

Konzeption und Inbetriebnahme einer Anlage und Modellbildung zur Raumheizungsregelung für den Betrieb mit Modellprädiktiver Regelung

Conception and startup operations of an installation and modelling for a space
heating control to run with model predictive control

Master-Thesis
im Studiengang Wirtschaftsingenieurwesen

zur Erlangung des akademischen Grades
Master of Science (M.Sc.)

vorgelegt von

Daniel Johannes Mayer
aus Sulzfeld

Erstkorrektor: Prof. Dr. Angelika Altmann-Dieses

Zweitkorrektor: Prof. Dr.-Ing. Marco-Braun

Matr.-Nr.: 51968

E-Mail: daniel-j-mayer@gmx.de

Bearbeitungszeitraum: 17.03.2015 – 29.02.2016

Tag der Einreichung: 17.03.2016

Fakultät für Wirtschaftswissenschaften
Hochschule Karlsruhe – Technik und Wirtschaft
2016

Kurzfassung

Abstract

This text should be *italic*

Schlüsselwortliste: Test1, Test2, Test3

Keywords: *Test1, Test2, Test3*

Danksagung

Inhaltsverzeichnis

Tabellenverzeichnis	III
Abbildungsverzeichnis	IV
Quelltextverzeichnis	V
Symbolverzeichnis	VI
1 Einleitung	1
1.1 Motivation und Problemstellung	1
1.2 Zielsetzung und Aufbau der Arbeit	2
2 Theoretische Grundlagen	3
2.1 Model Predictive Control	3
2.2 Technische Grundlagen zur Kommunikation mit Bussystemen	3
2.2.1 Bussysteme	3
2.2.1.1 Informationsaustausch	4
2.2.1.2 Netzwerk und Topologie	4
2.2.1.3 Buszugriffsverfahren	7
2.2.1.4 Datensicherung	8
2.2.1.5 Schnittstellen	9
2.2.2 OSI-Kommunikationsmodell	13
2.2.3 Modbus Kommunikationstechnologie	18
2.3 Technische Grundlagen zur Modellbildung	25
2.3.1 Thermodynamische Systeme	25
2.3.2 Erster Hauptsatz der Thermodynamik	27
2.3.3 Wärmeübertragung	28
2.4 Technische Grundlagen zur Solar- und Gebäudetechnik	29
2.4.1 Außenklima und Komponenten	29
2.4.1.1 Die Außenlufttemperatur	30
2.4.1.2 Sonnenstrahlung	31
2.4.2 Gebäudetechnik Glas und Wärmedurchgangskoeffizienten	31
3 Anlagendesign	32
3.1 Analyse der Anforderungen	32
3.1.1 Einsatzziele und Rahmenbedingungen	32
3.1.2 Definition der Anforderungen	33
3.2 Ziel und die Idee der Anlage	38
3.2.1 Aufgabe der Anlage	38
3.2.2 Idee der Anlage	38

3.3 Konzept und Planung	39
3.3.1 Netzwerkarchitektur	40
3.3.2 Erfassung der Raumtemperatur	40
3.3.3 Steuerung des Heizkörpers	42
3.4 Umsetzung und Installation	43
3.5 Software	43
4 Modellbildung des Raumes	44
4.1 Anforderungen an das Raummodell	44
4.2 Das Grundmodell des Raumes	45
4.3 Modellerweiterung durch Berücksichtigung der realen Umgebung	46
4.4 Modellerweiterung durch Berücksichtigung der räumlichen Gegebenheiten	48
4.5 Modellerweiterung durch Berücksichtigung von Störgrößen und der Sonneneinstrahlung	50
4.6 Validierung des Modells	50
4.7 Anpassung des Modells mit Parameterschätzung	50
5 Schlussbetrachtung	51
5.1 Fazit	51
5.2 Ausblick und Ansatzpunkte für weitere Arbeiten	51
A Modelle	52
A.1 Raummodell	52
B Datenblätter	57
B.1 THERMASGARD RTM1-Modbus Raumtemperaturfühler von S+S Regeltechnik	57
B.2 Webthermograph 8x von WuT	59
B.3 MULTICAL 602 Wärmemengenzähler von Kamstrup	63
C Modbusadressmapping	65
C.1 MULTICAL 602 Wärmemengenzähler von Kamstrup	65
C.2 THERMASGARD RTM1-Modbus Raumtemperaturfühler von S+S Regeltechnik	72
Literaturverzeichnis	75

Tabellenverzeichnis

Tab. 3.1: Umsetzung der Ziele in Anforderungen der Anlage	34
Tab. 4.1: Eigenschaften des Raummodells	46
Tab. 4.2: Weitere Eigenschaften des Raummodells	48

Abbildungsverzeichnis

Abb. 2.1:	Bus-Struktur	5
Abb. 2.2:	Impulsverzerrung auf einer Leitung	6
Abb. 2.3:	Baumstruktur	7
Abb. 2.4:	<i>Cyclic Redundancy Check</i>	10
Abb. 2.5:	Parallele und serielle Datenübertragung	11
Abb. 2.6:	Spannungspegel und Stecker der EIA 232-Schnittstelle	12
Abb. 2.7:	Spannungspegel und Stecker der EIA 485-Schnittstelle	12
Abb. 2.8:	Die sieben Schichten des Open System Interconnection Modells	15
Abb. 2.9:	Die vier Dienstvorgänge	16
Abb. 2.10:	Die Modbus Kommunikation im OSI-Referenzmodell	19
Abb. 2.11:	Allgemeiner Rahmen für Telegramme nach dem Modbus Anwendungsprotokoll	19
Abb. 2.12:	Transaktion mit dem Modbus Protokoll	20
Abb. 2.13:	Datenmodell und Adressierung nach dem Modbus Protokoll	21
Abb. 2.14:	Serielle Kommunikation über Modbus RTU	23
Abb. 2.15:	Die Modbus TCP/IP Kommunikationsarchitektur	24
Abb. 2.16:	Anangepasster Rahmen für Telegramme nach dem Modbus TCP/IP Protokoll	24
Abb. 2.17:	Komponenten des Außenklimas	31
Abb. 3.1:	Raumskizze K004A vom K Gebäude der Hochschule Karlsruhe – Technik und Wirtschaft	35
Abb. 3.2:	Prinzipskizze eines technischen Systems zur Raumtemperaturregelung des Raumes K004b	39
Abb. 3.3:	Aufbau des Netzwerks	40
Abb. 3.4:	Verteilung der Raumtemperaturfühler	41
Abb. 4.1:	Grundmodell eines Raumes	45
Abb. 4.2:	Erweitertes Raummodell	49
Abb. 4.3:	Erweitertes Raummodell	50

Quelltextverzeichnis

4.1 Einfaches Gleichungssystem für das Grundmodell des Raumes in Modelica	46
4.2 Erweitertes Gleichungssystem Modell des Raumes unter Berücksichtigung der realen Umgebung in Modelica	47
4.3 Erweitertes Gleichungssystem Modell des Raumes unter Berücksichtigung der räumlichen Gegebenheiten in Modelica	48
4.4 Erweitertes Gleichungssystem Modell des Raumes unter Berücksichtigung der Sonneneinstrahlung und Störgrößen	50
listings/room_model_listing.mo	52

Symbolverzeichnis

Hinweis: Bei der Angabe der Symbole soll sich auf die Wesentlichen beschränkt werden. Die jeweils zutreffende Bedeutung ergibt sich entweder aus dem Kontext oder ist explizit im Text angegeben.

Formelzeichen

$A_{exchange}$	Wärmeaustauschoberfläche [m^2]
c_p	Spezifische Wärmekapazität eines Stoffes [$\frac{J}{kg*K}$]
E	Gesamtenergie eines Systems [J]
E_{kin}	Kinetische Energie eines Systems [J]
E_{pot}	Potenzielle Energie eines Systems [J]
f_{max}	Maximale Übertragungsfrequenz [Hz]
T_0	Celsius Nullpunkt bei [273, 15K]
T	Kelvin Temperatur [K]
m	Masse [kg]
\dot{m}	Massenstrom [$\frac{kg}{s}$]
m_{sys}	Masse eines Systems [kg]
P	Leistung [W]
p_{high}	High-Pegel für ein Signal, entspricht logischer „1“
p_{low}	Low-Pegel für ein Signal, entspricht logischer „0“
Q	Wärme [J]
\dot{Q}	Wärmestrom [W]
t	Celsius Temperatur [$^{\circ}C$]
Δt_{Imp}	Impulsverzerrung [s]
$U - Wert$	Materialabhängiger Wärmedurchgangskoeffizient [$\frac{W}{K*m^2}$]
U	Innere Energie eines Systems [J]
W	Arbeit [J]

Griechische Buchstaben

κ	Erwartete Umsetzungsschwierigkeit
----------	-----------------------------------

Ω	Systemelement (Raum, Organisation, Technik)
ρ_n	Gewichtung des Wandlungspotentialmerkmals n
$\lambda_{abs,n}$	Absolute Teilweichtigkeit des Merkmals n
$\lambda_{rel,n}$	Relative Teilweichtigkeit des Merkmals n

Lateinische Buchstaben

kt	Kundentakt
$rw_\emptyset(t)$	Mittlere Lagerreichweite in Periode
t_{AZ}	Verfügbare Arbeitszeit in Periode
$T_{\emptyset BZ}$	Mittlere Gesamtbearbeitungszeit
$T_{\emptyset DLZ}$	Mittlere Gesamtdurchlaufzeit eines Produkts
$T_{\emptyset PZ}$	Mittlere Gesamtprozesszeit
$x_{\emptyset B}(t)$	Mittlerer Bedarf in Periode
$x_{\emptyset LB}$	Mittlerer Lagerbestand

Mathematische Operatoren

$(a; b]$	Halboffenes Intervall
\Leftrightarrow	Genau dann, wenn
\cong	Näherungsweise
\forall	Für alle
\in	Ist Element von
\mathbb{N}^+	Menge der natürlichen Zahlen ohne 0
\mathbb{R}	Menge der reellen Zahlen

„Erfolgreich zu sein setzt zwei Dinge voraus: Klare Ziele und den brennenden Wunsch, sie zu erreichen.“
— JOHANN WOLFGANG VON GOETHE

1 Einleitung

1.1 Motivation und Problemstellung

Von allgemein solare Anwendung hin zu Forschung HSKA wird das Ziel sein.

Hinführung und Forschung der HSKA aufzeigen und daher einordnen Forschen derzeit bereits auf dem Gebiet (MPC mit) der solaren Anwendungen. Derzeit wird eine große Solaranlage zur Kühlung des Atriums installiert. Die Anlage nutzt die in Solarkollektoren gewonnene Wärmeenergie um eine Adsorptionskälteanlage anzutreiben. Die Anlage ist aufgrund ihrer Größe träge in der Reaktion und bisher noch im Aufbau. Komplementär dazu soll eine kleine Anlage mit solarem Bezug aufgebaut werden um die Forschung auch für ergänzende Dinge betreiben zu können.

Die Modellprädiktive Regelung beschäftigt sich damit, ein technisches System oder allgemeiner einen Prozess – im mathematisch exakten Sinne – optimal zu regeln.

Im Dialog mit den Projektverantwortlichen der Hochschule Karlsruhe für die Erforschung von solaren Anwendungen, in Person von Herrn ADRIAN BÜRGER und MARKUS BOHLAYER, die Einsatzziele gemeinsam erarbeitet und sind im Detail in Kapitel 3.1 ausgeführt.

Wichtigste Ergebnisse aus Dialog ist die Motivation dieser Forschungsarbeit:

Die Motivation dieser Arbeit ist also, einen komplementären Forschungsbeitrag für solare Anwendungen zu leisten im Hinblick auf die große Anlage.

Dass heißt jene Chancen zu nutzen, die die große Anlage nicht bieten kann einen möglichst großen Nutzen zu realisieren, also was die große Anlage nicht leisten kann und die große Anlage zu entlasten im Sinne von Empfindlichkeit/Versuche.

Außerdem sollen möglichst Erfahrungen gesammelt werden, die beim Anlauf der großen Anlage von Nutzen sein können.

!!!!!!!!!!!!!! Forschungsfrage, vorher Begriffe klären: MPC, Anlage, Modell, Steuerung/Regelung Als Forschungsfrage und Problemstellung dieser Arbeit ergibt sich die Frage, wie eine Anlage und ein Modell derselben aufgebaut sein müssen, um die Regelung einer Raumtemperatur mit Modellprädiktiver Regelung zu ermöglichen.
!!!!!!!!!

1.2 Zielsetzung und Aufbau der Arbeit

Das übergeordnete Ziel dieser Arbeit ist es, eine Test- und Anwendungsumgebung zu schaffen, um die Forschung der Hochschule Karlsruhe auf dem Gebiet der Modellprädiktiven Regelung von solaren Anwendungen weitere Forschung weiter voran zu treiben und komplementär zu ergänzen.

Aus der Forschungsfrage lässt sich als konkretes Ziel der Arbeit ableiten, die Konzeption, Planung und Umsetzung einer Anlage und Bildung eines Modells zur Regelung einer Raumtemperatur mit Modellprädiktiver Regelung. Also ein Labor Weiterhin soll mit dieser Arbeit Know-How und Erfahrung generiert werden das bei der weiteren Forschung von Nutzen sein kann/ist.

Im Rahmen dieser Arbeit werden – anschließend an die Einleitung in 1 – die theoretischen Grundlagen in 2 ausgeführt. Zunächst wird die grundlegenden Theorie zu Model Predictive Control vorgestellt bevor anschließend weitere technische Grundlagen erklärt, welche für das weitere Verständnis dieser Arbeit benötigt werden.

Danach wird das technische System in Kapitel 3 Schritt für Schritt entwickelt, ausgehend von der Idee und den räumlichen Gegebenheiten/Nebenbedingungen, und weiter konkretisiert bis zur realisierten Umsetzung in eine funktionierende Anlage.

Dementsprechend werden zunächst das Konzept, die Planung und die technische Umsetzung der konkreten Anlage dargestellt, bevor anschließend die theoretischen Grundlagen von Model Predictive Control und zur Modellbildung erläutert werden. Anschließend wird das Modell für Model Predictive Control gebildet und ein erstes grobes Konzept zur Steuerung der Raumtemperatur vorgestellt. Abschließend wird eine Validierung des Modells versucht und findet eine Anpassung des Modells statt damit es künftig mit Model Predictive Control genutzt werden kann.

„Theorie ist die Mutter der Praxis.“

— LOUIS PASTEUR

2 Theoretische Grundlagen

In diesem Kapitel werden die theoretischen Grundlagen erläutert die benötigt werden, um die Forschungsfrage zu beantworten. Zunächst MPC dann eher technisch

2.1 Model Predictive Control

Model Predictive Control an sich ist eine Methodik zur Steuerung von Systemen. Diese versucht zunächst, zu sich periodisch wiederholenden, diskreten Zeitpunkten das Verhalten eines Systems in der Zukunft – also einer immer gleich weit in die Zukunft hineinreichenden Periode – zu beschreiben. Hierzu bedient Model Predictive Control sich der Kenntnis des aktuellen Zustandes und eines physikalischen-mathematischen Modells des Systems, um dessen zukünftiges Verhalten „vorherzusagen“ bzw. abzubilden. Des Weiteren wird versucht das Verhalten des Systems mit minimalem Aufwand zu beeinflussen, um einem eigens- oder vordefinierten Zielkriterium zu folgen beziehungsweise diesem zu entsprechen.

2.2 Technische Grundlagen zur Kommunikation mit Bussystemen

In diesem Kapitel werden die Grundlagen von Hard- und Software beleuchtet die für die Kommunikation der Steuerung mit den einzelnen Anlagenteilen benötigt werden. Diese umfassen zunächst Bussysteme im Allgemeinen, dann das OSI Modell für techn. Kommunikation und werden anhand des spezifischen/konkreten Anwendungsfalls Modbus erläutert.

Die Einführung wird an der Struktur von Schnell u. Wiedemann [2006] anlehnen.

2.2.1 Bussysteme

Um allgemein Prozesse zu überwachen, zu steuern oder regeln zu können, müssen zwischen den verschiedenen Prozessbeteiligten/einheiten Informationen ausgetauscht werden. Im Kontext dieser Arbeit gilt es einen Prozess zur Temperatursteuerung/Halten zu regeln. Die Prozessbeteiligten sind hierbei technische Bauteile, Aktoren und Sensoren sowie ein Steuerungsrechner/Controller, die zusammen ein technisches System bilden, dass im Folgenden als Anlage bezeichnet wird. Die Anlage zeichnet sich dadurch aus, dass sie eine eigenständige funktionale Einheit bildet einen eigenen Zweck

verfolgt, die Raumtemperaturregelung und einen Mehrwert gegenüber ihrer Einzelteile hat, was dem Zusammenspiel der einzelnen Bauteile und Geräte entspricht. Um den Mehrwert zu realisieren und den Zweck zu erfüllen, werden Kommunikationssysteme benötigt. Mit Hilfe derer kann Kommunikation erfolgen und die Anlage ihren Zweck erreichen. Diese Kommunikationssysteme werden von technischer Seite oftmals als Bussysteme realisiert/umgesetzt. Bussysteme lassen sich anhand ihrer verschiedenen Ausprägungen von Merkmalen klassifizieren. Im folgenden Abschnitt werden zunächst die Merkmale von Bussystemen zum allgemeinen Verständnis dargestellt bevor auf die konkreten Ausprägungen des später eingesetzten Modbus-Bussystems eingegangen wird.

2.2.1.1 Informationsaustausch

Die Informationen über einen Prozess und dessen Zustand, werden auf höherer Ebene durch Daten und auf unterster Ebene durch einzelne Bits repräsentiert. Der Austausch dieser Daten zwischen den einzelnen Geräten findet in Form von Telegrammen statt. Ein Telegramm besteht grundsätzlich aus den zu übertragenden Daten und zusätzlich aus Informationen zur Übertragung. Die Daten werden vor der Übertragung in Rahmen, sogenannte data frames, eingeteilt, deren genauer Aufbau abhängig vom verwendeten Kommunikationsprotokoll innerhalb des Netzwerks ist [Schnell u. Wiedemann, 2006, S. 11f.]. Der Aufbau eines Telegramms am Beispiel des Modbus-Kommunikationsprotokolls wird in Abschnitt 2.2.3 erläutert und ist in Abb. 2.11 graphisch dargestellt.

2.2.1.2 Netzwerk und Topologie

Werden einzelne Prozesseinheiten miteinander über sogenannte Verbindungsleitungen, die zur Übertragung Informationen genutzt werden können, verknüpft, entstehen dabei Netzwerke und die Prozesseinheiten werden als Teilnehmer des Netzwerks bezeichnet. Ein Netzwerk lässt sich in einzelne Segmente einteilen und kann je nach Ausführung der Verbindungsleitungen und Anzahl der Teilnehmer unterschiedlich ausgeprägt sein. Anhand der geometrischen Anordnung lassen sich die folgenden, verschiedenen Netzwerktopologien unterscheiden.

Die einfachste Art, um eine Verbindung zwischen zwei Teilnehmern eines Netzwerks herzustellen, ist die sogenannte Zweipunktverbindung. Dazu sind die Netzwerkteilnehmer durch eine direkte Leitung miteinander verbunden. Jedoch steigt mit der Anzahl von Teilnehmern auch der Verbindungsaufwand überproportional an, um bei solchen vermaschten Netzwerk alle Teilnehmer miteinander zu verbinden. Dies hat für große, vermaschte Netze zur Folge, dass eine unübersichtliche große Anzahl von Schnittstellen, ein extrem hoher Verkabelungsaufwand und damit verbundene hohe Kosten entstehen. Um diese Kosten zu vermeiden, ergeben sich noch verschiedene andere Möglichkeiten zur Anordnung von Teilnehmern in Netzwerken [Schnell u. Wiedemann, 2006, S. 1f.].

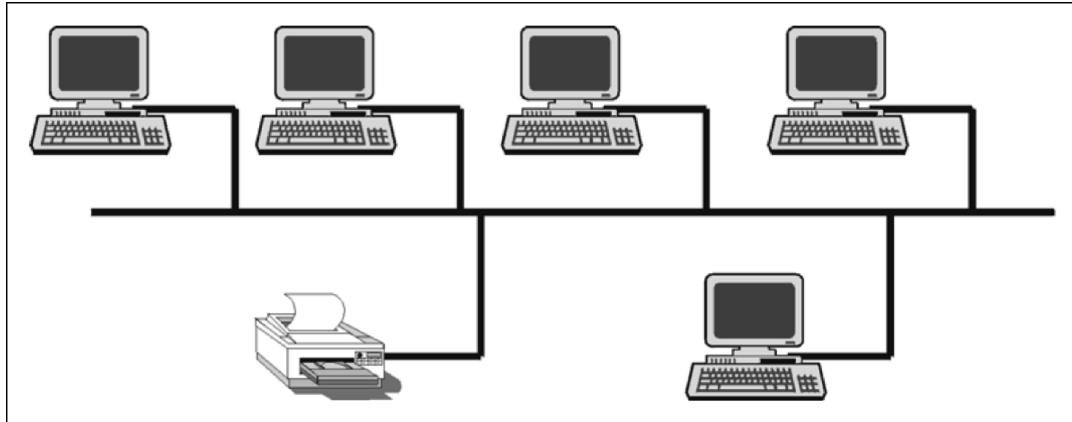


Abb. 2.1: Bus-Struktur aus [Schnell u. Wiedemann, 2006, S. 3]

Um dem hohen Verkabelungsaufwand zu vermeiden, wird bei großen Netzwerken zu einer Liniensstruktur übergegangen, die auch als Bus-Struktur bezeichnet wird und in Abb. 2.1 visualisiert ist. Charakteristisch für die Bus-Struktur ist, dass alle Teilnehmer entlang einer langen Verbindungsleitung, dem sogenannten Buskabel, angeordnet sind. Sie sind mit Hilfe von kurzen Stichleitungen an das gemeinsame Buskabel angebunden, über das die gesamte Kommunikation im Netzwerk erfolgt. Durch diese Anordnung wird der Verkabelungsaufwand sowie die Anzahl an Schnittstellen, insbesondere für sehr große Netzwerke, stark reduziert. Jedoch wird durch Nutzung einer gemeinsamen Kommunikationsleitung die gleichzeitige Kommunikation von Teilnehmern erschwert und es müssen sogenannte Buszugriffsverfahren definiert werden, welche lediglich Regeln für Zugriff auf den Bus festlegen. Weiterhin müssen durch die Parallelschaltung alle Teilnehmer ständig alle Sendungen mitverfolgen, wodurch der Sender stark belastet wird. Die Busleitungslängen sind meist sehr lange¹ und da die Länge auf die zu übertragende Wellenlänge bezogen nicht mehr vernachlässigbar klein ist, müssen Reflexionen durch Leitungsabschlusswiderstände an den beiden Enden der Busleitung unterbunden werden. Außerdem werden die Leitungslängen und die Teilnehmer je Netzwerksegment begrenzt??S. 3f.]schn06.

Ein weiterer begrenzender Faktor für die Leitungslänge ist der Fakt, dass die maximale Übertragungslänge und die maximale Übertragungsrate miteinander verknüpft sind und sich gegenseitig beschränken. Der Leitungs- und Kapazitätswiderstand einer Leitung hängen von der Länge der Leitung ab und lassen sich durch das Ersatzschaltbild eines RC-Gliedes repräsentieren, wie in Abbildung Abb. 2.2 a) zu sehen ist. Durch die beiden Widerstände entsteht auf der Leitung eine Impulsverzerrung Δt_{Imp} , die somit mittelbar von der Leitungslänge abhängt. Je länger die Leitung wird, desto größer werden auch beiden Widerstände. Durch die erhöhte Leitungskapazität $C_{Leitung}$ erhöht sich die Ladezeit und gleichzeitig sinkt durch den erhöhten Leitungswiderstand $R_{Leitung}$ die Lastspannung U_G . Damit vergrößert sich die Impulsverzerrung Δt_{Imp} , wie in Abb. 2.2 b) und Abb. 2.2 c) dargestellt. Dadurch wird die maximale Frequenz

¹ Diese reichen von mehreren hundert Metern bis teilweise in den Kilometerbereich, je nach Art und Einsatzort der Anwendung.

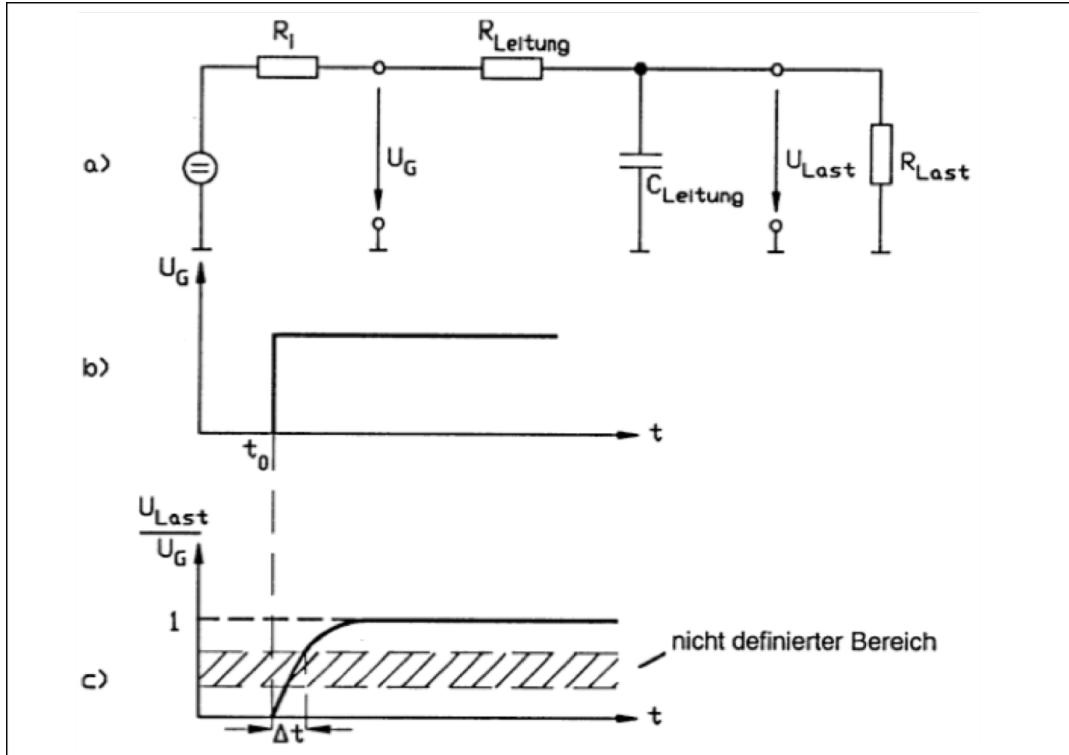


Abb. 2.2: Impulsverzerrung auf einer Leitung: a) Ersatzschaltbild der Anordnung b) Ausgangsspannung des Generators c) Empfängerspannung aus [Schnell u. Wiedemann, 2006, S. 4]

f_{max} der Datenübertragung auf den Kehrwert der Impulsverzerrung $f_{max} = \frac{1}{\Delta t_{Imp}}$ beschränkt, da ansonsten der Empfänger den Wechsel des logischen Zustandes nicht mehr registrieren kann. In der Praxis bedeutet dies, dass die maximale Übertragungslänge und die maximale Übertragungsrate miteinander verknüpft sind und sich gegenseitig beschränken [Schnell u. Wiedemann, 2006, S. 4f.].

Um die Begrenzung der Leitungslängen zu korrigieren, wurde die Bus-Struktur zu einer Baumstruktur weiterentwickelt, welche in Abb. 2.3 dargestellt ist. Darin werden einzelne Netzwerk-Segmente, also einzelne Bus-Strukturen, durch Verstärkerelemente, sogenannte Repeatere, zu einem großen Netzwerk verknüpft. Um damit jedoch größere Flächen als mit der Bus-Struktur zu vernetzen und gleichzeitig die maximale Leitungslänge und die maximale Anzahl der Busteilnehmer zu vergrößern, wird jedoch eine galvanische Trennung der Teilnehmer voneinander benötigt [Schnell u. Wiedemann, 2006, S.5 f.]. Die Besonderheit in dieser Struktur liegt also darin, dass sich durch Ihren Aufbau bestehende Bus-Strukturen auch nachträglich einfach erweitern oder miteinander verknüpfen lassen. Die Bauteile zur Erweiterung von Netzwerken werden im Abschnitt 2.2.1.5 Schnittstellen vorgestellt.

Weitere wichtige Netzwerk Topologien, die für das Verständnis dieser Arbeit keine weitere Relevanz haben, sind die Ring- und die Stern-Struktur. Die Ring-Struktur ist durch einen physikalischer Ring von Zweipunktverbindungen aufgebaut und gekennzeichnet durch die Kommunikation der Teilnehmer übereinander hinweg. Die Stern-Topologie hingegen ist um eine Zentralstation herum ausgebaut, die mit al-

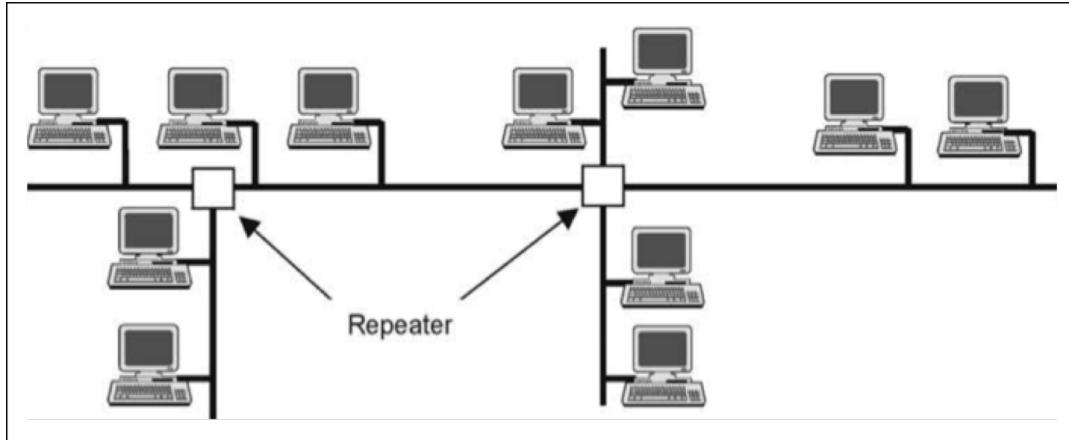


Abb. 2.3: Baumstruktur aus [Schnell u. Wiedemann, 2006, S. 5]

len Teilnehmer verbunden ist und über die gesamte Kommunikation abläuft. Der interessierte Leser findet in [Schnell u. Wiedemann, 2006, S. 6f.] detailliertere Ausführungen.

