



# Control Unit

*Technische Dokumentation*

**Autor:** Dipl.-Ing. Holger Heuser

**Revision:** 1.0 (1)

**Zuletzt geändert:** 25. März 2008

# Inhaltsverzeichnis

Abbildungsverzeichnis.....	4
Tabellenverzeichnis.....	5
1 Einleitung.....	6
2 Hardware.....	8
2.1 Schaltung.....	8
2.2 Pin-Konfiguration.....	11
2.3 Dimensionen und Platinenlayout.....	15
3 Steuerung.....	18
3.1 Kommunikation.....	18
3.1.1 Virtuelle LANs.....	19
3.1.2 Modul-Adressierung.....	20
3.1.3 Magic Packet.....	21
3.1.4 Central Control Unit Adresse.....	21
3.1.5 Bridge-Mode.....	22
3.1.6 Empfangspuffer.....	22
3.1.7 Steuerbefehle.....	22
3.1.8 Abfragen.....	23
3.1.9 Statusmeldungen.....	23
3.1.10 Makros.....	24
3.1.11 Fehlerbehandlung.....	24
3.2 Resets und Neustarts.....	25
3.3 Konfigurationsmanagement.....	26
3.4 Externe Beschaltung.....	27
3.5 Zeitsynchronisation.....	29
3.6 Software-Management.....	30
4 Komponenten.....	34
4.1 EEPROM-Support.....	34
4.2 Externer Reset.....	35
4.3 Buzzer.....	35
4.4 Funkmodul.....	37
4.5 CAN-Bus.....	39
4.6 Infrarotschnittstelle.....	39
4.6.1 Konfiguration der Fernbedienung.....	40
4.6.2 Infrarot-Adressen und Hot-Keys.....	42
4.6.3 Bedienung.....	43

4.7	LCD-Display.....	45
4.8	Logische Eingänge.....	46
4.9	Analoge Eingänge.....	48
4.10	Digitale Eingänge.....	51
4.10.1	Dallas Digitales Thermometer.....	54
4.11	Schalter.....	56
4.12	Dimmer.....	57
4.13	Abstraktes Modul.....	60
4.14	Rollladensteuerung.....	61
4.15	Drehgeber.....	63
4.16	Bedienoberfläche.....	65
4.16.1	Menü.....	67
4.16.2	Steuerung.....	68
4.16.3	Thermostat.....	70
4.16.4	Konfigurationsdownload.....	72
4.17	Autonome Steuerung.....	73
5	Tabellen und Übersichten.....	93
5.1	Klemm- und Steckverbinder.....	93
5.2	Befehlsreferenz.....	94
5.3	Steuercodes.....	96
5.4	IR-Bediencodes.....	97
5.5	GUI-Konfiguration.....	98
5.6	Objekttypen der autonomen Steuerung.....	99
	Literaturverzeichnis.....	103

## Abbildungsverzeichnis

Abb. 2.1: Control Unit in der Draufsicht.....	8
Abb. 2.2: Schaltplan der Control Unit.....	9
Abb. 2.3: Abmessungen der Control Unit.....	15
Abb. 2.4: Platinenlayout der Control Unit Oberseite.....	16
Abb. 2.5: Platinenlayout der Control Unit Unterseite.....	17
Abb. 3.1: Aufbau eines Datenpakets.....	18
Abb. 4.1: Allgemeines Werteformat bei analogen Eingängen.....	50
Abb. 4.2: Konkretes Werteformat bei analogen Eingängen.....	51
Abb. 4.3: Allgemeines Werteformat bei digitalen Eingängen.....	53
Abb. 4.4: Werteformat beim digitalen Thermometer (DS18B20).....	55
Abb. 4.5: Werteformat beim digitalen Thermometer (DS18S20).....	56
Abb. 4.6: Steuerelement Menü.....	67
Abb. 4.7: Steuerelement Steuerung 1.....	68
Abb. 4.8: Steuerelement Steuerung 2.....	68
Abb. 4.9: Steuerelement Steuerung 3.....	69
Abb. 4.10: Steuerelement Thermostat 1.....	70
Abb. 4.11: Steuerelement Thermostat 2.....	70
Abb. 4.12: Steuerelement Thermostat 3.....	71

## Tabellenverzeichnis

Tabelle 2.1: Belegung der CAN Klemmleiste.....	11
Tabelle 2.2: Belegung der 3-pol. Platinensteckverbinder.....	12
Tabelle 2.3: Belegung der Stiftleiste JP2.....	12
Tabelle 2.4: Belegung der Stiftleiste JP6.....	13
Tabelle 2.5: Belegung der Stiftleiste JP7.....	14
Tabelle 3.1: Konfiguration der Geräte-Typen.....	29
Tabelle 3.2: Softwarekomponenten.....	31
Tabelle 4.1: Signalarten.....	35
Tabelle 4.2: Buzzerlevel.....	36
Tabelle 4.3: IR-Kommandos.....	41
Tabelle 4.4: Aktivierungs-codes von logischen Eingängen.....	47
Tabelle 4.5: Voreingestellte Aktivierungszeiten bei logischen Eingängen.....	48
Tabelle 4.6: Ereigniscodierung bei analogen Eingängen.....	50
Tabelle 4.7: Typen digitaler Sensoren.....	52
Tabelle 4.8: Ereigniscodierung bei digitalen Eingängen.....	54
Tabelle 4.9: Temperaturcodierung beim digitalen Thermometer (DS18B20).....	55
Tabelle 4.10: Temperaturcodierung beim digitalen Thermometer (DS18S20).....	56
Tabelle 4.11: Typen abstrakter Objekte.....	61
Tabelle 4.12: AS Codierung der Zeitbasis bei der Ausschaltverzögerung.....	82
Tabelle 4.13: AS Codierung der Zeitbasis bei der Einschaltverzögerung.....	82
Tabelle 4.14: AS Codierung der Zeitbasis bei der Einschaltbegrenzung.....	83
Tabelle 4.15: AS Codierung der Zeitbasis bei der zeitgesteuerten Weiterleitung.....	84
Tabelle 5.1: Pin-Belegung der Klemm- und Steckverbinder.....	93
Tabelle 5.2: Befehlsreferenz.....	96
Tabelle 5.3: Steuer-codes.....	96
Tabelle 5.4: IR-Bediencodes.....	97
Tabelle 5.5: Objekttypen der autonomen Steuerung.....	102

# 1 Einleitung

Die Control Unit stellt innerhalb des HAP (Home Automation Project) die zentrale Komponente dar. Sie nimmt unterschiedlichste Steuer- und Regelaufgaben wahr und arbeitet dabei völlig autonom. So ist es mit Hilfe der Control Unit möglich, Leuchten zu steuern bzw. zu dimmen, alle möglichen Arten von Geräten zu steuern und zu schalten, sowie Messwerte von Sensoren zu erfassen, zu verarbeiten oder weiterzuleiten. Sie bietet eine Infrarot-Schnittstelle zu ihrer Steuerung an und kann mit Hilfe eines LCD (Liquid Crystal Display) und entsprechenden Eingabekomponenten eine Mensch-Maschine-Schnittstelle realisieren. So lassen sich prinzipiell alle im Haus bzw. in der Wohnung anfallenden Steuerungen und Regelungen realisieren.

Da die Zahl der anschließbaren Geräte durch die Anzahl an Ein- und Ausgängen des verwendeten Mikrocontrollers begrenzt ist, werden für eine Heimsteuerung gewöhnlich mehrere Control Units benötigt. Damit diese untereinander Informationen über den Status der jeweils angeschlossenen Geräte und weitere Nachrichten, die zum gemeinsamen Bewältigen einer komplexen Steueraufgabe nötig sind, austauschen können, besitzt eine Control Unit mehrere Kommunikationsschnittstellen bzw. kann mit diesen bestückt werden. So wird in den meisten Fällen auf einen CAN-Bus (Controller Area Network Bus) zurückgegriffen. Es ist aber auch die Kommunikation über Funk möglich.

Für Konfigurationsarbeiten, zur Verwaltung der kompletten Steuerungsanlage, zum Fehlermanagement und zur Steuerung von Geräten über Webbrowser, PDAs, Handys etc. wird eine Central Control Unit eingesetzt, bei der es sich um ein Serversystem mit spezieller Steuerungssoftware handelt. Damit eine Control Unit mit der Central Control Unit kommunizieren kann, wird eine speziell konfigurierte Control Unit eingesetzt, die als Brücke zwischen dem von den Control Units verwendeten Kommunikationssystem und der seriellen oder USB-Schnittstelle der Central Control Unit arbeitet.

Im Allgemeinen wird man die Control Units in Hutschienengehäusen in der Elektroverteilung unterbringen. Dies ist insbesondere dann sinnvoll, wenn zur Ansteuerung von Geräten Leistungsstufen wie Relais oder Dimmerstufen benötigt werden. Die räumliche Nähe erleichtert dann die Verkabelungsarbeiten erheblich. Die

Bauform der Control Unit lässt aber auch die Unterbringung in Schalterdosen zu, was vor allem für die Realisierung einer Mensch-Maschine-Schnittstelle sinnvoll ist, da sich eine solche meist in den Wohnräumen befindet.

Welche Geräte wo und wie an die Control Unit angeschlossen werden können, ist bis auf wenige Restriktionen frei konfigurierbar. Des weiteren ist das Verhalten einer Control Unit in ihrer Umgebung durch komplexe Konfigurationsmöglichkeiten einstellbar. Insbesondere die Konfiguration einer Mensch-Maschine-Schnittstelle ist recht aufwändig. Zur besseren Handhabung und zur Vermeidung von Fehlern steht ein Softwarewerkzeug für diese Arbeiten zur Verfügung, welches die Konfiguration einer Control Unit recht komfortabel gestaltet und nicht nur eine Control Unit dabei berücksichtigt, sondern das System als Ganzes. Die von diesem Werkzeug erzeugten Konfigurationen werden dann in EEPROMs auf der Control Unit gespeichert, so dass diese prinzipiell autark, also ohne die Central Control Unit funktionieren können und auch nach einem Stromausfall problemlos wieder funktionieren.

Der Aufbau und die Funktionsweise der Hardware wird im nächsten Kapitel beschrieben (siehe Kapitel 2 Hardware). Es folgt die Beschreibung des grundlegenden Verhaltens einer Control Unit, ihr Kommunikationsverhalten sowie deren Handhabung hinsichtlich Konfiguration, Management und Wartung (siehe Kapitel 3 Steuerung). Die unterschiedlichen Software Komponenten, die auf einer Control Unit integriert sein können, werden im Kapitel 4 Komponenten erörtert. Schließlich finden sich in Kapitel 5 Tabellen und Übersichten noch einmal Zusammenfassungen aus vorigen Kapiteln bzw. sind einige Sachverhalte anders zusammengestellt, um einen schnelleren Überblick zu bekommen bzw. schnell gezielte Informationen zu finden.

## 2 Hardware

Unter einer Control Unit wird im Allgemeinen nur die fertig bestückte Platine verstanden. Ob diese in einem Hutschienengehäuse zum Einbau in der Elektroverteilung, in einem gewöhnlichen Gehäuse, als Aufsatz einer Leistungsstufe oder direkt in einer Schalterdose untergebracht wird, ist vom jeweiligen Einsatzzweck abhängig. Unten stehende Abbildung zeigt die Control Unit in Draufsicht.



Abb. 2.1: Control Unit in der Draufsicht

### 2.1 Schaltung

Der Schaltplan der Control Unit ist auf der nächsten Seite dargestellt. Auf Basis dieses Plans sollen die wichtigsten Komponenten der Unit beschrieben und im folgenden Kapitel auch die Pin-Konfiguration erläutert werden.



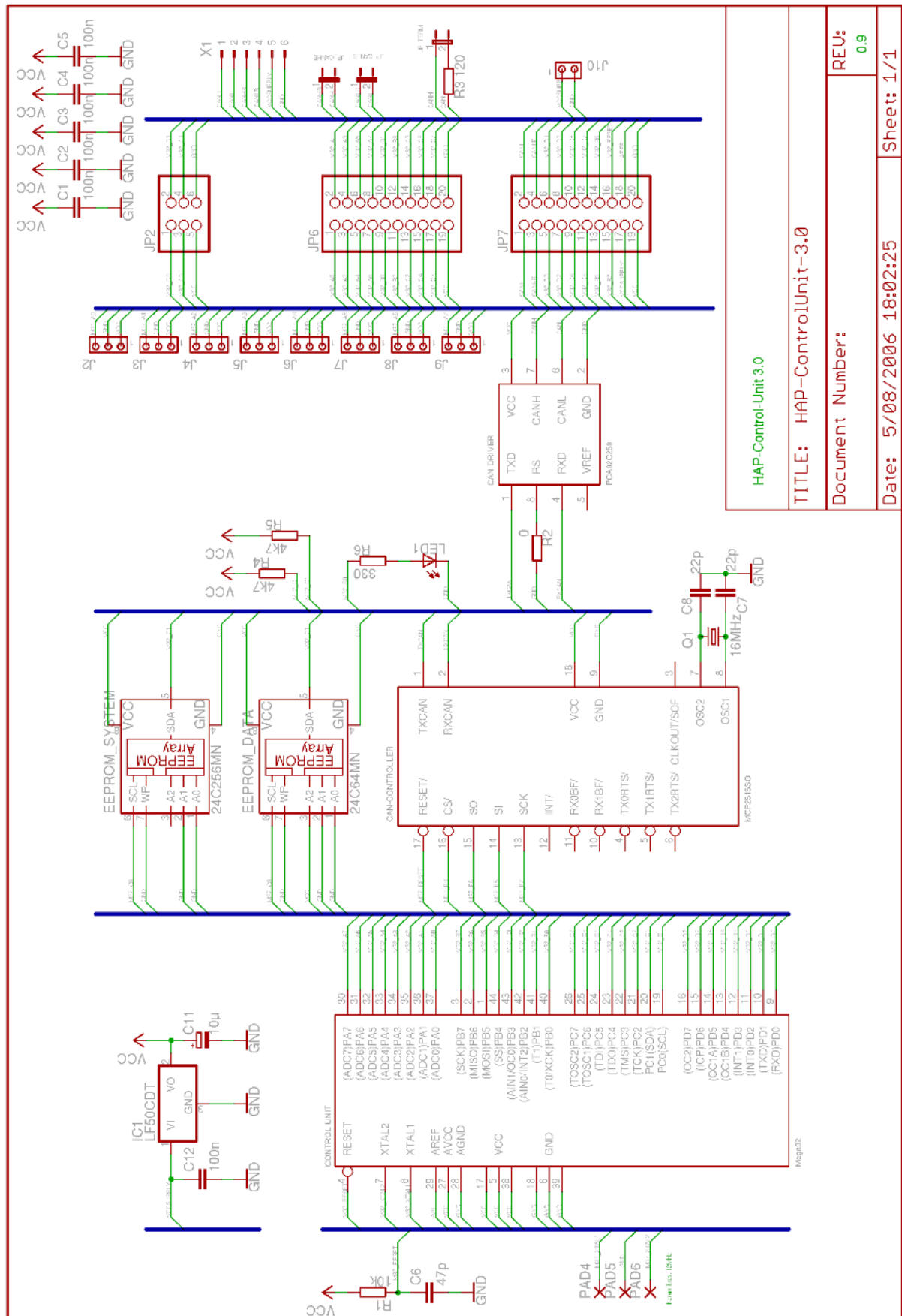


Abb. 2.2: Schaltplan der Control Unit

In den folgenden Ausführungen wird davon ausgegangen, dass der Schaltplan im Querformat vorliegt, die Schrift also korrekt zum Betrachter ausgerichtet ist. Zu den Details, die auf diesem Plan nicht mehr zu erkennen sind, sei auf den Originalplan verwiesen.

In der oberen linken Ecke des Plans erkennt man die Spannungsversorgung, die mit Hilfe eines Festspannungsreglers die außen anliegende 12 V Gleichspannung in eine intern verwendete 5 V Gleichspannung (VCC) wandelt und diese glättet.

Darunter findet man das Herzstück der Control Unit, den Mikrocontroller. Hier wird ein ATmega32 von ATMEL verwendet. Der Reset-Pin wird nicht verwendet und deshalb über R1 mit VCC verbunden. PAD4 bis PAD6 dienen der Aufnahme eines externen Quarz. Dieser muss eine Frequenz von 12 MHz aufweisen und legt damit die Taktfrequenz des Mikrocontroller auf eben diesen Wert fest.

In der Bildmitte oben sieht man zwei externe EEPROMs der Größe 64 KBit und 256 KBit, die über ein TWI (Two Wire Interface) mit dem Mikrocontroller verbunden sind. Neben dem im Mikrocontroller integrierten EEPROM dienen diese dem Speichern der Konfigurationsdaten der Control Unit und deren Firmware Update.

Rechts unterhalb der EEPROMs ist eine LED zu erkennen, die als optische Signaleinrichtung dient und mit Pin B0 des Mikrocontrollers verbunden ist. An den selben Pin kann auch ein Summer als akustische Unterstützung angeschlossen werden (siehe auch Kapitel 4.3 Buzzer).

Direkt unterhalb der EEPROMs findet man den CAN-Controller (Controller Area Network Controller), der zur Steuerung des CAN-Bus benötigt wird. Dieser bezieht seine Taktfrequenz von einem separaten 16 MHz Quarz (unten rechts) und ist über ein SPI (Serial Peripheral Interface) mit dem Mikrocontroller gekoppelt. Über den rechts daneben befindlichen CAN-Treiber wird dann direkt der CAN-Bus betrieben.

Die drei Jumper am rechten Bildrand stehen von oben nach unten für CANHBack, CANLBack und CANTerm. Ist CANHBack gesteckt, so wird der Anschluss CANH an der Schraubleiste X1 nach CANHBack geschliffen, ansonsten nicht. Für CANLBack gilt dies analog. Der Jumper CANTerm terminiert den CAN-Bus, das heißt, R3 wird zwischen CANH und CANL geschaltet.

## 2.2 Pin-Konfiguration

Die Control Unit besitzt unterschiedliche Steckverbinder und Klemmleisten, die alle für einen spezifischen Zweck zugeschnitten worden sind. Im Folgenden wird deren Belegung und Verwendung näher beschrieben (siehe als Referenz hierzu Abbildung 2.1 Control Unit in der Draufsicht, Kapitel 2.1 Schaltung und 2.3 Dimensionen und Platinenlayout).

Die Klemmleiste X1 dient dem Anschluss an den CAN-Bus. Von links nach rechts ergibt sich die Belegung der Klemmleiste wie folgt.

Klemme	Beschreibung
1	Ground
2	Versorgungsspannung (+12 V)
3	CAN Low Back
4	CAN High Back
5	CAN Low
6	CAN High

*Tabelle 2.1: Belegung der CAN Klemmleiste*

Wird die Control Unit über den CAN-Bus mit Strom versorgt, so ist die Versorgungsspannung an den Klemmen 1 und 2 anzulegen. Soll die Control Unit über eine Stichleitung an den CAN-Bus angeschlossen werden, so sind die Adern CAN Low und CAN High an Klemme 5 und 6 anzulegen. Soll der CAN-Bus über die Control Unit geschliffen werden, so sind die entsprechenden Jumper zu setzen (siehe Kapitel 2.1 Schaltung) und über die Klemmen 3 und 4 ist der Bus wieder abzugreifen.

Alternativ zur Spannungsversorgung über den CAN-Bus kann die Versorgungsspannung auch über den 2-pol. Platinensteckverbinder J10 zugeführt werden (außen liegender Pin ist +12 V, innen liegender Pin ist Ground). Wird die Control Unit aber über den CAN-Bus versorgt, so kann die Versorgungsspannung über J10 für andere Zwecke abgegriffen werden.

Insbesondere, weil der Anschluss analoger und digitaler Sensoren auf Port A des Mikrocontrollers beschränkt ist (siehe Kapitel 4.9 Analoge Eingänge und 4.10 Digitale Eingänge), ist jeder Pin von Port A über einen eigenen Platinensteckverbinder, an

dem zusätzlich auch Ground und VCC anliegt, nach außen geführt. Folgende Tabelle zeigt die Stift-Belegung von unten nach oben und von rechts nach links.

<b>Steckverb.</b>	<b>Links</b>	<b>Mitte</b>	<b>Rechts</b>
J2	Pin A0	Ground	VCC
J3	Pin A1	Ground	VCC
J4	Pin A2	Ground	VCC
J5	Pin A3	Ground	VCC
J6	Pin A4	Ground	VCC
J7	Pin A5	Ground	VCC
J8	Pin A6	Ground	VCC
J9	Pin A7	Ground	VCC

*Tabelle 2.2: Belegung der 3-pol. Platinensteckverbinder*

Die restlichen Pins des Mikrocontrollers sind über die drei Stiftleisten nach außen geführt. Über passende Pfostenstecker kann dann ein Flachbandkabel angeschlossen werden. Die Stift-Belegung der 6-pol. Leiste JP2, die am oberen Bildrand zu erkennen ist, kann folgender Tabelle entnommen werden. Die Stifte werden von unten nach oben und von links nach rechts gezählt.

<b>Stift</b>	<b>Beschreibung</b>
1	Pin D0
2	Pin D1
3	Pin C0
4	Pin C1
5	VCC
6	Ground

*Tabelle 2.3: Belegung der Stiftleiste JP2*

Die 20-pol. Leiste JP6 befindet sich in der Bildmitte. Es wird von rechts nach links und von unten nach oben nummeriert.

<b>Stift</b>	<b>Beschreibung</b>
1	Pin A0
2	Pin A1
3	Pin A2
4	Pin A3
5	Pin A4
6	Pin A5
7	Pin A6
8	Pin A7
9	Pin B0
10	Pin B1
11	Pin B2
12	Pin B3
13	Pin C2
14	Pin C3
15	Pin C4
16	Pin C5
17	Pin C6
18	Pin C7
19	VCC
20	Ground

*Tabelle 2.4: Belegung der Stiftleiste JP6*

Die 20-pol. Leiste JP7 befindet sich am linken Bildrand. Es wird von rechts nach links und von unten nach oben nummeriert.

<b>Stift</b>	<b>Beschreibung</b>
1	CAN High
2	CAN Low
3	CAN High Back
4	CAN Low Back
5	Pin D0
6	Pin D1
7	Pin D2
8	Pin D3
9	Pin D4
10	Pin D5
11	Pin D6
12	Pin D7
13	Pin B5
14	Pin B6
15	Pin B7
16	Reset
17	Versorgungsspannung (+12 V)
18	AREF
19	VCC
20	Ground

*Tabelle 2.5: Belegung der Stiftleiste JP7*

Zu Stift 16 und 18 sei auf das Datenblatt des Mikrocontrollers Kapitel System Control and Reset und Analog to Digital Converter verwiesen.

## 2.3 Dimensionen und Platinenlayout

Die Maße der Control Unit, insbesondere die Position der Bohrlöcher, können nachstehender Skizze entnommen werden. Der linke Teil zeigt die Draufsicht, wobei die Platine die selbe Orientierung aufweist, wie in Abb. 2.1: Control Unit in der Draufsicht. In der Seitenansicht rechts ist die Bestückung der Platine nur grob skizziert, wobei die Maße jedoch mit den maximalen realen Maßen übereinstimmen.

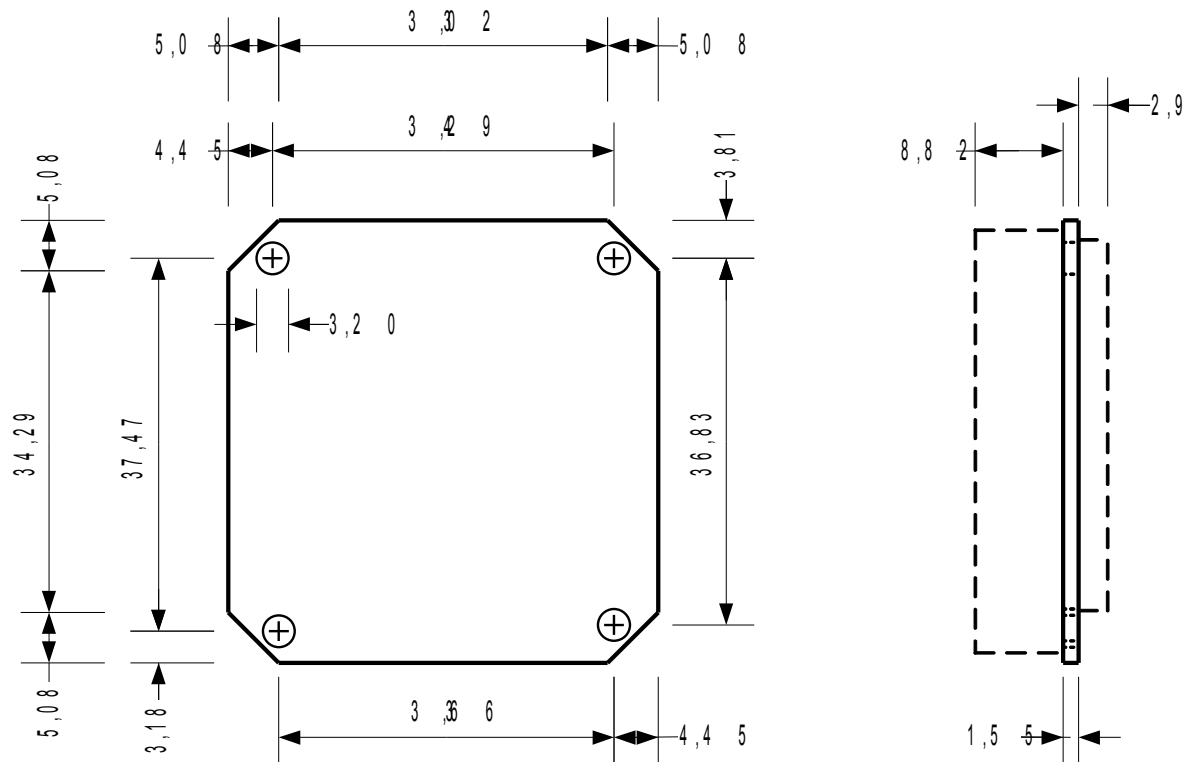


Abb. 2.3: Abmessungen der Control Unit

Die folgende Abbildung zeigt das Platinenlayout von oben. Alle Klemm- und Steckverbindungen befinden sich auf der Oberseite der Platine. Man erkennt ferner in der Bildmitte den CAN-Controller, rechts daneben den CAN-Treiber und oberhalb vom CAN-Controller den 16 MHz Quarz für den Controller-Takt.

