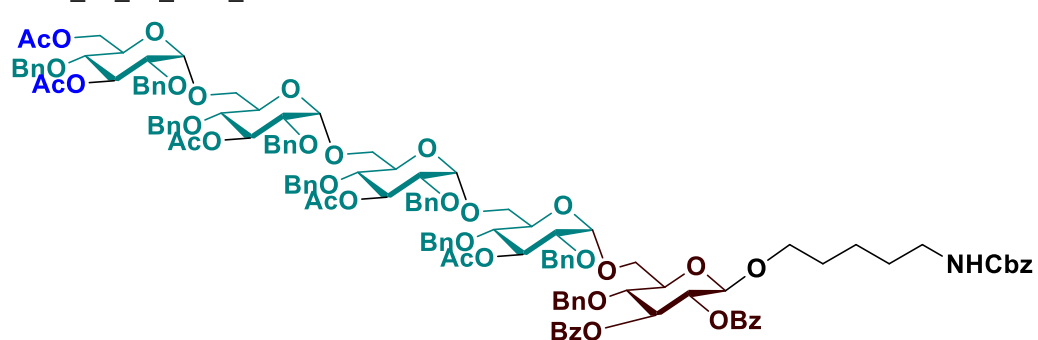
2014_01_14_Auto_128_I



2014_01_21_Auto_182



2014_01_22_Auto_183



6. Beschreibung der Prozessmodule

Die Module starten mit den wie folgt beschriebenen Standard-Parametern auf.

Alle Prozessparameter können individuell angepasst werden.

Glykosylierung mit Glycosyl Phosphaten

Aktivator: 62.5 mM Lösung TMSOTf in trockenem DCM

Aktion	Zyklen	Lösungsmittel	Reagent 1	Reagent 2	T (°C)	Verweilzeit
Wash	1	DCM	TMSOTf		-30	2 min
Wash	1	DCM			-30	25 s
Glycosylierung	1-3	DCM	BB (62.5 µmol)	TMSOTf (62.5 µmol)	-35/-30	5 min
					-15	30 / 35 min
Wash	3	DCM			-15	15 s
Wash	3	DCM			25	15 s

Tabelle 4: Modul: Glykosylierung mit Glycosyl Phosphaten, Standardparameter

Beschreibung:

Das Resin (16,9 µmol Hydroxylgruppe) wird in DCM (2 ml) gequollen, und die Temperatur des Reaktionsgefäßes auf -30°C eingestellt. Vor der Glykosylierungsreaktion wird das Resin mit TMSOTf in DCM und dann nur DCM gewaschen. Für die Glykosylierungsreaktion wurde das DCM entwässert und die Lösung von Phosphat BB (3,7 äquiv. in 1 ml DCM) dem Reaktionsgefäß zugeführt. Nachdem Erreichen der Zieltemperatur startet die Additionsreaktion durch Zugabe von TMSOTf in DCM (3,7 äquiv. in 1 ml DCM). Die Glykosylierung findet für 5 min bei -30 ° C oder -35 ° C und dann bei -15 ° C für 30 oder 35 Minuten statt. Anschließend wird die Lösung abgezogen und das Harz dreimal mit DCM gewaschen. Die gesamte Prozedur wird einmal, zweimal oder dreimal durchgeführt, um die Umwandlung der Akzeptorstellen zu verbessern. Danach wird das Resin dreimal mit DCM bei 25 ° C gewaschen.

Glykosylierung mit Thioglycosiden

Aktivator: Lösung von NIS (75mM) und TfOH (7.5 mM) in DCM/dioxane

Aktion	Zyklen	Lösungsmittel	Reagent 1	Reagent 2	T (°C)	Verweilzeit
Wash	1	DCM	TMSOTf		-30	2 min
Wash	1	DCM			-30	25 s
Glycosylierung	1	DCM	BB (62.5 µmol)	NIS (75 µmol) TfOH (7.5 µmol)	-55	5 min
					-30	40 min
Wash	3	DCM			-15	15 s
Glycosylierung	1	DCM	BB (62.5 µmol)	NIS (75 µmol) TfOH (7.5 µmol)	-55	5 min
					-30	40 min
Wash	3	DCM			-15	15 s
Wash	3	DCM			25	15 s

Tabelle 5: Modul: Glykosylierung mit Thioglycosiden, Standardparameter

Beschreibung:

Das Resin (16,9 µmol Hydroxylgruppen) wird in DCM (2 ml) gequollen, dabei wird die Temperatur des Reaktionsgefäßes auf -30°C eingestellt. Vor der Glykosylierungsreaktion wird das Harz mit TMSOTf in DCM und DCM gewaschen. Für die Glykosylierungsreaktion wird das DCM entwässert und eine Lösung von Thioglycosid BB (3,7 äquiv. in 1 ml DCM) in das Reaktionsgefäß gegeben. Nach Erreichen der Zieltemperatur wird die Reaktion durch Zugabe von NIS (4,44 äquiv.) und TfOH (0,44 äquiv.) in DCM / Dioxan (2: 1) gestartet. Die Glykosylierung erfolgt für 5 min bei -55 ° C und dann für 40 min bei -30 ° C. Anschließend wird die Lösung abgetropft und das Harz mit DCM gewaschen. Das gesamte Verfahren wird einmal wiederholt, um die vollständige Umwandlung aller Akzeptorstellen sicherzustellen. Anschließend wird das Resin dreimal mit DCM bei 25 ° C gewaschen.

Entschützung: Fmoc

Entschützungslösung: 20% NEt_3 in trockenem DMF

Aktion	Zyklen	Lösungsmittel	Reagent	T (°C)	Verweilzeit
Wash	3	DMF		25	15 s
Deprotection	3	DMF	NEt_3	25	5 min
Wash	3	DMF		25	15 s
Wash	3	THF		25	15 s
Wash	3	DCM		25	15 s

Tabelle 6: Modul: Entschützung mit Fmoc, Standardparameter

Beschreibung:

Das Resin wird gewaschen mit DMF, anschließend in 2 ml DMF gequollen. Die Temperatur des Reaktionsgefäßes wird auf 25°C eingestellt. Vor dem Entschützen wird das DMF abgetropft und das Resin dreimal mit DMF gewaschen. Für die Fmoc-Schutzgruppenabspaltung werden 2 ml 20% Et_3N Lösung in DMF dem Reaktionsgefäß zugeführt. Nach 5 min wurde die Lösung abgetropft und das gesamte Verfahren zweimal wiederholt. Nach Beendigung der Fmoc-Schutzgruppenabspaltung wird das Resin mit DMF, THF und DCM gewaschen.

Entschützung: Lev

Entschützungslösung: 0.15 M Hydrazin Acetat Lösung in Py/AcOH/H₂O (4:1:0.25)

Aktion	Zyklen	Lösungsmittel	Reagent	T (°C)	Verweilzeit
Wash	3	DCM		25	15 s
Deprotection	3	Py/AcOH/H ₂ O (4:1:0.25)	Hydrazin	25	30 min
Wash	3	DMF		25	15 s
Wash	3	THF		25	15 s
Wash	3	DCM		25	15 s

Tabelle 7: Modul: Entschützung mit Lev, Standardparameter

Beschreibung:

Vor dem Entschützen wird das Resin mit DCM dreimal gewaschen, in 1,3 ml DCM gequollen und die Temperatur des Reaktionsgefäßes auf 25°C eingestellt. Für Lev-Schutzgruppenabspaltung wird 0,8 ml einer Lösung von 150 mM Hydrazin Acetat in Py/AcOH/H₂O (4:1:0.25) das Reaktionsgefäß zugeführt. Nach 30 min wird die Lösung abgetropft und der Entschützungsschritt zweimal wiederholt. Nach Abschluss der Lev-Entschützung, wird das Harz mit DCM, DMF, THF jeweils dreimal gewaschen.

Capping: Benzoylation

Benzoylation Lösung: Lösung von Benzoic Anhydrid (0.5 M) und DMAP (0.25 M) in anhydriertes DCE

Aktion	Zyklen	Lösungsmittel	Reagent	T (°C)	Verweilzeit
Wash	3	Pyridine		25	15 s
Benzoylation	3	Pyridine	Bz ₂ O (0.5 mmol) DMAP (0.25 mmol)	40	30 min
Wash	3	DCM		25	15 s

Tabelle 8: Modul: Capping, Benzoylation, Standardparameter

Beschreibung:

Bei einer Temperatur von 25°C wird das Resin dreimal mit Pyridin (2 ml) gewaschen. Für die Benzoylierung wird die Temperatur auf 40 ° C eingestellt und dem Reaktionsgefäß 2 ml Pyridin, sowie 1 ml Lösung von 0,5 M Benzoesäureanhydrid und 0,25 M DMAP in DCE zugeführt. Nach 30 min wird die Reaktionslösung abgetropft und das Ganze zweimal wiederholt. Nachdem das Verkappen abgeschlossen ist, wird das Resin mit DCM gewaschen.