2.2.1.3 Buszugriffsverfahren

Die meisten Netzwerktopologien kommunizieren gemeinsam über eine Verbindungsleitung. Daher werden Regeln für den Zugriff definiert, um eine reibungslose Kommunikation zu ermöglichen. Die Buszugriffsverfahren lassen sich in zwei Gruppen aufteilen, der kontrollierten und zufälligen Verfahren [Schnell u. Wiedemann, 2006, S. 19].

Bei den kontrollierten Verfahren ist der Sender bereits vor Sendebeginn eindeutig bestimmt und eine Zuteilung des Busses ist nicht notwendig. Der Buszugriff findet entweder zentral innerhalb einer Zentralstation statt, bei sogenannten Master/Slave-Verfahren, oder wird dezentral durch Steuereinheiten vorgenommen, wie z.B. beim Tokenring und Tokenbus. Ein solches Verfahren heißt echtzeitfähig, wenn die Zykluszeit zur Datenübertragung berechenbar ist, aufgrund einer Beschränkung der Länge des Übertragungsintervalls und der maximalen Datenlänge. Bei zufälligen Buszugriffsverfahren greifen die Teilnehmer bei Bedarf auf die Verbindungsleitung zu und müssen sicherstellen, dass diese nicht gerade von einem anderen Teilnehmer belegt ist. Da nicht vorhergesehen werden kann, an welchem Zeitpunkt Informationen übertragen werden kann keine Echtzeitfähigkeit erreicht werden [Schnell u. Wiedemann, 2006, S. 19].

Das Master/Slave-Verfahren besteht in der Regel aus einer Bussteuerungseinheit, dem sogenannten Master, und mehreren passiven Teilnehmern, den Slaves. Die Kommunikation wird ausschließlich vom Master initiiert, der die Verbindung zu den Slaves aktiv durch ein Request herstellt, in welchem die angeforderten Daten spezifiziert sind. Die Slaves treten nur nach Anfragen in Aktion und antworten darauf unmittelbar mit einer Response, die die angeforderten Daten des Masters enthält. In der Regel erfolgt die Kommunikation zyklisch zu allen Slaves gleichzeitig (Polling), damit

der Master ein umfassendes und aktuelles Bild über den Systemzustand bekommt. Dadurch ergeben sich einfache Slaves, die günstig in den Bus eingebunden werden können, weil die gesamte „Intelligenz“ im Master implementiert ist. Jedoch gilt es bei diesem Verfahren zu beachten, dass der Informationsaustausch zwischen verschiedenen Slaves längere Zeit in Anspruch nehmen kann und bei einem Ausfall des Masters das gesamte Bussystem stillliegt [Schnell u. Wiedemann, 2006, S. 19ff.]. Auf eine weitere Beschreibung der übrigen Verfahren wird aufgrund der fehlenden Relevanz für diese Arbeit verzichtet, der interessierte Leser findet jedoch bei Schnell u. Wiedemann [2006] im Kapitel Buszugriffsverfahren ausführliche Informationen.

2.2.1.4 Datensicherung

Bei der Übertragung von Informationen besteht die Gefahr von Störungen, welche sich als Fehler in einer Nachricht durch Invertierung von Bits äußern. Störungen sind in der Regel technischer Art, wie zum Beispiel elektromagnetische Störsignale, Rauschen oder Potentialdifferenzen, und gegen einen Großteil lassen verschiedene Vorkehrungen treffen. Dadurch ergibt sich die Möglichkeit Störungen vorzubeugen oder diese nach Einsatz zu beseitigen. Der erste Ansatz ist also eine Verminderung des Auftretens durch technische Vorkehrungen, wie zum Beispiel Schirmung der Kabel, galvanische Trennung von Netzwerken oder differenzielle Signalübertragung. Der zweite Ansatz beschäftigt sich mit der Überwachung des Nachrichtenverkehrs und dem Ausbessern/Gegenmaßnahmen bei Fehlern [Schnell u. Wiedemann, 2006, S. 30].

Auf dem Gebiet der Buskommunikation entspricht wird eine Information oder Nachricht oder Telegramm codiert. Es werden stets Transparente Codes betrachtet, welche bitorientiert sind. Dabei sind jegliche Kombination von Bits erlaubt und man kann allein aus der Folge der Blitz nicht auf einen Fehler schließen. Definition Telegramm und Bit(0 und 1)

Die Vorkehrungen technischer Art Können Sind in aller Regel in den Spezifikationen der einzelnen Bussysteme enthalten, daher Werden diese nicht weiter beschrieben und lediglich die Überwachung und Ausbesserung des Nachrichtenverkehrs erläutert. Bei der Übermittlung von Nachrichten können drei Arten von Fehlern auftreten: Der Fehler kam erkennbar und korrigierbar sein, Erkennbar und nicht korrigierbar Oder nicht erkennbar und damit auch nicht korrigierbar.

Kann der Fehler erkannt werden ist bereits ein großer Teil der Arbeit getan.

Fehlermaße sind die Bitfehlerrate $p = \text{Anzahl fehlerhafter bits} / \text{gesamtzahl gesendete bits}$, schlechterer Wert ist $p=0,5$, da durch invertierung immer wieder umstellbar. üblich in Technik $p = 10^{-4}$. ARQ(Error detection and automatic request repeat) ist normale Reaktion auf erkannte Fehler, dass eine einfache Wiederholung der Übertragung.

Restfehlerrate R ist eine wichtige Kennzahl weil, sie die unerkannten, fehlerhaften Bitfolgen die nach der Anwendung von Fehlererkennungsstrategien noch verbleiben

misst. $R = \text{Anzahl unerkannt fehlerhafter Bitkombinationen}/\text{Gesamtlänge in Bits der Information}$. \rightarrow Maß für Unversehrtheit der Daten

Telegrammeffizienz auch wichtig, weil sie eine Aussage erlaubt wie viel Informationen transportiert werden können und sie steht im Gegensatz zur Restfehlerquote, je sicherer Übertragung desto weniger Effizienz. $E = \text{fehlerfreie Infobits}/\text{Gesamtzahl übertragene bits} (\text{incl. Adresse Error check usw.})$ [Schnell u. Wiedemann, 2006, S. 31ff.]

Die genauen Berechnungen der Wahrscheinlichkeiten und der mittleren Zeit zwischen zwei Fehlern sind und noch viel mehr Details finden sich im Kapitel Datensicherung in [Schnell u. Wiedemann, 2006, S. 31f.] zu finden.

Um Fehler systematisch zu erkennen, existieren verschiedene Fehlererkennungsstrategien. Die einfachste ist der Paritätsbit, der lediglich die Quersumme des Telegramms angibt, $P=0$ für eine ungerade und $p=1$ für eine gerade Quersumme. Damit können diejenigen Fehler entdeckt werden, die eine ungerade Anzahl an Bitflips haben, die mit geraden leider nicht. Eine Erweiterung der Paritätssicherung ist die Blocksicherung, bei der die Paritäten über ein Array aus mehreren Telegrammen überprüft werden kann [Schnell u. Wiedemann, 2006, S. 34f.].

Beim sogenannten Cyclic Redundancy Check, in der Literatur häufig als CRC-Check bezeichnet, wird ein Telegramm als Zahl aufgefasst. Im Sender wird diese Zahl durch das Generatorpolynom G geteilt. Das Ergebnis wird verworfen, lediglich der Rest bei der Division wird an das Telegramm angehängt. Der Empfänger dividiert das empfangene Telegramm durch dasselbe Polynom G und falls sich ein Rest von 0 ergibt, war die Übertragung fehlerfrei. Dadurch lassen sich abhängig vom Generatorpolynom G unterschiedliche Güten der Fehlererkennung realisieren. In Abb. 2.4 sind die Vorgänge beim Cyclic Redundancy Check graphisch zusammengefasst.

2.2.1.5 Schnittstellen

Die physikalische Übertragung der Daten/Telegramme kann über verschiedenste Schnittstellen geschehen und erfolgt binär. Je nach Topologie des Bussystems und Protokoll, kann die Datenübertragung seriell oder parallel geschehen wie in Abb. 2.5 zu sehen. Bei einer parallelen Datenübertragung werden immer mehrere Bits gleichzeitig übertragen, weshalb eine hohe Übertragungsgeschwindigkeit erreicht werden kann, jedoch eine aufwändige im Sinne eines vermaschten Netzes notwendig ist. Daher erfolgt in der Praxis eher eine serielle Datenübertragung, also die einzelne Übertragung von Bits nacheinander über eine Leitung, wie sie zum Beispiel mit einer Bus-Struktur möglich ist [Schleicher, 2008, S. 13].

Bei der binären Datenübertragung werden lediglich zwei Zustände unterschieden, deren Signalwerte einen High-Pegel p_{high} und einen Low-Pegel p_{low} besitzen, die durch einen Bereich, in dem das Signal nicht definiert ist, voneinander getrennt sind [Schleicher, 2008, S. 9]. Die Geschwindigkeit der Datenübertragung wird in der Einheit Baud gemessen und definiert als übertragene Bits pro Sekunde (Bps) [Schleicher, 2008,

CRC (Cyclic Redundancy Check)	
1. Die Nachricht sei I	
Beispiel dezimal: $I = 14$	binär: $I = 110101$
2. Das Prüfpolynom sei G	
$G = \dots a_4 \cdot x^4 + a_3 \cdot x^3 + a_2 \cdot x^2 + a_1 \cdot x + a_0 \cdot x^0$	
d.: $\dots 0 \cdot x + 3 \cdot x^0 = 3$	b.: $\dots 1 \cdot x^3 + 0 \cdot x^2 + 1 \cdot x + 1 \cdot x^0 = 1011$
3. Der Hilfsvektor ist dann $H = 100 \dots 0$	
d.: $H = 10$	b.: $H = 10000$
4. Die Information I wird mit H multipliziert:	
$B = I \cdot H$	
d.: $B = 140$	b.: $B = 1101010000$
5. Das Produkt B wird durch G dividiert:	
$\frac{B}{G} = Q + \frac{R}{G}$	
d.: $\frac{B}{G} = \frac{140}{3} = 46 + \frac{2}{3}$	b.: $\frac{B}{G} = \frac{1101010000}{1011}$
	$= 1111011 + \frac{101}{1011}$
6. Der Rest R wird B hinzugefügt und das Ganze gesendet	
d.: $B + R = 142$	b.: $B + R = 1101010101$
7. Der Empfänger bildet	
$(B - R) : G$	
d.: $(B - R) : G = 138 : 3$	b.: $(B - R) : G = (B + R) : G = 1101010101 : 1011$
$= 46, R = 0$	$= 1111011, R = 0$
8. Es bedeutet $R = 0$ fehlerfreie Übertragung	

Abb. 2.4: Cyclic Redundancy Check aus [Schnell u. Wiedemann, 2006, S. 38]

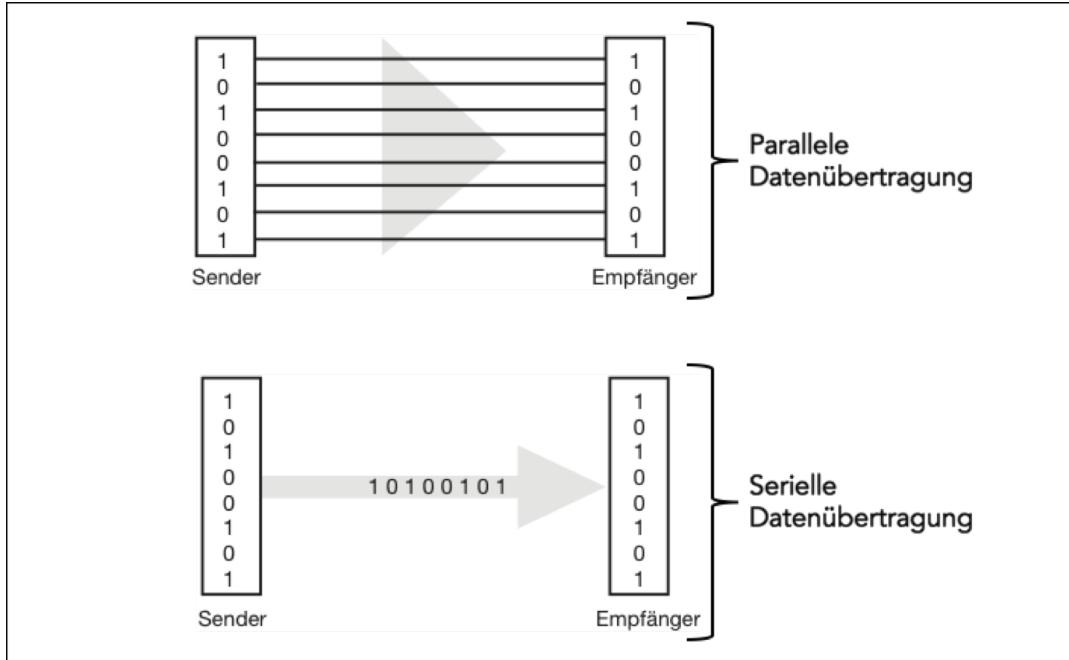


Abb. 2.5: Parallele und serielle Datenübertragung aus [Schleicher, 2008, S. 13]

S. 22].

Die gängigsten Übertragungsverfahren sind elektrische und optische Schnittstellen. Die elektrischen Schnittstellen gliedern sich wiederum in Strom- und Spannungsschnittstellen auf. Im weiteren Verlauf der Arbeit sind lediglich die Spannungsschnittstellen EIA² 232 und EIA 485, welche auch als EIA 232 und EIA 485 bezeichnet werden, relevant, weshalb der interessierte Leser weitere Informationen zu weiteren Schnittstellen in [Schleicher, 2008, S. 13ff.] und [Schnell u. Wiedemann, 2006, S. 57ff.] findet.

Die EIA 232 Schnittstelle ist wie bereits erwähnt eine Spannungsschnittstelle, die für Punkt-zu-Punkt Verbindungen geeignet ist und deren Pegel in Abb. 2.6 zu sehen sind. Die Pegel sind für Spannungen zwischen $-3V < p_{high} < -15V$ als logische „1“, der Low-Pegel für Spannungen zwischen $-3V < p_{low} < -15V$ als die logische „0“ definiert. Im Intervall $[-3V, 3V]$ ist das Signal nicht definiert, weshalb dieser Bereich möglichst schnell durchlaufen werden sollte. Da der Signalpegel von der Datenleitung hin zur Masse gemessen wird kann er nicht symmetrisch sein und ist damit erdunsymmetrisch. [Schnell u. Wiedemann, 2006, S. 57f.]

Typischerweise steht eine solche Schnittstelle jedem Rechner als COM-Port zur Verfügung und in der Automatisierungstechnik werden nur die RxD (Receive Data), TxD (Transmit Data) und GND Leitungen, die ein gemeinsames Bezugspotenzial definiert, verwendet. Wichtig bei der Verkabelung ist, dass die übertragende Leitung TxD immer mit der empfangenden Leitung RxD verbunden ist [Schleicher, 2008, S. 14f.].

Die EIA 485 Schnittstelle ist ebenfalls eine Spannungsschnittstelle, die jedoch für

² Abkürzung der Normen und Standards die von der Electronic Industries Alliance entwickelt wurden.

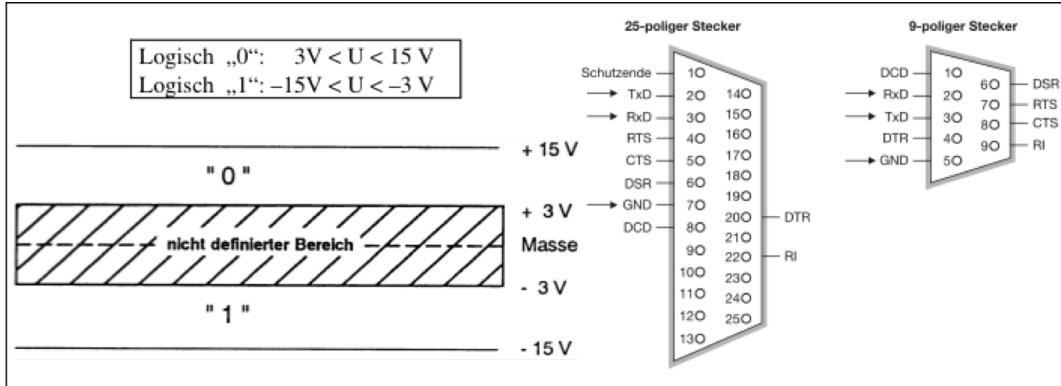


Abb. 2.6: Links: Spannungspegel EIA 232-Schnittstelle aus [Schnell u. Wiedemann, 2006, S. 57]
Rechts: Stecker EIA 232-Schnittstelle aus [Schleicher, 2008, S. 14]

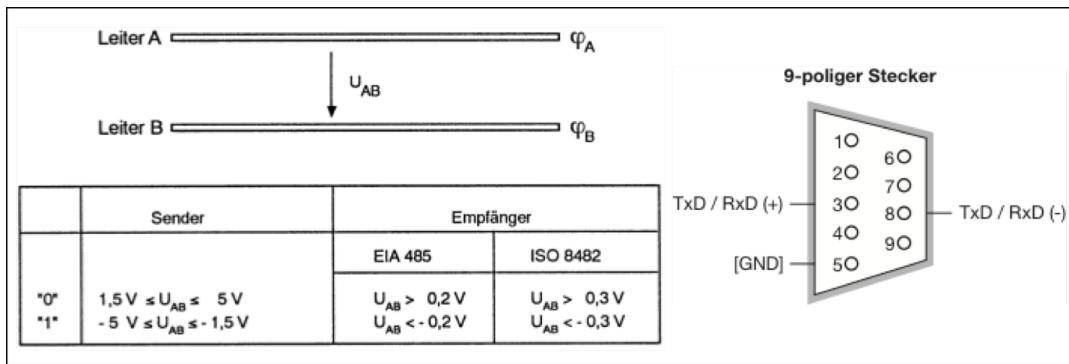


Abb. 2.7: Links: Spannungspegel EIA 485-Schnittstelle aus [Schnell u. Wiedemann, 2006, S. 60]
Rechts: Stecker EIA 485-Schnittstelle aus [Schleicher, 2008, S. 19]

Mehrpunktverbindungen geeignet ist und die in der Norm ISO 8482 beschrieben ist. Die Signalübertragung erfolgt über zwei Übertragungsleitungen, die in der Regel als verdrilltes und abgeschirmtes Zweidrahtleitung ausgeführt sind. Die Signalpegel entsprechen der Differenzspannung U_{AB} zwischen den beiden Leitungen, die innerhalb des Intervalls von $[-7V, 12V]$ bezogen auf die Masse liegen muss. Durch das verdrillte Leitungspaar, wirken sich mögliche Störgrößen auf die Spannung beider Leitungen gleichermaßen aus, wodurch die Spannungsdifferenz unverändert bleibt und eine erhöhte Störfestigkeit gegenüber der EIA 232-Schnittstelle erreicht wird. Bei der Pegelfestlegung werden für Empfänger und Sender gibt es verschiedene Vorgaben, wie in Abb. 2.7 graphisch dargestellt. So müssen die Sender eine Differenzspannung zwischen $-1,5V < U_{AB} < -5V$ für logische „1“ und $1,5V < U_{AB} < -5V$ für logische „0“ leisten können. Empfänger hingegen müssen in der Lage sein Spannungsdifferenzen von $U_{AB} < -0,3V$ als logische „1“ und $0,3 < U_{AB} < -0,3V$ als logische „0“ zu detektieren und zu interpretieren [Schnell u. Wiedemann, 2006, S. 59ff.].

Es besteht ein geringer Installationsaufwand und die Datenübertragung erfolgt über die TxD/RxD + und TxD/RxD - Leitungen, ein gemeinsames Bezugspotential über die GND Leitung wird nicht zwingend benötigt, jedoch wird ein Leitungsabschluss an beiden Enden des Buskabels benötigt. [Schleicher, 2008, S. 19f.].

Um Netzwerk zu erweitern, können verschiedene Bauteile eingesetzt werden. Sogenann-

te Repeater sind aktive Bauteile, die lediglich einen kurze Zeitverzögerung verursachen und nur Netzwerke der selben Schnittstelle miteinander verbinden kann. Er ist ein aktives Bauteil, dass das Datensignal lediglich verstärkt, indem er die empfangenen Bits blind kopiert und verstärkt auf das angeschlossene Netzwerk überträgt, weshalb er für die Kommunikationsteilnehmer unsichtbar ist [Schnell u. Wiedemann, 2006, S. 79f.].

Eine Möglichkeit um Netzwerke verschiedener Art zu verbinden bieten Bridges und Gateways. Erstere werden auch als Schnittstellenumsetzer bezeichnet und kommen zum Einsatz, wenn trotz gleichem Übertragungsprotokoll genutzt unterschiedliche physikalische Schnittstellen oder Übertragungsmedien genutzt werden sollen [Schnell u. Wiedemann, 2006, S. 80f.]. Er erkennt die Datenflussrichtung automatisch und wandelt die Pegel der Schnittstellen in den jeweils anderen um [Schleicher, 2008, S. 21]. Die sogenannten Gateways dienen der Kopplung von Netzwerken, die verschiedene Architekturen aufweisen, also neben unterschiedlichen physikalischen Schnittstellen auch andere Übertragungsprotokolle verwenden. Das Gateway ist demnach umfassender als die Bridge und erweitert deren Funktionen um die Übersetzung der Signale von einem Übertragungsprotokoll in das jeweils andere [Schnell u. Wiedemann, 2006, S. 84f.].

2.2.2 OSI-Kommunikationsmodell

Aufgrund der großen Anzahl verschiedener technischer Systeme existieren auch viele verschiedene Arten der Kommunikation untereinander. Bei der genaueren Betrachtung der Kommunikation wird ersichtlich, dass diese oftmals ähnlich abläuft und sich durch ein Meta-Schema beschreiben lässt [Schnell u. Wiedemann, 2006, S. 8]. Um die Kommunikation auch über verschiedenen Systeme hinweg zu ermöglichen und sie zu formalisieren, wurde von der *International Organization for Standardization* 1984 ein abstraktes Referenz-Modell entwickelt, dass in der *ISO-Norm 7498-1* beschrieben ist. Es dient der Entwicklung und Verbesserung von Standards für den Informationsaustausch sowie als Referenz für bestehende Standard um eine gewisse Konsistenz zu wahren [osi, 1996, S. 1]. Das Ziel bei dem Entwurf des Modells war es, eine Menge von Standards zu schaffen um autonomen Systemen die Kommunikation untereinander zu ermöglichen [osi, 1996, S. 4].

Das sogenannte Open System Interconnection Modell wird zunächst allgemein erläutert, da die Kommunikation von technischen Systemen im Rahmen der Arbeit eine zentrale Rolle spielt, und wird anschließend im Anwendungskontext mit den eingesetzten Protokollen und Schnittstellen referenziert.

Zunächst wird im Standard definiert, womit sich das Modell beschäftigt und abgegrenzt welche Aspekte im Modell keine Berücksichtigung finden [osi, 1996, S. 3]:

„OSI is concerned with the exchange of information between open systems
(and not the internal functioning of each individual real open system).“

Das OSI-Modell beschäftigt sich also zentral mit dem Austausch von Informationen zwischen verschiedenen offenen Systemen und allen dabei anfallenden Aktivitäten. Diese sind sehr umfangreich und lassen sich in folgende Bereiche gliedern [osi, 1996, S. 3f.]:

- Der Austausch von Informationen zwischen offenen Systemen,
- die physischen Medien zur Verbindung von offenen Systemen und deren Transportmöglichkeit von Informationen,
- die Vernetzung von offenen Systemen,
- die Interaktion zwischen offenen Systemen und deren Fähigkeit zur Kooperation bei der Datenübertragung.

Bezogen auf den Austausch von Informationen überschneiden sich die physische Verbindung und die Vernetzung und entsprechen zusammen der Infrastruktur und deren Architektur, die zur Übertragung zur Verfügung steht. Die Interaktion umfasst weitaus mehr Aufgaben: Neben der Synchronisation der Prozesse, die Daten austauschen wollen, muss auch die Darstellung der auszutauschenden Daten und eventuell notwendige Transformationen beachtet werden, um eine Kompatibilität unterschiedlicher Systeme zu erreichen. Weitere wichtige Aufgaben sind die Datenspeicherung, deren Integrität und die Sicherheit beim Austausch hinsichtlich Fehler und Einsicht von Außen [osi, 1996, S. 4]. Es ist leicht zu erkennen, dass die technische Kommunikation einen sehr umfangreichen und komplizierten Prozess darstellt. Daher wird der Kommunikationsprozess im OSI-Modell stark abstrahiert und in sieben abstrakte Ebenen gegliedert. Die einzelnen Ebenen sind in Abb. 2.8 dargestellt und dienen dazu verschiedene Aufgaben des Kommunikationsprozesses in Teilaufgaben zusammenzufassen.

Die Ebenen werden Schichten genannt und haben klar definierte Aufgaben und Schnittstellen zu ihren Nachbarschichten. An diesen Schnittstellen werden Dienste bereitgestellt, die von den anderen Ebenen genutzt werden können. Durch diesen Aufbau können einzelne Schichten einfach bearbeitet oder ausgetauscht werden, ohne die Gesamtfunktionalität zu gefährden. Außerdem kann ein System auch aus Komponenten verschiedener Hersteller zusammengesetzt werden, womit diese Architektur nachweislich als Basis für offene Systeme dient. In Abb. 2.8 ist ebenfalls dargestellt, dass die Schichten eins bis vier auch als Übertragungsschichten beziehungsweise Transportsystem zusammengefasst werden, weil sie für die Datenübertragung zwischen Systemen als gemeinsame Aufgabe haben. Die Schichten fünf bis sieben werden als Anwendungsschichten bezeichnet weil sie bei der Datenübertragung die Zusammenarbeit zwischen der Anwendersoftware und dem Betriebssystem sicherstellen [Schnell u. Wiedemann, 2006, S. 8f.].

Die Schnittstellen/Dienste zwischen den Schichten werden als Service Access Points bezeichnet und besitzen jeweils eine eindeutige Adresse, die oberhalb liegende Schicht ist der Service user, da er den Service daer unterhalb liegenden Schicht nutzt, dem service provider. Die Dienste können in verbindungsorientierte und verbindungsunab-

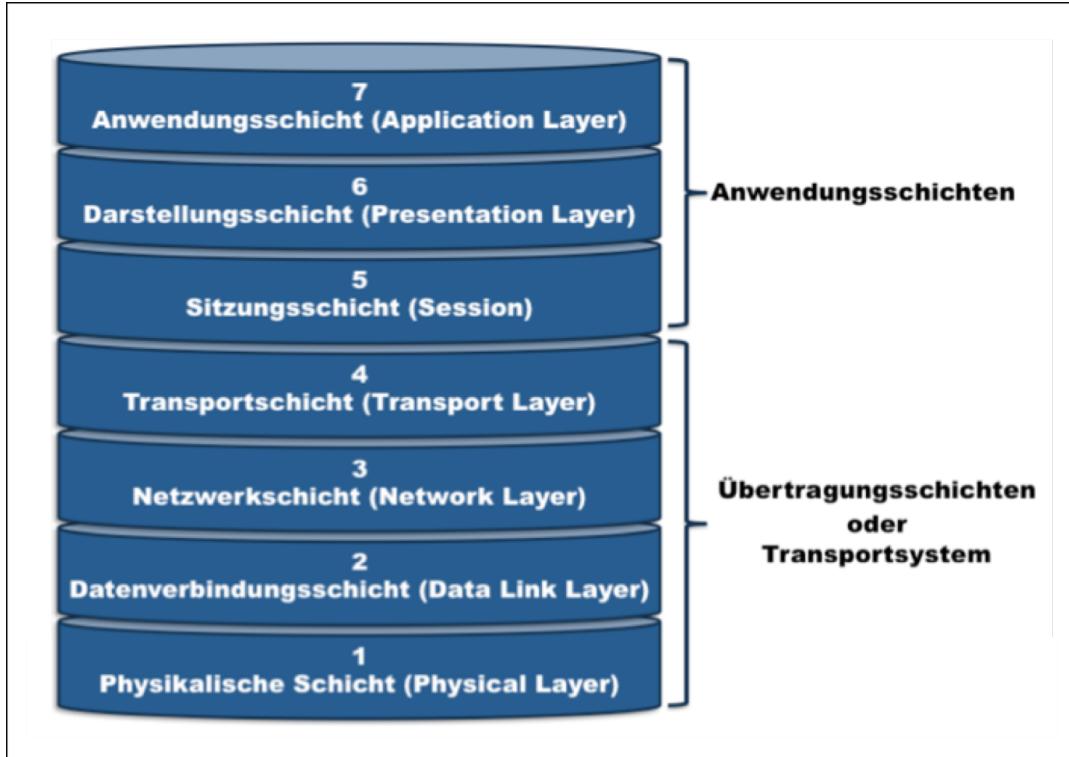


Abb. 2.8: Die sieben Schichten des *Open System Interconnection* Modells verändert nach [Schnell u. Wiedemann, 2006, S. 10] und [osi, 1996, S. 28]

hängige unterschieden werden. Für den Datenausausch stehen folgende Dienste zur Verfügung Bei der Abhandlung der Dienstaufgaben stehen die vier Dienstvorgänge der Client/Server Architektur zur Verfügung, die zusammengefasst in Abb. 2.9 abgebildet sind [mod, 2006a, S. 2f.] :

request - Anforderung indication - Meldung response - Antwort confirmation - Bestätigung Bestätigten Diensten stehen alle vier Vorgänge zur Verfügung, unbestätigten lediglich die Anforderung und Meldung. Typische Dienste sind Connect, disconnect, data

[Schnell u. Wiedemann, 2006, S. 14f.].