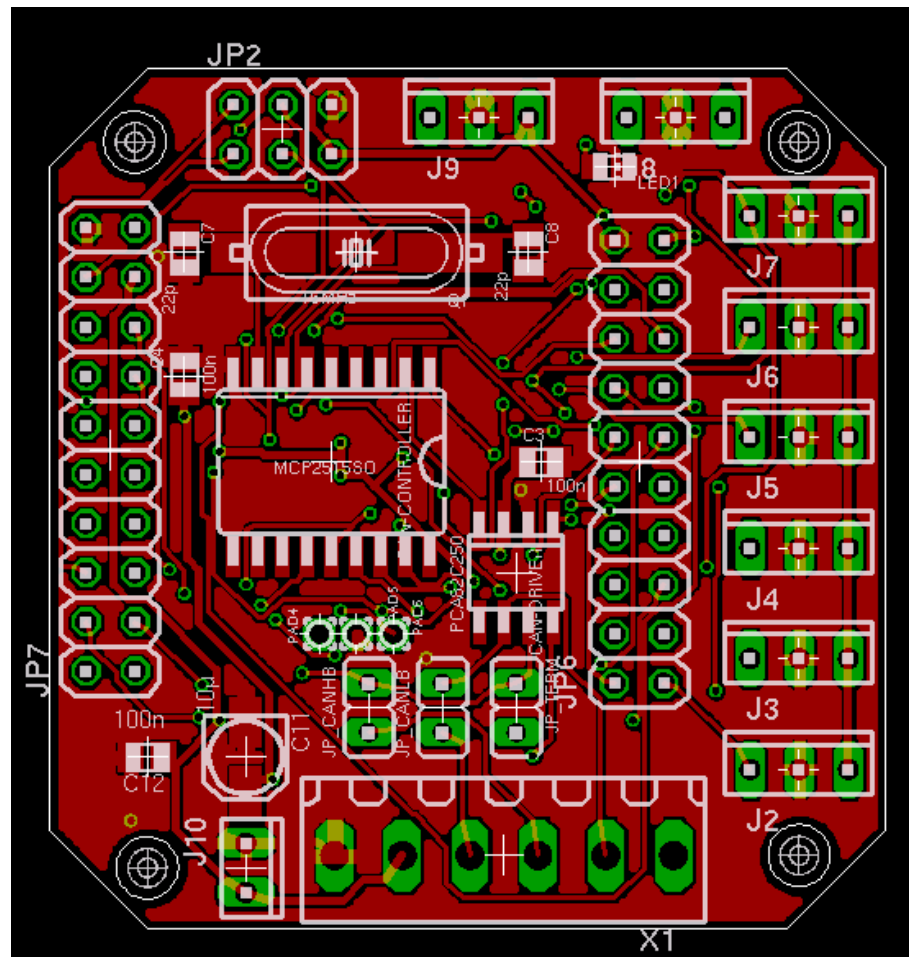


Abb. 2.4: Platinenlayout der Control Unit Oberseite



Der Mikrocontroller ist auf der Unterseite angebracht. Auf dem Layout findet man ihn in der Bildmitte. Oberhalb davon sind die beiden EEPROMs zu erkennen und in der linken unteren Ecke der Festspannungsregler.

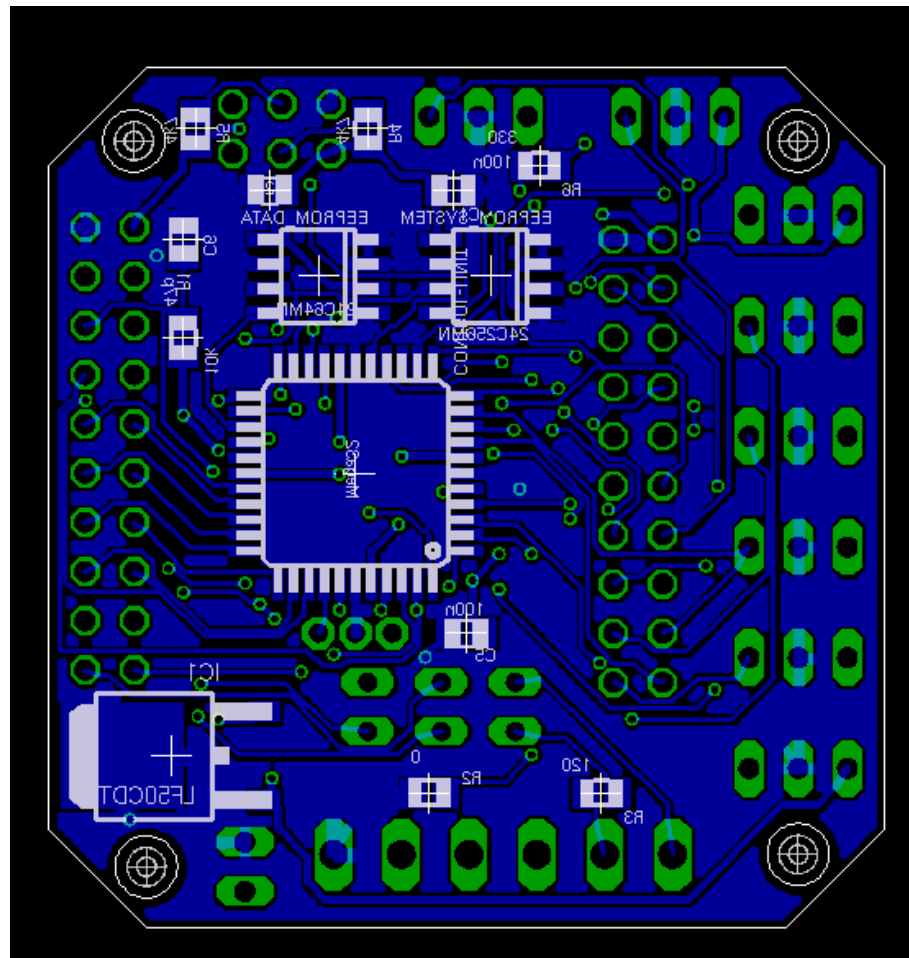


Abb. 2.5: Platinenlayout der Control Unit Unterseite

### 3 Steuerung

Das Steuermodul bestimmt das grundsätzliche Verhalten einer Control Unit, es steuert und verbindet die einzelnen Komponenten, es stellt die Kommunikation mit anderen Control Units und der Central Control Unit sicher und bietet die Möglichkeit des Firmware-Updates. Es handelt sich um eine reine Softwarekomponente, die in jeder Firmware enthalten ist.

#### 3.1 Kommunikation

Zur Kommunikation der Control Units untereinander und mit der Central Control Unit werden zwischen diesen Datenpakete ausgetauscht. Dies kann über verschiedene Schnittstellen - z. B. CAN-Bus, USB-Schnittstelle, Funk - erfolgen. Ein solches Datenpaket hat immer eine Länge von acht Byte und hat folgenden prinzipiellen Aufbau:

VLAN	Source	Dest	MType	Device	Value0	Value1	Value2
------	--------	------	-------	--------	--------	--------	--------

*Abb. 3.1: Aufbau eines Datenpakets*

- VLAN  
Dieses Byte kennzeichnet das virtuelle LAN – Local Area Network – in dem sich das Datenpaket bewegen soll.
- Source  
Source ist die Moduladresse der Control Unit bzw. der Central Control Unit, die das Datenpaket gesendet hat.
- Dest  
Dest ist die Moduladresse der Control Unit bzw. der Central Control Unit, an die das Datenpaket geschickt werden soll.
- MType  
Dieses Byte spezifiziert den Typ des Datenpakets. Es lässt z. B. erkennen, ob es sich um einen Steuer- oder einen Konfigurationsbefehl handelt.
- Device

Dieses Byte gibt an, für welches auf einer Control Unit konfigurierten Geräts das Paket bestimmt ist bzw. welches Gerät das Paket initiiert hat. Ob es sich hierbei um eine Quell- oder Zieladresse handelt wird dabei durch MType bestimmt. In einigen Ausnahmefällen wird dieses Byte aber nicht als Geräteadresse verwendet, sondern trägt vom jeweiligen Anwendungsfall abhängige Informationen, was auch durch MType bestimmt wird.

- Value0 – Value2

Diese drei Bytes werden je nach Datenpaket mit ganz bestimmten Werten gefüllt, die sich so global nicht beschreiben lassen.

Empfängt ein Modul ein Datenpaket und verarbeitet es korrekt, so quittiert es dieses, indem es das Paket mit den selben Werten an den Sender zurückschickt. Als Source wird dabei die eigene Moduladresse und als Dest die des Senders eingesetzt. Der MType wird um 1 inkrementiert. Konnte das Paket nicht erfolgreich verarbeitet werden, wird MType um 2 inkrementiert. In Ausnahmefällen wird von diesem Vorgehen abgewichen. Dann wird an den entsprechenden Stellen explizit darauf hingewiesen.

Im folgenden werden Datenpakete immer durch acht Zahlenwerte dargestellt, die durch Leerzeichen getrennt sind:

34 1 20 0 200 80 100 0

Hier ist 34 das VLAN, in dem sich das Datenpaket bewegt, 1 die Adresse des sendenden Moduls, 20 die Adresse des empfangenden Moduls und die 0 steht für einen Set-Befehl, der an das Gerät mit der Adresse 200 gerichtet ist. Dieses soll innerhalb einer bestimmten Zeit, die in den letzten beiden Bytes codiert ist, den Wert 80 annehmen.

Sind einige dieser Bytes in der jeweiligen Beschreibung nicht genau anzugeben, so werden sie durch entsprechende Variablen dargestellt:

VLAN Source Dest 0 Device 80 100 0

### **3.1.1 Virtuelle LANs**

Jedes lokale physikalische Netzwerk – z. B. ein CAN-Bus oder ein auf einer

bestimmten Frequenz betriebenes Funknetz – kann logisch in kleinere Einheiten zerlegt werden, zwischen denen prinzipiell kein Datenaustausch stattfinden kann. Diese logischen Netzwerke verhalten sich in jeder Hinsicht wie ihre physikalischen Äquivalente.

Zwei Module können nur dann Datenpakete untereinander austauschen, wenn die jeweiligen Schnittstellen, über die der Austausch stattfindet, sich im selben VLAN befinden. Das bedeutet, jeder Schnittstelle wird eine VLAN-ID zugeordnet. Und wenn ein Datenpaket über eine Schnittstelle versendet wird, dann wird eben diese ID in das erste Feld des Pakets eingetragen. Ebenso kann ein Datenpaket nur über eine Schnittstelle empfangen werden, dessen VLAN-ID mit der im Paket übereinstimmt.

Zur Konfiguration der VLAN-IDs sei auf die entsprechenden Kapitel verwiesen, in denen die Schnittstellen beschrieben werden.

### **3.1.2 Modul-Adressierung**

Jedes Modul wird über seine eindeutige Modul-Adresse angesprochen. Das heißt, nur wenn eben diese Adresse im Dest-Feld eines Datenpakets steht, wird das Modul dieses Paket empfangen. Ebenso sorgt die Steuerung dafür, dass diese Adresse ins Source-Feld eines Datenpakets eingetragen wird, wenn ein solches Paket vom Modul versendet wird.

Die Moduladresse kann über folgenden Befehl konfiguriert werden:

```
VLAN Source Dest 76 5 ModulAddress 0 0
```

Ein noch nicht konfiguriertes Modul hat die Adresse 0.

Die Moduladresse kann Werte zwischen 0 und 239 annehmen. Bei den Adressen von 240 bis 255 handelt es sich um Multicast-Adressen, welche somit als Moduladressen nicht zulässig sind. Jede dieser Adressen kann gewissermaßen als Gruppe aufgefasst werden. Wird ein Datenpaket an eine solche Gruppe adressiert, so empfängt dieses Paket jedes Modul, welches Mitglied dieser Gruppe ist. Die Mitgliedschaft in den einzelnen Gruppen kann über folgenden Befehl festgelegt werden:

```
VLAN Source Dest 76 13 LByte HByte 0
```

LByte und HByte bilden zusammen einen 16-Bit Wert, wobei ein gesetztes Bit die Mitgliedschaft in der entsprechenden Gruppe kennzeichnet. Bit0 entspricht dabei der Adresse 240, Bit1 der Adresse 241, usw..

Per Voreinstellung ist jedes Modul und ausschließlich Mitglied der Gruppe 255, wobei sich diese Adresse dann wie eine Broadcast-Adresse verhält.

### **3.1.3 Magic Packet**

Ist ein Modul nicht mehr erreichbar, weil die aktuelle Moduladresse nicht bekannt ist oder gibt es innerhalb eines virtuellen LANs mindestens zwei Module mit der gleichen Adresse, kann das entsprechende Modul angesprochen bzw. die Eindeutigkeit wieder hergestellt werden, indem ein so genanntes Magic Packet eingesetzt wird. Hiermit kann dem Modul eine neue Moduladresse zugewiesen werden ohne die aktuelle Adresse als Dest zu verwenden:

```
VLAN Source 255 124 ModulAddress HwAddr1 HwAddr2 HwAddr3
```

Hierbei ist ModulAddress die neu zu vergebende Moduladresse und HwAddr $x$  das  $x$ -te Byte der Hardware-Adresse des Moduls. Die Hardware-Adresse ist 4 Byte lang, wobei die einzelnen Bytes von 0 bis 3 nummeriert werden. Sie findet sich auf einem Aufkleber auf der Platine der Control Unit, wobei die einzelnen Bytes durch Punkte getrennt sind.

### **3.1.4 Central Control Unit Adresse**

Die Central Control Unit wird wie eine Control Unit über eine Moduladresse angesprochen. Damit ein Modul initiiert mit der Central Control Unit kommunizieren kann, muss deren Adresse dem Modul bekannt gemacht werden. Dies geschieht mit folgendem Konfigurationsbefehl:

```
VLAN Source Dest 76 6 CCUAddress 0 0
```

Der Wertebereich für die CCUAddress unterliegt dabei den selben Restriktionen wie die Moduladresse und liegt damit zwischen 0 und 239. Die Voreinstellung ist 255 und muss damit unbedingt geändert werden.

### **3.1.5 Bridge-Mode**

Eine Control Unit lässt sich auch als Vermittler zwischen verschiedenen LANs einsetzen, wenn sie im Bridge-Mode betrieben wird. Dieser lässt sich mit

VLAN Source Dest 76 10 Bridge-Mode 0 0

einstellen, wobei Bridge-Mode=0 für den Normalbetrieb und Bridge-Mode=1 für den aktivierten Bridge-Mode steht. Die Voreinstellung ist Bridge-Mode=0.

Wird eine Control Unit im Bridge-Mode betrieben, so wird jedes Datenpaket, welches über eine bestimmte Schnittstelle empfangen wird und nicht für die Control Unit selbst bestimmt ist, über alle anderen verfügbaren Schnittstellen unverändert weiter geschickt. Lediglich die VLAN-ID wird gemäß der auf der Schnittstelle konfigurierten angepasst.

Empfangene Broadcasts und Multicasts werden grundsätzlich weiter versendet.

### **3.1.6 Empfangspuffer**

Es kann vorkommen, dass eine Control Unit schneller Datenpakete empfängt, als sie diese verarbeiten kann. Zu diesem Zweck ist jede Control Unit mit einem Empfangspuffer ausgestattet, der per Voreinstellung 4 Pakete zwischenspeichern kann. Die Länge dieses Empfangspuffers lässt sich über folgenden Befehl festlegen:

VLAN Source Dest 76 96 Length 0 0

Length spezifiziert dabei die Länge des Empfangspuffers. Damit die Einstellung wirksam wird, ist auf dem Modul ein Reset auszuführen.

Bei der Länge ist ein Kompromiss zu finden zwischen der Betriebssicherheit – der Puffer muss groß genug sein, damit keine Pakete verloren gehen – und den Anwendungsmöglichkeiten der Control Unit – jedes zwischen zu speichernde Paket benötigt 8 Byte RAM, der statisch verwaltet wird und deshalb anderen Komponenten nicht mehr zur Verfügung steht.

### **3.1.7 Steuerbefehle**

Dieser Nachrichtentyp wird verwendet, um ein an die Control Unit angeschlossenes

Gerät auf einen bestimmten Wert zu setzen. Gegebenenfalls werden weitere Steuerparameter übergeben. Ein Steuerbefehl hat folgende allgemeine Form:

VLAN Source Dest 0 Device Value LParam HParam

Device ist die Geräte-Adresse des zu steuernden Geräts, Value der Wert, den das Gerät annehmen soll und LParam und HParam bilden zusammen einen 16-Bit Wert der Steuerparameter enthält. Jedes Gerät bzw. jede Komponente der Software behandelt die letzten drei Bytes etwas anders. Hier sei auf die entsprechenden Kapitel verwiesen.

### **3.1.8 Abfragen**

Um den Wert bzw. den Status, den ein Gerät aktuell hat, abzufragen, kommt dieser Nachrichtentyp zum Tragen. Die allgemeine Form ist:

VLAN Source Dest 8 Device 0 0 Select

Device ist die Geräte-Adresse des Geräts, dessen Status abgefragt werden soll. Manche Geräte haben mehrere Werte bzw. Status, dann wird durch Select der entsprechende Wert selektiert. Beantwortet wird eine solche Nachricht wie folgt:

VLAN Source Dest 9 Device LValue HValue Select

Es ändert sich also lediglich der Nachrichtentyp und in dem 16-Bit Wert, der durch LValue und HValue gebildet wird, ist das Ergebnis der Abfrage enthalten. Auch hier werden die letzten drei Byte von jeder Komponente individuell genutzt, was den entsprechenden Kapiteln zu entnehmen ist.

### **3.1.9 Statusmeldungen**

Statusmeldungen werden von unterschiedlichsten Geräten einer Control Unit beim Eintreffen bestimmter Ereignisse generiert. Andere Komponenten, wie z. B. die autonome Steuerung, verarbeiten dann solche Statusmeldungen. Die allgemeine Form einer Statusmeldung ist die folgende:

VLAN Source Dest 16 Device LValue HValue Ext

Device ist die Geräte-Adresse des Geräts, welches die Statusmeldung auslöste,

LValue und HValue bilden zusammen einen 16-Bit Wert, der normalerweise dem aktuellen Status des Geräts entspricht und Ext ist eine komponentenabhängige Erweiterung. Insbesondere die letzten drei Byte werden von den verschiedenen Komponenten unterschiedlich verwendet, was in den entsprechenden Kapiteln nachzulesen ist.

### **3.1.10 Makros**

Makros sind zunächst einmal nichts anderes als ein bestimmter Nachrichtentyp. Ein Makro hat folgenden Aufbau:

VLAN Source Dest 24 0 LMakro HMakro 0

LMakro und HMakro bilden zusammen dabei einen 16-Bit Wert, der die Makro-Nummer spezifiziert.

Unterschiedliche Komponenten der Control Unit können Makros generieren, z. B. die Infrarotschnittstelle und unterschiedliche Komponenten können auf den Empfang eines Makros reagieren, z. B. die autonome Steuerung. Details hierzu sind den entsprechenden Kapiteln zu entnehmen.

Auch die Central Control Unit kann Makros generieren und auch auf sie reagieren, indem gewöhnlich eine komplexe Steuer- bzw. Regelaufgabe durch sie angestoßen wird.

### **3.1.11 Fehlerbehandlung**

Neben der „normalen“ Fehlerbehandlung, wie sie in Kapitel 3.1 Kommunikation beschrieben ist, gibt es noch einige Sonderfälle, die in diesem Kapitel erörtert werden sollen.

Wird durch eine lokale Ansteuerung – Autonome Steuerung, Bedienoberfläche oder Infrarot Fernbedienung – ein Fehler beim Setzen eines Werts provoziert, so schickt die entsprechende Control Unit eine Fehlermeldung folgender Gestalt:

VLAN Source Dest 2 Device Value LDelay HDelay

Source und Dest sind hier gleich und repräsentieren die Moduladresse der Control Unit, auf der der Fehler aufgetreten ist. Dies leuchtet ein, wenn man sich verdeutlicht,



dass in Source die Control Unit steht, die angesteuert wurde und in Dest die ansteuernde Control Unit, was bei einer lokalen Ansteuerung das selbe ist. Die restlichen Werte entsprechen denen eines klassischen Setz-Befehls und sind selbstredend.

Steuert eine lokale Control Unit eine entfernte andere oder richtet an sie eine Anfrage und bekommt von dieser keine Antwort, so setzt die lokale Control Unit, wie folgt eine Fehlermeldung ab:

VLAN LModul CCU 127 1 RModul RMTyp RDevice

LModul ist die Adresse des lokalen Moduls, CCU die Adresse der Central Control Unit, RModul ist die Adresse des entfernten Moduls, RMTyp ist der Nachrichtentyp der ursprünglichen Nachricht und RDevice das entsprechende Gerät. Die 127 repräsentiert einen Protokollfehler und die 1 steht für „No Response“.

Wird versucht ein Gerät anzusteuern, welches zu dieser Zeit nicht bereit ist – z. B. ein Rollladen, der gerade dabei ist in eine zuvor bestimmte Position zu fahren, reagiert die zugehörige Control Unit mit folgender Meldung:

VLAN Modul CCU 127 2 Device 0 0

Modul ist dabei die Adresse der entsprechenden Control Unit, CCU die Adresse der Central Control Unit und Device die Geräte-Adresse. Die 127 repräsentiert einen Protokollfehler und die 2 steht für „Device Busy“.

### **3.2 Resets und Neustarts**

Es gibt insgesamt drei verschiedene Möglichkeiten auf einer Control Unit einen Reset auszuführen. Ein normaler Software-Reset – er wird im Allgemeinen nur als Reset bezeichnet – wird mit dem Befehl

VLAN Source Dest 76 1 0 0 0

eingeleitet. Dadurch initialisiert sich die Software neu und startet mit den aktuellen Konfigurationsparametern. Diese werden jedoch nicht aus dem EEPROM geladen. Das heißt, wurden Parameter geändert aber noch nicht gespeichert, so werden sie dennoch berücksichtigt und nicht durch gespeicherte Parameter überschrieben. Einige Änderungen der Konfiguration verlangen im Anschluss einen solchen Reset

um wirksam zu werden. In diesen Fällen wird dann explizit darauf hingewiesen.

Einen so genannten Full-Reset leitet man mit dem Befehl

```
VLAN Source Dest 76 2 0 0 0
```

ein. Die Control Unit wird daraufhin komplett neu gestartet, das heißt, es findet auch eine Hardware-Initialisierung statt. Dieser Reset entspricht dem Aus- und wieder Einschalten des Moduls. Geänderte Konfigurationsparameter, die noch nicht gespeichert wurden, gehen dabei verloren.

Um die Control Unit mit den voreingestellten Konfigurationsparametern neu zu starten (Konfigurationsreset) verwendet man folgenden Befehl:

```
VLAN Source Dest 76 3 0 0 0
```

Die Control Unit befindet sich danach in dem selben Zustand wie nach dem aller ersten Start nach der Auslieferung.

### **3.3 Konfigurationsmanagement**

Möchte man die aktuelle Konfiguration dauerhaft im EEPROM speichern, so dass sie auch nach einem Neustart wieder zur Verfügung steht, so wendet man folgenden Befehl an:

```
VLAN Source Dest 76 8 0 0 0
```

Um die Konfiguration aus dem EEPROM zu laden und damit die aktuelle Konfiguration zu überschreiben, kommt folgender Befehl zur Anwendung:

```
VLAN Source Dest 76 9 0 0 0
```

Im Allgemeinen muss danach ein Reset durchgeführt werden, um die geladene Konfiguration aktiv werden zu lassen. Es sei denn, die geladene Konfiguration unterscheidet sich von der vorherigen nur durch Parameter, die einen solchen Reset nicht erfordern.

Mit welcher Konfiguration ein Modul nach dem Start bzw. Neustart hochfährt, wird durch den Startmodus festgelegt, der wie folgt gesetzt werden kann:

```
VLAN Source Dest 76 4 StartMode 0 0
```

Damit die Einstellung nach dem Neustart auch greift, muss die Konfiguration vorher gespeichert werden. Ist StartMode=217, so startet das Modul mit der Konfiguration, die sich zu diesem Zeitpunkt im EEPROM befindet. Ist der StartMode=179, so startet das Modul mit der Default-Konfiguration. Lediglich die Konfigurationsparameter, die die Steuerung, den Buzzer, das Funkmodul und das CAN-Modul betreffen, bleiben erhalten. Bei jedem anderen Wert startet das Modul mit der kompletten Default-Konfiguration, also derjenigen, wie sie unmittelbar nach der Auslieferung vorhanden war.

### **3.4 Externe Beschaltung**

Die Ein- und Ausgänge der Control Unit können mit unterschiedlichen externen Beschaltungen versehen werden. Hierzu gehören z. B. Dimmer- oder Relaisstufen. Je nach Ausführung sind bereits einige Beschaltungen auf der Platine der Control Unit integriert, wie z. B. das TWI (Two Wire Interface) oder das SPI (Serial Peripheral Interface), welches zur Ansteuerung des CAN-Bus (Controller Area Network Bus) verwendet wird.

Welche Beschaltung eines Ein- bzw. Ausgangs vorgenommen wurde, muss der Software über Konfigurationsbefehle mitgeteilt werden. Die auf der Platine bereits vorhandenen Beschaltungen sind bereits vorkonfiguriert, sofern die dazu nötigen Software-Komponenten installiert wurden. Dadurch weist ein nicht konfiguriertes Modul bereits eine gewisse Basisfunktionalität auf, die die Inbetriebnahme erheblich erleichtert.