Im Folgenden wird kurz auf die einzelnen Schichten von Unten nach oben eingegangen bevor das Zusammenwirken der einzelnen Schichten anhand eines Beispiels verdeutlicht wird.

Die erste, physikalische Schicht stellt die mechanischen und elektrischen Möglichkeiten zur physischen Verbindung von Systemen zur Verfügung, um die Datenübertragung der einzelnen Bits zu ermöglichen [osi, 1996, S. 49f.]. Sie legt also die mechanischen und elektrischen Eigenschaften der Übertragung fest, also die Endsystemkopplung (Stecker), die Kabelspezifikationen und die Zuordnung der Anschlüsse sowie die Art der Codierung und die Spannungspegel zur Übertragung. In der Regel werden dazu bestehende Normen genutzt, wie zum Beispiel die elektrischen Übertragungsstrecke nach EIA 485-Norm, welche im Folgenden noch erläutert wird.

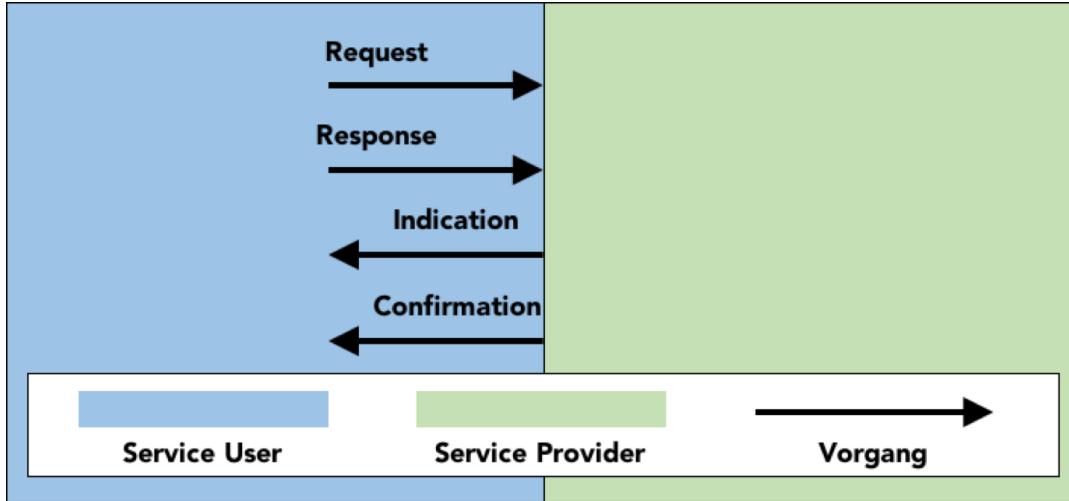


Abb. 2.9: Die vier Dienstvorgänge

Ein wichtiger Aspekt der Schicht ist es, dass die Spezifikation der Strecke und nicht das physikalisches Medium selbst Teil der Schicht eins ist, denn die Kommunikation ist unabhängig von der konkreten Ausprägung der Schicht [Schnell u. Wiedemann, 2006, S. 9].

Die zweite Schicht betrachtet Kommunikation zwischen zwei systemen. Deshalb stellt die Datenverbindungsschicht, stellt funktionale und prozedurale Möglichkeiten für den verbindungsauflauf/trennung erhaltung und den Transfer von Dateneinheiten Verfügung. Ermöglicht dem Netzwerkschicht die Kontrolle über die Verbindung von Data circuits physikalisch, sowie fehlerabfangen der physikalischen schicht [osi, 1996, S. 46f.] Aufgabe ist sicherer transport von Station zu station. Datensicherung -> Verpacken um Übertragungsfehler erkenntlich zu machen in data frames. In Frames sind die maximale Anzahl Datenbits für Rohdaten spezifiziert, weiterhin wird Information zur Übertragung hinzugefügt. Die zusatzinfo kann Prüfsumme und Anfang und Ende des Rahmens enthalten oder quittierung eines telegramms und dient dazu fehlerhafte Übertragung oder etwas verloren gegangen zu überprüfen. MAC mit Schicht eins, LLC mit Schicht drei. Wuelle Wiki bisher MAC regelt den Zugriff auf das ohyische Medium zur Kommunikation, kontrolliert oder konkurriert. LLC verteilt die Daten passend in Schicht drei und gibt die Daten von schicht drei an passende MAC für schicht eins weiter und fügt Diese Infos von oben hinzu (Adressen Empfänger und Sender und zusatzinfo wie control für Steuerung von Datenfluss oder so).

Wichtig Die schicht hat jedoch keine Kenntnis über Inhalte der Daten![Schnell u. Wiedemann, 2006, S. 9ff.]

verbindungsloser Dienst heisst keine Verbindung zwischen Kommunikationspartnern, Datenpakete werden wie Brief ganz in Netzwerk gespeist mit Zieladresse versehen und weitertransportiert, ohne beeinflussung des transportweges durch benutzer des netzwerkdienstes. Später ist Modbus ein Beispiel dazu erklärt. verbindungsorientierte Dienste heisst ein virtueller Kanal zwischen kommunikationspartnern wird zur Verfüzung gestellt, eingeriechtet: Verbindungsauflauf, Datenautausch, Verbindungsabbau,

wie telefongespräch. [Schnell u. Wiedemann, 2006, S. 11f.]

Die dritte Schicht beschäftigt sich mit dem Netzwerk als Ganzes. Die Aufgaben der Netzwerkschicht hängen ein wenig ab von verbindungsorientierung Daher beschäftigt sie sich mit dem Aufbau , der erhaltung und Datenaustausch und dem trennen von Netzwerk Verbindungen zwischen offenen systemen i Netzwerk, also schnittstellen. weiterhin ist sie für den Transport von Daten im netzwerk zustaändig, also insbesondere auch für die Festlegung der Route(Wegsteuerung) der Daten im Netzwerk [osi, 1996, S. 41f.]. Also Kontrolle von Verkehr im Netzwerk, d.h. Anzahl Pakete im Netzwerk, Staus [Schnell u. Wiedemann, 2006, S. 11f.]

Die vierte Schicht ist die Transportschicht und zuständig für die transparente Übertragung von Daten zwischen Prozessen und ist völlig unabängig bzw losgelöst von den Gedanken an Kosten und Verlässlichkeit der Datenübertragung, da dies aufgaben der unteren schichten sind. Sie kümmert sich um die optimale nutzung von Netzwerk services/Nutzung [osi, 1996, S. 37f.]. Adressierung der Teilnehmer, Aufbau und Abbau für Transportverbindung wzischen Kom.Partner prozessen(Sammel Einzel Mehrere), Fehlerbehandlung verbindnung und flusskontrolle, Synchronisierung der dtanetaus-tauschenden Prozesse.Zerlegung der Daten aus Sitzungsschicht in Transportierbare Einheiten. Internetworking, Umsetzung verschiedener Protokolle Gateway Aufgaben. Aufbau Verbindung legt Art fest, Punkt zu Punkt oder Broadcast/Multicast (Alle bzw einige Teilnehmer gleichzeitig) [Schnell u. Wiedemann, 2006, S. 12f.].

Die fünfte Schicht, die Sitzungsschicht, startet eine Sitzungsverbindung mit bestimmter Adresse wenn dieser Prozess von einer höheren Schicht angefordert wird. Diese Verbindung dient dazu, den dialog von kooperiernden Prozessen auf der eines höheren Darstellungsbene durch eine Sitzungsverbindung zu synchronisieren und deren Daten-austausch zu organisieren. verknüpft die Sitzungsadressen mit den Transportadressen, also die Anwendungsschichten mit dem Transportsystem [osi, 1996, S. 35]. Benutzung des Transportsystems über die Schnittstelle zur Transportschicht. Je nach Funktionen der höheren Schichten entsprechender Funktionsumfang BCS Basic Combined Subset - Verbindungssteuerung und Datenübertragung BAS Basic Activity Subset - Aktivitätsverwaltung BSS Basic synchronized Subset - Synchronisierung [Schnell u. Wiedemann, 2006, S. 13].

Die sechste Schicht, die Darstellungsschicht ist nach [osi, 1996, S. 33f.] für die Darstellung der Daten die von Anwendung-Entitäten entweder kommuniziert oder bei deren Kommunikation referenziert werden. Sie stellt außerdem eine gemeinsame Represantion der übertragenen Daten dar zwischen Anwendungs-Entitäten und befreit diese dadurch von Syntaxabhängigkeiten. [Schnell u. Wiedemann, 2006, S. 13f.] stellt fest, dass die Dienste die der Darstellung der transferierten Daten dienen wie die Codierung der zu übertragenden Daten, der verwendete Zeichensatz und die Darstellung der Daten auf dem Bildschirm oder Drucker. Semantik/Syntax beim Nachrichtenaustausch und der beiden Kommunizierenden Prozesse. Evtl Komprimierung um Zeit und Kosten zu sparen .

Die siebte und letzte Schicht stellt lediglich eine Möglichkeit für Anwendungsprozesse zur Verfügung um auf die OSI Umgebung zuzugreifen. Jeder Anwendung stellt im OSI genau einen Anwendungsprozess dar, verschiedene Anwendungsprozesse für verschiedene Anwendungen und vice versa [osi, 1996, S. 32] stellt Funktionen bereit, mit denen der Benutzer auf das Kommunikationssystem zugreifen kann, wobei der Benutzer idR ein Computerprogramm und kein Mensch ist. [Schnell u. Wiedemann, 2006, S. 14]

Im folgenden werden die verwendeten Modbus Protokolle und Spezifikation ind Anwendung Berzug zum OSI Modell gebracht um den praktischen Nutzen davon klar zu machen.

2.2.3 Modbus Kommunikationstechnologie

Zunächst erfolgt die Einordnung ins OSI Modell und anschließend eine Klassifizierung der Spezifikationen gemäß der zuvor von Bussystemen.

Das Modbus Protokoll teilt sich auf verschiedene Protokolle auf, zum einen auf das Application Layer Messaging Protocol auf oberster Ebene, welches auf das Modbus Over Serial Line Protocol sowie das Ethernet TCP/IP Protokoll aufbaut. Das Application Layer Messaging Protocol lässt sich im OSI Referenzmodell in die siebte und oberste Schicht (Anwendungsschicht) einordnen wie in Abb. 2.10 dargestellt. Bei der Nutzung des Modbus over Serial line Protocol implementiert diese die zweite Schicht und die Ebenen drei bis sechs sind leer implementiert. Als physikalische Schicht werden die Übertragungsstandards nach EIA 485 oder nach EIA 232 [mod, 2006b, S. 2]. Anstatt des Over serial line kann auch das Modbus Messaging On TCP/IP Protocol verwendet, dann implementiert dieses zusammen mit dem dem TCP/IP Standard zusammen die Netzwerkschicht im OSI-Referenzmodell. Das Ethernet implementiert dabei die Datensicherungsschicht Nummer zwei und dessen physikalische Spezifikationen die unterste physikalische Schicht und deren Schnittstellen. Die Ebenen vier bis sechs sind auch bei Nutzung dieser Kommunikationsweise leer implementiert [mod, 2012, S. 2f.]. Eine Einführung zum Ethernet und TCP/IP Standard findet sich in Schnell u. Wiedemann [2006], detaillierte Ausführungen dazu in Furrer [2003].

Nun wird das Modbus Application Layer Messaging Protocol erläutert. Anschließend werden die Modbus Protokolle der unteren Ebenen erläutert bis hin zu den physikalischen Schnittstellen.

Das Modbus Application Layer Messaging protocol ist ein Protokoll, dass verschiedene Netzwerke und Bussysteme zur Master/Slave Kommunikationen von verbundenen Geräten nutzen kann [mod, 2012, S. 2f.], jedoch übernimmt im Rahmen des Protokolls der Master die Rolle des Clients und die Slaves die Rollen als Server, da der Client alleinig die Anfragen stellt und die Server lediglich auf Anfragen antworten. Das Modbus Protokoll definiert eine gemeinsame Telegrammstruktur, bezogen auf Inhalte und Rahmen der Nachricht. Damit ermöglicht es die Kommunikation zwischen verschie-

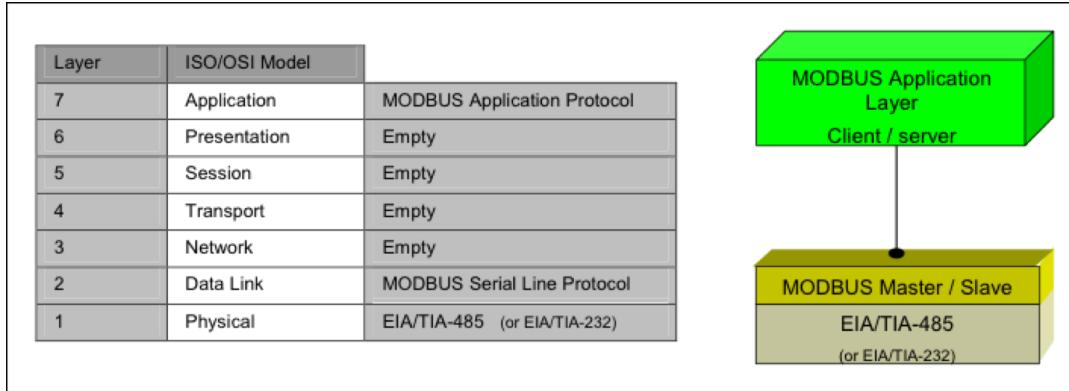


Abb. 2.10: Die Modbus Kommunikation im OSI-Referenzmodell aus [mod, 2006b, S. 5]

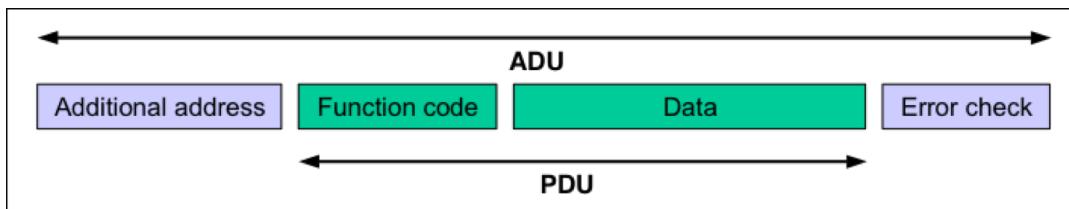


Abb. 2.11: Allgemeiner Rahmen für Telegramme nach dem Modbus Anwendungsprotokoll aus [mod, 2012, S. 3]

denen Geräten innerhalb eines Netzwerks, welche unabhängig von Art und Typ des darunter liegenden Netzwerks sind. Es beschreibt außerdem wie die Kommunikation abläuft, wie der Client eine Anfrage an ein Server stellt und wie diese auf die Anfragen reagieren und antworteten und beschreibt wie Fehler bei der Übertragung entdeckt und darauf hingewiesen werden [MODICON, 96, S. 2f.]. Die Kommunikation kann dabei über eine serielle Leitung nach EIA 485 sowie über Ethernet Netzwerk erfolgen. Über Gateways, wie in Abschnitt 2.2 bereits erläutert, kann die Kommunikation auch über verschiedene Typen von Bussystemen oder Netzwerken geschehen [mod, 2012, S. 3f.].

Das Modbus Protokoll stellt ein allgemeines Gerüst für Telegramme zur Verfügung, welches in Abb. 2.11 dargestellt ist. Die einfache Protocol Data Unit (PDU) enthält die eigentlichen Informationen die ausgetauscht werden sollen und sind daher unabhängig vom Netzwerk oder dem Bussystems. Zu den Informationen gehören die eigentlichen Daten, welche leer sein können oder Daten enthalten die der Slave, in Rahmen von Modbus auch als Server bezeichnet, benötigt, und ein Funktionscode, der ein Byte groß ist und beschreibt welche Art der Aktion Reaktion gefordert ist/wird. Die sogenannte Application Data Unit (ADU) enthält zusätzlich zur PDU weitere Informationen zur Adressierung im Netzwerk und zur Fehlererkennung und ist daher abhängig vom Netzwerk. Der Buszugriff ist damit als Master-Slave geregelt.

Für die Kommunikation wird innerhalb des Masters, der im Rahmen von Modbus auch als Client bezeichnet wird, die Kommunikation durch eine ADU initialisiert. Das genaue Format der ADU wird ist nach Modbus Protokollspezifikation festgelegt. Das Schema einer Transaktion läuft nach dem Prinzip in Abb. 2.12 ab. Wenn eine ADU

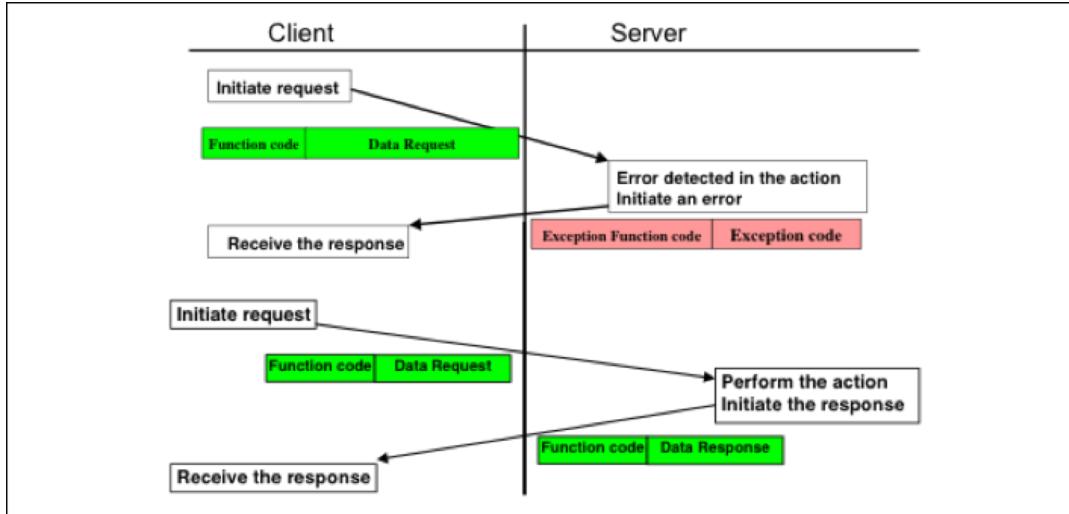


Abb. 2.12: Transaktion mit dem Modbus Protokoll nach [mod, 2012, S. 4]

fehlerfrei empfangen wurde nutzt der Server das Funktioncode Feld um anzugeben, ob seine Antwort ebenfalls eine normale, fehlerfrei Antwort, durch eine einfaches Echo des empfangenen Funktionscodes, ist oder ob ein irgendein Fehler aufgetreten ist durch einen Exception code. Die Größe der PDU ist grundsätzlich durch die serielle Kommunikation begrenzt auf 256 Bytes, da jedoch noch zwei Bytes für einen Cyclic Redundancy Check und ein Byte für die Server Adresse reserviert werden müssen ist die PDU auf 253 Bytes begrenzt. Ein weiterer, wichtiger Aspekt ist, dass Modbus die „big-Endian“ Codierung/Repräsentation für Daten verwendet verwendet, falls der numerische Wert größer als ein einzelner Byte ist [mod, 2012, S. 3ff.]. Anhand des Beispielder Uhrzeit kann die Bedeutung der Big-Endian Repräsentation einfach erläutert werden: Die Daten werden so aufgeteilt, dass zunächst die Daten mit der höchsten Wertigkeit, also den Stunden zuerst gesendet werden, anschließend den Minuten und zum Schluss die Sekunden, unabhängig von deren numerischem Wert. Also werden nacheinander die Code 03, 50, 12 empfangen bedeutet dues dass die Uhrzeit 03:50:12 ist. Der interessierte Leser wird für eine weitere Ausführungen auch zu little-Endian Darstellung in Bertrand Blanc [2005].

Das Modbus Datenmodell basiert darauf, dass auf die Daten in vier verschiedenen Tabellen mit unterschiedlichen Funktionen und Datenobjekten zugegriffen werden kann:

- In die Discretes Input, welche Single Bit Objekte enthält und lediglich gelesen werden kann,
- die Coils, welche ebenfalls Single Bit Objekte enthält jedoch gelesen und beschrieben werden darf,
- die Input Registers, die Datenobjekte als 16-Bit Wort enthält und wiederum nur gelesen werden kann
- und die Holding Registers, dessen 16-Bit Wort Objekte wiederum gelesen und beschrieben werden dürfen.

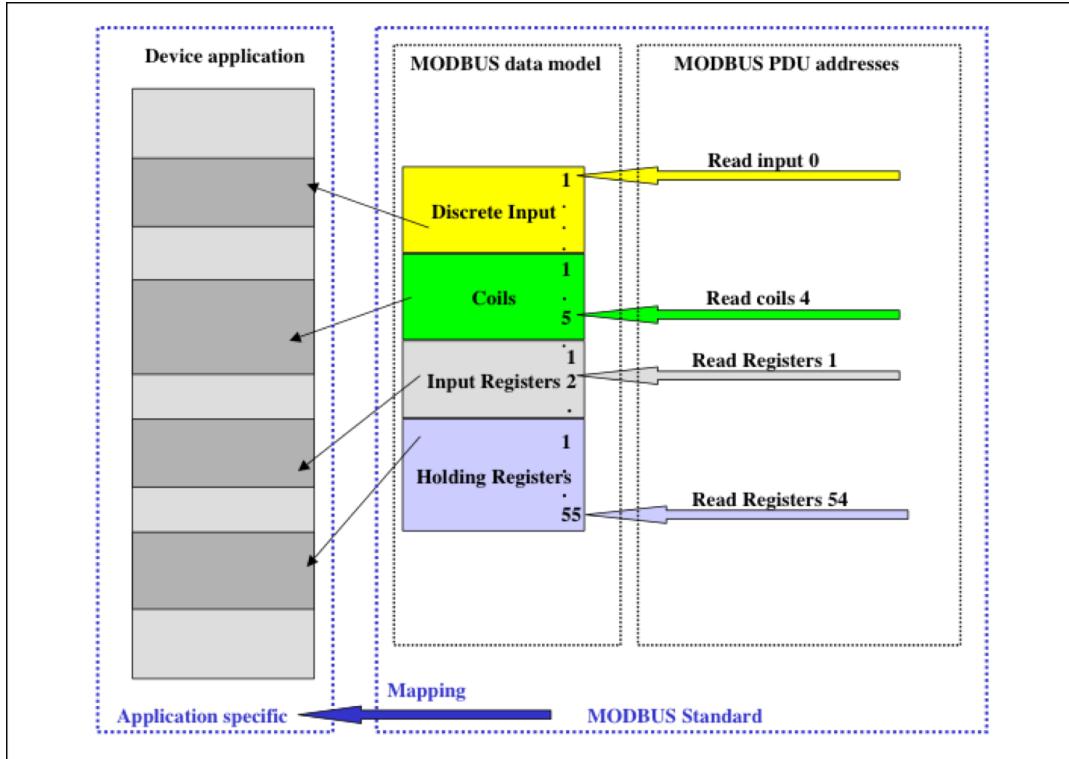


Abb. 2.13: Datenmodell und Adressierung nach dem Modbus Protokoll aus [mod, 2012, S. 8]

16 bit wort entspricht einer Folge von 16 bits/binärzeichen das also einen dezimalen Zahlenwert zwischen 0 und 65.536. Nach der IEC 61131-3 Norm entspricht es in iim Rechner einem Integer Wert.

Die Daten selbst können auch innerhalb einer Tabelle abgelegt sein, müssen lediglich über die vier angegebenen Tabellen zugreifbar sein, also die Referenz auf die Daten muss gegeben sein, der Rest ist egal. Jede Tabelle besitzt Daten adressiert zwischen 0 und 65535 und ist damit auch nach oben begrenzt was Daten angeht. Welche Daten wo genau stehen, also unter welcher Adresse kann von Gerät zu Gerät unterschiedlich sein und wird vom Gerätehersteller festgelegt. [mod, 2012, S. 6ff.]

Die Funktionscodes starten bei 1 und können bis 255 genutzt werden und sind wie bereits angesprochen in der PDU enthalten und definieren welche Aktion ein Server ausführen soll. In erster Linie dienen sie dem Datenzugriff in den Tabellen, können aber auch für Diagnosen oder nutzerdefinierte Aktionen benutzt werden. Daher lassen sich die Funktionscodes in drei große Gruppen aufteilen, die öffentlichen, nutzerdefinierbaren und die reservierten Funktionscodes. Die öffentlichen sind wohldefiniert, unique, dokumentiert und sind auf Konformität getestet und sind daher einfach, schnell und sicher nutzbar. Die vom Nutzer definierbaren Funktionscodes können genutzt werden um von den öffentlich bereitgestellten nicht bereitgestellte Funktionen eigens zu definieren/nutzbar zu machen. Die reservierten Codes sind von wenigen Unternehmen, welche an der Entwicklung des Modbus Protokolls beteiligt waren, für deren hinterlassene Produkte reserviert und daher nicht öffentlich nutzbar [mod, 2012, S. 10ff.].

Als nächstes folgen die beiden verschiedenen Modbus Over Serial Line Protocol und Modbus Messaging On TCP/IP Protocol

Beim Modbus Over Serial Line Protocol können die Daten über zwei unterschiedliche Modi übertragen werden, der RTU und ASCII Modus. Näheres Der ASCII Modus ist optional und wird in mod [2006b] detailliert beschrieben. Der RTU Modus wird von allen modbusfähigen Komponenten unterstützt und spezifiziert das folgende Format zur Übertragung der einzelnen Bytes: Jede Byteübertragung beginnt mit einem Startbit, auf das zu übertragende Byte, bestehend aus acht einzelnen Bits, folgt, bevor die Übertragung optional von einem Paritätsbit und einem Stopppbit beziehungsweise lediglich von zwei Stopppbits abgeschlossen wird. Dabei wird jedes zu übertragende Byte als zwei 4-bit hexadezimales Zeichen übertragen[mod, 2006b, S. 12f.]. Die Paritätsprüfung ist optional und dient der Fehlerüberprüfung des Telegramms, wie bereits in Abschnitt 2.2 erläutert. Der Rahmen eines gesamten Modbus RTU Telegramms besteht aus der Slave Adresse, die für jeden Slave eindeutig ist und zwischen 1 und 247 liegt, und dem Function Code, die jeweils aus einem Byte bestehen. Darauf folgen die eigentlichen Informationen für die 0 bis 252 Bytes vorgesehen sind. Abgeschlossen wird der Rahmen durch ein CRC Feld, dass aus einem CRC Low und einem CRC High byte aufgebaut ist und dazu dient das Telegramm auf Fehler zu überprüfen. Der Ablauf und Vorgang des CRC Checks ist detailliert in mod [2006b] beschrieben. Die Übertragung eines Telegramms erfolgt byteweise, wie zuvor beschrieben. Die Datensicherung findet also durch Parität und CRC auf verschiedenen Ebenen statt. Die genaue Übertragungszeit eines Bytes und einer Nachricht hängt von der Baudrate ab. Um den Beginn und den Abschluss eines RTU Rahmens eindeutig zu definieren, geschieht dies in Abhängigkeit von der Übertragungsgeschwindigkeit. Zwischen einzelnen Bytes innerhalb eines Rahmens folgt ein stilles Intervall, dass je nach Länge angibt ob das Telegramm beendet ist. Auf eine Intervall kleiner gleich der anderthalbfachen Übertragung eines Bytes folgt eine weiteres Byte. Ist das stille Intervall länger als die dreieinhalbfache Byteübertragungszeit markiert dies das Ende eines Telegramms und den Beginn eines nächsten Telegramms [mod, 2006b, S. 13]. Diese Zusammenhänge sind zur Veranschaulichung in Abb. 2.14 zusammengefasst.

Der Implementierungsleitfaden legt auch die Spezifikationen der physikalischen Schicht fest, die nun folgen. Er schlägt vor die EIA 438 Schnittstelle als elektrisches/physikalisches Interface zu verwenden, erlaubt aber auch weiterhin die Implementierung durch die EIA 232 Schnittstelle, beides über ein verdrilltes Leiterpaar. Weiterhin werden die Datenraten von 9.600 und 192.000 bps und eine Even Parität bei der Byteübertragung als Standard festgelegt. Die Standardverdrahtung der Komponenten erfolgt bei beiden elektrischen Standards über ein verdrilltes Leiterpaar und einer gemeinsamen Verbindungsleitung common. Die beiden Leitungen des verdrillten Paars werden mit D1, welche auch als D+ oder A Leitung bezeichnet wird, und D0, welche auch als D- oder B Leitung bezeichnet wird, bezeichnet. Ein Standard Netzwerk besteht aus maximal 32 Teilnehmern, dass durch den Einsatz von Repeatern auch vergrößert werden kann. Außerdem wird die Bus-Struktur als Topologie beschrieben, nach der

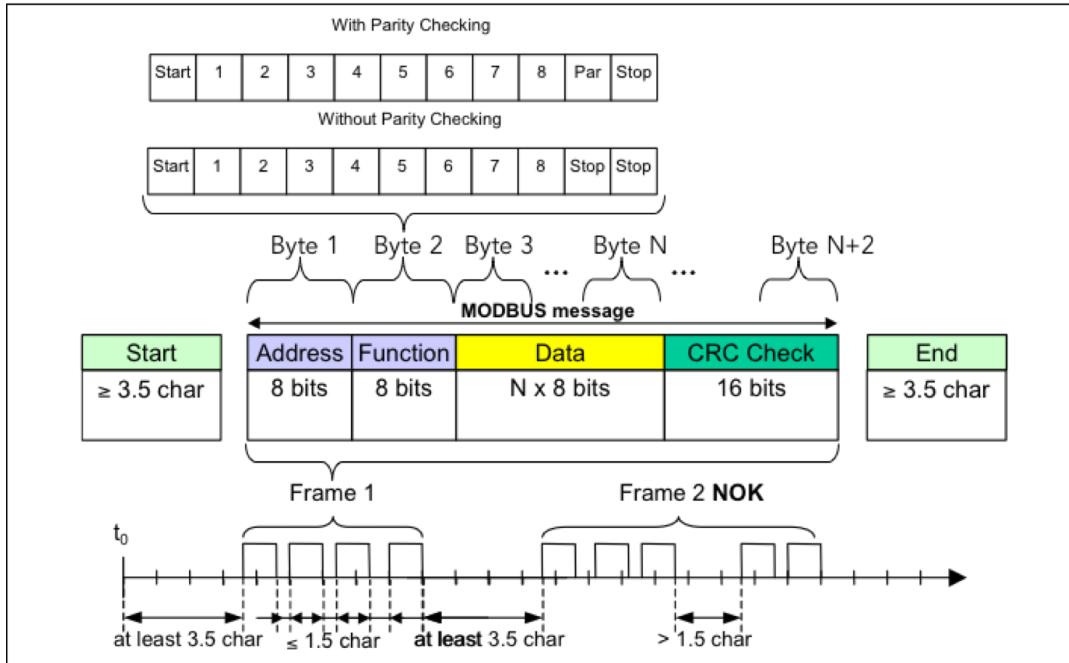


Abb. 2.14: Serielle Kommunikation über Modbus RTU nach [mod, 2006b, S. 12f.]

die einzelnen Komponenten im Netzwerk angeordnet werden, unter der Voraussetzung, dass die Busleitung an beiden Enden durch einen Widerstand von 150 Ohm zwischen der D0 und D1 Leitung abgeschlossen werden [mod, 2006b, S. 20ff.]. Die Verbindung der Kabel kann im einfachsten Fall durch Schraubklemmen erfolgen, jedoch können auch genormte mechanische Interfaces, also Standard Steckverbindungen genutzt werden, deren Einsatz und Verkabelung/Anschlüsse in [mod, 2006b, S. 29ff.] detailliert beschrieben sind.