Ein solcher Konfigurationsbefehl hat die allgemeine Form:

VLAN Source Dest 64 Pin Value0 Value1 0

Pin spezifiziert den zu konfigurierenden Pin des Mikrocontrollers (A0 = 0 bis D7 = 31). Dann gibt Value0 (Code + Offset) den Typ der externen Beschaltung gemäß nachstehender Tabelle und Value1 die Geräte-Adresse des angeschlossenen Geräts an.

Code	Offset	Geräte-Typ	Voreinstell.	
			Pin	Adr.
0	0	Statische Ausgabe (Pegel Low)	s. u.	s. u.
	1	Statische Ausgabe (Pegel High)		
2	0	Buzzer	B0	8
3	0	Infrarot-Empfänger	D4	28
4	0	Serielle Schnittstelle (Receiver)	D0	24
	1	Serielle Schnittstelle (Transmitter)	D1	25
6	0	Nulldurchgangserkenn. Netzspannung/Ext. Reset	D2	26
8	0	Serial Peripheral Interface (Slave Select)	B4	12
9	0	Serial Peripheral Interface (Master Out - Slave In)	B5	13
10	0	Serial Peripheral Interface (Master In - Slave Out)	B6	14
11	0	Serial Peripheral Interface (System Clock)	B7	15
12	0	Two Wire Interface (System Clock)	C0	16
13	0	Two Wire Interface (System Data)	C1	17
16	0	Geschaltetes Gerät		
32	0	Analoger Eingang		
40	0	Digitaler Eingang		
48	0	LCD (Display Data 0)		
	1	LCD (Display Data 1)		
	2	LCD (Display Data 2)		
	3	LCD (Display Data 3)		
	4	LCD (Display Data 4)		
	5	LCD (Display Data 5)		
	6	LCD (Display Data 6)		
	7	LCD (Display Data 7)		
	8	LCD (Read/Write)		
	9	LCD (Register Select)		
	10	LCD (Enable)		
	11	LCD (Hintergrundbeleuchtung)		
64	0	Dimmer		
	1	Dimmer und Soft-Delay		
	2	Dimmer und Lange Zündimpulse		
	4	Dimmer und Schaltrestriktion		
	8	Dimmer und Phasenabschnittsteuerung		
128	0	Logischer Eingang		

	1	Logischer Eingang und Steigende Flanke		
	2	Logischer Eingang und Fallende Flanke		
	4	Logischer Eingang und Prellfreiheit		
	8	Logischer Eingang und Kurze Aktivierung		
	12	Logischer Eingang und Lange Aktivierung		
	16	Logischer Eingang und Pull-up Widerstand		
	32	Logischer Eingang und Prellfreiheit erzwingen		

*Tabelle 3.1: Konfiguration der Geräte-Typen*

Alle Pins, die in obiger Tabelle in der Spalte Voreinstellung nicht aufgeführt sind, werden standardmäßig als Statische Ausgabe (Pegel Low) konfiguriert und bekommen die Geräte-Adresse, die der Nummer des Pins entspricht (A0 = 0 bis D7 = 31).

Addiert man zu dem zu konfigurierenden Pin 32, so gibt Value0 die Moduladresse an, an die das an dem entsprechenden Pin konfigurierte Gerät seine Statusmeldungen verschickt. Die Voreinstellung ist hier 0, was das Versenden jeglicher Statusmeldungen unterbindet. Entspricht die Adresse der Moduladresse des Moduls, auf dem das Gerät konfiguriert ist, wird die Statusmeldung nur lokal verarbeitet und nicht versendet. Soll eine Statusmeldung lediglich von der Central Control Unit verarbeitet werden, so wird deren Adresse hier eingesetzt. Analoges gilt für jede andere Control Unit. Broadcasts und Multicasts können hier in gewohnter Weise verwendet werden.

### **3.5 Zeitsynchronisation**

In einigen Komponenten der Control Unit ist es erforderlich, die aktuelle Tageszeit inklusive Wochentag verfügbar zu haben, um verschiedene Steueraufgaben adäquat durchführen zu können. Aus diesem Grund ist in die Control Unit eine vollständige Uhr integriert worden. Zum Einstellen der Uhrzeit kann der folgende Befehl verwendet werden:

VLAN Source Dest 120 Hundredth Second Minute HourDay

HourDay spezifiziert dabei den aktuellen Wochentag – die niederwertigen drei Bits beginnend mit Montag=0 – und die momentane Stunde – die höherwertigen fünf Bits.

Minute steht dann für die aktuell vergangenen Minuten, Second für die Sekunden und Hundredth für die hundertstel Sekunden. Die Voreinstellung ist bei allen Werten 0.

Nach jedem Neustart oder Reset versendet die Control Unit eine so genannte Synchronisationsanfrage der folgenden Form:

VLAN Source Dest 123 0 0 0 0

Erreicht diese Anfrage einen so genannten Time-Server, so antwortet dieser mit einer Nachricht zum Einstellen der Uhrzeit, so wie es am Anfang des Kapitels beschrieben ist.

Eine Control Unit kann gleichzeitig auch ein Time-Server sein. Dies lässt sich mit dem Befehl

VLAN Source Dest 76 7 TimeServer 0 0

festlegen. Ist TimeServer=0 ist die Control Unit kein Time-Server. Ist TimeServer=1 übernimmt das Modul die Funktion eines Time-Servers. Die Voreinstellung ist 0.

### **3.6 Software-Management**

Da auch die Firmware der Control Unit einem ständigen Entwicklungsprozess unterliegt, existieren eine ganze Reihe unterschiedlicher Versionen. Um nun herauszufinden welche Firmware Version gerade auf einer Control Unit installiert ist, kann man dieser die Nachricht

VLAN Source Dest 76 28 0 0 0

schicken. In dem entsprechenden Antwortpaket wird dann als Value0 die Major-Nummer, als Value1 die Minor-Nummer und als Value2 die Phase der Version zurückgesendet.

Ferner besteht die Firmware aus verschiedenen Komponenten, die die Ansteuerung unterschiedlicher externer Beschaltungen bzw. Bauteile übernehmen oder sonstige Features zur Verfügung stellen. Schickt man einer Control Unit den Befehl

VLAN Source Dest 76 30 X 0 0

so sendet die Unit im Antwortpaket als Value1 das X-te Byte der Eigenschaft, die die in der Firmware enthaltenen Komponenten spezifiziert. Diese Eigenschaft besteht aus vier Byte, wobei vom niederwertigsten mit 0 begonnen wird zu zählen. Die Bedeutung der einzelnen Bits dieser Eigenschaft kann folgender Tabelle entnommen werden. Dabei ist die entsprechende Komponente vorhanden, wenn das zugehörige Bit gesetzt ist.

Bit	Komponente
0	EEPROM-Support
1	Externer Reset
2	Buzzer
3	Funkmodul
4	CAN-Bus
5	Infrarotschnittstelle
6-7	2 – LCD-Display (2 x 16 Zeichen) 3 – LCD-Display (3 x 16 Zeichen)
8	Logische Eingänge
9	Analoge Eingänge
10	Dallas Digitales Thermometer
11	Schalter
12	Dimmer
13	Rollladensteuerung
14-15	1 – Drehgeber (PEC11) 2 – Drehgeber (STEC)
16	Bedienoberfläche
17	Autonome Steuerung
18-31	Nicht verwendet

*Tabelle 3.2: Softwarekomponenten*

Im Folgenden ist mit Firmware stets die compilierte Version ohne zusätzliche Informationen, wie Adressen oder ähnliches, also der reine Byte-Code gemeint.

Möchte man eine neue Firmware Variante auf einer Control Unit installieren, so ist zunächst deren Größe zu spezifizieren. Dies geschieht mit dem Befehl

VLAN Source Dest 76 27 LByte HByte 0

wobei LByte und HByte einen 16-Bit Wert bilden, der die Größe der Firmware in Byte

angibt. Jetzt kann mit dem Befehl

```
VLAN Source Dest 56 0 16 0 0
```

der Firmware Download eingeleitet werden. Dieser Download erfolgt nun seitenbasiert, das heißt, es werden immer ganze Seiten von 64 Byte Größe übertragen und die Anzahl der übertragenen Seiten muss gerade sein. Eine nicht vollständig gefüllte Seite bzw. – siehe obige Bedingung nach gerader Seitenanzahl – eine völlig leere Seite wird mit 0xFF aufgefüllt. Die Firmware wird dadurch auf eine Größe gebracht, die durch 128 teilbar ist. Obige Spezifikation der Firmware Größe bezieht sich wohl gemerkt auf die tatsächliche Größe, also ohne die Füllbytes. Die Reihenfolge, in der die Seiten übertragen werden, ist prinzipiell egal. Eine Seite kann auch mehrfach übertragen werden, wenn z. B. beim vorherigen Übertragungsversuch ein Fehler aufgetreten ist. Um nun eine einzelne Seite zu downloaden wird zunächst ein Startpaket geschickt:

```
VLAN Source Dest 56 16 160 LSideAddr HSideAddr
```

LSideAddr und HSideAddr bilden dabei einen 16-Bit Wert, der die Startadresse der Seite in Byte angibt. Die erste Seite hat die Adresse 0x0000, die zweite Seite die Adresse 0x0040 usw.. Jetzt folgen 16 Datenpakete – 16 Pakete x 4 Byte = 64 Byte = 1 Seite – in der Form

```
VLAN Source Dest 60 Data0 Data1 Data2 Data3
```

```
VLAN Source Dest 60 Data4 Data5 Data6 Data7
```

```
VLAN Source Dest 60 Data8 Data9 Data10 Data11 usw.
```

Um die Übertragung einer Seite abzuschließen, folgt ein Paket mit Prüfsummen:

```
VLAN Source Dest 60 Checksum0 Checksum1 Checksum2 Checksum3
```

Sind die Prüfsummen korrekt wird dieses Paket mit einem Acknowledge beantwortet, ansonsten mit einem Error. Das Startpaket und die Datenpakete erhalten jeweils keine Bestätigung. Nun kann die nächste Seite übertragen werden oder aber die vorherige noch einmal, sollte das Prüfsummenpaket mit einem Fehler oder gar nicht bestätigt werden.

Die Prüfsummen werden über die Daten-Bytes gebildet, also über Device und Value0 bis Value2 und zwar so, dass die Summe aller vorherigen 16 Datenpakete plus das



Seitenstartpaket plus das Prüfsummenpaket modulo 256 Null ergibt. In Device werden also die Device-Werte summiert, in Value0 die Value0-Werte usw.. Dabei wird das Startpaket, die 16 Datenpakete und das Prüfsummenpaket mit einbezogen.

Abgeschlossen wird der Firmware-Download mit einem Download-Ende-Paket:

```
VLAN Source Dest 56 1 0 0 0
```

Diese Prozedur hat die Konfiguration der Control Unit verändert. Soll diese nach einem Reset erhalten bleiben – das ist für einen erfolgreichen Installationsvorgang unbedingt erforderlich –, muss sie zunächst gespeichert werden (siehe Kapitel 3.3 Konfigurationsmanagement).

Die neue Firmware befindet sich jetzt in einem externen EEPROM und muss noch in den Flash-Speicher des Mikrocontrollers übertragen werden. Dieser Zustand, also ob eine neue Firmware zur Übertragung in den Flash im EEPROM zur Verfügung steht, kann mit dem Befehl

```
VLAN Source Dest 76 24 0 0 0
```

erfragt werden. In dem entsprechenden Antwortpaket nimmt Value0 den Wert 0 an, wenn dem so ist und den Wert 255 im anderen Fall.

Nach dem Speichern der Konfiguration muss die Control Unit neu gestartet werden, es muss also ein so genannter Full-Reset durchgeführt werden (siehe Kapitel 3.2 Resets und Neustarts). Beim Neustart erkennt der Bootloader, dass eine neue Firmware zum Flashen im EEPROM zur Verfügung steht und führt den Flash-Vorgang durch. Dies kann je nach Größe der Firmware einige Zeit in Anspruch nehmen. Danach wird automatisch die neue Firmware zur Ausführung gebracht, was durch den charakteristischen Piep-Ton bzw. das Aufleuchten der Control-LED signalisiert wird.

## 4 Komponenten

Die Firmware der Control Unit besteht aus unterschiedlichen Komponenten oder Modulen, die für die Ansteuerung unterschiedlicher externer Beschaltungen bzw. für bestimmte Software-Features benötigt werden (siehe Kapitel 3.6 Software-Management). Im Allgemeinen sind nie alle Komponenten in einer Firmware enthalten. Dieses Kapitel beschreibt die unterschiedlichen Komponenten und erklärt deren Einsatzmöglichkeiten.

### 4.1 EEPROM-Support

Zusätzlich zu dem im Mikrocontroller enthaltenen EEPROM verfügt eine Control Unit über zwei externe EEPROMs, die zusätzlichen Speicherplatz bereitstellen. Dieser wird unter anderem für die Installation neuer Firmware (siehe Kapitel 3.6 Software-Management) und für die Speicherung der Konfigurationsdaten der Bedienoberfläche (siehe Kapitel 4.16 Bedienoberfläche) genutzt. Die beiden EEPROMs haben eine Größe von 32 KByte bzw. 8 KByte. Die Komponente EEPROM-Support stellt die Möglichkeit bereit, die Inhalte der EEPROMs über die verfügbaren Kommunikationsschnittstellen auszulesen.

Hierzu wird einer Control Unit folgende Nachricht gesendet:

VLAN Source Dest 28 EEPROMReadAddr LAddr HAddr Number

EEPROMReadAddr ist die Lese-Adresse des EEPROMs, welches ausgelesen werden soll. Das 32 KByte EEPROM hat die Lese-Adresse 161 und das 8 KByte EEPROM die Lese-Adresse 169. LAddr und HAddr bilden zusammen einen 16-Bit Wert, der die Adresse innerhalb des EEPROMs angibt, ab der gelesen werden soll. Diese Adresse ist Byte-basiert und beginnt bei 0. Number gibt schließlich die Anzahl Bytes an, die ausgelesen werden soll. Mit einem Lesebefehl können höchstens 64 Byte ausgelesen werden.

Die Control Unit schickt daraufhin die angeforderten Daten an den Sender in folgender Form zurück:

VLAN Source Dest 29 Data0 Data1 Data2 Data3

VLAN Source Dest 29 Data4 Data5 Data6 Data7

VLAN Source Dest 29 Data8 Data9 Data10 Data11 usw.

Es werden dabei so viele Pakete geschickt, wie nötig sind, um die angeforderten Bytes zu übermitteln. Abschließend wird das anfragende Paket noch einmal bestätigt oder es wird mit einem Fehler quittiert, wenn es nicht korrekt verarbeitet werden konnte.

## 4.2 Externer Reset

Je nach Ausführung der Hardware einer Control Unit ist auf der Platine ein Taster oder zwei Pins zum Setzen einer Brücke untergebracht, mit deren Hilfe ein Reset der Control Unit von Extern herbeigeführt werden kann. Gegebenenfalls kann ein zusätzlicher Taster am Modulgehäuse angebracht werden, der über die gegebenen Steckverbindungen angeschlossen werden kann.

Wird dieser Taster für mindestens eine Sekunde betätigt, führt die Control Unit einen Konfigurationsreset durch (siehe Kapitel 3.2 Resets und Neustarts). Dieser wird von einem Signalton bzw. dem Aufleuchten der Control-LED der Stärke „Panic“ begleitet. So besteht die Möglichkeit, die Control Unit händisch in den Auslieferungszustand zu versetzen, falls eine Kommunikation mit der Control Unit bzw. ein Ändern der Konfiguration nicht mehr möglich ist.

## 4.3 Buzzer

In den meisten Fällen ist an einer Control Unit eine akustische – in Form eines Buzzers – oder optische – in Form einer Control-LED – Signaleinrichtung angebracht. Diese dient neben der Signalisierung von bestimmten Systemzuständen vor allem der Unterstützung des Anwenders bei der Benutzung der Infrarot-Fernbedienung und der Bedienoberfläche, sowie als Rückmeldung nach erfolgreichem oder aber fehlgeschlagenen Ansteuern eines Geräts.

Es werden drei verschiedene Signalstärken unterschieden, die sich nur hinsichtlich ihrer Länge unterscheiden:

Signal	Dauer
Ack	1/10 s
Error	1 s
Panic	4,5 s

*Tabelle 4.1: Signalarten*

Welches Signal an welcher Stelle eingesetzt wird, ist in den entsprechenden Kapiteln beschrieben.

Ob in einer gegebenen Situation ein Signal gegeben wird ist vom jeweiligen Buzzerlevel abhängig. Dieser kann wie folgt konfiguriert werden:

VLAN Source Dest 76 16 LByte HByte 0

Dabei bilden LByte und HByte einen 16-Bit Wert, der den Buzzerlevel spezifiziert. Der Vorgabewert ist 0xFFFF. Jedes gesetzte Bit in diesem Wert sorgt für das Auslösen des Signals in der entsprechenden Situation gemäß folgender Tabelle:

Bit	Bedeutung
0	Systemmeldung
1	
2	
3	
4	IR-Tastendruck
5	IR-Befehlsbestätigung
6	IR-Fehler
7	IR-Lernbestätigung
8	GUI-Tastendruck
9	GUI-Befehlsbestätigung
10	GUI-Fehler
11	GUI-Drehereignis
12	
13	
14	
15	

*Tabelle 4.2: Buzzerlevel*

Die Bedeutung der einzelnen Bits soll im Folgenden noch näher erläutert werden:

- Systemmeldung

Hierzu zählen alle Ereignisse, die durch das System im Laufe des Betriebs selbst hervorgebracht werden, wie z. B. Neustart oder Systemfehler.

- **IR-Tastendruck**  
Ein Signal wird gegeben, wenn auf einer Fernbedienung eine Taste gedrückt wurde.
- **IR-Befehlsbestätigung**  
Ein über eine Fernbedienung abgesetzter Befehl wird bestätigt, sofern dieser erfolgreich ausgeführt werden konnte.
- **IR-Fehler**  
Ein Fehler wird signalisiert, wenn ein über eine Fernbedienung abgesetzter Befehl nicht erfolgreich verarbeitet werden konnte.
- **IR-Lernbestätigung**  
Das erfolgreiche Lernen eines Tastencodes einer Fernbedienung wird signalisiert.
- **GUI-Tastendruck**  
Ein Signal wird gegeben, wenn in Verbindung mit der Bedienoberfläche eine Taste gedrückt wird.
- **GUI-Befehlsbestätigung**  
Ein über die Bedienoberfläche abgesetzter Befehl wird bestätigt, sofern dieser erfolgreich ausgeführt werden konnte.
- **GUI-Fehler**  
Ein Fehler wird signalisiert, wenn ein über die Bedienoberfläche abgesetzter Befehl nicht erfolgreich verarbeitet werden konnte.

## **4.4 Funkmodul**

Eine mögliche Schnittstelle zur Kommunikation mit anderen Control Units bzw. mit der Central Control Unit stellt die Beschaltung mit einem Funkmodul dar. Dieses wird an die serielle Schnittstelle der Control Unit angeschlossen. Prinzipiell ist es auch möglich, statt Funk ein beliebiges anderes Übertragungsmedium – z. B. ein Kabel – zu benutzen. Ursprünglich wurde diese Komponente jedoch als Funkmodul konzipiert. Über diese Schnittstelle können dann die Nachrichten versendet werden, wie sie in diesem Dokument beschrieben werden.

Zunächst muss hierfür die entsprechende Pin-Konfiguration vorgenommen werden, wie sie in Kapitel 3.4 Externe Beschaltung beschrieben wurde. Anschließend wird die VLAN-ID für die Schnittstelle mit dem Befehl

```
VLAN Source Dest 76 12 VLAN-ID 0 0
```

festgelegt, wobei VLAN-ID eben diese ID angibt. Die Voreinstellung ist 0. Nun ist die Funk-Schnittstelle prinzipiell einsatzbereit.

Um die Sicherheit bei der Übertragung der Daten hinsichtlich des Abhörens und der gezielten Manipulation des Funkverkehrs zu erhöhen, besteht die Möglichkeit, die übertragenen Daten zu verschlüsseln. Ob der Funkverkehr verschlüsselt werden soll, kann über folgenden Befehl konfiguriert werden:

```
VLAN Source Dest 76 14 Mode 0 0
```

Ist Mode gleich 0, findet keine Verschlüsselung statt, ist Mode gleich 1, wird die gesamte Nachricht bis auf die VLAN-ID verschlüsselt und ist Mode gleich 3, so wird die gesamte Nachricht inklusive der VLAN-ID verschlüsselt. Per Voreinstellung findet keine Verschlüsselung statt. Um die Daten nun sinnvoll zu verschlüsseln, muss ein acht Byte langer Schlüssel – per Voreinstellung sind alle Bytes des Schlüssels mit 0 initialisiert – auf der Control Unit hinterlegt werden. Hierbei wird jedes Byte einzeln mit dem Befehl

```
VLAN Source Dest 76 15 Number Value 0
```

festgelegt, wobei Number die Nummer (0 bis 7) des festzulegenden Bytes angibt und Value das entsprechende Byte des Schlüssels repräsentiert. Es empfiehlt sich, zuerst den Schlüssel festzulegen und dann erst die Verschlüsselung einzuschalten, damit es in der Zwischenzeit nicht zu Kommunikationsproblemen kommt. Verschlüsselt bzw. entschlüsselt wird, in dem jedes einzelne Byte einer Nachricht mit dem entsprechenden Byte des Schlüssels „Exklusiv Oder“ verknüpft wird.

Liegen zwei Control Units soweit auseinander, dass die Sendeleistung des Funkmoduls nicht ausreicht, die Distanz zu überbrücken, lässt sich dazwischen eine Art „Funk-Hub“ installieren um die Distanzen zu verringern. Eine solche Control Unit leitet dann alle Nachrichten, die von entsprechend konfigurierten Control Units empfangen werden bzw. für diese bestimmt sind weiter. Sie tritt damit in gewisser

Weise als Stellvertreter für diese entfernten Control Units auf. Eine Control Unit kann als „Funk-Hub“ für bis zu vier andere Control Units fungieren, wobei lediglich die Moduladressen der entfernten Control Units auf der Control Unit, die als „Funk-Hub“ arbeitet, konfiguriert werden müssen. Dies geschieht mit folgendem Befehl:

```
VLAN Source Dest 76 11 Number Address 0
```

Number ist die Nummer (0 bis 3) der entfernten Control Unit und Address deren Moduladresse. Ist Address gleich 0 – das ist die Voreinstellung – dann ist diese Funktion deaktiviert. Ansonsten sind bei der Angabe der Moduladresse die Konventionen einzuhalten, wie sie im Kapitel 3.1.2 Modul-Adressierung beschrieben sind.

## **4.5 CAN-Bus**

Eine weitere mögliche Schnittstelle zur Kommunikation mit anderen Control Units ist der CAN-Bus (Controller Area Network Bus). Hierbei ist der Anschluss eines entsprechenden CAN-Bus Controllers an das SPI (Serial Peripheral Interface) der Control Unit nötig, der damit die Verbindung zwischen der Control Unit und dem eigentlichen Bus bildet. Über diesen Bus können dann die üblichen Nachrichten verschickt werden.

Zur Konfiguration der CAN-Bus Schnittstelle muss die Pin-Konfiguration des SPI gemäß Kapitel 3.4 Externe Beschaltung vorgenommen werden. Weiter ist lediglich die VLAN-ID der Schnittstelle mit folgendem Befehl zu konfigurieren:

```
VLAN Source Dest 76 18 VLAN-ID 0 0
```

VLAN-ID gibt dabei eben diese ID an. Die Voreinstellung ist 0. Damit ist die Konfiguration der CAN-Bus Schnittstelle abgeschlossen und diese betriebsbereit.

## **4.6 Infrarotschnittstelle**

Die Software bietet die Möglichkeit des Betriebs einer Infrarotschnittstelle. Nimmt man eine adäquate Beschaltung der Control Unit mit einem Infrarotempfänger vor, so kann man daran eine handelsübliche Fernbedienung betreiben und darüber Steuerbefehle für die gesamte HA-Umgebung absetzen.

Zur Pin-Konfiguration des Infrarotempfängers ist gemäß Kapitel 3.4 Externe Beschaltung vorzugehen.

#### **4.6.1 Konfiguration der Fernbedienung**

Eine Grundvoraussetzung, die die Fernbedienung erfüllen muss, ist, dass sie mit dem RC5-Code arbeitet. Dies ist bei den meisten Fernbedienungen der Fall. Ferner muss die Control Unit auf die daran zu betreibende Fernbedienung abgestimmt werden. So hat jede Fernbedienung eine Geräteadresse, die bei jedem Tastendruck mitgesendet wird. Diese ist, und sollte es auch, von Fernbedienung zu Fernbedienung unterschiedlich. Die Control Unit reagiert nur auf die Fernbedienung, dessen Geräteadresse mit der zuvor konfigurierten übereinstimmt. Damit wird verhindert, dass z. B. bei der Fernbedienung des Fernsehers die Control Unit mit angesprochen wird. Darüber hinaus sind die Tastencodes, die bei Betätigung einer Taste versendet werden nicht immer bei jeder Fernbedienung gleich. Darum ist es wichtig, auf der Control Unit eine Zuordnung der Tasten-Codes zu den entsprechenden Kommandos, die durch den Tastendruck ausgelöst werden sollen, vorzunehmen.

Die Abstimmung einer Control Unit auf eine entsprechende Fernbedienung wird mit folgendem Konfigurationsbefehl vorgenommen:

```
VLAN Source Dest 80 Button Command Device 0
```

Button ist hierbei der Tastencode einer bestimmten Taste, Command der Code des Kommandos, das durch den Tastendruck gegeben werden soll gemäß unten stehender Tabelle und Device die Geräteadresse der Fernbedienung.