XXXXXXXX

Beim Modbus Messaging on TCP/IP Protocol stellt die Möglichkeit zur Verfügung, Geräte, die über ein Ethernet miteinander verbunden sind, über ein Client/Server Modell kommunizieren zu lassen. Des Weiteren erlaubt es dieses Protokoll explizit über Bridges, Gateways oder Router Netzwerke miteinander zu verbinden. Dabei dürfen auch serielle Subnetzwerke zu verbinden und erlaubt auch zwischen diesen Endgeräten die Kommunikation [mod, 2006a, S. 2f.]. Diese Kommunikationsarchitektur ist auf in Abb. 2.15 dargestellt.

Außerdem ist eine leicht verschiedene ADU vorhanden, wie in Abbildung REF ADU TCP zu sehen. Modbus Application Protocol Header MBAP Header ist 7 bytes lang und enthält unit identifier ähnlich slave adress/id, adresse für modbus routing, ein bytecount, der die Länge des folgenden Telegramms angibt (inklusive Unit identifier und Daten) auch wenn gesplittet, CRC-32 error check, protocol identifier mit modbus 0, transaction identifier vom client, der nur kopiert wird vom server um Transaktionen einander zuzuordnen. Alle Kommunikation erfolgt über TCP Port 502 [mod, 2006a, S. 4f.]

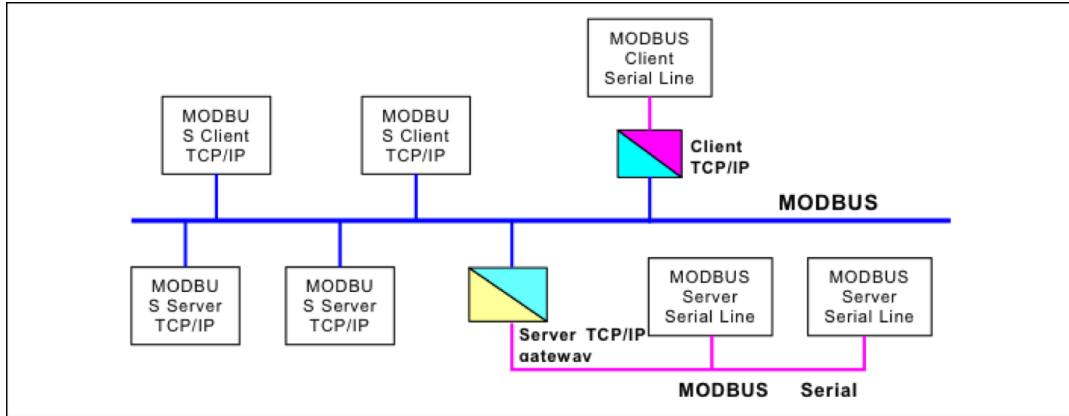


Abb. 2.15: Die Modbus TCP/IP Kommunikationsarchitektur aus [mod, 2006a, S. 4]

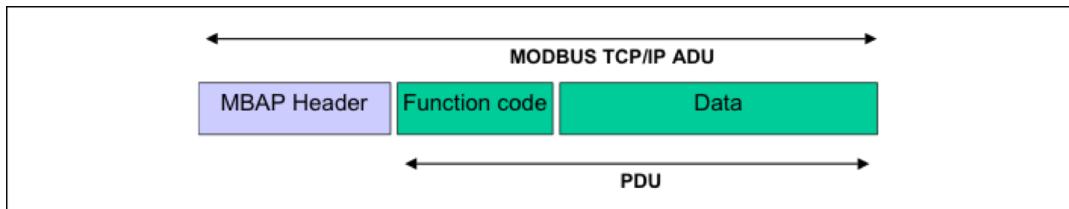


Abb. 2.16: Angepasster Rahmen für Telegramme nach dem Modbus TCP/IP Protokoll aus [mod, 2006a, S. 4]

Alle Modbus /TCP ADU werden via TCP zum Modbus registrierten Port 502 geschickt.

Modbus tcp Komponenten können sowohl client als auch server interface haben. [mod, 2006a, S. 7f.]

Der Modbus Client erlaubt den Informationsaustausch indem er eine ADU erstellt
Der Modbus Server wartet auf Anfragen über den TCP Port 502

Transmission control protocol TCP und IP ist Internet Protocol. TCP als Transportschicht verbindungsorientiert: Teilt Daten in Datenblöcke, sogenannte Pakete, zum Transport. IP regelt Netzwerkaufgaben, siehe OSI Modell Netzwerkschicht, und versendet die Daten über Telegramebservice und packt den MBAP Header an jedes Paket dran dran. Der Port erlaubt die parallele Nutzung von Ethernet Netzwerken, da er den Übertragungsprozess eindeutig kennzeichnet. Verbindungsorientiert heisst über ein Socket werden zwei prozesse miteinander verbunden und die empfangenen Telegrammen werden/müssen quittiert [Schnell u. Wiedemann, 2006, S. 16ff.]. Der interessierte Leser findet eine detaillierte Beschreibung des Ethernet TCP/IP Standards in Furrer [2003].

TCP übernimmt Netzwerkmanagement - TCP Management: Hauptaufgabe ist das connection management. Das managen von Verbindungen kann entweder durch ein Modul erfolgen oder durch die Nutzeranwendung selbst durch die Zugriffsüberwachung der sockets. Der Port 502 ist für Modbus Kommunikation reserviert, jedoch können auch andere Ports genutzt werden falls die Modbusgeräte eine Portkonfiguration unterstützen. Weiterhin wird der Datenfluss und der Einsatz der Netzwerkressourcen

überwacht. [mod, 2006a, S. 7ff.]

Generell, Verbindungsmanagement wichtig, da Modbus/TCP Kommunikation zwischen einem Server und einem Clienten eine Verbindung benötigt. Hinweise gibt der Guide, das die Verbindungen nicht dauernd geöffnet und geschlossen werden und auch mehrere viele Modbus Transaktionen während der Verbindung stattfinden. Außerdem sollte sich auf ein Minimum von Verbindungen beschränkt werden für den gleichen Server. [mod, 2006a, S. 9f.] Das Nutzer TCP Management umfasst folgende Aufgaben, die aktive und passive Herstellung von Verbindungen sowie das schließen dieser und das festlegen von maximalen Verbindungen [mod, 2006a, S. 11ff.]. Verbindungsherstellung über Ethernet IP, also der eindeutigen Adresse, des Geräts und Portnummer und die Socket Nummer. Der Socket ist ein Endpunkt innerhalb eines Rechners, über den die Kommunikation läuft und der einem Port eindeutig zugewiesen ist [mod, 2006a, S. 15f.] .

Physikalisch bedient sich der Ethernet Schnittstelle also einem normalen Netzwerkkabel.

Diese Kommunikationstechnologie und die verschiedenen Protokolle finden im Rahmen der Anlage in Kapitel dann Anwendung

2.3 Technische Grundlagen zur Modellbildung

In diesem Kapitel werden die technischen Grundlagen zur Bildung eines mathematischen Modells des Raumes erläutert. Themrodym systeme 1. HS thermo Wärmeübertragung

2.3.1 Thermodynamische Systeme

Im Raummodell müssen Energieströme, genauer betrachtet Wärmeströme, untersucht werden. Um dieses thermodynamischen Vorgänge mit Hilfe von Bilanzierungsgleichungen zu beschreiben, folgt zunächst ein kurze Einführung in die Thermodynamische Systembildung nach [Baehr u. Kabelac, 2012, S. 11ff.].

Thermodynamische Systeme werden durch den zu untersuchenden Raum abgegrenzt. Sie dienen dem Zweck der Bilanzierung von Massen- und Energieströmen und alles was diesen abgegrenzten Raum an den Systemgrenzen umgibt wird als Umgebung bezeichnet. Die begrenzenden Flächen können gedanklicher, physischer oder beider Natur zugleich sein, wichtig ist jedoch das die Systemgrenzen eindeutig festgelegt sind [Baehr u. Kabelac, 2012, S. 11].

Anhand der Eigenschaften von den Systemgrenzen lassen sich die thermodynamischen Systeme weiter differenzieren. Solche Systeme, deren Grenzen undurchlässig für Materie sind, werden als *geschlossene Systeme* bezeichnet und werden durch eine konstante Stoffmenge innerhalb des Systems gekennzeichnet. Die Grenzen eines geschlossenen

Systems sind meistens räumlich anhand eines fixen Volumens definiert, können aber auch beweglich sein, wie z.B. das Volumen einer vorgegebenen Stoffmenge unabhängig von dessen räumlicher Ausdehnung [Baehr u. Kabelac, 2012, S. 12].

Sind die Grenzen von thermodynamischen Systemen für Materie durchlässig, werden diese als *offene Systeme* bezeichnet. In der Regel werden diese von Stoffströmen durchflossen und durch räumlich festgelegte Grenzen beschrieben. Diese werden in der Literatur auch als *Kontrollraum* oder *Kontrollvolumen* bezeichnet [Baehr u. Kabelac, 2012, S. 12].

Ein *abgeschlossenes System* umfasst in der Regel mehrere Systeme oder ein einzelnes System und dessen Umgebung, so dass es zwischen den Grenzen des abgeschlossenen Systems und seiner Umgebung keine Wechselwirkungen gibt. Die Systemgrenzen werden also so gelegt, dass über sie hinweg keine beziehungsweise keine relevanten³ Flüsse von Materie und Energie [Baehr u. Kabelac, 2012, S. 13].

Nach der Abgrenzung folgt die *Beschreibung* von thermodynamischen Systemen und dessen *Eigenschaften*. Diese erfolgt durch *Variablen* und *physikalische Größen* die ein System kennzeichnen. Falls die Variablen feste Werte annehmen werden diese als *Zustandsgrößen* bezeichnet, da sie den *Zustand* eines Systems bestimmen [Baehr u. Kabelac, 2012, S. 13]. Im Rahmen der Modellbildung in Kapitel 4 ist es ausreichend die Vorgänge und Effekte auf systemischer Ebene zu betrachten, wodurch sich Modelle mit wenigen Variablen und physikalischen Größen beschreiben lassen.

Die Variablen lassen sich in *äußere Größen*, welche den mechanischen Zustand eines Systems beschreiben⁴, und *innere Größen* gliedern, welche den thermodynamischen Zustand, also die Eigenschaften der Materie innerhalb der Systemgrenzen, beschreiben [Baehr u. Kabelac, 2012, S.13 f.].

Innerhalb der Grenzen eines thermodynamischen Systems, und damit implizit auch für das Raummodell⁵ wird *Homogenität* angenommen. Dies bedeutet, dass die physikalischen Eigenschaften, wie zum Beispiel Temperatur und Druck, sowie die chemische Zusammensetzung an jeder Stelle innerhalb des Systems homogen ist, also die gleiche Ausprägung besitzt [Baehr u. Kabelac, 2012, S.15].

Da wir im Rahmen der Modellbildung Zustände betrachten müssen auch deren Änderungen genauer untersucht werden. Die *Zustandsänderungen* eines Systems werden durch Änderungen von Energie oder Materie über dessen Grenzen hinweg bedingt und finden meist im Austausch der Umgebung statt. Während einer solchen Änderung des Systemzustands wird ein Prozess durchlaufen, der eine zeitliche Abfolge von Ereignissen ist. Eine Änderung des Zustands eines Systems mit der gleichen Wirkung kann also durch verschiedene Prozesse bewirkt werden. Daher beschreibt ein *Prozess* nicht nur die Veränderung des Zustands sondern viel mehr die Beziehungen zwischen einem System und seiner Umgebung [Baehr u. Kabelac, 2012, S.21 f.].

³ Relevant im Sinne von kaum messbarer Fluss und nicht messbare Auswirkung auf das System.

⁴ Zum Beispiel die Koordinaten im Raum oder die relative Geschwindigkeit zum Beobachter)

⁵ Diese Annahme wird im Kapitel 5 noch überprüft und kritisch hinterfragt werden müssen

Ein Prozess kann aber auch innerhalb eines Systems stattfinden, dass heißt ohne äußere Einwirkungen. Dies geschieht zum durch das Aufheben innerer Hemmungen oder dem Wegfall Zwängen von Außen. Diese Prozesse laufen in abgeschlossenen Systemen meist von selbst ab und streben als Ziel einen ausgeglichenen, also homogenen, Endzustand an. *Ausgleichsprozesse* dienen somit dazu, einen *Gleichgewichtszustand* zu erreichen und repräsentieren Wechselwirkungen zwischen verschiedenen Teilen eines abgeschlossenen Systems. Dabei gleichen sich die Zustandsgrößen von einzelnen Subsystemen wie zum Beispiel der Druck oder die Temperatur einander an. Der Gleichgewichtszustand wird also durch die Zustände in den einzelnen Subsystemen bestimmt und ist dadurch charakterisiert, dass ein System diesen Zustand nicht von sich aus sondern nur durch äußere Eingriffe verlässt, zum Beispiel durch eine Veränderungen in der Umgebung. Die Erfahrung lehrt, dass ein System einem Gleichgewichtszustand entgegen strebt, wenn es sich selbst überlassen wird [Baehr u. Kabelac, 2012, S.22 f.]. Im Rahmen der Modellbildung in Kapitel 4 nehmen diese *Ausgleichsprozesse* eine zentrale Rolle ein, weil der Großteil an Änderungen von einzelnen Zustandsgrößen innerhalb des Raumes darauf zurückgeführt werden können.

2.3.2 Erster Hauptsatz der Thermodynamik

Der erste Hauptsatz der Thermodynamik wird im Folgenden als allgemeiner Energieerhaltungssatz formuliert und anschließend angewendet um eine Energiebilanzgleichung für geschlossene thermodynamische Systeme zu erhalten.

Der erste Hauptsatz der Thermodynamik erweitert den mechanischen Energieerhaltungssatz um die Energieformen Wärme und innere Energie. Er handelt ganz allgemein vom Prinzip der Energieerhaltung und dient er der Bilanzierung von Systemen [Baehr u. Kabelac, 2012, S. 43].

Die Gesamtenergie eines Systems E setzt sich zusammen aus der potenziellen E_{pot} und kinetischen Energie E_{kin} wie in der Mechanik und wird durch die innere Energie U ergänzt [Baehr u. Kabelac, 2012, S. 49]:

$$E := E_{pot} + E_{kin} + U \quad (\text{Gl. 1})$$

Im weiteren Verlauf der Arbeit werden nur ortsfeste Systeme betrachtet die sich dadurch auszeichnen, dass deren potenzielle Energie E_{pot} in etwa konstant ist. Weiterhin erfahren sie im betrachteten Intertialsystem Erde auch nur sehr kleine Änderungen in ihrer Geschwindigkeit, weshalb auch die kinetische Energie E_{kin} in etwa konstant ist. Da die Änderungen der mechanischen Energien in Bezug auf die Änderung der inneren Energie sehr klein sind werden im Folgenden nicht weiter betrachtet und die Gesamtenergie eines Systems E vereinfacht und lediglich aus der inneren Energie bestehend angenommen.

Die innere Energie hängt von der spezifischen Wärmekapazität c_p , der Masse eines

Systems m_{sys} und der Temperatur t beziehungsweise T ab [Baehr u. Kabelac, 2012, S. 54]:

$$U := m * c_p * T = m * c_p * t + u_0, \text{ mit } t = T - T_0 \quad (\text{Gl. 2})$$

Nach dem Prinzip der Energieerhaltung, kann die Energie eines Systems also weder erzeugt noch vernichtet werden sondern lediglich durch den Energietransport über dessen Grenzen hinweg verändert werden. Daraus ergeben sich folgende qualitative Formen des Energietransports [Baehr u. Kabelac, 2012, S. 48f.]:

- Die Arbeit W , die entweder von oder an einem System verrichtet wird, in differentieller Form die Leistung P .
- Die Wärme Q , die entweder in das System hinein- oder herausfließt, in differentieller Form der Wärmestrom \dot{Q} .
- Der Transport von Materie, also das Einbringen oder Wegnehmen von Masse m eines System, in differentieller Form die Materialflüsse \dot{m} .

Mit der zuvor getroffenen Annahme, dass die innere Energie des Systems entspricht, und unter Beachtung der Vorzeichenkonvention, welche besagt dass zugeführte Energie positiv und abgeführte Energie negativ zu bewerten ist, lassen sich die Änderungen der Energie eines Systems mit der folgenden Gleichung quantitativ beschreiben [Baehr u. Kabelac, 2012, S. 54]:

$$\begin{aligned} \Delta U &= Q + W + m_{in} * c_p * T_{in} - m_{out} * c_p * T_{out} \\ &\text{beziehungsweise in differentieller Form} \\ \frac{dU}{dt} &= \dot{U} = \dot{Q} + P + \sum \dot{m}_{in} * c_p * T_{in} - \sum \dot{m}_{out} * c_p * T_{out} \end{aligned} \quad (\text{Gl. 3})$$

2.3.3 Wärmeübertragung

Wärmeströme spielen bei der Modellbildung in Kapitel 4 eine wichtige Rolle, daher ist eine genauere Betrachtung dieser unumgänglich und im Folgenden werden die Grundlagen dazu erläutert.

Die Definition von Wärmeübertragung ist nach [Böckh u. Wetzel, 2014, S. 1] „[...] der Transfer der Energieform Wärme aufgrund einer Temperaturdifferenz. „Die Definition umfasst also einen zuvor beschriebenen Ausgleichsprozess und eine Änderung der inneren Energie eines thermodynamischen Systems. Die Wärmeübertragung kann nach Nußelt⁶ grundsätzlich durch zwei verschiedene Arten stattfinden [Böckh u. Wetzel, 2014, S. 3f.]:

- Durch Strahlung, bei der die Übertragung von Wärme ohne stofflichen Träger

⁶ Beschrieben in seinem Aufsatz „Das Grundgesetz des Wärmeüberganges“, 1915.

durch elektromagnetische Wellen zwischen Oberflächen erfolgt. Weil diese Art der Wärmeübertragung keine Relevanz für die weiteren Betrachtungen hat wird er interessierte Leser an dieser auf Böckh u. Wetzel [2014] verwiesen der diese Thematik detailliert ausführt.

- Durch Wärmeleitung, die sich wiederum in die Wärmeübertragung zwischen ruhenden Stoffen, und die Konvektion, die eine Wärmeübertragung zwischen einem ruhenden und einem strömenden Fluid beschreibt, aufteilen lässt.

Die übertragene Wärmemenge ist bei der reinen Wärmeleitung lediglich von den Stoffeigenschaften und der Temperaturdifferenz abhängig, bei der Konvektion hingegen, unabhängig davon ob erzwungen oder frei, hängt sie von der Strömung der Fluide ab. Die Konvektion ist ein Effekt zusätzlich zur reinen Wärmeleitung auftritt und ist im weiteren Verlauf der Arbeit nicht relevant und wird deshalb nicht detaillierter ausgeführt [Böckh u. Wetzel, 2014, S. 3f.]. Erfolgt der Wärmetransport stationär, dass heißt er ist von äußeren Anregungen bedingt und unabhängig von der Zeit, lässt er sich qualitativ einfach als konstanter Wärmestrom \dot{Q} beschreiben und gibt an wie viel Wärme pro Sekunde übertragen wird [Böckh u. Wetzel, 2014, S. 5ff.]. Der Wärmestrom ist wie zuvor bereits erwähnt von den Stoffeigenschaften abhängig, welche von der Wärmedurchgangszahl U – Wert⁷ und der Austauschoberfläche $A_{exchange}$, an der der Wärmeaustausch stattfindet. Typische U-Werte für verschiedene Materialien und Komponenten finden in der einschlägigen Literatur und beziehen sich bei der Übertragung durch eine Wand im europäischen Raum auf die Außenfläche [Böckh u. Wetzel, 2014, S. 28]. Damit lässt sich der Wärmestrom unter Berücksichtigung der Abhängigkeiten durch die kinetische Kopplungsgleichung quantifizieren [Böckh u. Wetzel, 2014, S. 6f.]:

$$\dot{Q} := u * A * (t_1 - t_2) \quad (\text{Gl. 4})$$

Unterschiedliche geometrische Ausprägungen, wie zum Beispiel ein Wärmeaustausch durch eine Wand oder ein Rohr hindurch, finden damit implizit bei der Austauschoberfläche Berücksichtigung.

2.4 Technische Grundlagen zur Solar- und Gebäudetechnik

2.4.1 Außenklima und Komponenten

Der Begriff Außenklima wird häufig im Zusammenhang mit dem Thema Umwelt und deren Einflüsse auf Gebäude gebraucht. Der allgemeine Begriff des Klimas wird von [Peter Häupl, 2013, S. 295] definiert als:

⁷ Der U-Wert wurde bis zu der Umstellung auf die europäischen Prüfnormen 2003 als k-Wert bezeichnet und ist unter dieser Bezeichnung noch häufig in der Literatur zu finden [Sack, 2004, S.1 f.]

„die Summe aller Umweltfaktoren, die unmittelbar oder mittelbar Einfluss nehmen auf die Gesundheit und das Befinden von Menschen und Tieren, auf die Entwicklung von Pflanzen sowie auf den Zustand von Lagergütern, Produktionsverfahren, Maschinen, Apparaten und Bauwerken.“

Daraus lässt sich ableiten, dass das Außenklima ein Aspekt des Klimas ist und den meteorologischen Umweltzustand außerhalb von Gebäuden, an einem bestimmten, lokalen Ort meint. Abhängig vom Außenklima stellt sich innerhalb von Gebäuden ein Innenklima ein, welches direkten Einfluss auf das Wohlbefinden von Menschen hat und wodurch der mittelbare Einfluss des Außenklimas gegeben ist. Um den Zustand zu beschreiben werden viele Zustandsgrößen herangezogen. Um einen Überblick zu bekommen, lassen sich diese in verschiedene Bereiche gliedern [Peter Häupl, 2013, S. 295f.] : Schall Licht Temperatur und Feuchte

Im Hinblick auf die Modellbildung in Kapitel 4 sind lediglich die Größen zur Beschreibung der Temperatur und des Lichts von Interesse, eine detaillierte Ausführung in die Bereiche Schall und Feuchte und deren Zustandsgrößen ist in Peter Häupl [2013] gegeben.

Je nach Größe des Gebietes wird von einem Regional- bzwziehungsweise Makroklima, das große Gebiete umfasst, oder von einem Lokal- beziehungsweise Mikroklima gesprochen, dass kleine Gebiete wie eine Straße oder einen Park umfasst und von deren Besonderheiten abhängig ist. So kann z.B. die Außentemperatur je nach Verschattungsgrad einer Straße lokal erhöht oder erniedrigt sein. Das Klima folgt in verschiedenen Regionen der Erde bestimmten Charakteristiken, welche sich in Klimazonen zusammenfassen lassen. Die Erde besteht aus vierzehn verschiedenen Klimazonen und in Europa wird von einem Übergangsklima gesprochen [Peter Häupl, 2013, S. 296f.].

Eine Übersicht der Außenklimakomponenten, die einen Einfluss auf die Gebäude-technik und damit auch auf die Raumtemperatur haben ist in Abb. 2.17 gegeben. Für die Modellbildung ist weiterhin eine Quantifizierung der relevanten Größen des Außenklimas in den Bereichen Licht und Temperatur erforderlich. Wie bereits im vorherigen Abschnitt erwähnt, werden Wärmeströme durch Temperaturdifferenzen bedingt, weshalb die Außenlufttemperatur *HierSymbol* einen großen Einfluss auf die Raumtemperatur hat und durch Messung quantifiziert werden muss. Des Weiteren werden die lichttechnischen Größen der kurzweligen direkten und diffusen Strahlung durch die beiden Strahlungsintensitäten G_{dif} und G_{dir} beschrieben, da sie Energie durch einen Wärmestrom in ein System einbringen und damit auch einen Einfluss ausüben.

2.4.1.1 Die Außenlufttemperatur

Eine Übersicht über die Außenlufttemperatur

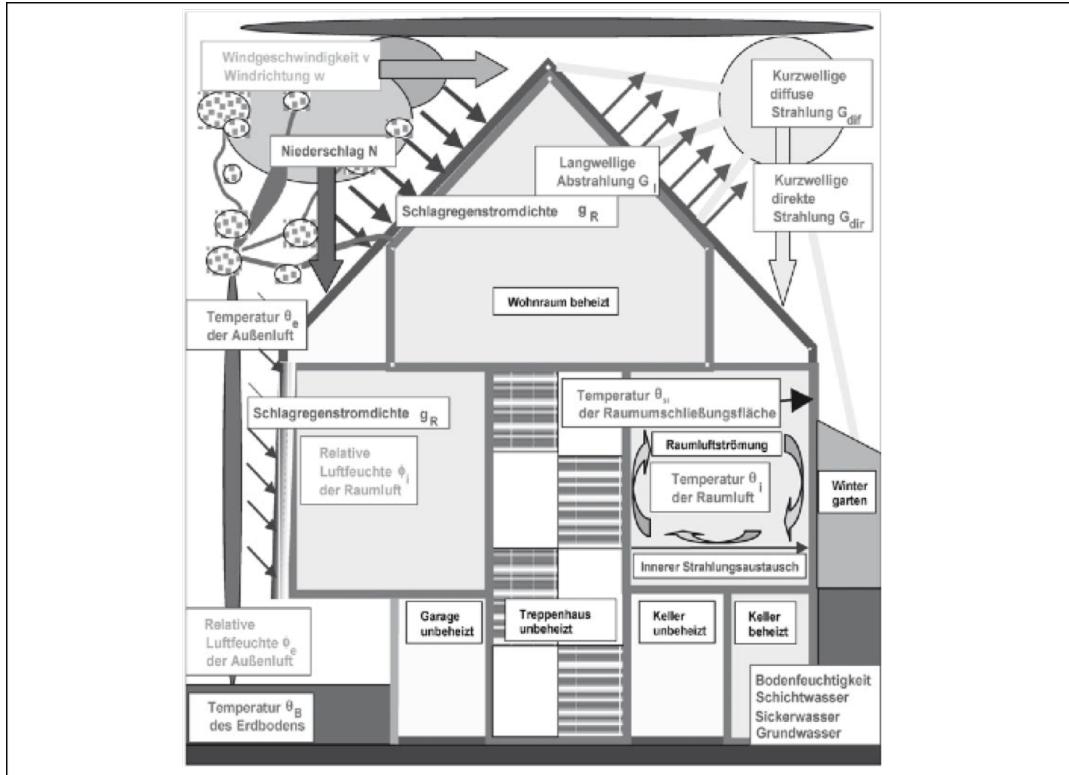


Abb. 2.17: Komponenten des Außenklimas aus [Peter Häupl, 2013, S. 298]

2.4.1.2 Sonnenstrahlung

Nicolai [2013] [Quaschning, 2011, S. 63ff.] [Kaltschmitt, 2013, S. 61ff.] [Peter Häupl, 2013, S. 315ff.] Bild

Algorithmus nach Reda [2008]

Nutzung Berechnung/Implementierung von pys

2.4.2 Gebäudephysik Glas und Wärmedurchgangskoeffizienten

Auf gehts

Glas nach [Peter Häupl, 2013, S. 61ff.] Durchlassgrad nach [Peter Häupl, 2013, S. 604ff.] Transmissionsgrad

„*Design is the appropriate combination of materials in order to solve a problem.*“
— CHARLES EAMES

3 Anlagendesign

Ziel dieses Kapitel ist es, eine Anlage zur Raumtemperaturregelung für den Betrieb mit Modellprädiktiver Regelung zu konzipieren, zu konkretisieren und im letzten Schritt umzusetzen. Dazu werden zunächst die Anforderungen an die Anlage analysiert und weiterhin die Vorgaben und Rahmenbedingungen zur Anlage von Seiten der Hochschule Karlsruhe spezifiziert und ausgeführt. Daraus wird eine Idee abgeleitet, die anschließend zu einem Konzept weiterentwickelt und in ein konkretes Anlagendesign umgesetzt wird. Dabei werden die einzelnen Anlagenteile und deren Funktionsweisen näher beschrieben und auf die realen Einsatzbedingungen ausgelegt. Abschließend wird die Installation und dabei aufgetretenen Besonderheiten der Anlage beschrieben.