<b>Code</b>	<b>Kommando</b>
0	Taste 0
1	Taste 1
2	Taste 2
3	Taste 3
4	Taste 4
5	Taste 5
6	Taste 6
7	Taste 7
8	Taste 8
9	Taste 9
12	Alles Einschalten
15	Alles Ausschalten
30	Makro
32	Plus
33	Minus
38	Enter
62	Ignorieren

*Tabelle 4.3: IR-Kommandos*

Jeder Taste bzw. jedem Tastencode ist nun entsprechend ein Kommando zuzuordnen. Soll eine Taste keine Funktion haben, so ordnet man ihr den Kommandocode 62 zu. Die Control Unit wird diese Taste dann ignorieren. Per Voreinstellung ist jedem möglichen Tastencode von 0 bis 63 der identische Kommandocode zugeordnet. Die Geräteadresse ist 0.

Da es sehr schwierig ist, die Geräteadresse und die Tastencodes einer Fernbedienung zu ermitteln, gibt es eine elegantere Möglichkeit obige Zuordnung vorzunehmen. Man gibt ein Kommando vor und ordnet den Tastencode zu, indem man die entsprechende Taste auf der Fernbedienung drückt. Die Control Unit lernt dann gewissermaßen die Codes und auch die Geräteadresse von der Fernbedienung.

Um nun einen solchen Lernvorgang einzuleiten, schickt man der Control Unit den Befehl

#### VLAN Source Dest 84 0 Command 0 0

Command ist auch hier wieder der Code des zuzuordnenden Kommandos (siehe obige Tabelle). Dieser Befehl wird erst einmal nicht im herkömmlichen Sinn von der Control Unit beantwortet. Wird nun innerhalb der nächsten 4 Sekunden eine Taste auf der Fernbedienung betätigt und von der Control Unit registriert, wird die mit dem Tastendruck übermittelte Geräteadresse gespeichert und der übermittelte Tastencode dem zuvor definierten Kommando zugeordnet. Die Signaleinrichtung bestätigt dies mit einem Ack und die Control Unit beantwortet den ursprünglichen Befehl in der Form

#### VLAN Source Dest 85 Button Command Device 0

wobei Button wieder der Tastencode, Command der Code des festgelegten Kommandos und Device die Geräteadresse der Fernbedienung ist. Konnte die Control Unit innerhalb der 4 Sekunden kein Signal von einer Fernbedienung empfangen, schickt sie folgende Fehlernachricht:

#### VLAN Source Dest 86 0 Command 0 0

Dabei ist Command wieder der Code des Kommandos, welches festgelegt werden sollte. Die Signaleinrichtung quittiert diesen Zustand mit einem Error.

### **4.6.2 Infrarot-Adressen und Hot-Keys**

Um ein Gerät ansteuern zu können, muss die Modul-Adresse der Control Unit, an der das Gerät angeschlossen ist und die Geräte-Adresse selbst bekannt sein. Da eine Modul-Adresse als auch eine Geräte-Adresse aus jeweils drei Ziffern besteht, wäre die Eingabe eines sechs stelligen Codes über die Fernbedienung nötig, um das zu steuernde Gerät eindeutig zu spezifizieren. Da dies einem Anwender nicht zuzumuten ist, lassen sich pro Infrarotschnittstelle 100 zwei stellige Codes (auch Infrarot-Adressen genannt) definieren, hinter denen sich jeweils wieder eine Modul- und eine Geräte-Adresse verbirgt, die an Stelle der sechs stelligen Codes benutzt werden.

Eine Geräte-Adresse und die zugehörige Modul-Adresse lassen sich einer Infrarot-Adresse wie folgt zuweisen:

#### VLAN Source Dest 68 IR Modul Device 0

IR ist dabei die Infrarot-Adresse, Modul die Modul-Adresse und Device die Geräte-Adresse. Per Voreinstellung wird jeder möglichen Infrarot-Adresse die Modul- und Geräte-Adresse 0 zugeordnet.

Ferner lassen sich 10 Hot-Keys definieren, die den 10 Ziffern der Fernbedienung entsprechen (siehe hierzu Kapitel 4.6.3 Bedienung). Einem Hot-Key wird eine Makro-Nummer zugeordnet und bei Betätigen dieses Hot-Keys wird ein Makro mit eben dieser Nummer generiert. Die Zuordnung findet über folgenden Befehl statt:

#### VLAN Source Dest 72 HotKey LMakro HMakro 0

HotKey ist die Nummer des Hot-Keys (0 bis 9) und LMakro und HMakro bilden zusammen einen 16-Bit Wert, der die Nummer des zugeordneten Makros repräsentiert. Allen Hot-Keys ist per Voreinstellung die Makro-Nummer 0 zugeordnet.

### **4.6.3 Bedienung**

Grundsätzlich gilt, dass wenn ein Befehl nicht binnen 8 Sekunden über die Fernbedienung eingegeben wird, die Control Unit die Befehlseingabe abbricht und auf eine neue Eingabe wartet.

Bei den folgenden Darstellungen der Infrarot-Befehle steht jeder Ausdruck für jeweils ein über die Fernbedienung eingegebenes Kommando (siehe Kapitel 4.6.1 Konfiguration der Fernbedienung).

Die kürzesten Befehle, die eingegeben werden können, bestehen aus nur einem Kommando:

#### Command

Bei Command kann es sich um das Kommando Plus oder Minus handeln, was dazu führt, dass dem zuletzt über die Fernbedienung angesteuerten Gerät die Nachricht Plus bzw. Minus geschickt wird, was z. B. einen Dimmer dazu veranlasst seine Helligkeit etwas zu steigern bzw. zu verringern. Werden diese Tasten dauerhaft gedrückt, so wird das entsprechende Kommando periodisch wiederholt, bis die Taste wieder losgelassen wird.

Über den Befehl

Hot-Key Enter

wird ein Makro mit der Nummer generiert, die dem entsprechenden Hot-Key zugeordnet wurde (siehe Kapitel 4.6.2 Infrarot-Adressen und Hot-Keys). Hot-Key kann dabei das Kommando 0 bis 9 sein.

Um ein Makro ganz allgemeiner Form zu generieren, können folgende Befehlsformen verwendet werden:

Makro Number0 Enter

Makro Number1 Number0 Enter

Makro Number2 Number1 Number0 Enter

Makro Number3 Number2 Number1 Number0 Enter

Die Makro-Nummer wird durch NumberX spezifiziert, wobei X die Stelle bzw. die Wertigkeit der Nummer im Zehnersystem ist. Number kann dabei das Kommando 0 bis 9 sein.

Hat ein Befehl die Form

IR1 IR0 Enter

so wird der Status des Geräts, das durch die Infrarot-Adresse IR gegeben ist invertiert.

Wird ein Befehl in der Form

IR1 IR0 Value Enter

einggegeben, so wird dem Gerät, das durch die Infrarot-Adresse IR gegeben ist, der Wert  $(\text{Value} + 1) * 10$  zugewiesen, es sei denn, Value ist das Kommando 0, dann ist der zugewiesene Wert ebenfalls 0.

Um den Wert, der dem anzusteuernenden Gerät zugewiesen werden soll, in nativer Form anzugeben, wird der Befehl wie folgt verwendet:

IR1 IR0 Value2 Value1 Value0 Enter

IR ist wieder die Infrarot-Adresse des zu steuernden Geräts und Value der zuzuweisende Wert.

Möchte man auch noch die Zeit spezifizieren, innerhalb derer das anzusteuernde Gerät den neuen Wert annehmen soll, ist folgender Befehl zu verwenden:

```
IR1 IR0 Value2 Value1 Value0 Time1 Time0 Enter
```

IR ist auch hier wieder die Infrarot-Adresse des zu steuernden Geräts, Value der zuzuweisende Wert und Time die Zeit in Sekunden innerhalb derer das anzusteuernde Gerät den spezifizierten Wert anzunehmen hat.

Eine Fehleingabe wird durch die Signaleinrichtung mit einem Error quittiert, eine erfolgreiche Eingabe eines Befehls analog mit einem Ack. Jedes Drücken einer Taste wird zusätzlich mit einem Ack bestätigt. Dies alles gilt natürlich nur dann, wenn der Buzzerlevel entsprechend konfiguriert wurde (siehe Kapitel 4.3 Buzzer).

## **4.7 LCD-Display**

Die Software einer Control Unit bietet die Möglichkeit zur Ansteuerung eines LCD (Liquid Crystal Display). Voraussetzung hierfür ist, dass das LCD über einen ST7036 Controller angesteuert wird, wie es z. B. bei den LCDs der DOG-Serie standardmäßig der Fall ist. Im Wesentlichen wird das LCD zu Visualisierungszwecken im Zusammenhang mit der Bedienoberfläche verwendet (siehe Kapitel 4.16 Bedienoberfläche). Prinzipiell kann das LCD aber von jeder Komponente angesteuert werden. Insbesondere lässt es sich über spezielle Nachrichten steuern, die an die Control Unit, an die das LCD angeschlossen ist, geschickt werden.

Es werden unterschiedliche Display-Typen unterstützt, was durch die installierte Software sichergestellt werden muss (siehe hierzu Kapitel 3.6 Software-Management).

Die Pin-Konfiguration ist gemäß Kapitel 3.4 Externe Beschaltung vorzunehmen, wobei einige Besonderheiten zu berücksichtigen sind:

- Es werden nur die Datenpins 0 bis 3 verwendet, wobei diese auf einem Port und hintereinander liegen müssen.

- Die Pins Read/Write und Register Select müssen ebenfalls auf einem Port und hintereinander liegen.
- Die Device-Adresse wird auf jedem Pin gleich gesetzt.

Zur Ansteuerung des LCD über die diversen Kommunikationsschnittstellen stehen zwei spezielle Nachrichtentypen zur Verfügung. Mit

VLAN Source Dest 88 Char0 Char1 Char2 Char3

können vier Zeichen (Char0 bis Char3) auf dem Display ausgegeben werden. Das Zeichen 160 (0xA0) wird ignoriert. Mit

VLAN Source Dest 91 Inst0 Inst1 Inst2 Inst3

lassen sich vier Instruktionen bzw. Steuercodes (Inst0 bis Inst3) an das Display senden, wobei die Instruktion 0 ignoriert wird. Ist Inst0 gleich 0, definiert Inst1 den Wert für die Hintergrundbeleuchtung. Ist der Wert 0 wird die Hintergrundbeleuchtung ausgeschaltet. Für jeden anderen Wert wird sie eingeschaltet. Die Antwortpakete bei diesem Nachrichtentyp (Ack/Error) entsprechen denen des vorigen Nachrichtentyps. Sie bilden somit hier eine Ausnahme. Nähere Informationen zu den Instruktionen bzw. Steuercodes sind der Dokumentation des LCD bzw. des ST7036 Controllers zu entnehmen.

## 4.8 Logische Eingänge

Logische Eingänge dienen der Erfassung von logischen Eingangssignalen, die also lediglich aus zwei unterscheidbaren Zuständen bestehen. Sie werden im Rahmen des HAP vorwiegend zur Erfassung von Schaltzuständen von Schaltern, Tastern oder Kontakten benutzt. Sie kommen aber auch als Eingänge für Drehgeber und für weitere Aufgaben, die der Erfassung logischer Signalpegel dienen, zum Einsatz.

Zur Pin-Konfiguration sei auf das Kapitel 3.4 Externe Beschaltung verwiesen. Wird die Option Pull-up Widerstand gewählt, so wird der interne Pull-up Widerstand aktiviert und in den meisten Fällen kann dann auf eine aufwändige externe Beschaltung verzichtet werden.

Der Pegel, der an einem Pin anliegt, der als logischer Eingang konfiguriert ist, wird alle 10 ms abgegriffen. Signalwechsel, die schneller stattfinden können also nicht

registriert werden.

Der Wert, den ein logischer Eingang annimmt, entspricht nicht unbedingt dem Pegel, der an dem entsprechenden Pin anliegt. Dies ist je nach Konfiguration unterschiedlich. Der Wert ist eher eine abstrakte Eigenschaft des logischen Eingangs, der seinen Status beschreibt und der noch durch die Information ergänzt wird, durch welches Ereignis er in diesen Status versetzt wurde.

Ein logischer Eingang arbeitet also ereignisorientiert. Ein Ereignis ist zunächst einmal die Änderung des an dem entsprechenden Pin anliegenden Pegel. Auf ein solches Ereignis wird nur reagiert, wenn die entsprechende Flanke bei der Pin-Konfiguration Berücksichtigung gefunden hat (siehe Kapitel 3.4 Externe Beschaltung). Zusätzlich kann bestimmt werden, dass der Pegel nach einem Wechsel eine bestimmte Zeit stabil bleiben muss. Hier wird in der Pin-Konfiguration zwischen Prellfreiheit, kurze und lange Aktivierung unterschieden. Ist die kurze Aktivierung konfiguriert, werden aber auch Ereignisse berücksichtigt, die lediglich der Prellfreiheit genügen. Ebenso werden bei der Konfiguration der langen Aktivierung auch Ereignisse der kurzen Aktivierung und der Prellfreiheit akzeptiert. Nur wenn diese konfigurierten Bedingungen erfüllt sind, wird ein Ereignis registriert und der Status bzw. der Wert des logischen Eingangs ändert sich. Dann wird auch eine Statusmeldung generiert, die eben diesen Wert enthält.

Dieser Wert ist ein 8-Bit Wert und wird im LValue der Statusmeldung bzw. der Abfrage untergebracht. Der HValue ist dabei 0. Ebenso ist Select bzw. Ext gleich 0 (siehe Kapitel 3.1.8 Abfragen und 3.1.9 Statusmeldungen). Bit 7 von LValue ist 0, wenn das Ereignis eine steigende und 1, wenn es eine fallende Flanke war. Bit 2 und 3 von LValue geben die Zeit, die der Pegel nach seiner Änderung stabil war, gemäß folgender Tabelle an:

Bit 3	Bit 2	Zeit
0	0	Keine Zeitrestriktion
0	1	Prellfreiheit
1	0	Kurze Aktivierung
1	1	Lange Aktivierung

*Tabelle 4.4: Aktivierungscodes von logischen Eingängen*

Die Werte entsprechen also dem Offset der Pin-Konfiguration.

Wird per Konfiguration die Prellfreiheit erzwungen, so wird für jede nicht konfigurierte Flanke dennoch ein Ereignis generiert, sofern die Prellfreiheit gegeben ist.

Den einzelnen Zeitrestriktionen können die entsprechenden Zeitspannen über folgenden Konfigurationsbefehl zugeordnet werden:

VLAN Source Dest 76 32 Code LTime HTime

Code ist der Code der Zeitrestriktion deren Zeitspanne festgelegt werden soll gemäß unten stehender Tabelle. LTime und HTime bilden zusammen einen 16-Bit Wert, der die festzulegende Zeitspanne in hundertstel Sekunden angibt. Die jeweilige Voreinstellung ist ebenfalls der Tabelle zu entnehmen.

Code	Zeitrestriktion	Voreinstellung
0	Keine Zeitrestriktion	0 s
1	Prellfreiheit	0,1 s
2	Kurze Aktivierung	0,5 s
3	Lange Aktivierung	1,5 s

*Tabelle 4.5: Voreingestellte Aktivierungszeiten bei logischen Eingängen*

## 4.9 Analoge Eingänge

Analoge Eingänge dienen der Messwerterfassung mit Hilfe analoger Sensoren. Darunter werden hier Sensoren verstanden, die über eine entsprechende externe Beschaltung an den in den Mikrocontroller der Control Unit integrierten Analog to Digital Converter angeschlossen werden können, der den analogen Messwert zur Weiterverarbeitung in einen digitalen verwandelt. Als Beispiel seien hier temperaturabhängige oder helligkeitsabhängige Widerstände genannt.

Über das reine Erfassen von Messwerten hinaus, können jedem analogen Eingang bis zu zwei Triggerpunkte zugewiesen werden, die bei Über- bzw. Unterschreiten – das ist von der jeweils aktuellen Konfiguration abhängig – Statusmeldungen generieren.

Zur Nutzung der analogen Eingänge ist die entsprechende Komponente der



Software zu aktivieren (siehe Kapitel 3.6 Software-Management).

Zur Pin-Konfiguration sei auf das Kapitel 3.4 Externe Beschaltung verwiesen. Hierbei ist zu beachten, dass die Sensoren ausschließlich an Port A angeschlossen werden können.

Wie oft ein Messwert vom Sensor abgefragt wird, wird durch die Sample-Rate definiert. Diese kann nicht höher sein, als zehn mal pro Sekunde. Konfiguriert wird sie über folgenden Befehl:

VLAN Source Dest 76 80 Addr LRate HRate

Addr ist dabei die Device-Adresse des analogen Eingangs – bei den folgenden Konfigurationsbefehlen ist das ebenfalls so – und LRate und HRate bilden zusammen einen 16-Bit Wert, der die Sample-Rate in Zehntelsekunden angibt. Die Voreinstellung ist 60 Sekunden.

Die Triggerpunkte und die zugehörige Hysterese sowie die Triggerflags werden mit dem Befehl

VLAN Source Dest 76 Code Addr LValue HValue

bzw.

VLAN Source Dest 76 Code Addr Hyst Flags

festgelegt. Ist Bit 0 von Code 0, wird der Triggerwert festgelegt, andernfalls die Hysterese und die Triggerflags. Bit 1-3 kodieren die Nummer des Triggers. Bit 4-7 sind 0010. Ist Code also 64 wird der Triggerwert des Triggers 0 festgelegt. Ist Code 65, legt man die Hysterese und die Triggerflags des Triggers 0 fest. LValue und HValue bilden zusammen einen 16-Bit Wert, der den Triggerwert repräsentiert. Hyst steht für die Hysterese und Flags für die Triggerflags. Der Triggerwert, die Hysterese und auch die Triggerflags werden per Voreinstellung auf 0 gesetzt.

Der Triggerwert weist dasselbe Format auf, wie der Wert des analogen Eingangs selbst. Dieses Format sei im Folgenden dargestellt:

LByte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	$2^3$	$2^2$	$2^1$	$2^0$	$2^{-1}$	$2^{-2}$	$2^{-3}$	$2^{-4}$

<b>HByte</b>	<b>Bit 15</b>	<b>Bit 14</b>	<b>Bit 13</b>	<b>Bit 12</b>	<b>Bit 11</b>	<b>Bit 10</b>	<b>Bit 9</b>	<b>Bit 8</b>
	$2^{11}$	$2^{10}$	$2^9$	$2^8$	$2^7$	$2^6$	$2^5$	$2^4$

Abb. 4.1: Allgemeines Werteformat bei analogen Eingängen

Die Hysterese besitzt das gleiche Format wie das LByte des Triggerwerts. Von den Triggerflags sind nur Bit 2 und Bit 3 relevant. Alle anderen Bits werden intern verwendet und sind bei der Konfiguration auf 0 zu setzen. Ist Bit 2 gesetzt, so wird der analoge Eingang eine Statusmeldung (siehe Kapitel 3.1.9 Statusmeldungen) generieren, sobald der aktuelle Messwert den Triggerwert unterschreitet. Eine weitere Statusmeldung folgt, wenn der Messwert mindestens wieder den Triggerwert plus Hysterese erreicht, nachdem der Triggerwert vorher unterschritten worden ist. Ist Bit 3 gesetzt, so generiert der analoge Eingang eine Statusmeldung, sobald der aktuelle Messwert den Triggerwert überschreitet. Eine weitere Statusmeldung folgt dann, wenn der Messwert mindestens wieder den Triggerwert minus Hysterese erreicht, nachdem der Triggerwert vorher überschritten worden ist. Sind diese Bits nicht gesetzt, bleiben die entsprechenden Statusmeldungen aus.

Die Statusmeldungen haben das Format, wie es in Kapitel 3.1.9 Statusmeldungen beschrieben ist. Während HValue und Ext 0 sind, codiert LValue den Trigger und das Ereignis, das zum Auslösen der Statusmeldung führte. Die Triggernummer des Triggers, der das Ereignis auslöste, wird in den Bits 0-2 codiert. Bit 3-5 sind immer 0. Bit 6 und Bit 7 codieren das Ereignis gemäß folgender Tabelle:

Bit 7	Bit 6	Ereignis
0	0	Der Triggerwert plus Hysterese wurde wieder erreicht, nachdem er zuvor unterschritten wurde.
0	1	Der Triggerwert wurde unterschritten.
1	0	Der Triggerwert minus Hysterese wurde wieder erreicht, nachdem er zuvor überschritten wurde.
1	1	Der Triggerwert wurde überschritten.

Tabelle 4.6: Ereigniscodierung bei analogen Eingängen

Soll der aktuelle Wert eines analogen Eingangs abgefragt werden (siehe Kapitel 3.1.8 Abfragen), so ist Select auf 0 zu setzen. Dann liefert LValue und HValue zusammen den 16-Bit Wert des Eingangs gemäß oben stehenden Formats. Hat

Select die Form 1010TT00 und wird für TT die Nummer eines Triggers eingesetzt, so liefert LValue und HValue zusammen den 16-Bit Wert des entsprechenden Triggers, ebenfalls gemäß oben stehenden Formats. Entspricht Select keinem dieser Muster, wird als Ergebnis 0xFFFF zurückgeliefert.

Die Triggerwerte lassen sich auch über gewöhnliche Steuerbefehle (siehe Kapitel 3.1.7 Steuerbefehle) setzen, indem Value auf 1010TT00, wobei TT für die Nummer des entsprechenden Triggers codiert, gesetzt wird. Dann bildet LParam und HParam zusammen den 16-Bit Wert, der dem Trigger zugewiesen wird.

Der integrierte Analog to Digital Converter hat eine Auflösung von 10 Bit. Davon codieren 8 Bits den Wert links vom Komma und 2 Bits stehen für den Wert rechts vom Komma. Das bedeutet, dass die Bits 0-1 und 12-15 des Werts des analogen Eingangs, so wie weiter oben beschrieben, nicht verwendet werden. Diese Bits sind immer 0. Dies führt dann zu folgendem Format:

<b>LByte</b>	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	$2^3$	$2^2$	$2^1$	$2^0$	$2^{-1}$	$2^{-2}$	0	0

<b>HByte</b>	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
	0	0	0	0	$2^7$	$2^6$	$2^5$	$2^4$

*Abb. 4.2: Konkretes Werteformat bei analogen Eingängen*

Die genaue Zuordnung der Spannungshöhen an den analogen Eingängen zu den digitalisierten Werten, sowie weitere Informationen über externe Beschaltungsmöglichkeiten, sind dem Datenblatt des verwendeten Mikrocontrollers zu entnehmen.

## 4.10 Digitale Eingänge

Digitale Eingänge dienen der Messwerterfassung mit Hilfe digitaler Sensoren. Unter digitalen Sensoren werden hier Sensoren verstanden, die über ein 1-Wire Interface verfügen, über welches der Messwert ausgelesen werden kann.

Über das reine Erfassen von Messwerten hinaus, können jedem digitalen Eingang

bis zu zwei Triggerpunkte zugewiesen werden, die bei Über- bzw. Unterschreiten – das ist von der jeweils aktuellen Konfiguration abhängig – Statusmeldungen generieren.

Gegebenenfalls macht die Benutzung eines bestimmten Sensors die Aktivierung einer bestimmten Softwarekomponente nötig (siehe hierzu Kapitel 3.6 Software-Management).

Zur Pin-Konfiguration sei auf das Kapitel 3.4 Externe Beschaltung verwiesen. Hierbei ist zu beachten, dass die Sensoren ausschließlich an Port A und nur einer pro Pin angeschlossen werden können.

Wie oft ein Messwert vom Sensor abgefragt wird, wird durch die Sample-Rate definiert. Diese kann nicht höher sein, als einmal pro Sekunde. Konfiguriert wird sie über folgenden Befehl:

VLAN Source Dest 76 144 Addr LRate HRate

Addr ist dabei die Device-Adresse des digitalen Eingangs – bei den folgenden Konfigurationsbefehlen ist das ebenfalls so – und LRate und HRate bilden zusammen einen 16-Bit Wert, der die Sample-Rate in Sekunden angibt. Die Voreinstellung ist 60 Sekunden. Mit

VLAN Source Dest 76 145 Addr Type 0

wird der Sensortyp gemäß unten stehender Tabelle spezifiziert, wobei die Voreinstellung 0 ist.

Type	Sensortyp
1	Dallas Digitales Thermometer DS18B20
2	Dallas Digitales Thermometer DS18S20

*Tabelle 4.7: Typen digitaler Sensoren*

Die Triggerpunkte und die zugehörige Hysterese sowie die Triggerflags werden mit dem Befehl

VLAN Source Dest 76 Code Addr LValue HValue

bzw.

## VLAN Source Dest 76 Code Addr Hyst Flags

festgelegt. Ist Bit 0 von Code 0, wird der Triggerwert festgelegt, andernfalls die Hysterese und die Triggerflags. Bit 1-3 kodieren die Nummer des Triggers. Bit 4-7 sind 0001. Ist Code also 128 wird der Triggerwert des Triggers 0 festgelegt. Ist Code 129, legt man die Hysterese und die Triggerflags des Triggers 0 fest. LValue und HValue bilden zusammen einen 16-Bit Wert, der den Triggerwert repräsentiert. Hyst steht für die Hysterese und Flags für die Triggerflags. Der Triggerwert, die Hysterese und auch die Triggerflags werden per Voreinstellung auf 0 gesetzt.

Der Triggerwert weist dasselbe Format auf, wie der Wert des digitalen Eingangs selbst. Dieses Format sei im Folgenden dargestellt:

<b>LByte</b>	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	$2^3$	$2^2$	$2^1$	$2^0$	$2^{-1}$	$2^{-2}$	$2^{-3}$	$2^{-4}$

<b>HByte</b>	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
	S	$2^{10}$	$2^9$	$2^8$	$2^7$	$2^6$	$2^5$	$2^4$

Abb. 4.3: Allgemeines Werteformat bei digitalen Eingängen

Das „S“ bei Bit 15 steht für „Sign“. Es handelt sich bei diesem Bit also um das Vorzeichenbit. Ist dieses 0, so handelt es sich um einen positiven Wert, andernfalls um einen negativen.