3.1 Analyse der Anforderungen

3.1.1 Einsatzziele und Rahmenbedingungen

Um die Anforderungen an eine Anlage zu bestimmen, die sich für die Anwendung mit Modellprädiktiver Regelung eignet, wird zunächst der Zweck und die Einsatzziele der Anlage untersucht und definiert. In Kapitel 1.1 wurde bereits darauf hingewiesen, dass es die Vorgabe von Seiten der Hochschule ist, die Einsatzziele in Einklang mit der großen Anlge zu bringen und komplementär zu wählen. Daher wurden im Dialog mit den Projektverantwortlichen⁸ für die Forschung im Bereich solarer Anwendungen an der Hochschule Karlsruhe gemeinsam konkrete Einsatzziele der Anlage erarbeitet. Als Ergebnis wurden die folgenden, konkreten Ziele vereinbart:

- Die Einarbeitung in die Thematiken Modellbildung, Kommunikation von technischen Systemen und Modellprädiktive Regelung soll durch eine praktisches Anwendung unterstützt werden.
- Es soll Know-how im Bereich der Kommunikation von technischen Systemen aufgebaut werden, insbesondere im Umgang mit der Software, der Hardware und zahlreichen Schnittstellen.
- Die Anlage soll eine hohe Funktionalität, also möglichst wartungsarm, und eine hohe Robustheit gegenüber Fehlern und Beschädigungen besitzen, da bei der Einarbeitung eine erhöhte Wahrscheinlichkeit der Fehlbedienung besteht und Schäden dadurch vermieden werden sollen.
- Es soll ein Vergleich verschiedener Regelungsmethodiken beim Einsatz von

⁸ In Person von Herrn ADRIAN BÜRGER und MARKUS BOHLAYER

Modellprädiktiver Regelung ermöglicht werden.

- Außerdem soll ein Vergleich von Ergebnissen bei der Variation von Steuerungsparametern sowie beim Einsatz verschiedener Steuerungs- und Regelungsalgorithmen ermöglicht werden.
- Des Weiteren soll die Anlage möglichst flexibel ansteuerbar und erweiterbar sein, damit der Grad der Komplexität anpassbar ist und die Anlage um weitere Funktionen oder Features ergänzt werden kann.
- Der temperaturerhöhende Effekt der Sonneneinstrahlung auf die Raumtemperatur soll untersucht werden können.
- Im Rahmen der Anwendungsforschung soll der Raum zur Temperaturregelung möglichst nahe an der Realität sein, also Störgrößen beinhalten und nicht ungenutzt beziehungsweise leerstehend sein.

Zusammenfassend wurde festgehalten, dass die Anlage als Forschungsumgebung für Entwicklungs-, Test- und Anwendungszwecke von verschiedenen Steuerungen und Regelungen dienen soll.

Weiterhin wurden von Seiten der Hochschule Karlsruhe⁹ Rahmenbedingungen definiert, die im Folgenden zusammengefasst sind:

- Der Raum K004a im K Gebäude der Hochschule Karlsruhe wird zur Installation der Anlage und Einrichtung der Forschungsumgebung zur Verfügung gestellt.
- Die Installation der Anlage muss mit minimalem baulichem und finanziellem Aufwand zu realisieren sein.
- Für die Kommunikation innerhalb der Anlage soll die Modbus Kommunikationstechnologie mit mindestens zwei verschiedenen Übertragungsprotokollen genutzt werden.
- Die Modellprädiktive Regelung soll mit Hilfe der Plattform JModelica.org erfolgen.

3.1.2 Definition der Anforderungen

Diese Einsatzziele und Rahmenbedingungen definieren implizit Anforderungen an eine Anlage, welche im Nachfolgenden explizit ausgeführt werden und aus Gründen der Übersichtlichkeit die wichtigsten in Tabelle Tab. 3.1 zusammengefasst sind.

⁹ In Person von Frau Professor ANGELIKA ALTMANN-DIESES, Herrn Professor MARCO BRAUN und Herrn ADRIAN BÜRGER

Einsatzziele & Rahmenbedingungen	Anforderungen
Raum K004a als Umgebung	<ul style="list-style-type: none"> – Die Anpassung der Anlage an K004a.
Minimaler baulicher Aufwand	<ul style="list-style-type: none"> – Die Nutzung bestehender Heizkörper anstatt einer Klimatisierung des Raumes.
Minimaler finanzieller Aufwand	<ul style="list-style-type: none"> – Die Beschränkung auf eine minimale Funktionalität und Anzahl der einzelnen Komponenten.
Modellprädiktive Regelung mit JModelica.org und CasADi	<ul style="list-style-type: none"> – Die Modellbildung erfolgt in Modelica. – Die Ansteuerung und Kommunikation innerhalb der Anlage soll in Python stattfinden.
Einsatz der Modbus Kommunikationstechnologie	<ul style="list-style-type: none"> – Die Kommunikation der Anlage erfolgt gemäß den Modbus RTU und TCP Protokollspezifikationen. – über Modbus TCP soll die Ansteuerung der Anlage innerhalb des gesamten lokalen Netzwerks möglich sein.
Flexible Ansteuerung der Anlage	
Einarbeitung in die Thematiken:	
<ul style="list-style-type: none"> – Modellbildung, – Kommunikation technischer Systeme, – und Modellprädiktive Regelung. 	<ul style="list-style-type: none"> – Komplexität ist notwendig, darf jedoch nicht zu hoch sein. – Es sind möglichst wenige thematische überschneidungen erwünscht, daher wird eine klare Struktur mit möglichst scharfer Trennung benötigt.
Know-how für Kommunikation technischer Systeme	
Vergleich von Ergebnissen durch:	
<ul style="list-style-type: none"> – die Variation von Steuerungsparametern, – den Einsatz verschiedener Regelungsmethodiken, – und den Einsatz verschiedener Algorithmen. 	<ul style="list-style-type: none"> – Die Reaktion des Systems muss schnell sowie günstig und einfach zu erfassen sein. – Der Einsatz von robusten und einfachen Bauteile. – Nutzung eines wartungsarmen Systems. – Eine einfache, modulare Erweiterbarkeit des Systems für weitere Schritte muss gegeben sein.
Hohe Funktionalität und Robustheit	
Erweiterbarkeit der Anlage	

Tab. 3.1: Umsetzung der Ziele in Anforderungen der Anlage

Die grundlegendste Anforderung an die Planung, ist die Anpassung an die Lage und die Gegebenheiten des Raumes K004a, welche in Abb. 3.1 skizziert sind. Der Raum befindet auf dem Campus der Hochschule Karlsruhe, an der südwestlichen Ecke des K Gebäudes im Erdgeschoss. Die südliche und westliche Wand teilt sich der Raum mit der Außenumgebung und wird im Folgenden als Außenwand bezeichnet. Die östliche und nördliche Wand sowie die Decke und der Boden des Raumes grenzen an andere Räume im K Gebäude. Außerdem ist in der südlichen Außenwand eine Fensterfront mit

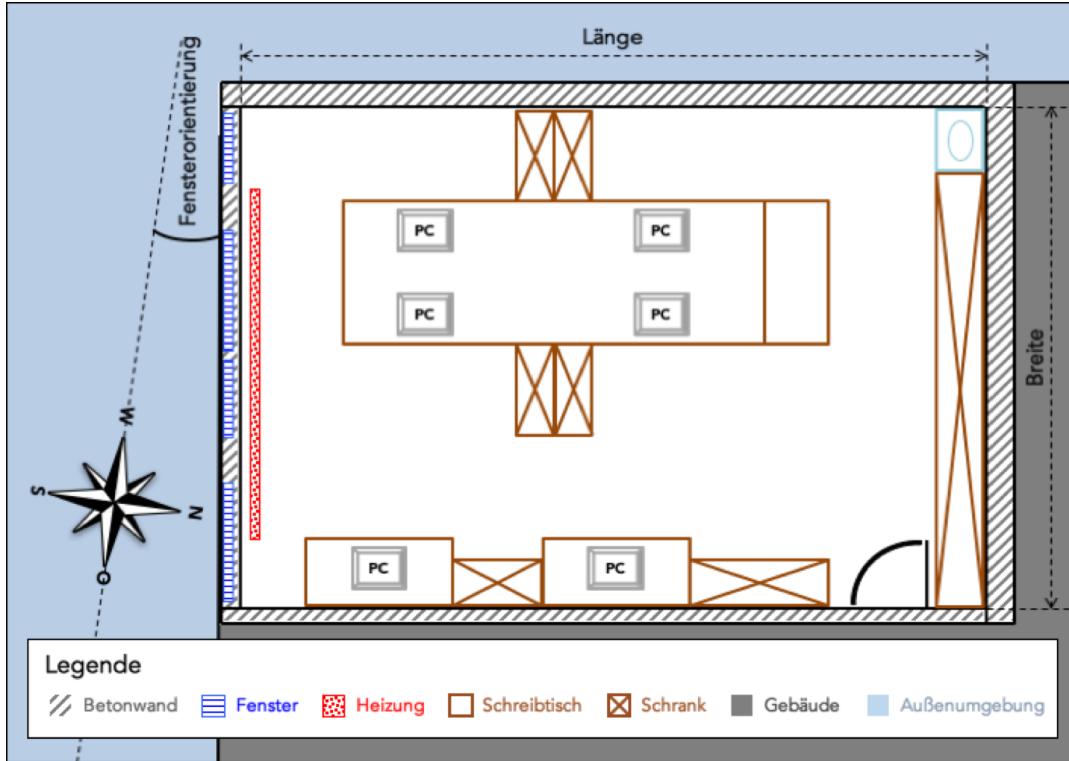


Abb. 3.1: Raumskizze K004A vom K Gebäude der Hochschule Karlsruhe – Technik und Wirtschaft

Jalousien zur Verschattung eingebaut und direkt darunter ein Heizkörper installiert.

Durch die Raumwahl werden bereits zwei wichtige Anforderungen erfüllt, denn durch die Fensterfront kann der Einfluss der Sonneneinstrahlung auf die Raumtemperatur untersucht werden und der bestehende Heizkörper kann in die Anlage integriert werden, um einen minimalen baulichen Aufwand sicherzustellen. Damit wird zunächst eine Klimatisierung zur Temperaturregelung des Raumes ausgeschlossen, weil diese mit einem erheblichen finanziellen und baulichen Aufwand verbunden ist. Um der Forderung nach einer Erweiterbarkeit nachzukommen, wird jedoch die Möglichkeit der Nachrüstung einer Klimatisierung und der Ansteuerung der Jalousien bei der Planung explizit berücksichtigt.

Durch die Nutzung des Raumes als Büro für wissenschaftliche Mitarbeiter befinden sich in K004a sechs Computerarbeitsplätze sowie das typische Büro-Mobiliar, wie in Abb. 3.1 abgebildet. Damit wird auch die Anforderung einer anwendungsnahen Umgebung erfüllt und durch die Menschen und Rechner sind verschiedene Störgrößen zu berücksichtigen.

Eine weitere, sehr einschränkende Vorgabe ist es, dass die Modellprädiktive Regelung unter Zuhilfenahme der Plattform JMODELICA.ORG erfolgen soll. JMODELICA.ORG ist eine kostenlose Open-Source Plattform zur Analyse, Simulation und Optimierung von komplexen, dynamischen Systemen, die auf der Modellierungssprache Modelica basiert. Aufbauend auf den mathematischen Modellen physikalischer Systeme in Modelica, lassen sich, dank der Unterstützung der Spracherweiterung Optimica, Optimierungsprobleme durch einfache Konstrukte das Optimierungsintervall, die Kostenfunktion

und die Nebenbedingungen einfach formulieren. JModelica.org wird über eine Python Nutzerschnittstelle genutzt und besitzt eine eigene Klasse für die Modellprädiktive Regelung, auch wenn diese bisher noch experimenteller Art ist [AB, 2015, S. 1f.]. Der Compiler kann die Modelle und Optimierungsprobleme in verschiedene Formate übersetzen. Zum einen in direkt ausführbaren C Code, der die Modellgleichungen und Optimierungsparameter enthält, und ein XML Code, der die Meta-Daten des Modells enthält. Zum anderen kann das Modell aber auch in ein *OptimizationProblem* Objekt transferiert werden, welches eine symbolische Repräsentation des Optimierungsproblems ist [AB, 2015, S. 12ff.].

Die *OptimizationProblem* Objekte können anschließend direkt mit den Optimierungs werkzeugen von CASADI bearbeitet und damit zur Lösung des Optimierungsproblems eingesetzt werden. CASADI ist ein Open-Source Softwaretool, das einzelne Bausteine für die numerische Optimierung im Allgemeinen und für die Optimalsteuerung im Speziellen zur Verfügung stellt. Es eignet sich besonders für die gradientenbasierte, numerische Optimierung von nichtlinearen Problemen, aufgrund seiner effizienten Ableitungserzeugung durch die Algorithmische Differentiation und der Möglichkeit zur Integration von gewöhnlichen Differentialgleichungen und differential-algebraischer Gleichungen. Die Interaktion mit dem Nutzer soll aus Stabilitätsgründen über die Python Schnittstelle erfolgen [Joel Andersson, 2015, S. 5f.]. Daraus resultieren weitere Anforderungen an das Modell, welche vor Beginn der Modellbildung im Kapitel 4 erörtert werden.

Weil die vorgegebene Softwareplattform zur Modellprädiktiven Regelung und deren Komponenten eine allesamt eine Schnittstelle für Python besitzen, soll die Ansteuerung und Kommunikation der gesamten Anlage in Python erfolgen. Python eignet sich hervorragend für diese Aufgabe, da es frei erhältlich und modular aufgebaut ist. Durch die Open-Source Lizenzierung ist es frei nutzbar und bietet sowohl durch die Standardbibliothek als auch durch eine Vielzahl an nutzerentwickelten Bibliotheken, die auch als Pakete bezeichnet werden, unzählige Anwendungsmöglichkeiten, zum Beispiel das pysolar Paket, das im Rahmen der Modellbildung eine Umrechnung der gemessenen Solarstrahlung in die wirkende am Fenster ermöglicht [van Rossum u. the Python development team, 2016, S. 2f.].

Der Vorgabe der Modbus Kommunikationsprotokolle für die Kommunikation ist von ebenfalls von großer Relevanz, da die komplementäre Forschungsanlage zur solaren Klimatisierung dasselbe Kommunikationsprotokoll unterstützt und durch die gewonnenen Erkenntnisse eine beschleunigte Inbetriebnahme erfolgen kann. Um das Know-how breit zu fächern, sollen die beiden Modbus RTU und Modbus TCP Protokolle Anwendung finden. Außerdem soll das Kommunikationsnetzwerk aus mehreren Subnetzwerken aufgebaut sein, die durch verschiedene elektrische und mechanische Schnittstellen implementiert werden. Für die Kommunikation über Modbus werden in den Python Paketen pymodbus, minimalmodbus und modbus-tk bereits Bausteine zur Verfügung gestellt.

Die Anlage sollte möglichst wenig Komplexität aufweisen, damit die Einarbeitung in die einzelnen Themengebiete Modellbildung, Kommunikation von technischen Systemen und Modellprädiktive Regelung vereinfacht wird. Dies soll durch eine klare Abgrenzung der verschiedenen Anlagenteile und deren Funktionen sowie durch die Strukturierung und den Aufbau der Anlage erreicht werden. Im Gegensatz dazu steht die Forderung nach dem Aufbau von Know-how auf dem Gebiet der Kommunikation technischer Systeme. Deshalb muss ein Kompromiss zwischen Verständlichkeit und Komplexität gefunden werden, um beiden Anforderungen gerecht zu werden. Dieser besteht darin, dass die Kommunikation nur über Modbus stattfindet, jedoch unter Nutzung von verschiedenen Hard- und Software-Schnittstellen, um einen gewissen Grad an Komplexität zu erreichen. Zudem können dadurch bereits Erfahrungen gesammelt werden, die für die Inbetriebnahme der solaren Klimatisierungsanlage sehr nützlich sind.

Um das Vergleichen von Ergebnissen zu ermöglichen, soll das System auf eine änderung der Steuergrößen schnell¹⁰ reagieren und die Reaktionen sollen möglichst einfach messbar sein. Konkret lässt sich daraus ableiten, dass die Aktorik auf Steuersignale möglichst ohne zeitliche Verzögerung agieren sowie die gesamte Anlage einen möglichst direkten Einfluss auf die Raumtemperatur haben soll. Die Messung der Raumtemperatur kann durch einfache Temperatursensoren erfolgen die ohne großen technischen und monetären Aufwand auskommen und damit die Anforderung erfüllen.

Es soll außerdem eine hohe Funktionalität erreicht werden, um die wissenschaftliche Arbeit – durch den Ausschluss von Fehlerquellen außerhalb der Forschung – zu erleichtern. Zusätzlich ist eine Robustheit gegenüber Bedienungsfehlern und Beschädigungen erforderlich, da durch den Test- und Entwicklungsauftrag der Anlage hierzu eine erhöhte Gefahr besteht. Dies lässt sich durch wartungsarme, robuste und einfache Bauteile sowie den Aufbau der Anlage berücksichtigen, was wiederum im Einklang mit der Forderung nach einer minimalen finanziellen Belastung steht.

Wie an passender Stelle bereits erwähnt, stellt Python Schnittstellen und Funktionalitäten für die Aufgaben der Anlage zur Verfügung. Darüberhinaus bietet sich Python, durch die weitgehende Unabhängigkeit vom Betriebssystem, als zentrales Steuerungstool an, um den Controller in einem nächsten Schritt durch einen vergleichsweise günstigen Einplatinenrechner zu realisieren, wie zum Beispiel ein Raspberry Pi. Daher basiert die folgende Planung und Auslegung der Anlage auf einer zentralen Steuerung in Python.

¹⁰ Im Kontrast zu den langsamen Reaktionen der solaren Klimatisierungsanlage, welche im oberen Minutenbereich liegen, bedeutet schnell in diesem Kontext im unteren Minutenbereich.

3.2 Ziel und die Idee der Anlage

3.2.1 Aufgabe der Anlage

Das Ziel ist es, die Temperatur innerhalb eines Raumes mit Hilfe eines technischen Systems zu regeln, welches den zuvor ausgeführten Anforderungen genügt.

Dazu gilt es zunächst, die Raumtemperatur über einfache Raumtemperaturfühler zu erfassen. Ist die Raumtemperatur bekannt so soll diese mit Hilfe eines Heizkörpers beeinflusst werden, damit sie einer vorgegebener Temperaturkurve folgen kann.

Die Heizleistung wiederum hängt von weiteren Eigenschaften des Heizkörpers ab und wird über das Einlassventil gesteuert. Sie wird von einem Wärmemengenzähler gemessen und indirekt über den Massenstrom und die Temperaturen des Heizwassers am Ein- und Auslass bestimmt. Dazu werden ein Durchflusssensor sowie zwei Temperatursensoren an den jeweiligen Enden des Heizkörpers eingesetzt. Um die Heizleistung zu steuern, muss also der Massenstrom gemessen und gesteuert werden sowie beide Temperaturen am Heizkörper bekannt sein. über das Ventil kann der Massenstrom mit Hilfe eines Stellantriebs eingestellt werden. Zusammen mit der Temperatur am Einlass und den Eigenschaften des Heizkörpers, lässt sich die Temperatur am Auslass und damit die aktuelle Heizleistung bestimmen. Dadurch wird eine gezielte Beeinflussung der Raumtemperatur ermöglicht.

Nachdem die Raumtemperatur bekannt ist und eine Möglichkeit für deren Manipulation besteht, wird weiterhin eine intelligente Steuerung benötigt. Dieser soll den Heizkörper möglichst ressourcenschonend einsetzen, um das Ziel der Temperaturregelung zu erreichen.

3.2.2 Idee der Anlage

Diese umfassende Aufgabe wird von einer Anlage übernommen, die sich dazu grob in drei Teile gliedern lässt. Den größten Umfang besitzt die Sensorik, die zur quantitativen Erfassung der Zustandsgrößen dient. Die aktive Beeinflussung dieser Größen erfolgt durch die Aktorik. Die Regelungsaufgabe wird von einem Controller durch seine interne Logik und der Koordination des Zusammenspiels zwischen Sensorik und Aktorik gelöst.

Um den Anforderungen gerecht zu werden, muss der logische Controller als zentrale Komponente der Anlage Python unterstützen und die gesamten Steuerungs- und Kommunikationsaufgaben übernehmen. Weiterhin müssen im Rahmen der Modellprädiktiven Regelung in regelmäßigen Zeitabständen wiederholt Optimalsteuerungsprobleme gelöst werden, weshalb eine ausreichende große Rechenkapazität benötigt wird. Um diesen Aufgaben übernehmen zu können und um keinen zusätzlichen finanziellen Aufwand zu generieren, wird zunächst ein freier Rechner der Hochschule Karlsruhe als Controller genutzt.

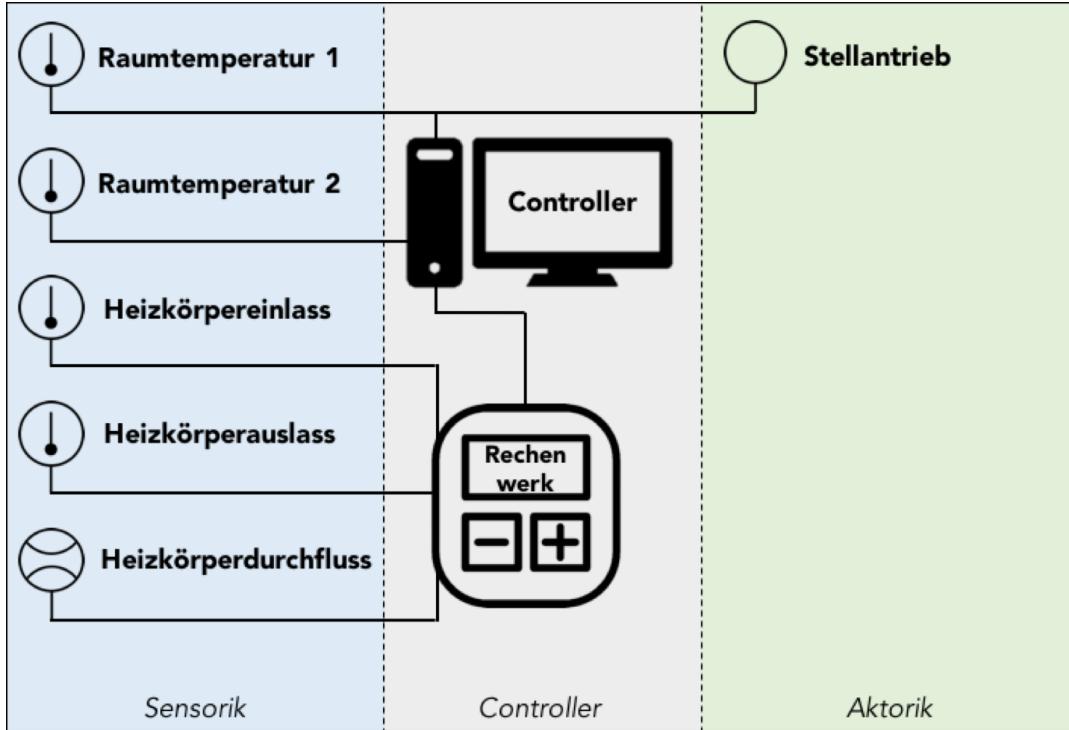


Abb. 3.2: Prinzipskizze eines technischen Systems zur Raumtemperaturregelung des Raumes K004b

Wie bereits beschrieben hat die Sensorik den größten Umfang, da sie neben der Raumtemperatur auch den Massenstrom und die Temperaturen am Heizkörper erfasst. Die Temperatur innerhalb des Raumes wird durch mehrere Sensoren erfasst, um die unterstellte Homogenität zu überprüfen und für die Modellberechnungen herzustellen. Für die Erfassung der Heizleistung wird ein Wärmemengenzähler eingesetzt, der wiederum aus einem Rechenwerk, zwei weiteren Temperatursensoren und einem Durchflusssensor aufgebaut ist. Für die Modellprädiktive Regelung werden die einzelnen Werte aller Sensoren benötigt, weshalb das Rechenwerk einen Zugriff darauf ermöglichen muss.

Die Aktorik umfasst lediglich die Ansteuerung des Heizungsventils. Wie zuvor bereits erwähnt wird dazu ein Stellantrieb genutzt, um das Ventil stufenlos zu öffnen und zu schließen und damit den Massenstrom bis hin zu einem Maximalwert zu steuern.

Eine Prinzipskizze dieser Idee ist in Abb. 3.2 graphisch dargestellt.

3.3 Konzept und Planung

Das folgende Konzept erfüllt die in Abschnitt 3.1 definierten Anforderungen und konkretisiert die zuvor geschilderte Idee. Zunächst das Netzwerk, dann xxx bis xxx

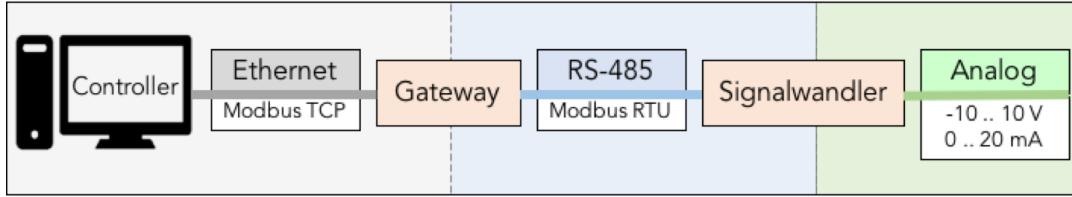


Abb. 3.3: Aufbau des Netzwerks

3.3.1 Netzwerkarchitektur

Der Rechner stellt das Zentrum der Anlage dar, deshalb werden darauf aufbauend die Kommunikationsleitungen und der Aufbau des Netzwerk geplant. Die Kommunikation erfolgt gemäß den Spezifikationen der beiden Modbus Protokolle RTU und TCP, deren Besonderheiten im Abschnitt 2.2.3 beschrieben wurden. Da der Rechner standardmäßig mit einem Ethernet-Netzwerkanschluss ausgestattet ist, wird der Rechner über ein Netzwerkkabel mit dem ersten Subnetzwerk verbunden. Das erste Netzwerk ist also ein lokales Netzwerk, das gemäß dem Ethernet/IP Protokoll ausgeführt ist und damit die Kommunikation über das Modbus TCP Protokoll ermöglicht.

Das zweite Subnetz ist ein serielles RS 485 Netzwerk und erlaubt den Einsatz des Modbus RTU Protokolls. Die beiden Netzwerke werden durch ein Gateway miteinander verbunden, welches die Übersetzung der Kommunikation in beiderlei Richtungen übernimmt. Die Übersetzung beinhaltet im Detail das Umwandeln der elektrischen Signale und der verschiedenen Modbus Telegramme ineinander.

Das Gateway bietet , um ein serielles Netzwerk in ein Lokales Netzwerk zu integrieren. Dadurch ist es möglich Modbus RTU Geräte von einem Modbus TCP/IP Clienten zu steuern. Das eingesetzte EX9132C-2-MTCP Gateway von EXPERTDAQ bietet hierzu eine einfache und kostengünstige Möglichkeit, da es über zwei serielle Ports verfügt. , ein EIA-232 und ein EIA 232/422/485 Port. Die genauen Spezifikationen können dem Datenblatt im Anhang REFXXXXXXXX entnommen werden. Damit kann

Im dritten Subnetz wird über analoge Spannungssignale ud Stromsignale kommuniziert, welche von einem Signalwandler erzeugt werden. Der Signalwandler wiederum ist kein Gateway, da die Spannungssignale nicht implizit übersetzt, sondern explizit durch Modbus RTU Befehlstelegramme festgelegt werden. Eine graphische Zusammenfassung der Netzwerkarchitektur ist in Abb. 3.3 abgebildet.

3.3.2 Erfassung der Raumtemperatur

Die Raumtemperatur kann einfach und günstig mit Raumtemperaturfühlern gemessen werden. Zudem soll der gemessene Wert über eine der beiden Modbus Schnittstellen dem Controller zur Verfügung gestellt werden. Innerhalb der Anlage kommen zwei THERMASGARD RTM1-MODBUS Raumtemperaturfühler ohne Display von S+S REGELTECHNIK zum Einsatz, weil sich diese durch weitere Eigenschaften, wie zum

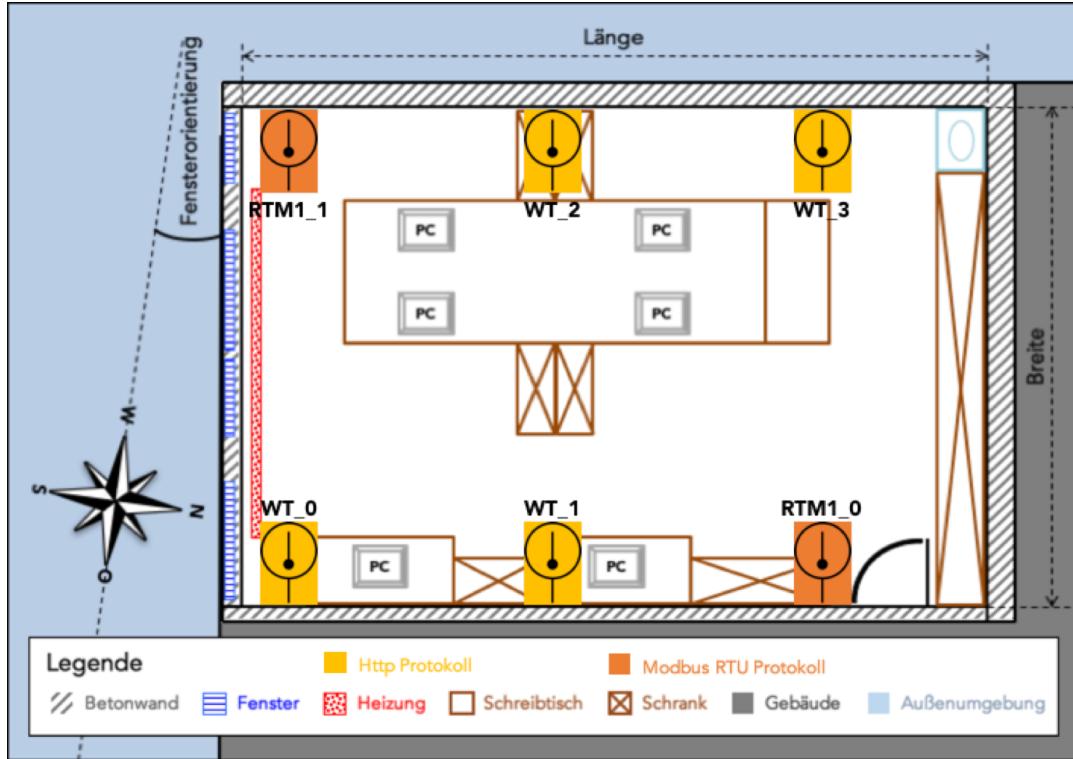


Abb. 3.4: Verteilung der Raumtemperaturfühler

Beispiel der Kalibrierfähigkeit, auszeichnen. Die detaillierten Funktionen und Daten können dem Datenblatt im Anhang B.1 entnommen werden.