Die Hysterese besitzt das gleiche Format wie das LByte des Triggerwerts. Von den Triggerflags sind nur Bit 2 und Bit 3 relevant. Alle anderen Bits werden intern verwendet und sind bei der Konfiguration auf 0 zu setzen. Ist Bit 2 gesetzt, so wird der digitale Eingang eine Statusmeldung (siehe Kapitel 3.1.9 Statusmeldungen) generieren, sobald der aktuelle Messwert den Triggerwert unterschreitet. Eine weitere Statusmeldung folgt, wenn der Messwert mindestens wieder den Triggerwert plus Hysterese erreicht, nachdem der Triggerwert vorher unterschritten worden ist. Ist Bit 3 gesetzt, so generiert der digitale Eingang eine Statusmeldung, sobald der aktuelle Messwert den Triggerwert überschreitet. Eine weitere Statusmeldung folgt dann, wenn der Messwert mindestens wieder den Triggerwert minus Hysterese erreicht, nachdem der Triggerwert vorher überschritten worden ist. Sind diese Bits

nicht gesetzt, bleiben die entsprechenden Statusmeldungen aus.

Die Statusmeldungen haben das Format, wie es in Kapitel 3.1.9 Statusmeldungen beschrieben ist. Während HValue und Ext 0 sind, codiert LValue den Trigger und das Ereignis, das zum Auslösen der Statusmeldung führte. Die Triggernummer des Triggers, der das Ereignis auslöste, wird in den Bits 0-2 codiert. Bit 3-5 sind immer 0. Bit 6 und Bit 7 codieren das Ereignis gemäß folgender Tabelle:

Bit 7	Bit 6	Ereignis
0	0	Der Triggerwert plus Hysterese wurde wieder erreicht, nachdem er zuvor unterschritten wurde.
0	1	Der Triggerwert wurde unterschritten.
1	0	Der Triggerwert minus Hysterese wurde wieder erreicht, nachdem er zuvor überschritten wurde.
1	1	Der Triggerwert wurde überschritten.

*Tabelle 4.8: Ereigniscodierung bei digitalen Eingängen*

Soll der aktuelle Wert eines digitalen Eingangs abgefragt werden (siehe Kapitel 3.1.8 Abfragen), so ist Select auf 0 zu setzen. Dann liefert LValue und HValue zusammen den 16-Bit Wert des Eingangs gemäß oben stehenden Formats. Hat Select die Form 1010TT00 und wird für TT die Nummer eines Triggers eingesetzt, so liefert LValue und HValue zusammen den 16-Bit Wert des entsprechenden Triggers, ebenfalls gemäß oben stehenden Formats. Entspricht Select keinem dieser Muster, wird als Ergebnis 0xFFFF zurückgeliefert.

Die Triggerwerte lassen sich auch über gewöhnliche Steuerbefehle (siehe Kapitel 3.1.7 Steuerbefehle) setzen, indem Value auf 1010TT00, wobei TT für die Nummer des entsprechenden Triggers codiert, gesetzt wird. Dann bildet LParam und HParam zusammen den 16-Bit Wert, der dem Trigger zugewiesen wird.

#### **4.10.1 Dallas Digitales Thermometer**

Beim Digitalen Thermometer von Dallas Semiconductor der DS1820 Reihe handelt es sich um einen digitalen Sensor, der die Umgebungstemperatur misst. Beim HAP werden zwei Varianten unterstützt, die sich gemäß ihrer Auflösung unterscheiden.

Der DS18B20 hat eine Auflösung von 12 Bit. Davon repräsentiert ein Bit das

Vorzeichen, 7 Bits codieren den Wert links vom Komma und 4 Bits stehen für den Wert rechts vom Komma. Das bedeutet, dass die Bits 11-14 des Werts des digitalen Eingangs, so wie in Kapitel 4.10 Digitale Eingänge beschrieben, nicht verwendet werden. Diese Bits nehmen den selben Wert an, wie Bit 15. Dies führt dann zu folgendem Format:

<b>LByte</b>	<b>Bit 7</b>	<b>Bit 6</b>	<b>Bit 5</b>	<b>Bit 4</b>	<b>Bit 3</b>	<b>Bit 2</b>	<b>Bit 1</b>	<b>Bit 0</b>
	$2^3$	$2^2$	$2^1$	$2^0$	$2^{-1}$	$2^{-2}$	$2^{-3}$	$2^{-4}$

<b>HByte</b>	<b>Bit 15</b>	<b>Bit 14</b>	<b>Bit 13</b>	<b>Bit 12</b>	<b>Bit 11</b>	<b>Bit 10</b>	<b>Bit 9</b>	<b>Bit 8</b>
	S	S	S	S	S	$2^6$	$2^5$	$2^4$

Abb. 4.4: Werteformat beim digitalen Thermometer (DS18B20)

Die nachfolgende Tabelle gibt einige Temperaturwerte und deren zugehöriges Bitmuster bzw. den entsprechenden hexadezimalen Wert wieder:

<b>Temperatur</b>	<b>Wert (binär)</b>	<b>Wert (hexadezimal)</b>
+125 °C	0000 0111 1101 0000	07D0
+85 °C	0000 0101 0101 0000	0550
+25,0625 °C	0000 0001 1001 0001	0191
+10,125 °C	0000 0000 1010 0010	00A2
+0,5 °C	0000 0000 0000 1000	0008
0 °C	0000 0000 0000 0000	0000
-0,5 °C	1111 1111 1111 1000	FFF8
-10,125 °C	1111 1111 0101 1110	FF5E
-25,0625 °C	1111 1110 0110 1111	FE6F
-55 °C	1111 1100 1001 0000	FC90

Tabelle 4.9: Temperaturcodierung beim digitalen Thermometer (DS18B20)

Der DS18S20 hat eine Auflösung von 9 Bit. Davon repräsentiert ein Bit das Vorzeichen, 7 Bits codieren den Wert links vom Komma und 1 Bit steht für den Wert rechts vom Komma. Das bedeutet, dass die Bits 11-14 des Werts des digitalen Eingangs, so wie in Kapitel 4.10 Digitale Eingänge beschrieben, nicht verwendet werden. Diese Bits nehmen den selben Wert an, wie Bit 15. Die Bits 0-2 sind immer

0. Dies führt dann zu folgendem Format:

<b>LByte</b>	<b>Bit 7</b>	<b>Bit 6</b>	<b>Bit 5</b>	<b>Bit 4</b>	<b>Bit 3</b>	<b>Bit 2</b>	<b>Bit 1</b>	<b>Bit 0</b>
	$2^3$	$2^2$	$2^1$	$2^0$	$2^{-1}$	0	0	0

<b>HByte</b>	<b>Bit 15</b>	<b>Bit 14</b>	<b>Bit 13</b>	<b>Bit 12</b>	<b>Bit 11</b>	<b>Bit 10</b>	<b>Bit 9</b>	<b>Bit 8</b>
	S	S	S	S	S	$2^6$	$2^5$	$2^4$

Abb. 4.5: Werteformat beim digitalen Thermometer (DS18S20)

Die nachfolgende Tabelle gibt einige Temperaturwerte und deren zugehöriges Bitmuster bzw. den entsprechenden hexadezimalen Wert wieder:

<b>Temperatur</b>	<b>Wert (binär)</b>	<b>Wert (hexadezimal)</b>
+85 °C	0000 0101 0101 0000	0550
+25 °C	0000 0001 1001 0000	0190
+0,5 °C	0000 0000 0000 1000	0008
0 °C	0000 0000 0000 0000	0000
-0,5 °C	1111 1111 1111 1000	FFF8
-25 °C	1111 1110 0111 0000	FE70
-55 °C	1111 1100 1001 0000	FC90

Tabelle 4.10: Temperaturcodierung beim digitalen Thermometer (DS18S20)

Für weitergehende Informationen über die DS1820 Reihe von Dallas sei auf die entsprechenden Datenblätter verwiesen.

## 4.11 Schalter

Geschaltete Ausgänge dienen der Ansteuerung einfacher Geräte, die lediglich die beiden Zustände aus- bzw. eingeschaltet kennen. Werden hierüber nicht nur einfache LEDs (Light Emitting Diode) gesteuert, dann im Allgemeinen Relais, die wiederum die eigentlichen Geräte schalten.

Um diese Komponente nutzen zu können, ist die entsprechende Option in der Software zu aktivieren (siehe Kapitel 3.6 Software-Management).



Zur Pin-Konfiguration sei auf das Kapitel 3.4 Externe Beschaltung verwiesen, wobei die Wahl des Pins keiner Restriktion unterliegt.

Ein geschalteter Ausgang kann Werte zwischen 0 und 100 annehmen – die Voreinstellung, also nach dem Einschalten einer Control Unit, ist 0, wobei jeder Wert größer 0 zum Einschalten des entsprechenden Ausgangs führt und einzig der Wert 0 den Ausgang ausschaltet.

Der Wert eines geschalteten Ausgangs kann über einen Steuerbefehl festgelegt werden (siehe Kapitel 3.1.7 Steuerbefehle). Für Value sind dabei die Werte von 0 bis 100 und der Wert 128 zulässig, der zum Invertieren des aktuellen Werts führt, das heißt, ist der aktuelle Wert größer 0, dann nimmt der Ausgang den Wert 0 an, andernfalls wird der Ausgang auf 100 gesetzt. LParam und HParam sind beide 0.

Wird der Wert eines geschalteten Ausgangs neu festgelegt – nicht unbedingt verändert, so generiert der Ausgang eine Statusmeldung (siehe Kapitel 3.1.9 Statusmeldungen). LValue ist dann der aktuelle Wert des Ausgangs, HValue und Ext sind 0.

Der aktuelle Wert eines geschalteten Ausgangs lässt sich auf herkömmliche Weise abfragen (siehe Kapitel 3.1.8 Abfragen). LValue enthält dann wieder den aktuellen Wert, HValue und Select sind 0.

## **4.12 Dimmer**

Gedimmte Ausgänge sind so konzipiert, dass sie immer nur einen bestimmten einstellbaren Prozentsatz der gesamten, einer Leuchte zur Verfügung stehenden Energie, dieser auch wirklich zukommen lassen. Dies erfordert die Beschaltung des entsprechenden Ausgangs mit einer Phasenanschnitt- bzw. einer Phasenabschnittsteuerung. Die Control Unit bzw. die gedimmten Ausgänge übernehmen dann die adäquate Ansteuerung der externen Beschaltung.

Um diese Komponente nutzen zu können, ist die entsprechende Option in der Software zu aktivieren (siehe Kapitel 3.6 Software-Management).

Zur Pin-Konfiguration sei auf das Kapitel 3.4 Externe Beschaltung verwiesen, wobei die Wahl des Pins keiner Restriktion unterliegt. Es muss jedoch sichergestellt sein, dass eine Nulldurchgangserkennung der Netzspannung stattfindet und auch

entsprechend konfiguriert ist (siehe auch hierzu Kapitel 3.4 Externe Beschaltung).

Ein gedimmter Ausgang kann Werte zwischen 0 und 100 annehmen, wobei diese Werte direkt mit dem Prozentsatz an Energie, die einem angeschlossenen Verbraucher zukommt, korrelieren. Die Voreinstellung, also nach dem Neustart der Control Unit, ist 0.

Der Wert eines gedimmten Ausgangs kann über einen Steuerbefehl festgelegt werden (siehe Kapitel 3.1.7 Steuerbefehle). Für Value sind dabei zunächst einmal die Werte zwischen 0 und 100 – siehe oben – zulässig. LParam und HParam bilden zusammen einen 16-Bit Wert, der die Zeit in Zehntelsekunden festlegt, in der die Helligkeit einer Leuchte vom aktuellen Wert auf den neuen geregelt wird. Gültige Werte sind hier 0 bis 6553.

Ist Value gleich 128 wird der aktuelle Wert des Dimmers invertiert. Das heißt, ist der aktuelle Wert größer 0, wird er auf 0 gesetzt. Andernfalls wird der Wert eingestellt, den der Dimmer hatte, bevor er ausgeschaltet wurde. Dies geschieht alles ohne Zeitverzögerung.

Ist Value gleich 129, so wird der aktuelle Wert des Dimmers um 2 % erhöht, ist er gleich 130 wird der Wert entsprechend erniedrigt. Auch hier gibt es keine Zeitverzögerung. Diese Funktion wird vor allem verwendet, wenn auf der Infrarot-Fernbedienung die Plus bzw. Minus Taste betätigt wird.

Es gibt vier weitere spezielle Steuercodes, die Verwendung finden, wenn ein Dimmer über einen Taster gesteuert wird. Ist Value gleich 133, so wird vom aktuellen Wert bis auf 100 % gedimmt, ist Value gleich 134, so wird analog auf 0 % gedimmt. Die Geschwindigkeit dieser Dimmvorgänge kann festgelegt werden, indem man die Zeit in Zehntelsekunden (Time) für einen vollständigen Dimmvorgang – also von 0 % auf 100 % bzw. umgekehrt – über folgenden Befehl konfiguriert:

VLAN Source Dest 76 36 Time 0 0

Die Voreinstellung ist 60, also 6 s. Der höchste zu vergebende Wert ist 255. Diese Einstellung gilt für alle auf der Control Unit konfigurierten Dimmer. Setzt man Value auf 135, so wird ein zuvor eingeleiteter Dimmvorgang wieder gestoppt. Mit Value gleich 136 werden die selben Vorgänge initiiert, wie bei Value gleich 133 und 134, nur immer abwechselnd. Wurde also zuletzt hochgedimmt, so wird jetzt

herabgedimmt und umgekehrt.

Wird der Wert eines gedimmten Ausgangs neu festgelegt – nicht unbedingt verändert, so generiert der Ausgang eine Statusmeldung (siehe Kapitel 3.1.9 Statusmeldungen). Dies gilt insbesondere auch dann, wenn ein Dimmvorgang durch den Steuercode 135 (siehe voriger Abschnitt) abgebrochen wurde. LValue ist dann der aktuelle Wert des Ausgangs, HValue und Ext sind 0.

Der aktuelle Wert eines gedimmten Ausgangs lässt sich auf herkömmliche Weise abfragen (siehe Kapitel 3.1.8 Abfragen). LValue enthält dann wieder den aktuellen Wert, HValue und Select sind 0.

Ist bei der Pin-Konfiguration (siehe Kapitel 3.4 Externe Beschaltung) die Option Soft-Delay aktiviert worden, so dauert jeder Dimmvorgang mindestens 3 Zehntelsekunden, egal ob eine kürzere Zeit in dem entsprechenden Steuerbefehl hinterlegt wurde. Es kann so ein smartes Ein- bzw. Ausschalten gewährleistet werden.

Die Option Lange Zündimpulse bewirkt, dass die Triacs der externen Beschaltung nicht mit kurzen Impulsen gezündet werden, sondern die Impulslänge bis zum nächsten Nulldurchgang der Netzspannung ausgedehnt wird. Ist die Option Phasenabschnittsteuerung (siehe unten) aktiviert, darf diese Option nicht aktiviert werden. Ist diese Option nicht aktiviert kann insbesondere die Länge der Zündimpulse über den Befehl

VLAN Source Dest 76 37 LTics HTics 0

eingestellt werden. LTics und HTics bilden dabei zusammen einen 16-Bit Wert, der die Länge der Zündimpulse in Tics angibt. Eine Halbwelle besteht aus 15000 Tics. Die Voreinstellung ist 60. Diese Einstellung gilt für alle auf der Control Unit konfigurierten Dimmer.

Eine weitere Option ist die Schaltrestriktion. Ein gedimmter Ausgang verhält sich bei deren Aktivierung genau wie ein geschalteter Ausgang. Ist der Wert des Ausgangs größer 0, so wird der Ausgang eingeschaltet, also auf 100 % gesetzt. So lässt sich der sichere Betrieb von Geräten gewährleisten, die nicht gedimmt werden können, aber dennoch über eine Dimmerstufe angesteuert werden können, wie z. B. Fernseher.

Die Option Phasenabschnittsteuerung muss aktiviert werden, wenn es sich bei der externen Beschaltung um eben eine solche handelt. Andernfalls wird von einer Beschaltung mit einer Phasenanschnittsteuerung ausgegangen.

#### **4.13 Abstraktes Modul**

In den vorigen Kapiteln wurden Software Komponenten beschrieben, die zur Ansteuerung einfacher Geräte, wie Leuchten oder Fernseher, bzw. zur Erfassung von Messwerten über einfache Sensoren, wie Helligkeitssensor oder Temperaturfühler, über entsprechende externe Beschaltungen genutzt werden können. Daneben gibt es Komponenten, die zur Steuerung komplexer Geräte, wie Rollläden oder Drehgeber genutzt werden. Diese unterscheiden sich von den Komponenten zur Steuerung einfacher Geräte dadurch, dass sie diese Geräte nicht direkt steuern, sondern sich eben dieser einfachen Komponenten bzw. Geräten bedienen.

Diese Geräte werden also nicht mit einem konkreten Pin verbunden, sondern existieren als rein logische bzw. abstrakte Objekte innerhalb einer Control Unit. Zur Verwaltung und zum Management dieser Objekte, sowie zu deren Konfiguration wird diese Komponente benötigt.

Es können bis zu 4 abstrakte Objekte konfiguriert werden. Mit folgendem Befehl wird ein neues abstraktes Objekt angelegt:

VLAN Source Dest 96 0 Addr Type SModul

Addr ist die Geräte-Adresse des abstrakten Objekts, Type der Typ (Code + Offset) des Objekts gemäß unten stehender Tabelle und SModul die Moduladresse, an die das Objekt seine Statusmeldungen verschickt. Ist diese 0, wird das Versenden jeglicher Statusmeldungen unterbunden. Entspricht die Adresse der Moduladresse des Moduls, auf dem das Objekt konfiguriert ist, wird die Statusmeldung nur lokal verarbeitet und nicht versendet. Soll eine Statusmeldung lediglich von der Central Control Unit verarbeitet werden, so wird deren Adresse hier eingesetzt. Analoges gilt für jede andere Control Unit. Broadcasts und Multicasts können hier in gewohnter Weise verwendet werden.

Code	Offset	Objekt-Typ
192	0	Rollladensteuerung
224	0	Drehgeber
240	0	GUI (Graphical User Interface)

*Tabelle 4.11: Typen abstrakter Objekte*

Die Konfiguration aller anderen Eigenschaften, die ein abstraktes Objekt haben kann, wird in den entsprechenden Kapiteln beschrieben.

## 4.14 Rollladensteuerung

Diese Komponente dient der Ansteuerung eines Rollladens, einer Leinwand oder eines in ihrer Steuerfunktion vergleichbaren Geräts.

Um diese Komponente nutzen zu können, ist die entsprechende Option in der Software zu aktivieren (siehe Kapitel 3.6 Software-Management).

Die Grundkonfiguration ist gemäß Kapitel 4.13 Abstraktes Modul vorzunehmen. Eine Rollladensteuerung basiert auf zwei elementaren Steuerungen zum Betrieb des entsprechenden Motors in Auf- und Abwärtsrichtung. Hier können sowohl geschaltete als auch gedimmte Ausgänge eingesetzt werden, sofern die Parametrierung auf das zu steuernde Gerät passt (siehe Kapitel 4.11 Schalter und 4.12 Dimmer). Die Rollladensteuerung schaltet den entsprechenden Ausgang solange ein, bis der eingestellte Wert für den Rollladen erreicht ist. Mit dem Befehl

VLAN Source Dest 96 RSAddr 0 Modul Addr

wird der Ausgang konfiguriert, der den Rollladenmotor zur Aufwärtsfahrt aktiviert. RSAddr ist dabei die Geräte-Adresse des Rollladens, so wie sie beim Anlegen des Rollladenobjekts vergeben wurde. Modul ist die Modul-Adresse des Ausgangs und Addr die Geräte-Adresse desselben. Analog wird der Ausgang zur Abwärtsfahrt des Rollladens wie folgt konfiguriert:

VLAN Source Dest 96 RSAddr 1 Modul Addr

Damit die Rollladensteuerung errechnen kann, wie lange der Motor in die entsprechende Richtung angesteuert werden muss, um die vorgegebene Position zu

erreichen, ist die maximale Fahrzeit des Rollladens (von 100 % auf 0 % - also eine Aufwärtsfahrt, die Abwärtsfahrt dauert gegebenenfalls, bedingt durch das Gewicht des Rollladens, nicht ganz so lange) zu konfigurieren. Dies geschieht mit dem Befehl

VLAN Source Dest 96 RSAddr 128 Time 0

Time gibt dabei die maximale Fahrzeit in Fünftelsekunden an. Mit dem Befehl

VLAN Source Dest 96 RSAddr 129 Type 0

wird der Typ des Rollladens festgelegt. Von Type wird nur Bit 0 verwendet. Ist es nicht gesetzt, verhält sich die Steuerung wie oben beschrieben. Andernfalls werden die Ausgänge über Impulse der Länge 3/5 s gesteuert. Das heißt, die Steuerlogik des Rollladenmotors bekommt einen Impuls zum Anfahren und einen weiteren, wenn der Motor wieder gestoppt werden soll.

Ein Rollladen kann Werte zwischen 0 und 100 annehmen, wobei diese Werte dem Prozentsatz der Geschlossenheit entsprechen, das heißt, 100 % steht für einen geschlossenen und 0 % für einen vollständig eingefahrenen Rollladen. Die Voreinstellung, also nach dem Neustart der Control Unit, ist 0.

Der Wert eines Rollladens kann über einen Steuerbefehl festgelegt werden (siehe Kapitel 3.1.7 Steuerbefehle). Für Value sind dabei zunächst einmal die Werte zwischen 0 und 100 – siehe oben – zulässig. LParam und HParam sind 0.

Ist Value gleich 128 wird der aktuelle Wert des Rollladens invertiert. Das heißt, ist der aktuelle Wert größer 0, wird er auf 0 gesetzt, der Rollladen also vollständig eingefahren. Andernfalls wird der Wert auf 100 gesetzt, der Rollladen also vollständig geschlossen. Erreicht der Steuerbefehl den Rollladen, während dieser gerade in Bewegung ist, so wird er gestoppt.

Es gibt vier weitere spezielle Steuercodes, die Verwendung finden, wenn ein Rollladen über einen Taster gesteuert wird. Ist Value gleich 133, so wird er vom aktuellen Wert bis auf 100 % gefahren, ist Value gleich 134, so wird analog auf 0 % gefahren. In beiden Fällen gilt, dass wenn der Rollladen beim Empfang des Steuerbefehls in Bewegung war, die Fahrt gestoppt wird. Setzt man Value auf 135, so wird eine zuvor eingeleitete Fahrt wieder gestoppt. Mit Value gleich 136 werden die selben Vorgänge initiiert, wie bei Value gleich 133 und 134, nur immer

abwechselnd. Wurde also zuletzt hochgefahren, so wird jetzt heruntergefahren und umgekehrt.

Wird der Wert eines Rollladens neu festgelegt – nicht unbedingt verändert, so generiert der Ausgang eine Statusmeldung (siehe Kapitel 3.1.9 Statusmeldungen). Dies gilt insbesondere auch dann, wenn eine Fahrt durch den Steuercode 135 (siehe voriger Abschnitt) abgebrochen wurde. LValue ist dann der aktuelle Wert des Rollladens, HValue und Ext sind 0.

Der aktuelle Wert eines Rollladens lässt sich auf herkömmliche Weise abfragen (siehe Kapitel 3.1.8 Abfragen). LValue enthält dann wieder den aktuellen Wert, HValue und Select sind 0.

Wird versucht, den Wert eines Rollladens neu festzulegen, während dieser sich in Bewegung befindet – oben beschriebene spezielle SteuerCodes sind davon ausgenommen, wird eine „Device Busy“ Fehlermeldung generiert (siehe Kapitel 3.1.11 Fehlerbehandlung).

## **4.15 Drehgeber**

Diese Komponente unterstützt den Betrieb von Drehgebern, wie sie im Allgemeinen als Eingabegerät im Zusammenhang mit GUIs (Graphical User Interfaces) eingesetzt werden. Sie liefern unterschiedliche Signale, wenn man in verschiedene Richtungen dreht oder den Taster betätigt. Diese Software-Komponente wandelt diese Signale in solche um, wie sie zur Weiterverarbeitung durch andere Geräte oder Bedienoberflächen benötigt werden.

Um diese Komponente nutzen zu können, ist die entsprechende Option in der Software zu aktivieren (siehe Kapitel 3.6 Software-Management). Insbesondere ist dabei auf den richtigen Typ von Drehgeber zu achten.

Die Grundkonfiguration ist gemäß Kapitel 4.13 Abstraktes Modul vorzunehmen. Ein Drehgeber hat 3 Anschlüsse, zwei für die Signale zur Erfassung der Drehung und einen für die Tastfunktion, hier willkürlich mit A, B und T bezeichnet. Zur Erfassung dieser Signale wird für jeden Anschluss ein logischer Eingang konfiguriert (siehe Kapitel 4.8 Logische Eingänge). Dem Drehgeber müssen diese Eingänge dann zugeordnet werden. Hierzu wird zunächst die Modul-Adresse der Control Unit

konfiguriert, an der der Drehgeber angeschlossen ist.

VLAN Source Dest 96 DGAddr 0 Modul 0

DGAddr ist die Geräte-Adresse des Drehgebers, so wie sie bei der Grundkonfiguration vergeben wurde und Modul die zu konfigurierende Modul-Adresse. Es folgen der Reihe nach die Geräte-Adressen der logischen Eingänge A, B und T.

VLAN Source Dest 96 DGAddr 1 AddrA 0

VLAN Source Dest 96 DGAddr 2 AddrB 0

VLAN Source Dest 96 DGAddr 3 AddrT 0

AddrX ist dabei die jeweils zu konfigurierende Geräte-Adresse. Schließlich muss noch das Gerät spezifiziert werden, das durch den Drehgeber angesteuert werden soll, also im Allgemeinen eine Bedienoberfläche. Hierzu wird der Befehl

VLAN Source Dest 96 DGAddr 4 Modul Addr

verwendet, wobei Modul die Modul-Adresse und Addr die Geräte-Adresse des anzusteuernenden Geräts ist.

Die logischen Eingänge der Anschlüsse A und B sind für Steigende Flanke, Fallende Flanke und Prellfreiheit zu konfigurieren. Der logische Eingang des Anschluss T ist so zu konfigurieren, dass sich das gewünschte Verhalten ergibt, es muss jedoch mindestens die Prellfreiheit konfiguriert werden. Ferner ist darauf zu achten, dass die Statusmeldungen, die von den logischen Eingängen verschickt werden, den Drehgeber auch erreichen, die entsprechende Modul-Adresse für diese Meldungen also korrekt konfiguriert wurde.

Detektiert der Drehgeber ein Ereignis – eine Drehung oder einen Tastendruck, so gibt er dieses Ereignis als Steuerbefehl (siehe Kapitel 3.1.7 Steuerbefehle) an das konfigurierte Ausgabegerät weiter. Ist das Ereignis eine Drehung, wird als Value die Drehrichtung angegeben (Links = 137, Rechts = 138), in LParam wird die Geschwindigkeit der Drehung hinterlegt (siehe unten) und HParam ist 0. Handelt es sich bei dem Ereignis um einen Tastendruck, steht in Value die Länge des Tastendrucks (Kurz = 139, Medium = 140, Lang = 141) und LParam und HParam



sind 0.

Die Drehgeschwindigkeit des Drehgebers kann individuell angepasst werden. Mit dem Befehl

```
VLAN Source Dest 96 DGAddr 5 Speed 0
```

lässt sie sich festlegen, wobei Speed eben den Wert der Drehgeschwindigkeit spezifiziert. Bei der Detektion eines Drehereignisses setzt der Drehgeber einen internen Zähler auf diesen konfigurierten Wert. Dann wird dieser alle Hundertstelsekunde decrementiert, bis ein weiteres Drehereignis detektiert wird oder der Zähler 0 geworden ist. Dieser Zählerstand wird dann, wie oben beschrieben, in LParam hinterlegt. Die Interpretation dieses Werts obliegt dann dem Zielgerät. Insofern sei hier auf die entsprechenden Kapitel verwiesen.

## **4.16 Bedienoberfläche**

Bei der Komponente Bedienoberfläche handelt es sich um eine frei konfigurierbare GUI (Graphical User Interface). Als Eingabegerät kommt ein Drehgeber zum Einsatz (siehe Kapitel 4.15 Drehgeber) und zur Ausgabe wird ein LCD-Display verwendet (siehe Kapitel 4.7 LCD-Display). Die GUI bietet die Möglichkeit, jedes beliebige Gerät zu steuern, Sensoren abzufragen, Regelungen zu visualisieren, wie z. B. Thermostate oder auch komplexe Steueraufgaben zu initiieren. Hierzu stellt die GUI verschiedene konfigurierbare Controls zur Verfügung, die über ein ebenfalls konfigurierbares Menü, was selbst auch wiederum ein Control darstellt, aufgerufen werden können.

Generell wird das Drehen am Drehgeber oder das Betätigen eines Tasters von der Signaleinrichtung mit einem Ack bestätigt. Ein Ack wird ebenfalls ausgelöst, wenn das Ansteuern eines Geräts erfolgreich war, andernfalls erfolgt ein Error. Dies alles gilt natürlich nur dann, wenn der Buzzerlevel entsprechend konfiguriert wurde (siehe Kapitel 4.3 Buzzer).

Um diese Komponente nutzen zu können, ist die entsprechende Option in der Software zu aktivieren (siehe Kapitel 3.6 Software-Management).

Die Grundkonfiguration ist gemäß Kapitel 4.13 Abstraktes Modul vorzunehmen. Die eigentlichen Konfigurationsdaten werden bedingt durch ihre Menge nicht auf

herkömmliche Weise behandelt, sondern als Datei zusammengefasst und in eines der externen EEPROMs geladen (siehe hierzu Kapitel 4.16.4 Konfigurationsdownload). Der Aufbau und die Struktur der Konfigurationsdatei wird im Folgenden behandelt, wobei zunächst auf allgemeine Definitionen und die globale Struktur eingegangen wird und die Details jeweils in den folgenden Unterkapiteln beschrieben werden, in denen die Funktionsweise der einzelnen Controls näher erläutert wird.

Die Konfigurationsdatei wird in ihrer hexadezimalen Darstellung in BNF (Backus-Naur-Form) beschrieben, wobei zunächst einige elementare Definitionen vorgenommen werden.

```
<Nibble> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F
<Byte> ::= <Nibble> <Nibble>
<Word> ::= <Byte> <Byte>
<Addr8> ::= <Byte>
<Addr16> ::= <Word>

<EString> ::= <CharCount> [<CharList>]
<CharList> ::= <Char> | <Char> <CharList>
<CharCount> ::= <Byte>
<Char> ::= <Byte>
```

Ein EString besteht also aus einer Liste von Zeichen, denen die Anzahl der Zeichen in der Liste vorangestellt wird. Bei einem Zeichen handelt es sich um den ASCII-Code eines auf dem LCD darstellbaren Zeichens.

```
<Config> ::= <RootCtrlAddr> <StdCtrlAddr> <StdCtrlTime> <CtrlList>
<RootCtrlAddr> ::= <Addr16>
<StdCtrlAddr> ::= <Addr16>
<StdCtrlTime> ::= <Byte>
<CtrlList> ::= <Ctrl> | <Ctrl> <CtrlList>
<Control> ::= <Menu> | <OutCtrl> | <Thermostat>
```

Die gesamte Konfiguration besteht also aus zwei Adresseinträgen, einem Zeiteintrag und einer Liste von Steuerelementen. RootCtrlAddr ist die Adresse, innerhalb der Konfiguration, des Steuerelements, welches nach einem Reset als erstes aufgerufen wird. StdCtrlAddr ist die Adresse des Steuerelements, das standardmäßig angezeigt wird. Das heißt, nach einer bestimmten Zeit der Inaktivität, die durch StdCtrlTime in Sekunden definiert ist, wird automatisch zu diesem Steuerelement gesprungen. Gleichzeitig wird die Hintergrundbeleuchtung des Displays wieder ausgeschaltet. Jedes beliebige GUI-Ereignis, z. B. Drehen am Drehgeber, schaltet die Hintergrundbeleuchtung ein. Zur Erläuterung der verschiedenen Steuerelemente sei

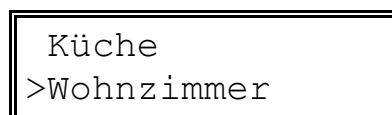
auf die nachfolgenden Unterkapitel verwiesen.

#### 4.16.1 Menü

Ein Menü ist eine Liste von Menüeinträgen, von denen jeder ein weiteres Steuerelement – welches auch wieder ein Menü sein kann – aufruft. Jeder Eintrag wird in einer Zeile des LCD dargestellt und der gerade selektierte durch einen Pfeil gekennzeichnet. Durch Rechtsdrehung am Drehgeber wird der nächste Menüeintrag selektiert, durch Linksdrehen der vorherige. Ist das Display für die Darstellung des gesamten Menüs zu klein, wird entsprechend gescrollt. Ein kurzer Tastendruck am Drehgeber bewirkt den Aufruf des Steuerelements, das mit dem gerade selektierten Eintrag verbunden ist.

Die Reihenfolge der einzelnen Menüaufrufe wird intern in einer History protokolliert, um bei einem Rücksprung zum aufrufenden Menü dessen letzten Status wieder rekonstruieren zu können. Die Länge der History ist auf 7 begrenzt. Das heißt, bei der Konfiguration der Menüstruktur ist darauf zu achten, dass jedes Steuerelement über maximal 7 Menüaufrufe erreicht werden kann. Ist die History erschöpft, ist kein weiterer Aufruf eines Steuerelements mehr möglich und die Signaleinrichtung meldet einen Fehler.

Eine beispielhafte Darstellung eines Menüs auf einem zweizeiligen LCD zeigt folgendes Bild:



*Abb. 4.6: Steuerelement Menü*

Die Konfiguration eines Menüs besitzt nachfolgende Struktur und Bedeutung:

```
<Menue> ::= <MCType> <MItemCount> <MItemList>
<MCType> ::= 01
<MItemCount> ::= <Byte>
<MItemList> ::= <MItem> | <MItem> <MItemList>
<MItem> ::= <MName> <MCtrlAddr>
<MName> ::= <EString>
<MCtrlAddr> ::= <Addr16>
```

Ein Menü besteht also aus dem Typ des Steuerelements der Anzahl der

Menüeinträge und der Liste der Menüeinträge selbst. Ein Menüeintrag wiederum besteht aus einem Bezeichner oder Namen, der höchstens 14 Zeichen lang sein darf und der Adresse, innerhalb der Konfiguration, des Steuerelements, welches durch den Menüeintrag aufgerufen werden soll.

#### 4.16.2 Steuerung

Dieses Steuerelement dient der Steuerung aller herkömmlicher Geräte. Darunter werden Geräte verstanden, deren Wert zwischen 0 % und 100 % einstellbar ist, also auch Schalter. Wird ein solches Steuerelement aufgerufen, ergibt sich bei einem zweizeiligen LCD zunächst folgende Anzeige:



Abb. 4.7: Steuerelement Steuerung 1

Im Hintergrund wird nun eine Abfrage auf das entsprechende Gerät generiert. Wird der aktuelle Wert des Geräts an die Bedienoberfläche zurückgeliefert, könnte die Anzeige wie folgt aussehen:

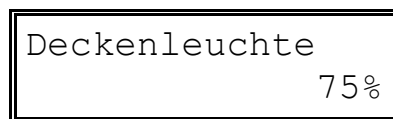


Abb. 4.8: Steuerelement Steuerung 2

In den meisten Fällen geschieht die Abfrage so schnell, dass die erste Anzeige nicht bewusst wahrgenommen wird. Ist die Abfrage erfolglos, bleibt die Anzeige auf erstem Bild stehen. Resultiert die fehlgeschlagene Abfrage aus dem Ausbleiben einer Antwort von einem entfernten Modul, wird ein Protokollfehler generiert (siehe Kapitel 3.1.11 Fehlerbehandlung).

Durch Drehen am Drehgeber wird das Steuerelement in den Bearbeitungsmodus gesetzt und es kann ein neuer Wert für das Gerät eingestellt werden. Dieser neue Wert wird parallel zum aktuellen im Display angezeigt.

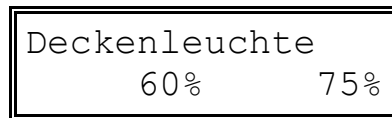


Abb. 4.9: Steuerelement Steuerung 3

Die Geschwindigkeit, mit der sich der neue Wert verändert, ist abhängig von der Drehgeschwindigkeit des Drehgebers. Der Wert für die Geschwindigkeit, den der Drehgeber an die GUI übermittelt (siehe Kapitel 4.15 Drehgeber), wird durch 10 geteilt, dann 1 addiert und bei einer Linksdrehung vom neuen Wert abgezogen und bei einer Rechtsdrehung zum neuen Wert addiert.

Ein kurzes Betätigen des Tasters weist nun die GUI an, den neu eingestellten Wert zu übernehmen und in den Normalmodus zurückzukehren. Ist das zu steuernde Gerät an der gleichen Control Unit angeschlossen, auf der auch die GUI läuft, erscheint der neu eingestellte Wert unten rechts in der Anzeige. Andernfalls werden zunächst wieder Bindestriche angezeigt und erst nach dem die Bestätigung von der entfernten Control Unit empfangen wird, werden die Bindestriche durch den neuen Wert ersetzt. Bleibt die Bestätigung aus, wird ein Protokollfehler generiert (siehe Kapitel 3.1.11 Fehlerbehandlung). Wird ein Tastendruck der Länge Medium erzeugt, wird in den Normalmodus gewechselt, ohne den neu eingestellten Wert zu übernehmen.

Befindet sich das Steuerelement im Normalmodus und der Taster wird kurz betätigt, so wird zum aufrufenden Steuerelement zurückgekehrt.

Die Konfiguration eines solchen Steuerelements ist wie folgt aufgebaut:

```
<OutCtrl> ::= <OCCType> <OCModul> <OCDevice> <OCName>
<OCCType> ::= 10
<OCModul> ::= <Addr8>
<OCDevice> ::= <Addr8>
<OCName> ::= <EString>
```

Sie besteht also aus dem Typ des Steuerelements, der Modul-Adresse der Control Unit, an der das zu steuernde Gerät angeschlossen ist, der Geräte-Adresse des zu steuernden Geräts und dem Namen, der im Display angezeigt werden soll.

### 4.16.3 Thermostat

Das Steuerelement Thermostat dient der Regelung eines Heizungsthermostats. Über einen Temperatursensor wird die Raumtemperatur erfasst und über eine autonome Steuerung ein Stellantrieb angesteuert. Hierbei wird die Funktionalität der Trigger der analogen und digitalen Eingänge (siehe Kapitel 4.9 Analoge Eingänge und 4.10 Digitale Eingänge) verwendet. Das Steuerelement dient der Anzeige der aktuellen Temperatur und dem Einstellen der entsprechenden Trigger. Nach dem Aufruf des Thermostats könnte die Anzeige auf einem zweizeiligen Display wie folgt aussehen:

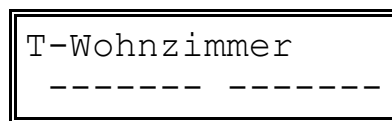


Abb. 4.10: Steuerelement Thermostat 1

Im Hintergrund wird nun der aktuelle Wert und der Triggerwert vom Temperatursensor abgefragt. Bei einer erfolgreichen Abfrage könnte die Anzeige dann so aussehen:

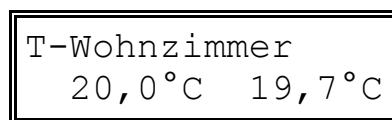


Abb. 4.11: Steuerelement Thermostat 2

In der rechten Ecke findet sich der aktuelle Temperaturwert und links der Sollwert, also der Triggerwert. In den meisten Fällen geschieht die Abfrage so schnell, dass die erste Anzeige nicht bewusst wahrgenommen wird. Ist die Abfrage erfolglos, bleibt die Anzeige auf erstem Bild stehen. Resultiert die fehlgeschlagene Abfrage aus dem Ausbleiben einer Antwort von einem entfernten Modul, wird ein Protokollfehler generiert (siehe Kapitel 3.1.11 Fehlerbehandlung).

Durch Drehen am Drehgeber wird das Steuerelement in den Bearbeitungsmodus gesetzt und der Sollwert kann entsprechend angepasst werden. Es könnte sich dann folgende Anzeige ergeben:

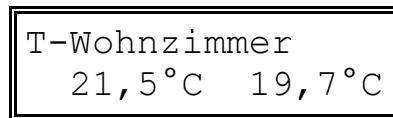


Abb. 4.12: Steuerelement Thermostat 3

Die Geschwindigkeit, mit der sich der Sollwert verändert, ist abhängig von der Drehgeschwindigkeit des Drehgebers. Der Wert für die Geschwindigkeit, den der Drehgeber an die GUI übermittelt (siehe Kapitel 4.15 Drehgeber), wird durch 10 geteilt, dann 1 addiert, nochmals durch zwei geteilt und bei einer Linksdrehung vom Sollwert abgezogen und bei einer Rechtsdrehung zum Sollwert addiert.

Ein kurzes Betätigen des Tasters weist nun die GUI an, den neuen Sollwert zu übernehmen und in den Normalmodus zurückzukehren. Ist der zum Temperatursensor gehörende Eingang an der gleichen Control Unit angeschlossen, auf der auch die GUI läuft, bleibt der neue Sollwert unverändert unten links in der Anzeige stehen. Andernfalls werden zunächst wieder Bindestriche angezeigt und erst nach dem die Bestätigung von der entfernten Control Unit empfangen wird, werden die Bindestriche durch den neuen Sollwert ersetzt. Bleibt die Bestätigung aus, wird ein Protokollfehler generiert (siehe Kapitel 3.1.11 Fehlerbehandlung). Wird ein Tastendruck der Länge Medium erzeugt, wird in den Normalmodus gewechselt, ohne den neuen Sollwert zu übernehmen.

Befindet sich das Steuerelement im Normalmodus und der Taster wird kurz betätigt, so wird zum aufrufenden Steuerelement zurückgekehrt.

Die Konfiguration eines Thermostats ist wie folgt aufgebaut:

```
<Thermostat> ::= <TSCType> <TSModul> <TSDevice> <TSRefresh> <TSName>
<TSCType> ::= 20
<TSModul> ::= <Addr8>
<TSDevice> ::= <Addr8>
<TSRefresh> ::= <Byte>
<TSName> ::= <EString>
```

Sie besteht also aus dem Typ des Steuerelements, der Modul-Adresse der Control Unit, an die der Temperatursensor angeschlossen ist, der Device-Adresse des Temperatursensors, der Zeit in Sekunden, in der die Temperaturanzeige aktualisiert werden soll und dem Namen, der im Display angezeigt werden soll.

#### 4.16.4 Konfigurationsdownload

Der Download der Konfigurationsdatei wird mit dem folgenden Befehl eingeleitet:

```
VLAN Source Dest 56 0 32 0 0
```

Dieser Download erfolgt seitenbasiert, das heißt, es werden immer ganze Seiten von 32 Byte Größe übertragen. Eine nicht vollständig gefüllte Seite wird mit 0xFF aufgefüllt. Die Reihenfolge, in der die Seiten übertragen werden, ist prinzipiell egal. Eine Seite kann auch mehrfach übertragen werden, wenn z. B. beim vorherigen Übertragungsversuch ein Fehler aufgetreten ist. Um nun eine einzelne Seite zu downloaden wird zunächst ein Startpaket geschickt:

```
VLAN Source Dest 56 16 168 LSideAddr HSideAddr
```

LSideAddr und HSideAddr bilden dabei einen 16-Bit Wert, der die Startadresse der Seite in Byte angibt. Die erste Seite hat die Adresse 0x0000, die zweite Seite die Adresse 0x0020 usw.. Jetzt folgen 8 Datenpakete – 8 Pakete x 4 Byte = 32 Byte = 1 Seite – in der Form

```
VLAN Source Dest 60 Data0 Data1 Data2 Data3
```

```
VLAN Source Dest 60 Data4 Data5 Data6 Data7
```

```
VLAN Source Dest 60 Data8 Data9 Data10 Data11 usw.
```

Um die Übertragung einer Seite abzuschließen, folgt ein Paket mit Prüfsummen:

```
VLAN Source Dest 60 Checksum0 Checksum1 Checksum2 Checksum3
```

Sind die Prüfsummen korrekt wird dieses Paket mit einem Acknowledge beantwortet, ansonsten mit einem Error. Das Startpaket und die Datenpakete erhalten jeweils keine Bestätigung. Nun kann die nächste Seite übertragen werden oder aber die vorherige noch einmal, sollte das Prüfsummenpaket mit einem Fehler oder gar nicht bestätigt werden.

Die Prüfsummen werden über die Daten-Bytes gebildet, also über Device und Value0 bis Value2 und zwar so, dass die Summe aller vorherigen 8 Datenpakete plus das Seitenstartpaket plus das Prüfsummenpaket modulo 256 Null ergibt. In Device werden also die Device-Werte summiert, in Value0 die Value0-Werte usw.. Dabei wird das Startpaket, die 8 Datenpakete und das Prüfsummenpaket mit einbezogen.



Abgeschlossen wird der Firmware-Download mit einem Download-Ende-Paket:

VLAN Source Dest 56 1 0 0 0

Nach dem Download der Konfigurationsdatei ist sofort ein Reset auszuführen, um die neue Konfiguration zu aktivieren und zu verhindern, dass die Control Unit durch das Ändern der Konfiguration in einen nicht definierten Zustand gerät.

## **4.17 Autonome Steuerung**

Bei der autonomen Steuerung handelt es sich um eine Komponente, die es ermöglicht, komplexe Steuer- und Regelaufgaben völlig autonom, das heißt ohne Hilfe der Central Control Unit, durchzuführen. Diese Steuerungen und Regelkreise lassen sich dabei nach dem Baukastenprinzip zusammensetzen. Das bedeutet, es gibt eine ganze Reihe wohl definierter Objekte, die sowohl Eingänge als auch einen Ausgang besitzen und die an den Eingängen anliegenden Werte nach einem festgelegten Schema in einen bestimmten Ausgangswert verwandeln. Indem man nun mehrere dieser Objekte zu Netzwerken verbindet, einige davon mit Eingangswerten versorgt, die man z. B. über logische, analoge oder digitale Eingänge bezieht, kann man den durch das Objektnetzwerk berechneten Ausgangswert zur Ansteuerung z. B. eines Dimmers oder Rollladens benutzen. Die Eingänge und auch die anzusteuernenden Geräte müssen dabei nicht unbedingt auf der selben Control Unit betrieben werden. Die Anzahl der zu konfigurierenden Objektnetzwerke ist dabei nur durch die maximale Anzahl der Objekte (64) begrenzt, die gleichzeitig auf einer Control Unit konfiguriert werden können.

Grundsätzlich haben die Werte, die durch die Objekte berechnet werden die Größe von einem Byte. Damit ist der Wertebereich auf das Intervall von 0 bis 255 festgelegt, wobei die 0 auch als Wahrheitswert False und die 255 als Wahrheitswert True interpretiert wird.

Ein Objekt wird durch bis zu vier Eigenschaften charakterisiert, die alle einzeln konfiguriert werden müssen. Die 0-te Eigenschaft spezifiziert dabei den Typ des Objekts. Die Funktionen der Eigenschaften 1 bis 3 sind individuell vom Objekttyp abhängig oder werden unter Umständen auch gar nicht verwendet. Sie lassen sich grundsätzlich in zwei Kategorien ordnen:

- Sie beschreiben einen Eingang eines Objekts. Dann gibt die Eigenschaft an, mit welchem Objekt dieser Eingang verbunden ist.
- Sie beschreiben keinen Eingang eines Objekts. Dann beschreibt die Eigenschaft einen allgemeinen Parameter, der das Objektverhalten in geeigneter Weise bestimmt.

Die gesamte Konfiguration der autonomen Steuerung wird mit einem einzigen Typ von Konfigurationsbefehl vorgenommen:

VLAN Source Dest 100 Object Prop Value Default

Object spezifiziert die Nummer des Objekts, auf welches sich der Konfigurationsbefehl bezieht. Es wird mit 0 beginnend aufsteigend nummeriert. Prop legt die Eigenschaft fest, die konfiguriert werden soll und liegt je nach Objekttyp zwischen 0 und 3. Value gibt schließlich den Wert an, den die spezifizierte Eigenschaft haben soll. Default ist bei dieser Art von Konfigurationsbefehl immer 0. Wird Default auf 255 gesetzt, so werden alle Eigenschaften aller Objekte mit 0 initialisiert.

Im Folgenden wird auf die verschiedenen Objekttypen und deren zugehörige Eigenschaften näher eingegangen:

### **No Operation – Typ 0**

Bei diesem Objekt gibt es keine weiteren Eigenschaften bzw. diese sind nicht relevant. Der Ausgangswert dieses Objekts ist stets 0. Wird ein Objekt nicht explizit konfiguriert, so ist es von diesem Typ. Um den Speicherplatzbedarf zur Laufzeit gering zu halten, sollte möglichst kein Objekt dieses Typs zwischen anderen Objekten in der Reihenfolge stehen. Stehen diese Objekte am Ende der Objektfolge, so werden sie nicht berücksichtigt und es wird auch kein Speicher dafür benötigt.

### **Timer (Minuziöse Ausführung) – Typ 32**

Der Ausgang dieses Objekts nimmt den Wert 255 an, wenn sich die Zeit in einem bestimmten abgeschlossenen minuziösen Intervall befindet. Andernfalls ist der Ausgang 0. Das Intervall wird definiert durch den Startwert innerhalb der Minute und die Länge des Intervalls. Der Startwert errechnet sich aus den

6 niederwertigen Bits der 1-ten Eigenschaft in Sekunden plus den 4 niederwertigen Bits der 2-ten Eigenschaft in 1/10 Sekunden. Die Länge des Intervalls ergibt sich aus den 2 höherwertigen Bits der 1-ten Eigenschaft multipliziert mit 256 plus des Werts der 3-ten Eigenschaft in 1/10 Sekunden.

### **Timer (Stündliche Ausführung) – Typ 33**

Der Ausgang dieses Objekts nimmt den Wert 255 an, wenn sich die Zeit in einem bestimmten abgeschlossenen stündlichen Intervall befindet. Andernfalls ist der Ausgang 0. Das Intervall wird definiert durch den Startwert innerhalb der Stunde und die Länge des Intervalls. Der Startwert errechnet sich aus den 6 niederwertigen Bits der 1-ten Eigenschaft in Minuten plus der 6 niederwertigen Bits der 2-ten Eigenschaft in Sekunden. Die Länge des Intervalls ergibt sich aus den 2 höherwertigen Bits der 1-ten Eigenschaft multipliziert mit 1024 plus den 2 höherwertigen Bits der 2-ten Eigenschaft multipliziert mit 256 plus des Werts der 3-ten Eigenschaft in Sekunden.

### **Timer (Tägliche Ausführung) – Typ 34**

Der Ausgang dieses Objekts nimmt den Wert 255 an, wenn sich die Zeit in einem bestimmten abgeschlossenen täglichen Intervall befindet. Andernfalls ist der Ausgang 0. Das Intervall wird definiert durch den Startwert innerhalb des Tages und die Länge des Intervalls. Der Startwert errechnet sich aus den 5 niederwertigen Bits der 1-ten Eigenschaft in Stunden plus der 6 niederwertigen Bits der 2-ten Eigenschaft in Minuten. Die Länge des Intervalls ergibt sich aus den 2 höherwertigen Bits der 2-ten Eigenschaft multipliziert mit 256 plus des Werts der 3-ten Eigenschaft in Minuten. Dieser Mechanismus funktioniert an Werktagen nur, wenn das Bit 5, an Samstagen, wenn das Bit 6 und an Sonntagen, wenn das Bit 7 der 1-ten Eigenschaft gesetzt ist.