Ursprünglich sollten die beiden Temperatursensoren auf einer mittig gedachten Achse im Raum in nord-südlicher Richtung platziert werden, jeweils am Ende des mittigen Schreibtischs. Jedoch wurden während der Installation der Anlage noch vier weitere Temperatursensoren zusammen mit einem Messumformer von Seiten der Hochschule Karlsruhe zur Verfügung gestellt. Bei den Sensoren handelt es sich um vier PT1000 Temperatursensoren die über den WEBTHERMOGRAPH 8X von WU T die Messwerte zur Verfügung stellen. Der Webthermograph besitzt keine Modbus Schnittstelle, jedoch kann er über ein Netzwerkkabel in das Ethernet Netzwerk integriert werden und unter Verwendung des HTTP Protokolls vom Controller ausgelesen werden. Um die Messwerte mit dem Controller auszulesen, bietet sich das Python Paket *httplib* an, welches das HTTP Protokoll implementiert und einfache Funktionen zur Verfügung stellt. Die Anordnung der Temperatursensoren wurde daher angepasst, sodass jeweils drei Sensoren an gegenüberliegenden Wänden angeordnet sind. Die Sensoren wurden entlang der westlichen Außenwand und der östlichen Innenwand gleichmäßig verteilt und auf einer Höhe von 2 m installiert. Die Identifizierung der Sensoren erfolgt über ID's, die zusammen mit der Anordnung in Abb. 3.4 visualisiert sind.

3.3.3 Steuerung des Heizkörpers

Um die Heizung mit den benötigten Sensoren auszustatten, bietet sich wie bereits erwähnt ein Wärmemengenzähler an. Die Temperaturmessung am Ein- und Auslass der Heizung gestaltet sich jedoch etwas aufwendiger als zuvor. Dazu werden Tauchhülsen im Vor- und Rücklauf des Heizkörpers benötigt, welche vom Heizwasser direkt umflossen werden. Die Tauchhülsen bedingen geringe bauliche Änderungen und können beispielsweise in einen Kugelhahn eingeschraubt werden. Sie sind jeweils mit einem Temperaturfühler bestückt und ermöglichen dadurch die Messung der Heizwassertemperatur. Der Einbau des Durchflusssensors erfordert ebenfalls bauliche Maßnahmen. Dazu wird im Vor- oder Rücklauf, je nach Spezifikation Herstellers, ein Stück Rohrleitung entfernt und durch den Durchflusssensor ersetzt. Dieser wird beim Einbau entweder fest integriert oder kann durch eine Anschlussverschraubung beziehungsweise einen Flanschanschluss einfach demontierbar sein.

In der Anlage kommt der Wärmemengenzähler MULTICAL 602 von KAMSTRUP zum Einsatz, da er neben einem geforderten Modbus Kommunikationsmodul weitere besondere Eigenschaften mit sich bringt. Die berechneten und gemessenen Werte können damit einfach einzeln ausgelesen werden. PT500 Sensoren Zum einen ist er kostengünstig und umfasst alle aufeinander abgestimmte Komponenten/Sensoren, zum anderen ist er wartungsfrei, da der Durchfluss über Ultraschall gemessen wird. Der Durchflusssensor ULTRAFLOW 54 ist bereits mit einer Tauchhülse ausgestattet, sodass lediglich eine weitere Tauchhülse montiert werden muss. Der ULTRAFLOW 54 ermittelt den Durchfluss, indem er die Laufzeitdifferenz zwischen wechselseitig gesendeten und empfangenen Signalen zweier integrierter Ultraschallsensoren auswertet.

Die Steuerung des Durchflusses im Heizkörper erfolgt über das Heizungsventil, welches mit Hilfe eines Stellantriebs geöffnet und geschlossen wird. Dazu kann ein Elektromotor eingesetzt werden, der sich durch eine schnelle Reaktion auszeichnet jedoch gleichzeitig mit hohen Anschaffungskosten verbunden ist. Außerdem kann auch ein thermoelektrisches Element als Antrieb genutzt werden, welches sich durch geringe Anschaffungskosten auszeichnet jedoch eine langsamere Reaktionszeit besitzt. Die Ansteuerung des Stellantriebs erfolgt üblicherweise unabhängig von der Ausführung durch ein digitales oder analoges Spannungs- oder Stromsignal.

Um dem Controller die Steuerung zu ermöglichen, wird ein Signalwandler eingesetzt der Modbustelegramme in elektrische Signale umwandeln kann. Dazu wird innerhalb der Anlage das *EX9024-M* Modul von EXPERTDAQ eingesetzt, welches sich leicht in ein serielles RS485-Netzwerk einbinden lässt. Das Modul ist Modbus RTU-fähig und hat vier analoge Ausgänge die Spannungen und Ströme in verschiedenen Bereichen ausgeben können. Da für den Stellantrieb lediglich ein Ausgang genutzt wird, besteht explizit die Möglichkeit einer einfachen Erweiterung der Anlage um 3 weitere analoge Komponenten, beispielsweise um die Jalousien zu bedienen. Der thermoelektrische Stellantrieb *ABNM-LIN* von *Danfoss* wird zur stufenlosen Betätigung des Ventils

eingesetzt. Die Ansteuerung erfolgt über ein 0-10 V Signal, innerhalb der Anlage vom EX9024-M bereitgestellt wird, welches Signal in einen proportionalen, linearen Stellweg umwandelt.

Dieses Konzept bietet durch immer darauf hingewiesen die explizite Möglichkeit zur Erweiterung auch die offene Architektur und die gewählten Komponenten langfristig auch die Möglichkeit die Anlage durch eine Klimatisierung zu erweitern, um die Raumtemperatur ganzjährig beziehungsweise in beide Richtungen zu regeln zu können. Außerdem besteht die Möglichkeit in diesem Zuge auch das öffnen und Schließen der Jalousien ebenfalls von der Anlage aus gesteuert werden kann.

Damit ist die Idee der Anlage umrissen und im nächsten Abschnitt wird ein Konzept vorgestellt, dass diese Idee implementiert.

3.4 Umsetzung und Installation

3.5 Software

„Das beste Modell für eine Katze ist eine Katze; möglichst dieselbe Katze.“
— NORBERT WIENER

4 Modellbildung des Raumes

Ziel dieses Kapitels ist es, ein hinreichend exaktes Modell zur Berechnung der Raumtemperatur, basierend auf den thermodynamischen Prozessen mit dessen Umgebung und der Anlage aus Kapitel 3, zu bilden, um damit und unter Zuhilfenahme der Anlage Modellpräditive Regelung zu ermöglichen. Dazu wird zunächst ein einfaches Grundmodell für einen hypothetischen Raum gebildet, dass anschließend schrittweise an den bestehenden Raum erweitert angepasst wird, bis eine die Qualität des Modells ausreichend ist.

Physikalisch motiviert. Anhand von Zustand, Steuergröße und Parameter/Eigenschaft

4.1 Anforderungen an das Raummodell

Da MPC die Lösung von Nichtlinearen Gleichungssystemen erfordert wird ein erhöhter Rechenbedarf benötigt. Daher sollte das Modell so einfach wie möglich gehalten werden. Die Krux liegt also darin, einen geeigneten Kompromiss zwischen Komplexität und Genauigkeit des Modells zu finden, der eine sinnvolle MPC Regelung ermöglicht. Des Weiteren werden gradientenbasierte Ableitungen zweifach bei der Optimierung/Lösung des LGS generiert weshalb auch keine Unstetigkeiten im Modell vorkommen dürfen. Die triviale Aufgabe ist die hinreichend genaue Beschreibung der Realität bzw realen Vorgänge. hinreichend definieren. Damit eine Steuerung mit Hilfe von Modellprädiktiver Regelung möglich ist, darf das Modell keine hohe Kompliziertheit aufweisen und sollte durch möglichst wenig Gleichungen trotzdem ein möglichst genaues Abbild der Realität abbilden. Das Modell soll zunächst so simpel wie möglich gestaltet werden um eine Optimierung mit Hilfe von MPC zu ermöglichen. Dessen Verfahren zur Optimierung sind gradientenbasiert und erfordern damit die Erzeugung von stetigen Ableitungen bis zum zweiten Grad. Daher soll die Komplexität des Modells zunächst sehr gering gehalten werden und dann Stück für Stück erhöht werden und die damit die Genauigkeit des Modells erhöht werden. Die Optimalsteuerung stellt in diesem Fall den begrenzenden Faktor dar, da die Optimierungsumgebung CasADi für dynamische Systeme nur unter JModelica.org läuft. Daher wird darauf aufbauend das benötigte Modell für die MPC in Modelica gebildet unter Berücksichtigung der Restriktionen bezüglich JModelica.

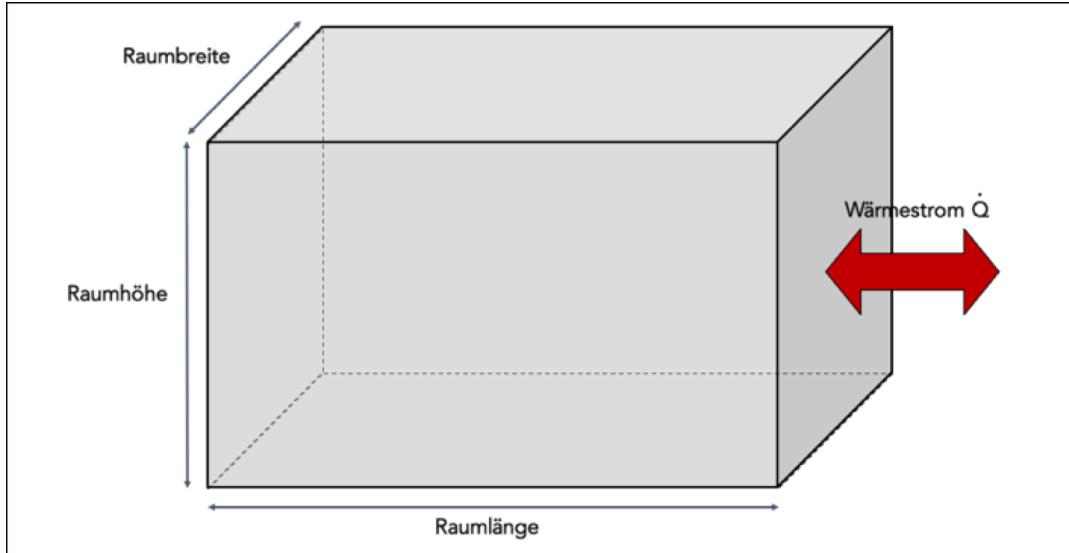


Abb. 4.1: Grundmodell eines Raumes

4.2 Das Grundmodell des Raumes

Um ein möglichst einfaches Grundmodell zu erhalten, wird zunächst ein hypothetischer Raum betrachtet. Dieser Raum bildet zusammen mit der ihn umgebenden Luft ein abgeschlossenes thermodynamisches System, wie in Kapitel 2.3 beschrieben. Der Raum ist selbst mit Luft gefüllt und wird zu allen sechs Seiten hin durch Wände begrenzt. Damit bildet der Raum ein geschlossenes System, da keine Massenströme über die Grenzen hinweg fließen können. An den Grenzflächen kann also lediglich Wärme zwischen der Umgebung und dem Raum ausgetauscht werden. Des Weiteren wird eine homogene Temperatur innerhalb des Raumes und der Umgebung angenommen, welche in der Realität eingeschwungenen Gleichgewichtszuständen innerhalb der beiden Teilsysteme entspricht. Um die Annahme für den Raum zu überprüfen, muss noch festgestellt werden auf welcher zeitlichen Skala der Einschwingvorgang für eine homogene Temperatur innerhalb des Raumes stattfindet und ob dieser damit eine Relevanz für die Modellbildung besitzt.

Zur Bestimmung der Temperatur innerhalb des Raumes, ausgehend von einer initialen Raumtemperatur und dem externen Steuerungsparameter der Umgebungstemperatur, muss der Ausgleichsprozess zwischen Raum und Umgebung untersucht werden, konkret der ausgetauschte Wärmestrom. Um diesen nach Gl. 4 zu berechnen, müssen zunächst die verschiedene modellrelevanten Eigenschaften des Raumes durch physikalische Größen und Variablen beschrieben werden. Zur Berechnung der Austauschoberfläche wird die Raumbreite, -länge und -höhe benötigt und weiterhin sind der U-Wert einer Betonwand, die spezifische Wärmekapazität und Dichte von Luft für die Bestimmung des Wärmestroms relevant. Diese modellrelevanten Eigenschaften sind allesamt mit ihren Zahlenwerten in Tabelle Tab. 4.1 zusammengefasst.

Modellrelevante Eigenschaften	Wert	Einheit
Raumbreite	7,81 ¹⁾	[m]
Raumlänge	5,78 ¹⁾	[m]
Raumhöhe	2,99 ¹⁾	[m]
Wärmedurchgangskoeffizient Betonwand	2,0 ²⁾	[$\frac{W}{m^2 \cdot K}$]
Spezifische Wärmekapazität von Luft	1.000,0 ³⁾	[$\frac{J}{kg \cdot K}$]
Dichte von Luft	1,25 ³⁾	[$\frac{kg}{m^3}$]

¹⁾Werte durch eigene Vermessung des Raumes K004b vom 07.12.2015.

²⁾Schätzwert, geschätzt nach [Recknagel, 2013, S. 409] mit Richtwerten aus [Recknagel, 2013, S. 194ff].

³⁾Tabellenwert aus [Peter Häupl, 2013, S. 139].

Tab. 4.1: Eigenschaften des Raummodells

Erfolgt nun die Bilanzierung des Raumes mit Hilfe des ersten Hauptsatzes der Thermodynamik nach Gl. 3 und die Berechnung der inneren Energie des Raumes nach Gl. 2 ergibt sich folgendes, einfaches Gleichungssystem zur Bestimmung der Raumtemperatur in Abhängigkeit vom Steuergrößen Außentemperatur im Grundmodell in Modelica:

```

1 equation
  /* calculate room volume */
3   room_volume = room_length * room_height * room_breadth;
  /* calculate room mass */
5   room_mass = room_volume * rho_air;
  /* calculate surface of heat exchange */
7   exchange_surface = 2 * (room_length * room_breadth) + 2 * (room_length * room_height) + 2
    * (room_breadth * room_height);
  /* calculate inner energy*/
9   room_u = room_mass * cp_air * room_temperature;
  /* calculate derivative of the inner energy */
11  der(room_u) = environment_qdot;
  /* calculate heatflow between room and environment */
13  environment_qdot = u_wall * exchange_surface * (environment_temperature - room_temperature
  );

```

Listing 4.1: Einfaches Gleichungssystem für das Grundmodell des Raumes in Modelica

Damit ist ein Grundmodell für einen Raum gebildet, wie in Abb. 4.1 graphisch dargestellt, um die Temperatur innerhalb eines Raumes zu berechnen. Dieses wird im Folgenden nun schrittweise erweitert und zum Abschluss überprüft, ob es der Realität genüge zu trägt.

4.3 Modellerweiterung durch Berücksichtigung der realen Umgebung

Im nächsten Schritt wird das einfache Raummodell zunächst an die reale Umgebung des Raums K004b angepasst. Die Lage von K004b ist in Abb. 3.1 ersichtlich und es ist zu

erkennen, dass der Raum lediglich zwei Außenwände besitzt, die an die Umgebungsluft grenzen: Die Wände in Richtung Süden und Westen. Die anderen beiden Wände, sowie die Decke und der Boden, grenzen an weitere Gebäudeteile des K-Gebäudes. Somit entspricht der Raum im Modell nach wie vor einem geschlossenen System und bildet weiterhin, zusammen mit dem umgebenden K-Gebäude und der Umgebungsluft, ein abgeschlossenes System. Jedoch müssen nun potenziell verschiedene Wärmeströme zwischen dem Raum und der Außenumgebung sowie dem Raum und dem K-Gebäude betrachtet werden. Da die fließenden Wärmeströme im Vergleich zur sehr großen Energie innerhalb des gesamten K-Gebäudes und der Umgebung nur verschwindend gering sind, wird der erwärmende beziehungsweise kühlende Effekt der Wärmeströme auf die beiden Teilsysteme vernachlässigt und es wird von konstanten, homogenen Temperaturen beider ausgegangen.

Durch diese Erweiterung des Modells hängt die Raumtemperatur nun von zwei Wärmeströmen und damit indirekt von zwei externen Steuergrößen, den Temperaturen in der Umgebung und im K-Gebäude, ab. Um die Wärmeströme separat berechnen zu können, wird die gesamte Oberfläche zum Wärmeaustausch aufgeteilt in die Austauschoberfläche mit der Umgebung und die Austauschoberfläche mit dem K-Gebäude. Des Weiteren werden im Modell die Temperatur der Außenumgebung und die Temperatur innerhalb des K-Gebäudes als externe Steuergrößen berücksichtigt. Das Gleichungssystem des Grundmodells in 4.1 erweitert sich also um folgende Änderungen:

```

1 equation
  [...]
3  /* calculate surface of heat exchange with the environment */
  environment_surface = room_length * room_height + room_breadth * room_height;
5  /* calculate surface of heat exchange with the remaining building */
  building_surface = 2 * (room_length * room_breadth) + room_length * room_height +
    room_breadth * room_height;
7  /* calculate derivative of the inner energy */
  der(room_u) = environment_qdot + building_qdot;
9  /* calculate heatflow between room and environment */
  environment_qdot = u_wall * environment_surface * (environment_temperature -
    room_temperature);
11 /* calculate heatflow between room and building */
  building_qdot = u_wall * building_surface * (building_temperature - room_temperature);

```

Listing 4.2: Erweitertes Gleichungssystem Modell des Raumes unter Berücksichtigung der realen Umgebung in Modelica

Damit wurde das Raummodell an die reale Umgebung angepasst und um die Temperatur innerhalb des Raumes zu bestimmen, wird nun neben der Ausgangstemperatur im Raum und der Umgebungstemperatur noch die Temperatur innerhalb des restlichen K-Gebäudes berücksichtigt. Im nächsten Schritt werden die realen, räumlichen Gegebenheiten im Modell abgebildet.

4.4 Modellerweiterung durch Berücksichtigung der räumlichen Gegebenheiten

Um das Modell an die realen Gegebenheiten des Raumes K004b anzupassen, müssen zwei bauliche Gegebenheiten beachtet werden. Wie in Abb. 3.1 dargestellt ist, ist in der südlichen Außenwand eine Fensterfront vorhanden. Da der U-Wert eines Fensters erheblich von dem U-Wert einer Wand abweicht, entsteht ein zusätzlicher Wärmestrom zwischen dem Raum und der Umgebung durch das Fenster hindurch. Das Öffnen und Schließen der Fenster mit daraus resultierenden Massenströmen wird jedoch zunächst nicht explizit berücksichtigt, weshalb das Raummodell weiterhin als geschlossenes System betrachtet wird. Des Weiteren ist es möglich, den Raum über einen Heizkörper zu beheizen. Mit dem Heizkörper, der zunächst als einfache Wärmequelle im Modell ergänzt wird, erhöht sich die Anzahl der externen Steuergrößen erneut, da die Temperatur innerhalb des Raumes auch von dieser abhängig ist.

Durch diese Erweiterungen werden auch weitere physikalische Größen zur Beschreibung der Eigenschaften des Raummodells benötigt. Wie bereits erwähnt werden die Eigenschaften um den U-Wert eines Fensters, sowie die Breite und Höhe der Fensterfront ergänzt wie in Tabelle Tab. 4.2 zusammengefasst.

Modellrelevante Eigenschaften	Wert	Einheit
Fensterbreite	7,0 ¹⁾	[m]
Fensterhöhe	2,08 ¹⁾	[m]
Wärmedurchgangskoeffizient Glas	4,0 ²⁾	[$\frac{W}{m^2 \cdot K}$]

¹⁾Werte durch eigene Vermessung des Raumes K004b vom 07.12.2015.

²⁾Tabellenwert, geschätzt nach [Hauser, 2000, S. 270ff.].

Tab. 4.2: Weitere Eigenschaften des Raummodells

Durch diese Anpassung verändert sich die Austauschoberfläche mit der Umgebung, die sich nun auf zwei Flächen mit verschiedenen Wärmedurchgangskoeffizienten verteilt. Des Weiteren wird eine Wärmequelle für die Heizung ergänzt, so dass sich folgende Änderungen des Gleichungssystems im Vergleich zum bisherigen Modell in 4.1 ergeben:

```

1  equation
2  [...]
3  /* calculate surface of heat exchange with the environment */
4  environment_surface = room_length * room_height + room_breadth * room_height -
5      window_surface;
6  /* calculate surface of heat exchange with the remaining building */
7  building_surface = 2 * (room_length * room_breadth) + room_length * room_height +
8      room_breadth * room_height;
9  /* calculate surface of window with the environment */
10 window_surface=(window_length*window_height);
11 /* calculate derivative of the inner energy */
12 der(room_u) = environment_qdot + building_qdot + window_qdot + radiator_qdot;

```

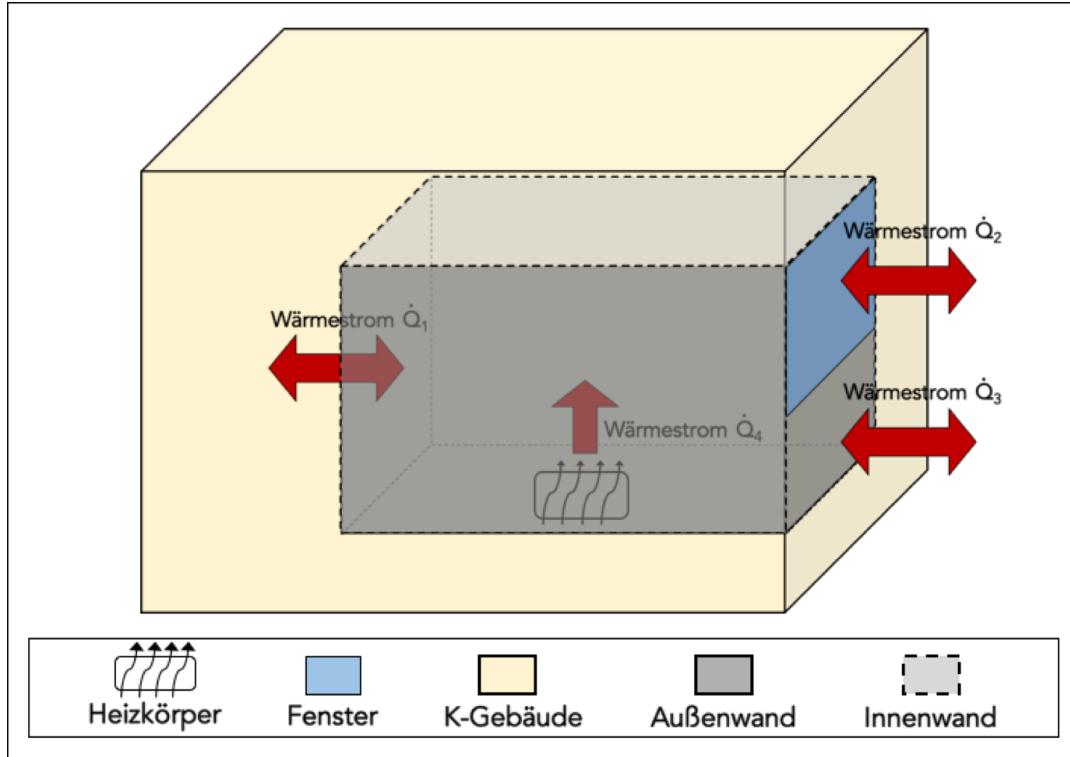


Abb. 4.2: Erweitertes Raummodell

```

12  /* calculate heatflow between room and environment through the walls */
13  environment_qdot = u_wall * environment_surface * (environment_temperature -
   room_temperature);
14  /* calculate heatflow between room and environment through the window */
15  building_qdot = u_glass * window_surface * (environment_temperature - room_temperature);
16  /* calculate heatflow between room and building */
17  building_qdot = u_wall * building_surface * (building_temperature - room_temperature);

```

Listing 4.3: Erweitertes Gleichungssystem Modell des Raumes unter Berücksichtigung der räumlichen Gegebenheiten in Modelica

Damit ist das Modell auch an die räumlichen Gegebenheiten angepasst und beschreibt dadurch in groben Zügen die realen Zusammenhänge. Allerdings kann eine Steuerung/Regelung eine Heizung nicht einfach beliebig als einfache Wärmequelle einsetzen, weshalb eine detailliertere, physikalisch motivierte Modellierung folgen wird. Die bisherigen Zusammenhänge des Modells sind in Abb. 4.2 graphisch dargestellt.

Bereits bei den Einsatzzielen der Anlage in 3.1 war gefordert, den Zusammenhang zwischen der Sonneneinstrahlung und der Raumtemperatur zu untersuchen sowie Störgrößen explizit in Kauf zu nehmen. Daher ist es passend, dass die Fensterfront in Richtung Süden ausgerichtet ist und der Raum K004b als Büro genutzt wird. Daher wird im nächsten Abschnitt das Modell an diese beiden Faktoren angepasst.

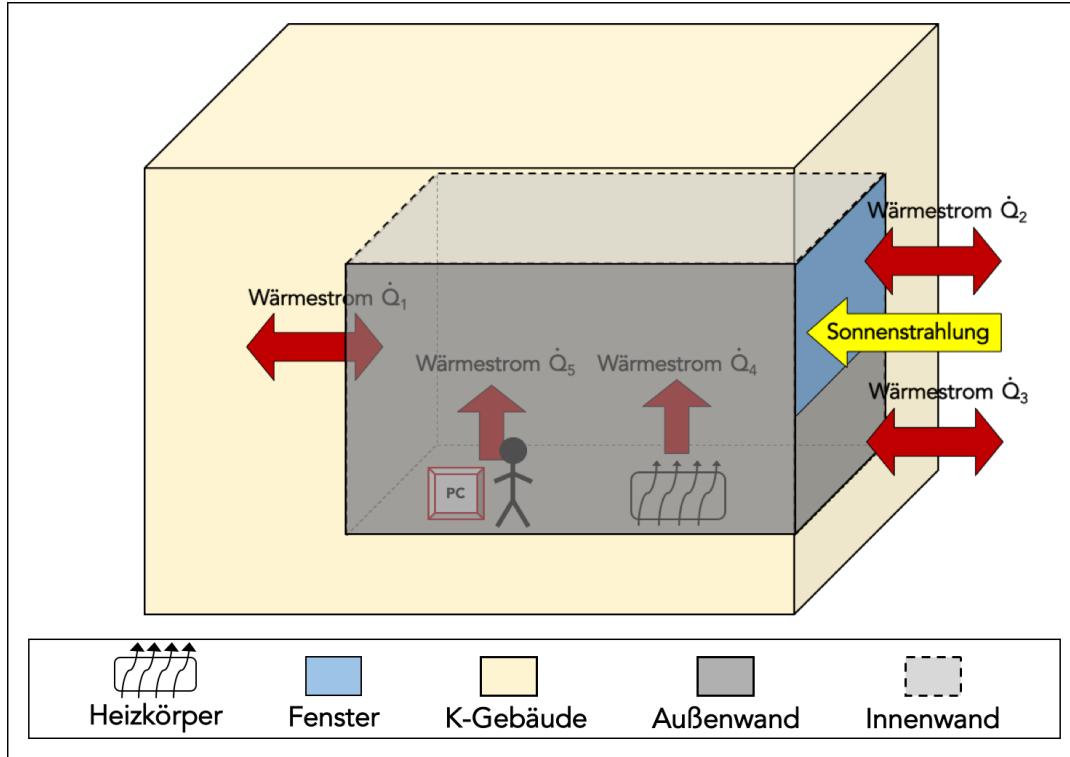


Abb. 4.3: Erweitertes Raummodell

4.5 Modellerweiterung durch Berücksichtigung von Störgrößen und der Sonneneinstrahlung

Der Raum K004b wird regulär als Büro genutzt, weshalb verschiedene Faktoren als Störgrößen in Bezug auf die Raumtemperatur betrachtet werden können. Zum einen wird durch die Menschen und deren Rechner weitere Wärme in den Raum eingebracht und zum Anderen werden die Fenster und die Türen manchmal geöffnet. Des Weiteren trifft insbesondere auf die südseitigen Fenster Sonnenstrahlung, welche ebenfalls Wärme in den Raum einbringen und damit einen Einfluss auf die Raumtemperatur haben.

```

equation
2 [...]
/* calculate derivative of the inner energy */
4 der(room_u) = environment_qdot + building_qdot + window_qdot + radiator_qdot + sun_qdot +
    otherfactors_qdot;

```

Listing 4.4: Erweitertes Gleichungssystem Modell des Raumes unter Berücksichtigung der Sonneneinstrahlung und Störgrößen

4.6 Validierung des Modells

4.7 Anpassung des Modells mit Parameterschätzung

„Knowing where things are, and why, is essential to rational decision making.“
— JACK DANGERMOND, Esri

5 Schlussbetrachtung

5.1 Fazit

5.2 Ausblick und Ansatzpunkte für weitere Arbeiten

Welche Art der Verwendung? MPC mit JModelica.org also deren mpc Klasse eigene in casadi etc?