### **Timer (Wöchentliche Ausführung) – Typ 35**

Der Ausgang dieses Objekts nimmt den Wert 255 an, wenn sich die Zeit in einem bestimmten abgeschlossenen wöchentlichen Intervall befindet. Andernfalls ist der Ausgang 0. Das Intervall wird definiert durch den Startwert innerhalb der Woche und die Länge des Intervalls. Der Startwert errechnet sich aus den 3 höherwertigen Bits der 1-ten Eigenschaft in Tagen plus die 5 niederwertigen Bits der 1-ten Eigenschaft in Stunden plus die 6 niederwertigen

Bits der 2-ten Eigenschaft in Minuten. Die Länge des Intervalls ergibt sich aus den 2 höherwertigen Bits der 2-ten Eigenschaft multipliziert mit 256 plus des Werts der 3-ten Eigenschaft in Minuten.

### **Aktiver Eingang – Typ 56**

Der Ausgang dieses Objekts präsentiert das Ergebnis der Abfrage eines bestimmten Gerätes. In der 1-ten Eigenschaft ist dabei die Modul-Adresse des Moduls zu hinterlegen, an dem eben dieses Gerät angeschlossen ist. Die 2-te Eigenschaft gibt die entsprechende Geräte-Adresse an. Die 3-te Eigenschaft definiert in 1/10 Sekunden das Intervall, in dem die Abfrage wiederholt, der Wert des Ausgangs also aktualisiert wird. Wenn insbesondere das Gerät nicht an dem Modul betrieben wird, auf dem die zu konfigurierende autonome Steuerung läuft, empfiehlt es sich, diesen Wert hinreichend groß zu wählen, um den Datenverkehr möglichst klein zu halten. Handelt es sich bei dem abzufragenden Gerät um einen analogen Eingang, so wird der 10 Bit große Ergebniswert auf einen 8 Bit großen gekürzt, indem die 2 niederwertigen Bits nicht beachtet werden. Bei der Abfrage eines digitalen Eingangs werden die 4 niederwertigen Bits nicht beachtet, also die Bits, die die Nachkommastellen repräsentieren.

### **Nicht flüchtiger passiver Eingang – Typ 60**

Der Ausgang dieses Objekts wird durch Empfangen einer Statusmeldung verändert. Die Modul-Adresse der Statusmeldung muss mit der Modul-Adresse des Moduls, auf dem die zu konfigurierende autonome Steuerung läuft, identisch sein oder es muss sich um einen Broad- bzw. Multicast handeln. Zusätzlich muss die Absender-Adresse mit der 1-ten Eigenschaft sowie die Geräte-Adresse der Statusmeldung mit der 2-ten Eigenschaft übereinstimmen. Handelt es sich bei dem sendenden Gerät um einen analogen Eingang, so wird der 10 Bit große Ergebniswert auf einen 8 Bit großen gekürzt, indem die 2 niederwertigen Bits nicht beachtet werden. Sendet die Statusmeldung ein digitaler Eingang, werden die 4 niederwertigen Bits nicht beachtet, also die Bits, die die Nachkommastellen repräsentieren.

Der Ausgang dieses Objekts wird ebenfalls durch Empfangen eines Makros verändert. Die Modul-Adresse des Makros muss mit der Modul-Adresse des

Moduls, auf dem die zu konfigurierende autonome Steuerung läuft, identisch sein oder es muss sich um einen Broad- bzw. Multicast handeln. Zusätzlich muss die Absender-Adresse mit der 1-ten Eigenschaft sowie das High-Byte der Makro-Nummer mit der 2-ten Eigenschaft übereinstimmen. Das Low-Byte der Makro-Nummer ist dann der Ergebniswert.

Nach einem Reset wird der Ausgang mit dem Wert der 3-ten Eigenschaft initialisiert.

### **Flüchtiger passiver Eingang – Typ 61**

Der Ausgang dieses Objekts wird durch Empfangen einer Statusmeldung verändert. Die Modul-Adresse der Statusmeldung muss mit der Modul-Adresse des Moduls, auf dem die zu konfigurierende autonome Steuerung läuft, identisch sein oder es muss sich um einen Broad- bzw. Multicast handeln. Zusätzlich muss die Absender-Adresse mit der 1-ten Eigenschaft sowie die Geräte-Adresse der Statusmeldung mit der 2-ten Eigenschaft übereinstimmen. Handelt es sich bei dem sendenden Gerät um einen analogen Eingang, so wird der 10 Bit große Ergebniswert auf einen 8 Bit großen gekürzt, indem die 2 niederwertigen Bits nicht beachtet werden. Sendet die Statusmeldung ein digitaler Eingang, werden die 4 niederwertigen Bits nicht beachtet, also die Bits, die die Nachkommastellen repräsentieren.

Der Ausgang dieses Objekts wird ebenfalls durch Empfangen eines Makros verändert. Die Modul-Adresse des Makros muss mit der Modul-Adresse des Moduls, auf dem die zu konfigurierende autonome Steuerung läuft, identisch sein oder es muss sich um einen Broad- bzw. Multicast handeln. Zusätzlich muss die Absender-Adresse mit der 1-ten Eigenschaft sowie das High-Byte der Makro-Nummer mit der 2-ten Eigenschaft übereinstimmen. Das Low-Byte der Makro-Nummer ist dann der Ergebniswert.

Nach der Verarbeitung des Ausgangswerts durch die autonome Steuerung oder nach einem Reset wird der Ausgang wieder auf den Wert der 3-ten Eigenschaft gesetzt.

### **Ausgangsmodifizierer (kein Eingang) – Typ 63**

Der Ausgangswert dieses Objekts ergibt sich als der Wert der 1-ten

Eigenschaft. Ist dieses Objekt einem Ausgang vorgeschaltet, so bestimmt die 2-te Eigenschaft (Low-Byte) und die niederwertigen 6 Bit der 3-ten Eigenschaft (High-Byte) die Verzögerung in 1/10 Sekunden, mit der eine am Ausgang angeschlossene Dimmerstufe den durch den Ausgang repräsentierten Wert annimmt. Ein Ausgang gibt den Steuerbefehl jedoch nur dann weiter, wenn der Eingangswert größer geworden (steigende Flanke) und Bit 7 der 3-ten Eigenschaft gesetzt ist oder der Eingangswert kleiner geworden (fallende Flanke) und Bit 6 der 3-ten Eigenschaft gesetzt ist.

#### **Bitweise Linksverschiebung (1 Eingang) – Typ 69**

Der Ausgangswert dieses Objekts ergibt sich durch eine bitweise Linksverschiebung des Eingangswerts. Die Nummer des Objekts, das als Eingang fungiert, wird in der 1-ten Eigenschaft hinterlegt. Die 2-te Eigenschaft kodiert die Anzahl der Bits, um die verschoben werden soll.

#### **Bitweise Rechtsverschiebung (1 Eingang) – Typ 70**

Der Ausgangswert dieses Objekts ergibt sich durch eine bitweise Rechtsverschiebung des Eingangswerts. Die Nummer des Objekts, das als Eingang fungiert, wird in der 1-ten Eigenschaft hinterlegt. Die 2-te Eigenschaft kodiert die Anzahl der Bits, um die verschoben werden soll.

#### **Vergleichsoperator „gleich“ (1 Eingang) – Typ 72**

Der Ausgang dieses Objekts nimmt den Wert 255 an, wenn der Wert des Eingangs gleich dem Wert der 2-ten Eigenschaft ist. Andernfalls ist der Ausgang 0. Die Nummer des Objektes, das als Eingang fungiert, wird in der 1-ten Eigenschaft hinterlegt.

#### **Vergleichsoperator „nicht gleich“ (1 Eingang) – Typ 73**

Der Ausgang dieses Objekts nimmt den Wert 255 an, wenn der Wert des Eingangs nicht gleich dem Wert der 2-ten Eigenschaft ist. Andernfalls ist der Ausgang 0. Die Nummer des Objektes, das als Eingang fungiert, wird in der 1-ten Eigenschaft hinterlegt.

#### **Vergleichsoperator „kleiner“ (1 Eingang) – Typ 74**

Der Ausgang dieses Objekts nimmt den Wert 255 an, wenn der Wert des Eingangs kleiner dem Wert der 2-ten Eigenschaft ist. Andernfalls ist der Ausgang 0. Die Nummer des Objektes, das als Eingang fungiert, wird in der 1-ten Eigenschaft hinterlegt.

#### **Vergleichsoperator „kleiner oder gleich“ (1 Eingang) – Typ 75**

Der Ausgang dieses Objekts nimmt den Wert 255 an, wenn der Wert des Eingangs kleiner oder gleich dem Wert der 2-ten Eigenschaft ist. Andernfalls ist der Ausgang 0. Die Nummer des Objekts, das als Eingang fungiert, wird in der 1-ten Eigenschaft hinterlegt.

#### **Vergleichsoperator „größer“ (1 Eingang) – Typ 76**

Der Ausgang dieses Objekts nimmt den Wert 255 an, wenn der Wert des Eingangs größer dem Wert der 2-ten Eigenschaft ist. Andernfalls ist der Ausgang 0. Die Nummer des Objekts, das als Eingang fungiert, wird in der 1-ten Eigenschaft hinterlegt.

#### **Vergleichsoperator „größer oder gleich“ (1 Eingang) – Typ 77**

Der Ausgang dieses Objekts nimmt den Wert 255 an, wenn der Wert des Eingangs größer oder gleich dem Wert der 2-ten Eigenschaft ist. Andernfalls ist der Ausgang 0. Die Nummer des Objekts, das als Eingang fungiert, wird in der 1-ten Eigenschaft hinterlegt.

#### **Addition (1 Eingang) – Typ 80**

Der Wert des Ausgangs dieses Objekts ergibt sich aus der Addition des Eingangswerts mit der 2-ten und 3-ten Eigenschaft. Die Nummer des Objekts, das als Eingang fungiert, wird in der 1-ten Eigenschaft hinterlegt.

#### **Subtraktion (1 Eingang) – Typ 81**

Der Wert des Ausgangs dieses Objekts ergibt sich durch Subtraktion der 2-ten und 3-ten Eigenschaft vom Eingangswert. Die Nummer des Objekts, das als Eingang fungiert, wird in der 1-ten Eigenschaft hinterlegt.

#### **Multiplikation (1 Eingang) – Typ 82**

Der Wert des Ausgangs dieses Objekts ergibt sich aus der Multiplikation des Eingangswerts mit der 2-ten und 3-ten Eigenschaft. Die Nummer des Objekts, das als Eingang fungiert, wird in der 1-ten Eigenschaft hinterlegt.

#### **Division (1 Eingang) – Typ 83**

Der Wert des Ausgangs dieses Objekts ergibt sich durch Division des Eingangswerts durch die 2-te und 3-te Eigenschaft. Die Nummer des Objekts, das als Eingang fungiert, wird in der 1-ten Eigenschaft hinterlegt.

#### **Multiplikation mit Offset – Typ 84**

Der Wert des Ausgangs dieses Objekts ergibt sich durch Addition der 2-ten Eigenschaft zum Eingangswert und anschließender Multiplikation mit der 3-ten Eigenschaft dividiert durch 16. Die Rundung auf einen ganzzahligen Wert wird am Ende vorgenommen. Die Nummer des Objekts, das als Eingang fungiert, wird in der 1-ten Eigenschaft hinterlegt.

#### **Division mit Offset – Typ 85**

Der Wert des Ausgangs dieses Objekts ergibt sich durch Addition der 2-ten Eigenschaft zum Eingangswert und anschließender Division der 3-ten Eigenschaft durch eben diesen Wert. Die Rundung auf einen ganzzahligen Wert wird am Ende vorgenommen. Die Nummer des Objekts, das als Eingang fungiert, wird in der 1-ten Eigenschaft hinterlegt.

#### **Multiplikation mit einer rationalen Zahl – Typ 86**

Der Wert des Ausgangs dieses Objekts ergibt sich durch Division des Eingangswerts mit der 2-ten Eigenschaft und anschließender Multiplikation mit der 3-ten Eigenschaft. Die Rundung auf einen ganzzahligen Wert wird am Ende vorgenommen. Die Nummer des Objekts, das als Eingang fungiert, wird in der 1-ten Eigenschaft hinterlegt.

#### **Regelglied 0 – Typ 96**

Der Wert des Ausgangs dieses Objekts wird jeweils um die Hälfte der Differenz zwischen dem Eingangswert und einem Referenzwert korrigiert. Er kann dabei nicht größer als 255 und nicht kleiner als 0 werden. Die Nummer



des Objekts, das als Eingang fungiert, wird in der 1-ten Eigenschaft hinterlegt. Der Referenzwert wird mit der 2-ten Eigenschaft initialisiert.

#### **Auf-/Ab-Steuerung 1-Taster-Lösung – Typ 100**

Dieses Objekt setzt die Eingangswerte, wie sie von einem entsprechend konfigurierten logischen Eingang geliefert werden, in entsprechende Steuercodes um, wie sie zur Ansteuerung eines Dimmers, eines Schalters oder eines Rollladens verwendet werden können. Die Nummer des Objekts, das als Eingang fungiert, wird in der 1-ten Eigenschaft hinterlegt. Die 2-te und 3-te Eigenschaft wird nicht verwendet.

#### **Weiterleitung „kleiner“ (1 Eingang) – Typ 104**

Der Ausgang dieses Objekts nimmt den kleinsten Wert vom Eingang, der 2-ten Eigenschaft und der 3-ten Eigenschaft an. Die Nummer des Objekts, das als Eingang fungiert, wird in der 1-ten Eigenschaft hinterlegt.

#### **Weiterleitung „mittleren“ (1 Eingang) – Typ 105**

Der Ausgang dieses Objekts nimmt den mittleren Wert vom Eingang, der 2-ten Eigenschaft und der 3-ten Eigenschaft an. Die Nummer des Objekts, das als Eingang fungiert, wird in der 1-ten Eigenschaft hinterlegt.

#### **Weiterleitung „größer“ (1 Eingang) – Typ 106**

Der Ausgang dieses Objekts nimmt den größten Wert vom Eingang, der 2-ten Eigenschaft und der 3-ten Eigenschaft an. Die Nummer des Objekts, das als Eingang fungiert, wird in der 1-ten Eigenschaft hinterlegt.

#### **Weiterleitung „gesteuert“ (1 Eingang) – Typ 107**

Der Ausgang dieses Objekts nimmt den Wert der 2-ten Eigenschaft an, wenn der Eingang 0 ist. Andernfalls wird der Wert der 3-ten Eigenschaft angenommen. Die Nummer des Objekts, das als Eingang fungiert, wird in der 1-ten Eigenschaft hinterlegt.

#### **Ausschaltverzögerung – Typ 112**

Ist der Eingangswert dieses Objekts größer 0, so nimmt der Ausgang eben

diesen Wert an und hält ihn für eine konfigurierbare Zeit, auch wenn der Eingangswert wieder 0 geworden ist. Die Nummer des Objekts, das als Eingang fungiert, wird in der 1-ten Eigenschaft hinterlegt. Die Zeit, um die der Ausschaltvorgang verzögert wird, wird in der 3-ten Eigenschaft hinterlegt, wobei die unteren 3 Bits der 2-ten Eigenschaft die Zeitbasis gemäß folgender Tabelle kodieren:

<b>Zeitbasis</b>	<b>Code</b>
1/10 Sek.	0
Sekunden	1
Minuten	2
Stunden	3
Tage	4
Wochen	5

*Tabelle 4.12: AS Codierung der Zeitbasis bei der Ausschaltverzögerung*

### **Einschaltverzögerung – Typ 113**

Ist der Eingangswert dieses Objekts größer 0, so wird dieser Wert erst nach einer konfigurierbaren Zeit auf den Ausgang übertragen. Ist der Eingangswert 0, so wird auch der Ausgang 0. Die Nummer des Objekts, das als Eingang fungiert, wird in der 1-ten Eigenschaft hinterlegt. Die Zeit, um die der Einschaltvorgang verzögert wird, wird in der 3-ten Eigenschaft hinterlegt, wobei die unteren 3 Bits der 2-ten Eigenschaft die Zeitbasis gemäß folgender Tabelle kodieren:

<b>Zeitbasis</b>	<b>Code</b>
1/10 Sek.	0
Sekunden	1
Minuten	2
Stunden	3
Tage	4
Wochen	5

*Tabelle 4.13: AS Codierung der Zeitbasis bei der Einschaltverzögerung*

## Einschaltbegrenzung – Typ 114

Ist der Eingangswert dieses Objekts größer 0 und eine konfigurierbare Zeitspanne noch nicht abgelaufen, so wird dieser Wert auf den Ausgang übertragen. Die Zeit fängt in dem Moment an abzulaufen, wenn der Eingangswert größer 0 wird. Ist der Eingangswert 0 bzw. die Zeit abgelaufen, so wird auch der Ausgang 0. Die Nummer des Objekts, das als Eingang fungiert, wird in der 1-ten Eigenschaft hinterlegt. Die Zeit, um die der Einschaltvorgang verzögert wird, wird in der 3-ten Eigenschaft hinterlegt, wobei die unteren 3 Bits der 2-ten Eigenschaft die Zeitbasis gemäß folgender Tabelle kodieren:

Zeitbasis	Code
1/10 Sek.	0
Sekunden	1
Minuten	2
Stunden	3
Tage	4
Wochen	5

*Tabelle 4.14: AS Codierung der Zeitbasis bei der Einschaltbegrenzung*

## Zeitgesteuerte Weiterleitung – Typ 115

Der Eingangswert dieses Objekts wird in einem zu konfigurierendem Intervall auf den Ausgang übertragen. Somit wird nicht jede Änderung des Eingangs vom Ausgang übernommen, sondern nur jene, die mit dem Ablauf dieses Intervalls zusammenfallen. Die Nummer des Objekts, das als Eingang fungiert, wird in der 1-ten Eigenschaft hinterlegt. Die Länge des Zeitintervalls wird in der 3-ten Eigenschaft hinterlegt, wobei die unteren 3 Bits der 2-ten Eigenschaft die Zeitbasis gemäß folgender Tabelle kodieren:

<b>Zeitbasis</b>	<b>Code</b>
1/10 Sek.	0
Sekunden	1
Minuten	2
Stunden	3
Tage	4
Wochen	5

*Tabelle 4.15: AS Codierung der Zeitbasis bei der zeitgesteuerten Weiterleitung*

### **Ausgang – Typ 120**

Ändert sich der Eingangswert dieses Objekts, so wird eben dieser Wert dividiert durch 2,55 (Der Wertebereich der autonomen Steuerung von 0 bis 255 wird auf 0 bis 100 begrenzt.) in Form eines Steuerbefehls weitergegeben. Die Nummer des Objekts, das als Eingang fungiert, wird in der 1-ten Eigenschaft hinterlegt. Die Modul-Adresse des Moduls, für das der Steuerbefehl bestimmt ist, wird in der 2-ten Eigenschaft gespeichert. Die 3-te Eigenschaft speichert schließlich die Geräte-Adresse des Zielgeräts. Wird dem Ausgang ein Ausgangsmodifizierer vorgeschaltet, so lässt sich dem Steuerbefehl neben dem eigentlichen Wert auch eine Verzögerungszeit mitgeben. Außerdem wird dann der Steuerbefehl nur unter bestimmten Bedingungen weitergegeben (siehe Objekttypen 63 und 127).

### **Nativer Ausgang – Typ 121**

Ändert sich der Eingangswert dieses Objekts, so wird eben dieser Wert in Form eines Steuerbefehls weitergegeben. Die Nummer des Objekts, das als Eingang fungiert, wird in der 1-ten Eigenschaft hinterlegt. Die Modul-Adresse des Moduls, für das der Steuerbefehl bestimmt ist, wird in der 2-ten Eigenschaft gespeichert. Die 3-te Eigenschaft speichert schließlich die Geräte-Adresse des Zielgerätes. Wird dem Ausgang ein Ausgangsmodifizierer vorgeschaltet, so lässt sich dem Steuerbefehl neben dem eigentlichen Wert auch eine Verzögerungszeit mitgeben. Außerdem wird dann der Steuerbefehl nur unter bestimmten Bedingungen weitergegeben (siehe Objekttypen 63 und 127).

### **Statusmeldungs Ausgang – Typ 122**

Ändert sich der Eingangswert dieses Objekts, so wird eben dieser Wert in Form einer Statusmeldung als Broadcast verschickt. Die Nummer des Objekts, das als Eingang fungiert, wird in der 1-ten Eigenschaft hinterlegt. Die 3-te Eigenschaft speichert die Geräte-Adresse, die in der Statusmeldung hinterlegt wird. Es handelt sich somit um die Geräte-Adresse des Statusmeldungs Ausgangs.

### **Ausgangsmodifizierer (1 Eingang) – Typ 127**

Der Ausgangswert dieses Objekts ergibt sich als dessen Eingangswert. Die Nummer des Objekts, das als Eingang fungiert, wird in der 1-ten Eigenschaft hinterlegt. Ist dieses Objekt einem Ausgang vorgeschaltet, so bestimmt die 2-te Eigenschaft (Low-Byte) und die niederwertigen 6 Bit der 3-ten Eigenschaft (High-Byte) die Verzögerung in 1/10 Sekunden, mit der eine am Ausgang angeschlossene Dimmerstufe den durch den Ausgang repräsentierten Wert annimmt. Ein Ausgang gibt den Steuerbefehl jedoch nur dann weiter, wenn der Eingangswert größer geworden (steigende Flanke) und Bit 7 der 3-ten Eigenschaft gesetzt ist oder der Eingangswert kleiner geworden (fallende Flanke) und Bit 6 der 3-ten Eigenschaft gesetzt ist.

### **Bitweise UND-Verknüpfung (2 Eingänge) – Typ 128**

Der Wert des Ausgangs dieses Objekts ergibt sich aus der bitweisen UND-Verknüpfung der beiden Eingänge mit der 3-ten Eigenschaft. Die Nummern der Objekte, die als Eingang fungieren, werden in der 1-ten und 2-ten Eigenschaft hinterlegt.

### **Bitweise ODER-Verknüpfung (2 Eingänge) – Typ 129**

Der Wert des Ausgangs dieses Objekts ergibt sich aus der bitweisen ODER-Verknüpfung der beiden Eingänge mit der 3-ten Eigenschaft. Die Nummern der Objekte, die als Eingang fungieren, werden in der 1-ten und 2-ten Eigenschaft hinterlegt.

### **Bitweise NICHT-UND-Verknüpfung (2 Eingänge) – Typ 130**

Der Wert des Ausgangs dieses Objekts ergibt sich aus der bitweisen NICHT-

UND-Verknüpfung der beiden Eingänge mit der 3-ten Eigenschaft. Die Nummern der Objekte, die als Eingang fungieren, werden in der 1-ten und 2-ten Eigenschaft hinterlegt.

#### **Bitweise NICHT-ODER-Verknüpfung (2 Eingänge) – Typ 131**

Der Wert des Ausgangs dieses Objekts ergibt sich aus der bitweisen NICHT-ODER-Verknüpfung der beiden Eingänge mit der 3-ten Eigenschaft. Die Nummern der Objekte, die als Eingang fungieren, werden in der 1-ten und 2-ten Eigenschaft hinterlegt.

#### **Bitweise EXKLUSIV-ODER-Verknüpfung (2 Eingänge) – Typ 132**

Der Wert des Ausgangs dieses Objekts ergibt sich aus der bitweisen EXKLUSIV-ODER-Verknüpfung der beiden Eingänge mit der 3-ten Eigenschaft. Die Nummern der Objekte, die als Eingang fungieren, werden in der 1-ten und 2-ten Eigenschaft hinterlegt.

#### **Bitweise Linksverschiebung (2 Eingänge) – Typ 133**

Der Ausgangswert dieses Objekts ergibt sich durch eine bitweise Linksverschiebung des 1-ten Eingangswerts. Der 2-te Eingangswert gibt die Anzahl der Bits an, um die verschoben werden soll. Die Nummern der Objekte, die als Eingang fungieren, werden in der 1-ten und 2-ten Eigenschaft hinterlegt.

#### **Bitweise Rechtsverschiebung (2 Eingänge) – Typ 134**

Der Ausgangswert dieses Objekts ergibt sich durch eine bitweise Rechtsverschiebung des 1-ten Eingangswerts. Der 2-te Eingangswert gibt die Anzahl der Bits an, um die verschoben werden soll. Die Nummern der Objekte, die als Eingang fungieren, werden in der 1-ten und 2-ten Eigenschaft hinterlegt.

#### **Vergleichsoperator „gleich“ (2 Eingänge) – Typ 136**

Der Ausgang dieses Objekts nimmt den Wert 255 an, wenn der Wert des 1-ten Eingangs gleich dem Wert des 2-ten Eingangs ist. Andernfalls ist der Ausgang 0. Die Nummern der Objekte, die als Eingang fungieren, werden in der 1-ten

und 2-ten Eigenschaft hinterlegt.

### **Vergleichsoperator „nicht gleich“ (2 Eingänge) – Typ 137**

Der Ausgang dieses Objekts nimmt den Wert 255 an, wenn der Wert des 1-ten Eingangs nicht gleich dem Wert des 2-ten Eingangs ist. Andernfalls ist der Ausgang 0. Die Nummern der Objekte, die als Eingang fungieren, werden in der 1-ten und 2-ten Eigenschaft hinterlegt.

### **Vergleichsoperator „kleiner“ (2 Eingänge) – Typ 138**

Der Ausgang dieses Objekts nimmt den Wert 255 an, wenn der Wert des 1-ten Eingangs kleiner dem Wert des 2-ten Eingangs ist. Andernfalls ist der Ausgang 0. Die Nummern der Objekte, die als Eingang fungieren, werden in der 1-ten und 2-ten Eigenschaft hinterlegt.

### **Vergleichsoperator „kleiner oder gleich“ (2 Eingänge) – Typ 139**

Der Ausgang dieses Objekts nimmt den Wert 255 an, wenn der Wert des 1-ten Eingangs kleiner oder gleich dem Wert des 2-ten Eingangs ist. Andernfalls ist der Ausgang 0. Die Nummern der Objekte, die als Eingang fungieren, werden in der 1-ten und 2-ten Eigenschaft hinterlegt.