Vergleich mit bestehenden Modellen möglich z.B. das THERAKLES Modell

A Modelle

A.1 Raummodell

```

1 package room_model_backup

3   connector Temperature
4     Modelica.SIunits.Conversions.NonSIunits.Temperature_degC t;
5   end Temperature;

7   connector HeatFlow
8     Modelica.SIunits.HeatFlowRate qdot;
9   end HeatFlow;

11  connector MassTemperature
12    Modelica.SIunits.Conversions.NonSIunits.Temperature_degC t;
13    Modelica.SIunits.MassFlowRate mdot;
14  end MassTemperature;

15  connector RadiantEnergyFluenceRate
16    Modelica.SIunits.DensityOfHeatFlowRate radiation_sun;
17  end RadiantEnergyFluenceRate;

19  model CV_Radiator "control volume for a discretized radiator"
20

21    /** parameter */
22    outer parameter Modelica.SIunits.CoefficientOfHeatTransfer u_radiator
23      "heat transfer coefficient of the radiator";
24    outer parameter Modelica.SIunits.SpecificHeatCapacity cp_water
25      "specific heat capacity of water";
26    outer Modelica.SIunits.Mass cv_m
27      "mass within one control volume";
28    outer Modelica.SIunits.Area exchange_surface
29      "surface of one control volume at which heat transfer takes place";
30    Modelica.SIunits.Energy cv_u
31      "inner energy of the control volume";
32    Modelica.SIunits.HeatFlowRate cv_qdot
33      "heatflowrate over the borders of the control volume";
34    Modelica.SIunits.Conversions.NonSIunits.Temperature_degC cv_temperature_out(start=21.2,
35      fixed=true)
36      "temperature of the fluid leaving the control volume";

37    /** states */
38    outer Modelica.SIunits.Conversions.NonSIunits.Temperature_degC room_temperature_cv
39      "temperature within the room";

41    /** controls */
42    Modelica.SIunits.Conversions.NonSIunits.Temperature_degC cv_temperature_in=inlet.t
43      "temperature of the fluid streaming in the control volume";
44    outer Modelica.SIunits.MassFlowRate mdot
45      "massflowrate within the radiator";

47    equation
48      /* calculate inner energy */
49      cv_u = cv_m * cp_water * cv_temperature_out;
50      /* calculate derival of the inner energy */
51      der(cv_u) = mdot * cp_water * (cv_temperature_in - cv_temperature_out) - cv_qdot;
52      /* calculate heatflowrate */
53

```

```
    cv_qdot = u_radiator * exchange_surface * (cv_temperature_out - room_temperature_cv);
55  /* commit calculated temperature */
56  outlet.t = cv_temperature_out;
57 end CV_Radiator;

59 model Radiator "model for a discretized radiator within a room"

61  /** parameter */
62  parameter Real radiator_element_number=106
63   "number of normed elements of which the radiator consists";
64  parameter Real radiator_tubes_element=2
65   "number of parallel tubes in one element";
66  parameter Integer cv_number = 20
67   "number of control volumes in which the radiator is discretized";
68  parameter Modelica.SIunits.Length radiator_element_length=0.045
69   "length of one element depending on type (Recknagel 2013/2014: Heizung und Klimatechnik
70   S.815ff.)";
71  parameter Modelica.SIunits.Height tube_length_vertical=0.4
72   "height of one element depending on type (Recknagel 2013/2014: Heizung und Klimatechnik
73   S.815ff.)";
74  parameter Modelica.SIunits.Diameter tube_diameter_horizontal=0.05
75   "diameter of the horizontal tubes depending on type (Recknagel 2013/2014: Heizung und
76   Klimatechnik S.815ff.)";
77  parameter Modelica.SIunits.Diameter tube_diameter_vertical=0.0255
78   "diameter of the vertical tubes depending on type (Recknagel 2013/2014: Heizung und
79   Klimatechnik S.815ff.)";
80  parameter Modelica.SIunits.Density rho_water = 1000
81   "density of water";
82  parameter Modelica.SIunits.Mass radiator_element_mass=0.35
83   "mass within one element of the radiator depending on type (Recknagel 2013/2014:
84   Heizung und Klimatechnik S.815ff.)";
85  inner parameter Modelica.SIunits.CoefficientOfHeatTransfer u_radiator = 12.9872
86   "heat transfer coefficient of the radiator";
87  inner parameter Modelica.SIunits.SpecificHeatCapacity cp_water = 4200
88   "specific heat capacity of water";
89  CV_Radiator[cv_number] cv_radiator
90   "array of control volumes to discretize the radiator";
91  Modelica.SIunits.HeatFlowRate radiator_qdot_out
92   "heatflow which is leaving the radiator";
93  Modelica.SIunits.Conversions.NonSIunits.Temperature_degC radiator_temperature_out
94   "calculated(predicted) temperature of the water leaving the radiator";
95  inner Modelica.SIunits.Mass cv_m
96   "mass within one control volume";
97  inner Modelica.SIunits.Area exchange_surface
98   "surface of one control volume at which h";

99  /** states */
100 inner Modelica.SIunits.Conversions.NonSIunits.Temperature_degC room_temperature_cv
101  "temperature within the room for the control volume";
102 outer Modelica.SIunits.Conversions.NonSIunits.Temperature_degC room_temperature
103  "temperature within the room from the room";

104  /** controls */
105 inner Modelica.SIunits.MassFlowRate mdot=inlet.mdot
106  "massflowrate within the radiator";
107 Modelica.SIunits.Conversions.NonSIunits.Temperature_degC radiator_inlet = inlet.t
108  "temperature of the inflowing fluid";

109 equation
110  /* calculate surface of one control volume */
```

```
109 exchange_surface = (radiator_element_number * radiator_element_length * 2 *
    Modelica.Constants.pi * tube_diameter_horizontal + radiator_element_number *
    radiator_tubes_element * tube_length_vertical * Modelica.Constants.pi *
    tube_diameter_vertical)/cv_number;
/* calculate mass within one control volume */
111 cv_m = (radiator_element_mass * radiator_element_number)/cv_number;
/* commit temperature of radiator fluid inlet to the first control volume */
113 cv_radiator[1].inlet.t = radiator_inlet;
/* commit fluid temperatures within the radiator control volumes */
115 for i in 1 : (cv_number-1) loop
    connect( cv_radiator[i].outlet, cv_radiator[i+1].inlet);
117 end for;
/* save fluid temperature of the last radiator control volumes */
119 radiator_temperature_out=cv_radiator[cv_number].outlet.t;
/* calculate and save the heatflowrate which is leaving the radiator*/
121 radiator_qdot_out = sum(cv_radiator.cv_qdot);
/* commit roomtemperature*/
123 room_temperature=room_temperature_cv;
outlet.mdot = mdot;
125 outlet.t = radiator_temperature_out;
end Radiator;
127
model Room_radiator_window "model of a room for mpc purpose with JModelica.org"
129
    /** parameter p **/
131     parameter Modelica.SIunits.Length room_length=7.81
        "length of the room";
133     parameter Modelica.SIunits.Breadth room_breadth=5.78
        "breadth of the room";
135     parameter Modelica.SIunits.Height room_height=2.99
        "height of the room";
137     parameter Modelica.SIunits.Length window_length=7
        "length of the window";
139     parameter Modelica.SIunits.Height window_height=2.08
        "height of the window";
141     parameter Modelica.SIunits.Density rho_air = 1.2
        "density of air";
143     parameter Modelica.SIunits.CoefficientOfHeatTransfer u_glass=2.0
        "heat transfer coefficient for glass ";
145     parameter Modelica.SIunits.CoefficientOfHeatTransfer u_wall=0.612986
        "heat transfer coefficient for the walls of the room";
147     parameter Modelica.SIunits.SpecificHeatCapacity cp_air=1000
        "specific heat capacity of air";
149 Radiator heating
    "instance of a radiator";
151 Window window
    "instance of a window";
153 Modelica.SIunits.Volume room_volume
    "volume of the room";
155 Modelica.SIunits.Mass room_mass
    "mass of air within the room";
157 Modelica.SIunits.Area building_surface
    "sum of contacting surfaces (walls) with other rooms of the building";
159 Modelica.SIunits.Area environment_surface
    "sum of contacting surfaces (walls) with the environment";
161 inner Modelica.SIunits.Area window_surface
    "sum of contacting surfaces (windows) with the environment";
163 Modelica.SIunits.Energy room_u
    "inner energy of the system room";
165 Modelica.SIunits.HeatFlowRate building_qdot
    "rate of heat flow with other rooms within the building";
```

```
167     Modelica.SIunits.HeatFlowRate environment_qdot
168     "rate of heat flow with the environment";
169     Modelica.SIunits.HeatFlowRate environment_qdot_wall
170     "rate of heat flow with the environment through the wall";
171     Modelica.SIunits.HeatFlowRate environment_qdot_window
172     "rate of heat flow with the environment through the window";
173     Modelica.SIunits.HeatFlowRate qdot_loss
174     "summed up rate of heatflow leaving the system";
175     Modelica.SIunits.HeatFlowRate radiator_qdot
176     "heat flow rate at the radiator surfaces streaming into the room";
177
178     /** states x **/
179     inner Modelica.SIunits.Conversions.NonSIunits.Temperature_degC room_temperature(start
180         =24, fixed=true)
181         "temperature within the room (Initially about 24 degree celsius)";
182
183     /** controls u **/
184     Modelica.SIunits.MassFlowRate mdot=inlet_radiator.mdot
185         "commitment of the massflowrate to the radiator";
186     Modelica.SIunits.Conversions.NonSIunits.Temperature_degC environment_temperature=
187         inlet_environment.t
188         "temperature of the environment";
189     Modelica.SIunits.Conversions.NonSIunits.Temperature_degC building_temperature=
190         inlet_building.t
191         "temperature of the rest of the building";
192     Modelica.SIunits.HeatFlowRate qdot_sun
193         "rate of heat flow brought in by the sun";
194     Modelica.SIunits.HeatFlowRate qdot_otherfactors=inlet_other.qdot
195         "rate of heat flow brought in by other factors (e.g. people, computer)";
196
197     equation
198
199     /* calculate room volume */
200     room_volume=room_length*room_height*room_breadth;
201
202     /* calculate room mass */
203     room_mass=room_volume*rho_air;
204
205     /* calculate surface of the room with other rooms of the building */
206     building_surface=(room_length*room_breadth*2)+(room_length*room_height*2);
207
208     /* calculate wall surface of the room with the environment */
209     environment_surface=(room_length*room_height)+(room_breadth*room_height)-window_surface;
210
211     /* calculate window surface of the room with the environment */
212     window_surface=(window_length*window_height);
213
214     /* calculate inner energy*/
215     room_u = room_mass * cp_air * room_temperature;
216
217     /* calculate derival of the inner energy */
218     der(room_u) = radiator_qdot + qdot_loss + qdot_sun + qdot_otherfactors;
219
220     /* sum up the lost heat flow */
221     qdot_loss = building_qdot+environment_qdot;
222
223     /* calculate lost heatflow with other rooms of the building */
224     building_qdot = u_wall * building_surface * (building_temperature-room_temperature);
225
226     /* sum up the lost heatflow with the environment */
227     environment_qdot = environment_qdot_window + environment_qdot_wall;
228
229     /* calculate lost heatflow with the environment through the window */
230     environment_qdot_window = u_glass * window_surface * (environment_temperature -
231         room_temperature);
232
233     /* calculate lost heatflow with the environment through the wall */
234     environment_qdot_wall = u_wall * environment_surface * (environment_temperature -
235         room_temperature);
236
237     /* commit the inflowing heat flow of the radiator */
238     radiator_qdot=heating.radiator_qdot_out;
239
240     /* connect radiator and sun with room */
241     connect(heating.inlet, inlet_radiator);
```

```
223  connect(inlet_sun, window.inlet_sun);
224  qdot_sun = window.outlet_room.qdot;
225 end Room_radiator_window;

227 model SourceTemp
228   Modelica.SIunits.Conversions.NonSIunits.Temperature_degC t=inlet;
229   Modelica.Blocks.Interfaces.RealInput inlet;
230   equation
231     outlet.t=t;
232   end SourceTemp;
233
234 model SourceHeat
235   Modelica.SIunits.HeatFlowRate qdot=inlet;
236   Modelica.Blocks.Interfaces.RealInput inlet;
237   equation
238     outlet.qdot=qdot;
239   end SourceHeat;

240 model SourceTempMass
241   Modelica.SIunits.Conversions.NonSIunits.Temperature_degC t=inlet_t;
242   Modelica.SIunits.MassFlowRate mdot=inlet_mdot;
243   Modelica.Blocks.Interfaces.RealInput inlet_t;
244   Modelica.Blocks.Interfaces.RealInput inlet_mdot;
245   equation
246     outlet.t=t;
247     outlet.mdot=mdot;
248   end SourceTempMass;

250 model Window
251   Modelica.SIunits.DensityOfHeatFlowRate radiation_arriving=inlet_sun.radiation_sun;
252   Modelica.SIunits.HeatFlowRate qdot_effective;
253   parameter Real window_transmission = 0.00687213;
254   outer Modelica.SIunits.Area window_surface;
255   equation
256     qdot_effective = radiation_arriving * window_transmission * window_surface;
257     qdot_effective = outlet_room.qdot;
258   end Window;

260
261 model SourceSun
262   Modelica.SIunits.DensityOfHeatFlowRate radiation_sun = inlet;
263   Modelica.Blocks.Interfaces.RealInput inlet;
264   RadiantEnergyFluenceRate outlet;
265   equation
266     outlet.radiation_sun = radiation_sun;
267   end SourceSun;
268
269 end room_model_backup;
```

B Datenblätter

B.1 THERMASGARD RTM1-Modbus Raumtemperaturfühler von S+S Regeltechnik

THERMASGARD® RTM1- Modbus

Raumbedien-Temperaturfühler ($\pm 3\%$), Aufputz,
für Temperatur, relative Feuchte, Taupunkt,
kalibrierfähig, mit Modbus-Anschluss

TECHNISCHE DATEN

Spannungsversorgung:	24 V AC ($\pm 20\%$) und 15...36 V DC ($\pm 10\%$)
Leistungsaufnahme:	< 1,0VA / 24 V DC < 2,2VA / 24 V AC
Sensor:	digitaler Feuchtesensor mit integriertem Temperatursensor, kleine Hysterese, hohe Langzeitstabilität
Datenpunkte:	Temperatur, relative Feuchte, Taupunkt, Sollwertpotentiometer
Messbereich:	0...+50 °C
Abweichung Temperatur:	$\pm 0,5K$ bei +20 °C
Nullpunkt-Offset:	$\pm 10^{\circ}\text{C}$, über Potentiometer einstellbar
Umgebungstemperatur:	Lagerung -35...+85 °C; Betrieb 0...+50 °C
Medium:	saubere Luft und nicht aggressiv, nicht brennbare Gase
Signalfilterung:	4 s / 32 s
Prozessanschluss:	mittels Schrauben
Gehäuse:	Kunststoff, Werkstoff ABS, Farbe Reinweiß (ähnlich RAL 9010)
Abmaße:	85 x 85 x 27 mm (Baldur 1)
Montage:	Wandmontage oder auf UP-Dose, Ø 55 mm, Unterteil mit 4-Löch, für Befestigung auf senkrecht oder waagerecht installierten UP-Dosen für Kabeleinführung hinten, mit Sollbruchstelle für Kabeleinführung oben/unten bei AP
zulässige Luftfeuchte:	< 95 % r. H., nicht kondensierende Luft
Schutzklass:	III (nach EN 60 730)
Schutzart:	IP 30 (nach EN 60 529)
Normen:	CE-Konformität, elektromagnetische Verträglichkeit nach EN 61326, nach EMV-Richtlinie 2004 / 108 / EC
Optional:	Display mit Beleuchtung, zweizeilig, programmierbar, Ausschnitt ca. 36 x 15 mm (B x H), zur Anzeige der Ist-Temperatur oder eines individuell programmierbaren Anzeigenwertes (Über die Modbuschnittstelle kann das Display sowohl im 7-Segment-Bereich, als auch im Dot-Matrix-Bereich individuell beschrieben werden.)




RTM1-Modbus
Standard

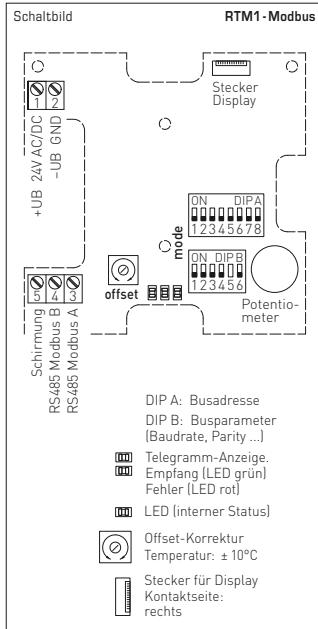


programmierbares
Display

RTM1-Modbus



RTM1-Modbus



Detailed description of the connection diagram:
 The diagram shows the RTM1-Modbus device with various connection points:
 - Power input: +UB 24VAC/DC, -UB GND.
 - Modbus communication: RS485 Modbus A (pins 2, 3) and RS485 Modbus B (pins 5, 4).
 - Address selection: DIP A (pins 1, 2) and DIP B (pins 3, 4).
 - Mode selection: mode (pin 5).
 - Offset adjustment: offset (pin 6).
 - Potentiometer: Potentiometer (pin 7).
 - Display connection: Stecker Display (pin 8).
 - LED status: LED (pin 9).
 - Error indicator: Fehler (pin 10).
 - Telegram display: Telegramm-Anzeige (pin 11).
 - Reception indicator: Empfang (pin 12).
 - Internal status: LED (pin 13).
 - Offset correction: Offset-Korrektur (pin 14).
 - Temperature range: Temperatur: $\pm 10^{\circ}\text{C}$ (pin 15).
 - Display connector: Stecker für Display (pin 16).
 - Contact side: rechts (pin 17).

S+S REGELTECHNIK

www.SplusS.de

+49 (0) 911 / 5 19 47-0

+49 (0) 911 / 5 19 47-70

Rev. 2016-V20 DE



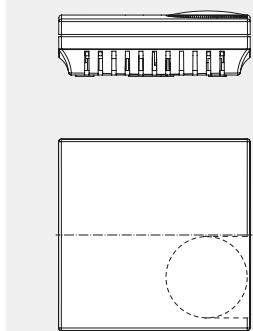
S+S REGELTECHNIK

NEW

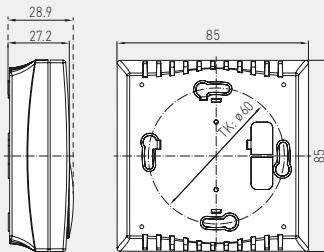
THERMASGARD® RTM1- Modbus

Raumbedien-Temperaturfühler ($\pm 3\%$), Aufputz,
für Temperatur, relative Feuchte, Taupunkt,
kalibrierfähig, mit Modbus-Anschluss

Maßzeichnung



Gehäuse Baldur1

RTM1- Modbus
mit DisplayRTM1- Modbus-P
mit Display
und PotentiometerRTM1- Modbus
mit DisplayRTM1- Modbus-P
mit PotentiometerRTM1- Modbus
ohne Display

THERMASGARD® RTM 1 - Modbus – Raumbedien-Temperaturfühler

Typ / WG01	Sensor	Ausgang	Ausstattung	Display	Art.-Nr.	Preis
RTM 1 - Modbus					IP 30	
RTM1-MODBUS	digital	Modbus	–		1101-42A6-0000-000	95,27 €
RTM1-MODBUS DISPLAY	digital	Modbus	–		■ 1101-42A6-2000-000	141,06 €
RTM 1 - P - Modbus					IP 30	
RTM1-MODBUS P	digital	Modbus	Potentiometer		1101-42A6-0001-005	125,27 €
RTM1-MODBUS P DISPLAY	digital	Modbus	Potentiometer		■ 1101-42A6-2001-005	225,59 €

B.2 Webthermograph 8x von WuT



Datenblatt:

Web-Thermograph 8x

Artikelnummer: 57608



Messen, loggen und melden...

Der Web-Thermograph 8x ist ein Messgerät welches **acht Temperaturwerte** erfasst und die Werte im Netzwerk zur Verfügung stellt. Das Gerät verfügt über einen integrierten Datenlogger und über zahlreiche Web- und Netzwerkdienste zur manuellen Abfrage der Messwerte, oder selbstständigem Versand von Meldungen.

Eigenschaften:

Sensor:

- **Acht Temperaturmesseingänge Pt100/Pt1000:**
 - Messbereich W&T Fühler: -50°C...180°C
 - Messeingang: -200°C...650°C

Konnektivität:

- Per Browser Temperaturen und Verläufe überwachen
- **Alarm und Berichtsfunktion:**
 - E-Mail zur Alarmierung oder als Berichtsfunktion
 - SNMP-Abfragen /-Alarm Traps
 - Bis zu 8 Alarmmeldungen konfigurierbar
- **Dynamische Integration in andere Webseiten:**
 - Direktzugriff auf aktuelle Messwerte via AJAX, JavaScript und Java-Applet
- **Weitere Software-Schnittstellen zur Einbindung in Ihre Systeme/Datenbanken:**

MULTICAL® 602

DATENBLATT

Verzeichnis

Rechenwerksfunktionen	3
Impulsaus- und Eingänge auf Modulen	10
Kabinetteinrichtung	11
Zugelassene Zählerdaten	12
Elektrische Daten	12
Mechanische Daten	15
Werkstoffbezeichnungen	15
Bestellvorschrift	16
Toleranzband	17
Maßskizzen	18
Zubehör	20

MULTICAL® 602

DATENBLATT

Rechenwerksfunktionen

Energieberechnung

MULTICAL® 602 berechnet die thermische Energie gemäß prEN 1434-1:2009, die die internationale Temperaturskala von 1990 (ITS-90) und die Druckdefinition von 16 bar verwendet.

Die Energieberechnung kann in vereinfachter Form wie folgt ausgedrückt werden:

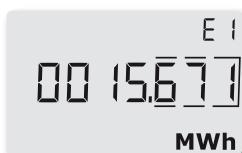
$$\text{Energie} = V \times \Delta\Theta \times k$$

V ist das zugeführte Wasservolumen

$\Delta\Theta$ ist die gemessene Temperaturdifferenz

k ist der Wärmekoeffizient des Wassers

Das Rechenwerk berechnet immer die Energie in [Wh]. Hiernach erfolgt die Umrechnung auf die gewählte Maßeinheit.



E [Wh] =	$V \times \Delta\Theta \times k \times 1000$
E [kWh] =	E [Wh] / 1.000
E [MWh] =	E [Wh] / 1.000.000
E [GJ] =	E [Wh] / 277.780
E [Gcal] =	E [Wh] / 1.163.100

Applikationstypen

MULTICAL® 602 arbeitet mit neun verschiedenen Energieformeln E1...E9, die alle bei jeder Integration parallel berechnet werden, unabhängig von der Konfiguration des Zählers.

Die Energiertypen E1 zu E9 werden wie folgt berechnet:

E1=V1(T1-T2)k Wärmeenergie (V1 in Vor- oder Rücklauf)

E2=V2(T1-T2)k Wärmeenergie (V2 in Rücklauf)

E3=V1(T2-T1)k Kälteenergie (V1 in Vor- oder Rücklauf)

E4=V1(T1-T3)k Vorlaufenergie

E5=V2(T2-T3)k Rücklaufenergie oder Zapfen von Rücklauf

E6=V2(T3-T4)k Zapfwasserenergie, separat

E7=V2(T1-T3)k Zapfwasserenergie von Vorlauf

E8=m³xT1 Die Grundlage für die Berechnung von volumenbasierten Durchschnittstemperaturen in der Vorlauf T1

E9=m³xT2 Die Grundlage für die Berechnung von volumenbasierten Durchschnittstemperaturen in der Rücklauf T2

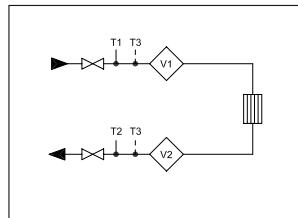
Somit kann MULTICAL® 602 die Wärme- und Kälteenergie der meisten Applikationen, sowohl geschlossener als offener Anlagen, berechnen.

Alle Energiertypen werden protokolliert und können konfigurationsabhängig angezeigt werden.

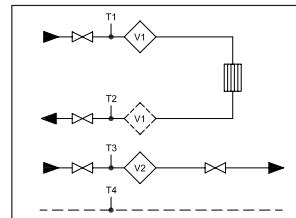
MULTICAL® 602

DATENBLATT

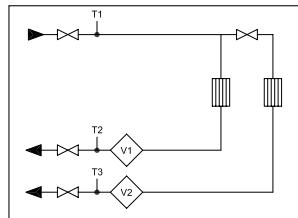
Rechenwerksfunktionen



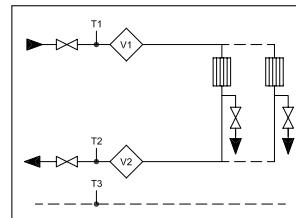
*Beispiel 1:
Geschlossenes thermisches System mit
einem Durchflusssensor*



*Beispiel 2:
Offenes Zweistrangsystem mit zwei
Durchflusssensoren*



*Beispiel 3:
Zwei Wärmekreise mit gemeinsamem
Vorlauf*



*Beispiel 4:
Offenes System mit zwei Durchflusssen-
soren*

Durchflusssmessung

MULTICAL® 602 berechnet den aktuellen Wasserdurchfluss nach zwei verschiedenen Prinzipien abhängig vom angeschlossenen Durchfluszzählertyp:

- Die Durchflussanzeige bei angeschlossenen elektronischen Durchfluszzählern wird alle 10 Sekunden aktualisiert.
- Die Durchflussanzeige bei angeschlossenen mechanischen Durchfluszzählern, normalerweise mit Reed-Schalter, wird auf der Basis einer Periodenzeitzählung berechnet und wird bei jedem Volumenimpuls aktualisiert.



B.3 MULTICAL 602 Wärmemengenzähler von Kamstrup

MULTICAL® 602

DATENBLATT



Wärme- und Kältezähler mit unbegrenzter Kommunikation

Anwendung

MULTICAL® 602 ist ein universelles Rechenwerk zur Wärme- und Kältemessung zusammen mit den meisten impulsgebenden Durchflusssensoren sowie einem 2- oder 4-Leiter Temperaturfühlerpaar. In Kombination mit dem Kamstrup Ultraschalldurchflusssensor ULTRAFLOW® bietet der Zähler noch erweiterte Funktionen. Dank seiner hohen Messgenauigkeit registriert der Zähler den genauen Verbrauch über die ganze Lebensdauer des Zählers. Der Zähler ist wartungsfrei, hat eine lange Lebensdauer, und garantiert so- mit minimale jährliche Betriebskosten.

MULTICAL® 602 wird zur Wärme-, Kälte- und kombinierten Wärme-/Kältemessung in allen wasserbasierten Anlagen mit Temperaturen von 2°C bis 180°C für Wärme und 2°C bis 50°C für Kälte verwendet.

Funktion

In Wärmeapplikationen wird MULTICAL® 602 zusammen mit dem Durchflusssensor ULTRAFLOW® 54 verwendet. Die Durchflussgrößen decken den Bereich von qp 0,6 m³/h bis qp 1.000 m³/h.

In Kälteapplikationen bis zu qp 100 m³/h wird MULTICAL® 602 zusammen

mit ULTRAFLOW® 14 und von qp 150 m³/h bis qp 1.000 m³/h zusammen mit ULTRAFLOW® 54 verwendet.

Das Rechenwerk kann an Durchflusssensoren bis zu qp 3.000 m³/h angeschlossen werden.

MULTICAL® 602 auszeichnet sich durch die komplette Auswahl von Kommunikationsmodulen und die eingebaute RTC (Echtzeituhr), die es leicht machen, den Zähler in allen Applikationen anzupassen, unabhängig von der Auslesemethode.

Für drahtgebundene Kommunikation kann der Zähler mit LON, SIOX, M-Bus, Datenmodul sowie den neuen Lösungen Metasys N2 und Ethernet/IP ausgestattet werden. Wird der Zähler in ein drahtloses Netzwerk integriert, können Sie Funk, Wireless M-Bus oder eines der neuen Module: GSM/GPRS oder High Power RadioRouter wählen.

Die InfoCodes und Datenlogger des Rechenwerks stellen ein unschätzbares Werkzeug zur Fehlersuche, Fehlerberichtigung und Analyse des Energieverbrauchs dar. Das Infologger überwacht ständig eine Reihe wichtiger Funktionen im Zähler, z.B. Fehler im Messsystem, Stromausfall, Leckage, Bersten, oder Montage des Durchflusssensors mit falscher Durch-

flussrichtung installiert ist. In solchen Fällen erscheinen eine blinkende "INFO" und eine InfoCode im Display.

MULTICAL® 602 speichert die Verbrauchsdaten jährlich, monatlich, täglich und stündlich, welches eine komplette Betriebsanalyse erleichtert.

Betriebsoptimierung

Bei eventuellem Stromausfall werden die Daten gespeichert, und die Abrechnung der Verbrauchsdaten wird somit sichergestellt. Durch die Batterieversorgung des Zählers wird die Lebensdauer auf bis zu 13 Jahre verlängert, einschl. Wireless M-Bus.

Schließlich garantiert MULTICAL® 602 mit ULTRAFLOW® und den genau gepaarten Temperaturfühlern präzise Messergebnisse eben bei minimalen Temperaturunterschieden. Die Langzeitstabilität und Genauigkeit des Durchflusssensors wird nicht von Durchflussgeschwindigkeit, Durchflusstörungen und Verschleiß beeinflusst, was für einen optimalen Betrieb sorgt.

MULTICAL® 602

DATENBLATT

Verzeichnis

Rechenwerksfunktionen	3
Impulsaus- und Eingänge auf Modulen	10
Kabinetteinrichtung	11
Zugelassene Zählerdaten	12
Elektrische Daten	12
Mechanische Daten	15
Werkstoffbezeichnungen	15
Bestellvorschrift	16
Toleranzband	17
Maßskizzen	18
Zubehör	20

C Modbusadressmapping

C.1 MULTICAL 602 Wärmemengenzähler von Kamstrup

Datenblatt

Modbus RTU Slave-Modul

Modbus-Kommunikationsmodul für MULTICAL® 62/601/602/6L2/6M2/801

- RTU-Kommunikation basiert auf RS-485
- Übertragungsgeschwindigkeit von bis zu 76.800 Bits/Sek.
- Einstellung von Programmierbaren Daten, der Kommunikationsgeschwindigkeit und der Parität
- Zwei Impulseingänge für zusätzliche Wasser- und Stromzähler
- RS-485 galvanisch getrennt vom Zähler



Beschreibung

Anwendung

Modbus ist ein offenes, weit verbreitetes und gut etabliertes, serielles Kommunikationsprotokoll, das im Rahmen der Gebäudeautomatisierung verwendet wird.

Das Modbus-Bodenmodul für MULTICAL® sichert eine einfache Integration von Wärme-, Kälte- und Wasserzählern mit Modbus-basierten Systemen.

Das Modbus-Modul wird in MULTICAL® installiert und zur Datenübertragung aus MULTICAL® Wärme-, Kälte- und Wasserzählern auf ein Modbus-System verwendet.

Funktionalität

Das Modbus-Modul kommuniziert als ein RTU*-Slave-Gerät auf RS-485.

Das Modbus-Modul überträgt eine Reihe von sowohl aktuellen als auch akkumulierten Daten.

Darüber hinaus können MULTICAL®-Infocodes für allgemeine Alarne, Durchflussfehler, Temperaturfehler, Wasserlecks, Rohrbrüche, Luft im System und falsche Durchflussrichtung auf das Modbus-System übertragen werden.

Die beiden Impulseingänge ermöglichen den Anschluss und die Auslesung von zwei zusätzlichen Zählern für z.B. Wasser und Strom mit Impulsausgang.

Betriebszuverlässigkeit

Der RS-485-Port des Modbus-Moduls ist vom Spannungspotenzial des Zählers galvanisch getrennt, was einen reibungslosen Betrieb sichert. Gleichzeitig ist die Gefahr einer Beeinflussung des Zählers durch Auswirkungen des RS-485-Ports auf ein Minimum reduziert.

Das Modbus-Modul ist in voller Übereinstimmung mit und in der MID-Zulassung für MULTICAL® enthalten.

Adressenbereich

Das Modul kann als einen Slave im Bereich 1-247 adressiert werden.

Als Standard entspricht die Modbus-Adresse den letzten drei Ziffern der Kundennummer.

Wenn die Kundennummer des Zählers eine Adresse ergibt, die größer als 247 ist, werden nur die beiden letzten Ziffern für die Modbus-Adresse des Moduls verwendet.

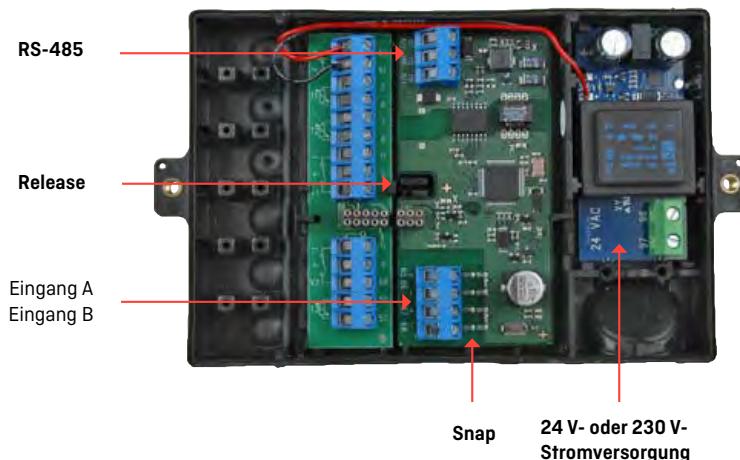
Zur Beachtung: Wenn die Kundennummer des Zählers mit 000 endet, wechselt das Modbus-Modul automatisch auf die Adresse 247.

*] RTU: Remote Terminal Unit

Konfiguration

Dateninhalt, Adresse, Baudrate und Parität des Modbus-Moduls kann über die optische Schnittstelle des Zählers und das PC-Programm MULTICAL® Module Programmer geändert werden.

Anschluss



Das Modul ist im Anschlussboden des Zählers wie folgt zu montieren:

Das Modul ist in den "Snap" schräg zu platzieren und dann in Richtung Anschlussboden zu drücken, bis das "Release" um der Hauptplatine des Zählers klickt.

Das Modul ist wie folgt zu entfernen:

Das Modul wird durch Drücken von "Release" nach oben und gleichzeitiges Ziehen der Anschlussklemmen oben auf dem Modul ausgelöst.

Elektrischer Anschluss:

Das Modbus-Modul wird automatisch mit dem Zähler über den 6-poligen Stecker direkt unter "Release" verbunden, wenn der Oberteil des Zählers montiert ist.

Das RS-485-Kabel wird über die Kabeleinführungen auf dem Anschlussboden angeschlossen.

Übertragene Daten

Das Modbus-Modul unterstützt zwei verschiedene Datagramme; das Standard-Datagramm und das alternative Datagramm.

Standard-Datagramm (alle Software-Versionen)

- Kundennummer [konfigurierbar]
- Seriennummer
- Akkumulierte Wärmeenergie [E1]
- Akkumulierte Kälteenergie [E3]
- Akkumulierter Volumendurchfluss [V1]
- Akkumulierter Volumendurchfluss [V2]
- Vorlauftemperatur [T1]
- Rücklauftemperatur [T2]
- Temperatur [T3]
- Aktueller Durchfluss [V1]
- Aktuelle Leistung
- Maximale Leistung
- Akkumulierte Werte aus weiteren Zählern über Impulseingang A und Impulseingang B
- MULTICAL®-Infocodes
- Programmnummer
- Konfig. Nummer 1 [DDDEE]
- Konfig. Nummer 2 [FFGGMN]
- Stundenzähler
- Zählertyp
- Tarif TA2
- Tarif TL2
- Tarif TA3
- Tarif TL3

Alternatives Datagramm (ab Software-Version C1)

- Kundennummer [konfigurierbar]
- Seriennummer
- Akkumulierte Wärmeenergie [E1]
- Regelenergie [E2]
- Akkumulierte Kälteenergie [E3]
- Aktueller Durchfluss Vorlauf [V1]
- Aktueller Durchfluss Rücklauf [V1]
- Akkumulierter Volumendurchfluss [V1]
- Akkumulierter Volumendurchfluss [V2]
- Vorlauftemperatur [T1]
- Rücklauftemperatur [T2]
- Temperatur [T3]
- Akkumulierte Werte aus zusätzlichen Zählern über Impulseingang A und Impulseingang B
- Masse [V1]
- Masse [V2]
- Aktuelle Leistung
- Maximale Leistung
- Programm-Version
- MULTICAL®-Infocodes
- Zählertyp
- Zählernummer
- Programmnummer
- Konfig. Nummer 1 [DDDEE]
- Konfig. Nummer 2 [FFGGMN]
- Stundenzähler
- Mediumcode [nur 6M2]

Technische Daten

Modbus-Funktionen

- Kommuniziert mit dem Bus mittels RS-485 [Standard 19200, 8, E, 1]
- Unterstützte Baudaten: 300, 2400, 9600, 19200, 38400, 76800
- Unterstützte Paritätseinstellungen: keine, gerade und ungerade
- Unterstützte Stoppbit-Einstellungen: ein und zwei
- Unterstützte Datenbits: 8
- Das Modul kann als einen Slave im Bereich 1-247 adressiert werden.
- Dateninhalt, Baudrate, Parität und Adresse werden in MULTICAL® gespeichert und können über einen optischen Lesekopf und das PC-Programm MULTICAL® Module Programmer geändert werden.
- Unterstützt RTU-Übertragungsmodus
- Unterstützte Funktionscodes und ihre möglichen Ausnahmecodes:
 - 0x03 Read Holding Registers with exception codes:
 - 0x02 – Illegal data address
 - 0x03 – Illegal data value
 - 0x04 Read Input Registers with exception codes:
 - 0x02 – Illegal data address
 - 0x03 – Illegal data value
 - 0x08 Diagnostics with exception code:
 - 0x01 – Illegal function
 - Subcode 0x01 Restart with exception code:
 - 0x03 – Illegal data value
 - 0x2B Encapsulated interface transport with exception code:
 - 0x01 – Illegal function
 - Subcode 0x0E Read Device Identification with exception code:
 - 0x03 – Illegal data value
 - 0x41 und 0x42 Reserviert als spezifischer Funktionscode.

Modbus-Datenmodellmapping von MULTICAL®-Werten

Der PDU-Adressenbereich von 0 bis 168 wird nachfolgend näher beschrieben. Die PDU-Adresse ist in Bytes angegeben, sodass ein 32-Bit Register ergeben wird, dass das folgende Register auf einer Adresse 4 Stellen höher platziert werden soll. Die Daten sind in 6 verschiedene Tabellen gegliedert. Die Daten in den jeweiligen Tabellen sind alle von der gleichen Größe und sollten in gleicher Weise ausgelegt werden. Eine genauere Beschreibung der einzelnen Spalten werden später gegeben.

Datenmodellmapping für Byte-adressierten Bereich, Standard-Datagramm

Memory	Memory (hex)	Individual description	Size in bytes	Table	Contents	Data type	Update status
0	0x0000	Heat energy E1	4	1	Values in float	IEEE Float - 32 bit	Dynamic
4	0x0004	Actual flow	4	1	Values in float	IEEE Float - 32 bit	Dynamic
8	0x0008	Volume V1	4	1	Values in float	IEEE Float - 32 bit	Dynamic
12	0x000C	Actual power	4	1	Values in float	IEEE Float - 32 bit	Dynamic
16	0x0010	Inlet temperature T1	4	1	Values in float	IEEE Float - 32 bit	Dynamic
20	0x0014	Outlet temperature T2	4	1	Values in float	IEEE Float - 32 bit	Dynamic
24	0x0018	Pulse input A	4	1	Values in float	IEEE Float - 32 bit	Dynamic
28	0x001C	Pulse input B	4	1	Values in float	IEEE Float - 32 bit	Dynamic
32	0x0020	Heat energy E1	2	2	Units	Word - 16 bit	Dynamic
34	0x0022	Actual flow	2	2	Units	Word - 16 bit	Dynamic
36	0x0024	Volume V1	2	2	Units	Word - 16 bit	Dynamic
38	0x0026	Actual power	2	2	Units	Word - 16 bit	Dynamic
40	0x0028	Heat energy E1	4	3	Values in integer	Double Word - 32 bit	Dynamic
44	0x002C	Actual flow	4	3	Values in integer	Double Word - 32 bit	Dynamic
48	0x0030	Volume V1	4	3	Values in integer	Double Word - 32 bit	Dynamic
52	0x0034	Actual power	4	3	Values in integer	Double Word - 32 bit	Dynamic
56	0x0038	Inlet temperature T1	4	3	Values in integer	Signed Double Word - 32 bit	Dynamic
60	0x003C	Outlet temperature T2	4	3	Values in integer	Signed Double Word - 32 bit	Dynamic
64	0x0040	Pulse input A	4	3	Values in integer	Double Word - 32 bit	Dynamic
68	0x0044	Pulse input B	4	3	Values in integer	Double Word - 32 bit	Dynamic
72	0x0048	Heat energy E1	2	4	Decimal	Word - 16 bit	Dynamic
74	0x004A	Actual flow	2	4	Decimal	Word - 16 bit	Dynamic
76	0x004C	Volume V1	2	4	Decimal	Word - 16 bit	Dynamic
78	0x004E	Actual power	2	4	Decimal	Word - 16 bit	Dynamic
80	0x0050	Pulse input A	2	4	Decimal	Word - 16 bit	Dynamic
82	0x0052	Pulse input B	2	4	Decimal	Word - 16 bit	Dynamic
84	0x0054	Version	2	5	Program version	Word - 16 bit	Static
86	0x0056	Info code	2	6	Info code	Word - 16 bit	Dynamic
88	0x0058	Reserved	4	N/A		IEEE Float - 32 bit	
92	0x005C	Cooling energy E3	4	8	Values in float	IEEE Float - 32 bit	Dynamic
96	0x0060	Volume - V2	4	8	Values in float	IEEE Float - 32 bit	Dynamic
100	0x0064	Temperature T3	4	8	Values in float	IEEE Float - 32 bit	Dynamic
104	0x0068	Cooling energy E3	2	9	Units	Word - 16 bit	Dynamic

Modbus-Datenmodellmapping von MULTICAL®-Werten

Memory	Memory (hex)	Individual description	Size in bytes	Table	Contents	Data type	Update status
106	0x006A	Volume - V2	2	9	Units	Word - 16 bit	Dynamic
108	0x006C	Cooling energy E3	4	10	Values in integer	Double Word - 32 bit	Dynamic
112	0x0070	Volume - V2	4	10	Values in integer	Double Word - 32 bit	Dynamic
116	0x0074	Temperature T3	4	10	Values in integer	Signed Double Word - 32 bit	Dynamic
120	0x0078	Cooling energy E3	2	11	Decimal	Word - 16 bit	Dynamic
122	0x007A	Volume - V2	2	11	Decimal	Word - 16 bit	Dynamic
124	0x007C	Max power	4	12	Values in float	IEEE Float - 32 bit	Dynamic
128	0x0080	Tarif 2	4	12	Values in float	IEEE Float - 32 bit	Dynamic
132	0x0084	Tarif 3	4	12	Values in float	IEEE Float - 32 bit	Dynamic
136	0x0088	Tarif limit 2	4	12	Values in float	IEEE Float - 32 bit	Static
140	0x008C	Tarif limit 3	4	12	Values in float	IEEE Float - 32 bit	Static
144	0x0090	Meter type	4	13	Parameters	Double Word - 32 bit	Static
148	0x0094	Meter number 1	4	13	Parameters	Double Word - 32 bit	Static
152	0x0098	Serial number	4	13	Parameters	Double Word - 32 bit	Static
156	0x009C	Program number	4	13	Parameters	Double Word - 32 bit	Static
160	0x00A0	Config number 1	4	13	Parameters	Double Word - 32 bit	Static
164	0x00A4	Config Number 2	4	13	Parameters	Double Word - 32 bit	Static
168	0x00A8	Hour counter	4	13	Parameters	Double Word - 32 bit	Dynamic

C.2 THERMASGARD RTM1-Modbus

Raumtemperaturfühler von S+S Regeltechnik

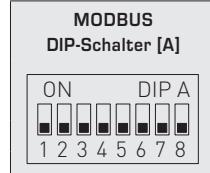
(D) THERMASGARD® RTM 1-Modbus | Konfiguration

Rev. 2016-V12 DE intro

BUSADRESSE

Busadresse (binärcodiert, Wertigkeit 1 bis 247 einstellbar)							
DIP 1	DIP 2	DIP 3	DIP 4	DIP 5	DIP 6	DIP 7	DIP 8
128	64	32	16	8	4	2	1
ON	ON	OFF	OFF	OFF	OFF	OFF	ON

Beispiel zeigt $128 + 64 + 1 = 193$ als Modbus-Adresse.



Die **Geräteadresse** im Bereich von **1 bis 247** (Binärformat) wird über den DIP-Schalter [A] eingestellt.
Schalterstellung Pos. 1 bis 8 – siehe Tabelle auf Rückseite!

Die Adresse 0 ist für Broadcast-Meldungen reserviert, die Adressen größer 247 dürfen nicht belegt werden und werden vom Gerät ignoriert. Die DIP-Schalter sind binärcodiert mit folgender Wertigkeit:

DIP 1 = **128** DIP 1 = **ON**

DIP 2 = **64** DIP 2 = **ON**

DIP 3 = **32** DIP 3 = **OFF**

DIP 4 = **16** DIP 4 = **OFF**

DIP 5 = **8** DIP 5 = **OFF**

DIP 6 = **4** DIP 6 = **OFF**

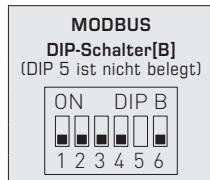
DIP 7 = **2** DIP 7 = **OFF**

DIP 8 = **1** DIP 8 = **ON**

folgt die Modbus-Adresse **128 + 64 + 1 = 193**

BUSPARAMETER

Baudrate (einstellbar)	DIP 1	DIP 2
9600 Baud	ON	OFF
19200 Baud	ON	ON
38400 Baud	OFF	ON
reserviert	OFF	OFF



Parity (einstellbar)	DIP 3
EVEN (gerade)	ON
ODD (ungerade)	OFF

Parity-Sicherung (mit/ohne einstellbar)	DIP 4
aktiv (1 Stopbit)	ON
inaktiv (2 Stopbits)	OFF

Busabschluss (mit/ohne einstellbar)	DIP 6
aktiv	ON
inaktiv	OFF

Die **Baudrate** (Übertragungsgeschwindigkeit) wird über Pos. 1 und 2 des DIP-Schalters [B] eingestellt.
Einstellbar sind **9600 Baud**, **19200 Baud** oder **38400 Baud** – siehe Tabelle!

Die **Parity** wird über Pos. 3 des DIP-Schalters [B] eingestellt.

Einstellbar sind **EVEN (gerade)** oder **ODD (ungerade)** – siehe Tabelle!

Die **Parity-Sicherung** wird über Pos. 4 des DIP-Schalters [B] aktiviert.

Einstellbar ist Parity-Sicherung **aktiv (1 Stopbit)** oder **inaktiv (2 Stopbits)**, d.h. keine Parity-Sicherung – siehe Tabelle!

Pos. 5 des DIP-Schalters [B] ist nicht belegt!

Der **Busabschluss** wird über Pos. 6 des DIP-Schalters [B] aktiviert.

Einstellbar ist **aktiv** (Busabschlusswiderstand von 120 Ohm) oder **inaktiv** (ohne Busabschluss) – siehe Tabelle!

Bei Änderung der Busparameter und Busadresse werden bei Geräten mit **Displayanzeige** die entsprechenden Einstellungen im Display für ca. 30 Sekunden angezeigt.

KOMMUNIKATIONSANZEIGE

Die Kommunikation wird über 2 LED-Anzeigen signalisiert. Fehlerfrei empfangene Telegramme werden unabhängig von der Geräteadresse durch Aufleuchten der grünen Anzeige signalisiert. Fehlerhafte Telegramme oder ausgelöste Modbus Exception-Telegramme werden durch das Aufleuchten der roten Anzeige dargestellt.

DIAGNOSE

Fehlerdiagnosefunktion mitintegriert

 **THERMASGARD® RTM 1-Modbus** | Telegramme

Function 04 Read Input Register

Register	Parameter		Data Type	Value	Range
3x0001	Temperatur	4 s Abtastung	Signed 16 Bit	-350...+800	-35.0 ... +80.0 °C
3x0002	Temperatur	Filterung	Signed 16 Bit	-350...+800	-35.0 ... +80.0 °C
3x0003	Sollwert Potentiometer		Signed 16 Bit	0...1000	0...100.0 %
3x0004	Taupunkt	Berechneter Wert	Signed 16 Bit	0... 500	0...+50.0 °C
3x0005	Relative Feuchte	Filterung	Signed 16 Bit	0...1000	0...100.0 % r.H.

Function 08 Diagnostics

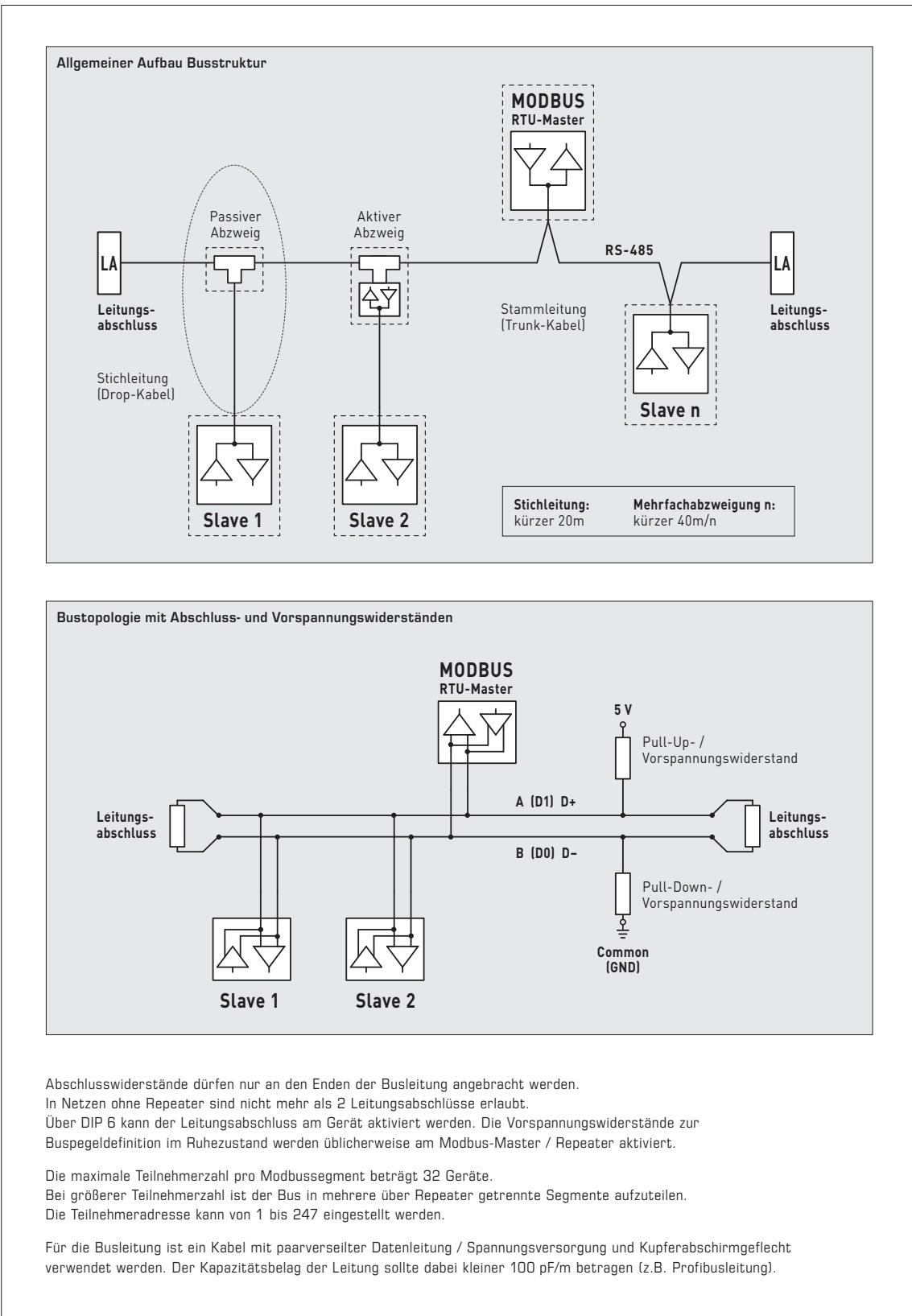
Folgende Sub Function Codes werden unterstützt

Sub Function Code	Parameter	Data Type	Antwort
00	Echo der Sendedaten (Loopback)		Echodata
01	Neustart Modbus (Reset Listen Only Mode)		Echo Telegramm
04	Aktivierung Listen Only Mode		Keine Antwort
10	Lösche Zähler		Echo Telegramm
11	Zähler Bustelegramme	Unsigned 16 Bit	alle gültigen Bustelegramme
12	Zähler Kommunikationsfehler (Parity, CRC, Framefehler, etc.)	Unsigned 16 Bit	fehlerhafte Bustelegramme
13	Zähler Exception-Meldungen	Unsigned 16 Bit	Fehlerzähler
14	Zähler Slave-Telegramme	Unsigned 16 Bit	Slave-Telegramme
15	Zähler Telegramme ohne Antwort	Unsigned 16 Bit	Broadcastmeldungen (Adresse 0)

Function 17 Report Slave ID

Aufbau Antworttelegramm

Byte Nr.	Parameter	Data Type	Antwort
00	Byteanzahl	Unsigned 8 Bit	6
01	Slave ID (Device Typ)	Unsigned 8 Bit	4 = THERMASGARD® MODBUS
02	Slave ID (Device Class)	Unsigned 8 Bit	10 = THERMASGARD® / THERMASREG®
03	Status	Unsigned 8 Bit	255 = RUN, 0 = STOP
04	Versionsnummer (Release)	Unsigned 8 Bit	1...9
05	Versionsnummer (Version)	Unsigned 8 Bit	1...99
06	Versionsnummer (Index)	Unsigned 8 Bit	1



Literaturverzeichnis

[pys] *Pysolar by pingswept*

[osi 1996] INTERNATIONAL ORGANIZATION FOR STANDARDIZATION: ISO 7498-1: Information technology – Open Systems Interconnection – Basic Reference Model: The basic model. Genf, Switzerland, März 1996. – ISO 7498-1 Standard

[mod 2006a] MODBUS ORGANIZATION: Modbus messaging on TCP/IP implementation guide v1.0b. Hopkinton, USA, Oktober 2006. – Forschungsbericht

[mod 2006b] MODBUS ORGANIZATION: Modbus over serial line specification and implementation guide V1.02. Hopkinton, USA, Dezember 2006. – Forschungsbericht

[mod 2012] MODBUS ORGANIZATION: Modbus application protocol specification v1.1b3. Hopkinton, USA, April 2012. – Forschungsbericht

[AB 2015] AB, Modelon (Hrsg.): *JModelica.org User Guide - Version 1.17*. Ideon Science Park, SE-223 70 Lund : Modelon AB, 2015

[Baehr u. Kabelac 2012] BAEHR, Hans ; KABELAC, Stephan: *Thermodynamik: Grundlagen und technische Anwendungen*. 15. Auflage 2012. Berlin, Heidelberg : Springer Berlin Heidelberg, 2012

[Bertrand Blanc 2005] BERTRAND BLANC, Bob M.: *Endianness or Where is Byte 0?* <http://3bc.bertrand-blanc.com/endianess05.pdf>, December 2005

[Böckh u. Wetzel 2014] BÖCKH, Peter v. ; WETZEL, Thomas: *Wärmeübertragung: Grundlagen und Praxis*. 5. überarbeitete und erweiterte Auflage 2014. Berlin, Heidelberg : Springer Vieweg, 2014

[Furrer 2003] FURRER, Frank J.: *Industrieautomation mit Ethernet-TCP/IP und Web-Technologie*. 3. neu bearbeitete und erweiterte Auflage. Heidelberg: Hüthig, 2003

[Hauser 2000] HAUSER, Höttges K G.: Bauphysik in Kürze: U-Werte von Fenstern. In: *Bauphysik* 22 (2000), S. S.270–273

[Joel Andersson 2015] JOEL ANDERSSON, Moritz D. Joris Gillis G. Joris Gillis: *User Documentation for CasADI v2.2.0+1.cf4be18*. Joel Andersson, Joris Gillis, Moritz Diehl, 2015

[Kaltschmitt 2013] KALTSCHMITT, Wolfgang ;Wiese A. Martin ;Streicher ;. Martin ;Streicher: *Erneuerbare Energien : Systemtechnik, Wirtschaftlichkeit, Umweltaspekte*. 5. erweiterte Auflage. Springer Berlin Heidelberg, 2013

[MODICON 96] MODICON, Inc.: *Modicon Modbus Protocol Reference Guide*. One High Street, North Andover, Massachusetts 01845: MODICON, Inc., Industrial Automation Systems, June 96

[Nicolai 2013] NICOLAI, Dr. A.: Physikalische Grundlagen des thermischen Raummodells THE-RAKLES / Technische Universität Dresden, Fakultät Architektur, Institut für Bauklimatik. 2013. – Forschungsbericht

- [Peter Häupl 2013] PETER HÄUPL, Christian Kölzow Olaf Riese Anton Maas Gerrit Höfker Christian N. Martin Homann ; WILLEMS, Wolfgang (Hrsg.): *Lehrbuch der Bauphysik : Schall - Wärme - Feuchte - Licht - Brand - Klima.* Springer Vieweg, 2013
- [Quaschning 2011] QUASCHNING, Volker: *Regenerative Energiesysteme : Technologie, Berechnung, Simulation.* Bd. 7. aktualisierte Auflage. München : Hanser, 2011
- [Recknagel 2013] RECKNAGEL, Schramek Sprenger: *(Taschenbuch für Heizung + Klimatechnik) : 76.2013/14.* Oldenbourg Industrieverlag, 2013
- [Reda 2008] REDA, A. I. ; A. I. ; Andreas: Solar Position Algorithm for Solar Radiation Applications (Revised) / National Renewable Energy Laboratory (NREL), Golden, CO. 2008. – Forschungsbericht
- [van Rossum u. the Python development team 2016] ROSSUM, Guido van ; TEAM the Python d.: *Python Frequently Asked Questions.* Release 2.7.11. Python Software Foundation, 2016
- [Sack 2004] SACK, Dipl.-Phys. N.: Von k zu U - Was ändert sich bei Fensterrahmen und -profilen? / ift Rosenheim Bauphysik. 2004. – Forschungsbericht
- [Schleicher 2008] SCHLEICHER, Manfred: *Digitale Schnittstellen und Bussysteme.* JUMO, 2008
- [Schnell u. Wiedemann 2006] SCHNELL, Gerhard ; WIEDEMANN, Bernhard: *Bussysteme in der Automatisierungs- und Prozesstechnik : Grundlagen, Systeme und Trends der industriellen Kommunikation.* 6. überarbeitete und aktualisierte Auflage. Wiesbaden : Vieweg+Teubner, 2006

Eidesstattliche Erklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und nur unter Benutzung der angegebenen Quellen und Hilfsmittel angefertigt habe. Alle Textstellen, die wörtlich oder sinngemäß aus veröffentlichten oder nicht veröffentlichten Quellen entnommen wurden, sind als solche kenntlich gemacht. Die Arbeit hat in gleicher oder ähnlicher Form keiner anderen Prüfungsbehörde vorgelegen.

Kandel, den 25. März 2016

Daniel Johannes Mayer