### **Vergleichsoperator „größer“ (2 Eingänge) – Typ 140**

Der Ausgang dieses Objekts nimmt den Wert 255 an, wenn der Wert des 1-ten Eingangs größer dem Wert des 2-ten Eingangs ist. Andernfalls ist der Ausgang 0. Die Nummern der Objekte, die als Eingang fungieren, werden in der 1-ten und 2-ten Eigenschaft hinterlegt.

### **Vergleichsoperator „größer oder gleich“ (2 Eingänge) – Typ 141**

Der Ausgang dieses Objekts nimmt den Wert 255 an, wenn der Wert des 1-ten Eingangs größer oder gleich dem Wert des 2-ten Eingangs ist. Andernfalls ist der Ausgang 0. Die Nummern der Objekte, die als Eingang fungieren, werden in der 1-ten und 2-ten Eigenschaft hinterlegt.

### **Addition (2 Eingänge) – Typ 144**

Der Wert des Ausgangs dieses Objekts ergibt sich aus der Addition der beiden

Eingangswerte mit der 3-ten Eigenschaft. Die Nummern der Objekte, die als Eingang fungieren, werden in der 1-ten und 2-ten Eigenschaft hinterlegt.

#### **Subtraktion (2 Eingänge) – Typ 145**

Der Wert des Ausgangs dieses Objekts ergibt sich durch Subtraktion des 2-ten Eingangs und der 3-ten Eigenschaft vom 1-ten Eingang. Die Nummern der Objekte, die als Eingang fungieren, werden in der 1-ten und 2-ten Eigenschaft hinterlegt.

#### **Multiplikation (2 Eingänge) – Typ 146**

Der Wert des Ausgangs dieses Objekts ergibt sich aus der Multiplikation der beiden Eingangswerte mit der 3-ten Eigenschaft. Die Nummern der Objekte, die als Eingang fungieren, werden in der 1-ten und 2-ten Eigenschaft hinterlegt.

#### **Division (2 Eingänge) – Typ 147**

Der Wert des Ausgangs dieses Objekts ergibt sich durch Division des 1-ten Eingangswerts durch den 2-ten Eingang und die 3-te Eigenschaft. Die Nummern der Objekte, die als Eingang fungieren, werden in der 1-ten und 2-ten Eigenschaft hinterlegt.

#### **Flipflop0 (2 Eingänge) – Typ 152**

Der Ausgang dieses Objekts nimmt den Wert der 3-ten Eigenschaft an, wenn der 1-te Eingangswert größer 0 ist. Ist der 2-te Eingangswert größer 0, so wird der Ausgang des Objekts auf 0 gesetzt. Der 2-te Eingang ist gegenüber dem 1-ten Eingang priorisiert.

#### **Flipflop1 (2 Eingänge) – Typ 153**

Der Ausgang dieses Objekts nimmt den Wert der 3-ten Eigenschaft an, wenn der 1-te Eingangswert größer 0 ist. Ist der 2-te Eingangswert größer 0, so wird der Ausgang des Objekts auf 0 gesetzt. Der 1-te Eingang ist gegenüber dem 2-ten Eingang priorisiert.

#### **Flipflop2 (2 Eingänge) – Typ 154**



Der Ausgang dieses Objekts nimmt den Wert der 3-ten Eigenschaft an, wenn der 1-te Eingangswert größer 0 ist. Ist der 2-te Eingangswert größer 0, so wird der Ausgang des Objekts auf 0 gesetzt. Ist der 1-te Eingangswert und der 2-te Eingangswert größer 0, so wird der Ausgang invertiert, das heißt, ist er gleich 0, so nimmt er den Wert der 3-ten Eigenschaft an, sonst wird er 0.

#### **Flipflop3 (2 Eingänge) – Typ 155**

Der Ausgang dieses Objekts nimmt den Wert der 3-ten Eigenschaft an, wenn der 1-te Eingangswert gerade größer 0 geworden ist (steigende Flanke). Ist der 2-te Eingangswert gerade größer 0 geworden (steigende Flanke), so wird der Ausgang des Objekts auf 0 gesetzt. Der 2-te Eingang ist gegenüber dem 1-ten Eingang priorisiert.

#### **Flipflop4 (2 Eingänge) – Typ 156**

Der Ausgang dieses Objekts nimmt den Wert der 3-ten Eigenschaft an, wenn der 1-te Eingangswert gerade größer 0 geworden ist (steigende Flanke). Ist der 2-te Eingangswert gerade größer 0 geworden (steigende Flanke), so wird der Ausgang des Objekts auf 0 gesetzt. Der 1-te Eingang ist gegenüber dem 2-ten Eingang priorisiert.

#### **Flipflop5 (2 Eingänge) – Typ 157**

Der Ausgang dieses Objekts nimmt den Wert der 3-ten Eigenschaft an, wenn der 1-te Eingangswert gerade größer 0 geworden ist (steigende Flanke). Ist der 2-te Eingangswert gerade größer 0 geworden (steigende Flanke), so wird der Ausgang des Objekts auf 0 gesetzt. Ist der 1-te Eingangswert und der 2-te Eingangswert gerade größer 0 geworden (steigende Flanke), so wird der Ausgang invertiert, das heißt, ist er gleich 0, so nimmt er den Wert der 3-ten Eigenschaft an, sonst wird er 0.

#### **Auf-/Ab-Steuerung 2-Taster-Lösung – Typ 164**

Dieses Objekt setzt die Eingangswerte, wie sie von entsprechend konfigurierten logischen Eingängen geliefert werden, in entsprechende SteuerCodes um, wie sie zur Ansteuerung eines Dimmers oder Schalters verwendet werden können. Die Nummern der Objekte, die als Eingang

fungieren, werden in der 1-ten und 2-ten Eigenschaft hinterlegt. Die 3-te Eigenschaft wird nicht verwendet.

#### **Auf-/Ab-Steuerung 2-Taster-Lösung (Rollladen) – Typ 165**

Dieses Objekt setzt die Eingangswerte, wie sie von entsprechend konfigurierten logischen Eingängen geliefert werden, in entsprechende Steuercodes um, wie sie zur Ansteuerung eines Rollladens verwendet werden können. Die Nummern der Objekte, die als Eingang fungieren, werden in der 1-ten und 2-ten Eigenschaft hinterlegt. Die 3-te Eigenschaft wird nicht verwendet.

#### **Weiterleitung „kleiner“ (2 Eingänge) – Typ 168**

Der Ausgang dieses Objekts nimmt den kleinsten Wert vom 1-ten Eingang, dem 2-ten Eingang und der 3-ten Eigenschaft an. Die Nummern der Objekte, die als Eingang fungieren, werden in der 1-ten und 2-ten Eigenschaft hinterlegt.

#### **Weiterleitung „mittleren“ (2 Eingänge) – Typ 169**

Der Ausgang dieses Objekts nimmt den mittleren Wert vom 1-ten Eingang, dem 2-ten Eingang und der 3-ten Eigenschaft an. Die Nummern der Objekte, die als Eingang fungieren, werden in der 1-ten und 2-ten Eigenschaft hinterlegt.

#### **Weiterleitung „größer“ (2 Eingänge) – Typ 170**

Der Ausgang dieses Objekts nimmt den größten Wert vom 1-ten Eingang, dem 2-ten Eingang und der 3-ten Eigenschaft an. Die Nummern der Objekte, die als Eingang fungieren, werden in der 1-ten und 2-ten Eigenschaft hinterlegt.

#### **Weiterleitung „gesteuert“ (2 Eingänge) – Typ 171**

Der Ausgang dieses Objekts nimmt den Wert des 2-ten Eingangs an, wenn der 1-te Eingang 0 ist. Andernfalls wird der Wert der 3-ten Eigenschaft angenommen. Die Nummern der Objekte, die als Eingang fungieren, werden in der 1-ten und 2-ten Eigenschaft hinterlegt.

#### **Bitweise UND-Verknüpfung (3 Eingänge) – Typ 192**

Der Wert des Ausgangs dieses Objekts ergibt sich aus der bitweisen UND-Verknüpfung der drei Eingänge. Die Nummern der Objekte, die als Eingang fungieren, werden in der 1-ten, 2-ten und 3-ten Eigenschaft hinterlegt.

#### **Bitweise ODER-Verknüpfung (3 Eingänge) – Typ 193**

Der Wert des Ausgangs dieses Objekts ergibt sich aus der bitweisen ODER-Verknüpfung der drei Eingänge. Die Nummern der Objekte, die als Eingang fungieren, werden in der 1-ten, 2-ten und 3-ten Eigenschaft hinterlegt.

#### **Bitweise NICHT-UND-Verknüpfung (3 Eingänge) – Typ 194**

Der Wert des Ausgangs dieses Objekts ergibt sich aus der bitweisen NICHT-UND-Verknüpfung der drei Eingänge. Die Nummern der Objekte, die als Eingang fungieren, werden in der 1-ten, 2-ten und 3-ten Eigenschaft hinterlegt.

#### **Bitweise NICHT-ODER-Verknüpfung (3 Eingänge) – Typ 195**

Der Wert des Ausgangs dieses Objekts ergibt sich aus der bitweisen NICHT-ODER-Verknüpfung der drei Eingänge. Die Nummern der Objekte, die als Eingang fungieren, werden in der 1-ten, 2-ten und 3-ten Eigenschaft hinterlegt.

#### **Bitweise EXKLUSIV-ODER-Verknüpfung (3 Eingänge) – Typ 196**

Der Wert des Ausgangs dieses Objekts ergibt sich aus der bitweisen EXKLUSIV-ODER-Verknüpfung der drei Eingänge. Die Nummern der Objekte, die als Eingang fungieren, werden in der 1-ten, 2-ten und 3-ten Eigenschaft hinterlegt.

#### **Addition (3 Eingänge) – Typ 208**

Der Wert des Ausgangs dieses Objekts ergibt sich aus der Addition der drei Eingangswerte. Die Nummern der Objekte, die als Eingang fungieren, werden in der 1-ten, 2-ten und 3-ten Eigenschaft hinterlegt.

#### **Subtraktion (3 Eingänge) – Typ 209**

Der Wert des Ausgangs dieses Objekts ergibt sich durch Subtraktion des 2-ten und 3-ten Eingangs vom 1-ten Eingang. Die Nummern der Objekte, die als Eingang fungieren, werden in der 1-ten, 2-ten und 3-ten Eigenschaft hinterlegt.

### **Multiplikation (3 Eingänge) – Typ 210**

Der Wert des Ausgangs dieses Objekts ergibt sich aus der Multiplikation der drei Eingangswerte. Die Nummern der Objekte, die als Eingang fungieren, werden in der 1-ten, 2-ten und 3-ten Eigenschaft hinterlegt.

### **Division (3 Eingänge) – Typ 211**

Der Wert des Ausgangs dieses Objekts ergibt sich durch Division des 1-ten Eingangswerts durch den 2-ten und 3-ten Eingang. Die Nummern der Objekte, die als Eingang fungieren, werden in der 1-ten, 2-ten und 3-ten Eigenschaft hinterlegt.

### **Flipflop0 (3 Eingänge) – Typ 216**

Der Ausgang dieses Objekts nimmt den Wert des 3-ten Eingangs an, wenn der 1-te Eingangswert größer 0 ist. Ist der 2-te Eingangswert größer 0, so wird der Ausgang des Objekts auf 0 gesetzt. Der 2-te Eingang ist gegenüber dem 1-ten Eingang priorisiert.

### **Flipflop1 (3 Eingänge) – Typ 217**

Der Ausgang dieses Objekts nimmt den Wert des 3-ten Eingangs an, wenn der 1-te Eingangswert größer 0 ist. Ist der 2-te Eingangswert größer 0, so wird der Ausgang des Objekts auf 0 gesetzt. Der 1-te Eingang ist gegenüber dem 2-ten Eingang priorisiert.

### **Flipflop2 (3 Eingänge) – Typ 218**

Der Ausgang dieses Objekts nimmt den Wert des 3-ten Eingangs an, wenn der 1-te Eingangswert größer 0 ist. Ist der 2-te Eingangswert größer 0, so wird der Ausgang des Objekts auf 0 gesetzt. Ist der 1-te Eingangswert und der 2-te Eingangswert größer 0, so wird der Ausgang invertiert, das heißt, ist er gleich 0, so nimmt er den Wert des 3-ten Eingangs an, sonst wird er 0.

### **Flipflop3 (3 Eingänge) – Typ 219**

Der Ausgang dieses Objekts nimmt den Wert des 3-ten Eingangs an, wenn der 1-te Eingangswert gerade größer 0 geworden ist (steigende Flanke). Ist der 2-te Eingangswert gerade größer 0 geworden (steigende Flanke), so wird

der Ausgang des Objekts auf 0 gesetzt. Der 2-te Eingang ist gegenüber dem 1-ten Eingang priorisiert.

#### **Flipflop4 (3 Eingänge) – Typ 220**

Der Ausgang dieses Objekts nimmt den Wert des 3-ten Eingangs an, wenn der 1-te Eingangswert gerade größer 0 geworden ist (steigende Flanke). Ist der 2-te Eingangswert gerade größer 0 geworden (steigende Flanke), so wird der Ausgang des Objekts auf 0 gesetzt. Der 1-te Eingang ist gegenüber dem 2-ten Eingang priorisiert.

#### **Flipflop5 (3 Eingänge) – Typ 221**

Der Ausgang dieses Objekts nimmt den Wert des 3-ten Eingangs an, wenn der 1-te Eingangswert gerade größer 0 geworden ist (steigende Flanke). Ist der 2-te Eingangswert gerade größer 0 geworden (steigende Flanke), so wird der Ausgang des Objekts auf 0 gesetzt. Ist der 1-te Eingangswert und der 2-te Eingangswert gerade größer 0 geworden (steigende Flanke), so wird der Ausgang invertiert, das heißt, ist er gleich 0, so nimmt er den Wert des 3-ten Eingangs an, sonst wird er 0.

#### **Weiterleitung „kleiner“ (3 Eingänge) – Typ 232**

Der Ausgang dieses Objekts nimmt den kleinsten Wert der Eingänge an. Die Nummern der Objekte, die als Eingang fungieren, werden in der 1-ten, 2-ten und 3-ten Eigenschaft hinterlegt.

#### **Weiterleitung „mittleren“ (3 Eingänge) – Typ 233**

Der Ausgang dieses Objekts nimmt den mittleren Wert der Eingänge an. Die Nummern der Objekte, die als Eingang fungieren, werden in der 1-ten, 2-ten und 3-ten Eigenschaft hinterlegt.

#### **Weiterleitung „größer“ (3 Eingänge) – Typ 234**

Der Ausgang dieses Objekts nimmt den größten Wert der Eingänge an. Die Nummern der Objekte, die als Eingang fungieren, werden in der 1-ten, 2-ten und 3-ten Eigenschaft hinterlegt.

#### **Weiterleitung „gesteuert“ (3 Eingänge) – Typ 235**

Der Ausgang dieses Objekts nimmt den Wert des 2-ten Eingangs an, wenn der 1-te Eingang 0 ist. Andernfalls wird der Wert des 3-ten Eingangs angenommen. Die Nummern der Objekte, die als Eingang fungieren, werden in der 1-ten, 2-ten und 3-ten Eigenschaft hinterlegt.

## 5 Tabellen und Übersichten

Dieses Kapitel soll als Schnellreferenz verstanden werden. Die Informationen werden nur sehr knapp und stark verdichtet dargestellt. Für detaillierte Informationen sei auf die entsprechenden Kapitel verwiesen.

### 5.1 Klemm- und Steckverbinder

Nachstehende Tabelle gibt eine Übersicht über die Belegung der Anschlussleisten der Control Unit. Bei den Platinenverbindern J2 – J9 ist nur der Pin angegeben, Gnd und VCC wurde weggelassen. Die Reihenfolge der Pins eines jeden Verbinders ist von links nach recht der Überschrift zu entnehmen.

Klemml. X1		Pin/Gnd/VCC		Stiftleiste JP2		Stiftleiste JP6		Stiftleiste JP7	
Kl.	Beschr.	SV	Beschr.	St.	Beschr.	St.	Beschr.	St.	Beschr.
1	GND	J2	A0	1	D0	1	A0	1	CANH
2	VCCSupp	J3	A1	2	D1	2	A1	2	CANL
3	CANLB	J4	A2	3	C0	3	A2	3	CANHB
4	CANHB	J5	A3	4	C1	4	A3	4	CANLB
5	CANL	J6	A4	5	VCC	5	A4	5	D0
6	CANH	J7	A5	6	GND	6	A5	6	D1
		J8	A6			7	A6	7	D2
		J9	A7			8	A7	8	D3
						9	B0	9	D4
						10	B1	10	D5
						11	B2	11	D6
						12	B3	12	D7
						13	C2	13	B5
						14	C3	14	B6
						15	C4	15	B7
						16	C5	16	Reset
						17	C6	17	VCCSupp
						18	C7	18	AREF
						19	VCC	19	VCC
						20	GND	20	GND

Tabelle 5.1: Pin-Belegung der Klemm- und Steckverbinder

## 5.2 Befehlsreferenz

In folgender Übersicht sind alle Befehle aufgeführt, die von einer Control Unit verarbeitet werden können. Die ersten drei Bytes jeder Nachricht werden dabei weggelassen. Ebenso sind die Antwort- und Fehlerpakete nur aufgeführt, wenn sie vom Standard abweichen.

Message-Type		Device		Value0	Value1	Value2
ID	Name	ID	Name			
0	Set		Device	Value	LParam	HParam
8	Query		Device	0	0	Select
9	QueryAck		Device	LValue	HValue	Select
16	Status		Device	LValue	HValue	Ext
24	Makro		0	LMakro	HMakro	0
28	EEPROM		161/169	LAddr	HAddr	Number
29	EEPROMAck		DataA	DataB	DataC	DataD
56	ControlProt	0	StartProt	16/32	0	0
		1	EndProt	0	0	0
		16	StartPage	160/168	LSideAddr	HSideAddr
60	RawData		DataA	DataB	DataC	DataD
64	ConfigIO		Pin	Type	Device	0
			Pin + 32	StatusMod	0	0
68	ConfigIRAddr		IR	Modul	Device	0
72	ConfigIRHotKeys		HotKey	LMakro	HMakro	0
76	Control	1	SystemReset	0	0	0
		2	SystemFullReset	0	0	0
		3	ConfigReset	0	0	0
		4	SetStartMode	StartMode	0	0
		5	SetModulAddr	ModulAddr	0	0
		6	SetCCUAddr	CCUAddr	0	0
		7	SetTimeServer	TimeSrv	0	0
		8	SaveConfig	0	0	0
		9	LoadConfig	0	0	0
		10	SetBridgeMode	BMode	0	0
		11	SetREAddr	Number	Addr	0
		12	SetFLANID	VLAN-ID	0	0
		13	SetMCG	LByte	HByte	0



		14	SetEncMode	Mode	0	0
		15	SetEncKey	Number	Value	0
		16	SetBuzzerLevel	LByte	HByte	0
		18	SetCLANID	VLAN-ID	0	0
		24	SSGetFlash	0	0	0
		27	SSSetSize	LByte	HByte	0
		28	SSGetVersion	0	0	0
		30	SSGetCO	X	0	0
		32	SetLIPrell	Code	LTime	HTime
		36	SetDMCtrlDelay	Time	0	0
		37	SetDMZD	LTics	HTics	0
		64	SetAIProp +Offs.	Addr	LValue	HValue
				Addr	Hyst	Flags
		80	SetAIPropSRate	Addr	LRate	HRate
		96	SetRecBufLength	Length	0	0
		128	SetDIProp + Offs.	Addr	LValue	HValue
				Addr	Hyst	Flags
		144	SetDIPropSRate	Addr	LRate	HRate
		145	SetDIPropType	Addr	Type	0
77	ControlAck	24	SSGetFlash	Flag	0	0
		28	SSGetVersion	Major	Minor	Phase
		30	SSGetCO	X	CO	0
80	ConfigIRTrans		Button	Command	Device	0
84	ConfigIRLearn		0	Command	0	0
85	ConfIRLearnAck		Button	Command	Device	0
88	DisplayData		Char0	Char1	Char2	Char3
91	DisplayControl		Inst0	Inst1	Inst2	Inst3
96	ConfigAM		0	Addr	Type	StatusMod
			RSAddr	0	Modul	Addr
			RSAddr	1	Modul	Addr
			RSAddr	128	Time	0
			RSAddr	129	Type	0
			DGAddr	0	Modul	0
			DGAddr	1	AddrA	0
			DGAddr	2	AddrB	0
			DGAddr	3	AddrT	0

			DGAddr	4	Modul	Addr
			DGAddr	5	Speed	0
100	ConfigAS		Object	Prop	Value	Default
120	TCSet		Hundredth	Second	Minute	HourDay
123	TCSynchReq		0	0	0	0
124	MagicPacket		ModulAddr	HWAddr1	HWAddr2	HWAddr3
127	ProtErr		1	RModul	RMTyp	RDevice
			2	Device	0	0

*Tabelle 5.2: Befehlsreferenz*

### 5.3 Steuercodes

Neben den Werten zwischen 0 und 100, die einem Gerät zugewiesen werden können, gibt es noch diverse Codes, die einem Gerät ein bestimmtes Verhalten aufzwingen. Diese Codes sind in folgender Tabelle noch einmal zusammengefasst.

Code	Funktion
0 - 100	Normal Value
128	StatusInvert
129	Plus
130	Minus
131	AllOn
132	AllOff
133	ControlUp
134	ControlDown
135	ControlStop
136	ControlStart
137	Left
138	Right
139	PressShort
140	PressMedium
141	PressLong
142	Refresh
255	NOP

*Tabelle 5.3: Steuercodes*

## 5.4 IR-Bediencodes

Folgende Tabelle fasst die Befehle, die über eine IR-Fernbedienung eingegeben werden können, zusammen.

1. Taste	2. Taste	3. Taste	4. Taste	5. Taste	6. Taste	7. Taste	8. Taste
Comm.							
HotKey	Enter						
Makro	Number0	Enter					
Makro	Number1	Number0	Enter				
Makro	Number2	Number1	Number0	Enter			
Makro	Number3	Number2	Number1	Number0	Enter		
IR1	IR0	Enter					
IR1	IR0	Value	Enter				
IR1	IR0	Value2	Value1	Value0	Enter		
IR1	IR0	Value2	Value1	Value0	Time1	Time0	Enter

*Tabelle 5.4: IR-Bediencodes*

## 5.5 GUI-Konfiguration

Die Struktur und der Aufbau der GUI-Konfiguration sei im Folgenden noch einmal zusammenfassend dargestellt.

```
<Nibble> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F
<Byte> ::= <Nibble> <Nibble>
<Word> ::= <Byte> <Byte>
<Addr8> ::= <Byte>
<Addr16> ::= <Word>

<EString> ::= <CharCount> [<CharList>]
<CharList> ::= <Char> | <Char> <CharList>
<CharCount> ::= <Byte>
<Char> ::= <Byte>

<Config> ::= <RootCtrlAddr> <StdCtrlAddr> <CtrlList>
<RootCtrlAddr> ::= <Addr16>
<StdCtrlAddr> ::= <Addr16>
<CtrlList> ::= <Ctrl> | <Ctrl> <CtrlList>
<Control> ::= <Menue> | <OutCtrl> | <Thermostat>

<Menue> ::= <MCTYPE> <MItemCount> <MItemList>
<MCTYPE> ::= 01
<MItemCount> ::= <Byte>
<MItemList> ::= <MItem> | <MItem> <MItemList>
<MItem> ::= <MName> <MCtrlAddr>
<MName> ::= <EString>
<MCtrlAddr> ::= <Addr16>

<OutCtrl> ::= <OCCType> <OCModul> <OCDevice> <OCName>
<OCCType> ::= 10
<OCModul> ::= <Addr8>
<OCDevice> ::= <Addr8>
<OCName> ::= <EString>

<Thermostat> ::= <TSCType> <TSModul> <TSDevice> <TSRefresh> <TSName>
<TSCType> ::= 20
<TSModul> ::= <Addr8>
<TSDevice> ::= <Addr8>
<TSRefresh> ::= <Byte>
<TSName> ::= <EString>
```

## 5.6 Objekttypen der autonomen Steuerung

Folgende Tabelle fasst die Objekttypen der autonomen Steuerung und deren zu konfigurierenden Eigenschaften zusammen.

Typ		1. Eigenschaft	2. Eigenschaft	3. Eigenschaft
ID	Bezeichnung			
0	No Operation	0	0	0
32	Timer Minute	B0-5 S (s) B6-7 Int (256/10s)	B0-3 S (1/10s)	B0-7 Int (1/10s)
33	Timer Stunde	B0-5 S (min) B6-7 Int (1024s)	B0-5 S (s) B6-7 Int (256s)	B0-7 Int (s)
34	Timer Tag	B0-4 S (h)  B5 Werktag B6 Samstag B7 Sonntag	B0-5 S (min) B6-7 Int (256min)	B0-7 Int (min)
35	Timer Woche	B0-4 S (h) B5-7 S (d)	B0-5 S (min) B6-7 Int (256min)	B0-7 Int (min)
56	Aktiver Eingang	Modul-Adresse	Device-Adresse	Abfrageintervall (1/10s)
60	Nicht flüchtig. pass. Eingang	Modul-Adresse	Device-Adr. bzw. HByte Makronum.	Default-Value
61	Flüchtiger pass. Eingang	Modul-Adresse	Device-Adr. bzw. HByte Makronum.	Default-Value
63	Ausgangs-modifizierer	Default-Value	B0-7 Del (1/10s)	B0-5 Del (256/10s) B6 fallende Flanke B7 steigende Fl.
69	Bitweise Linksversch.	Eingang	Anzahl Bits	0
70	Bitweise Rechtsversch.	Eingang	Anzahl Bits	0
72	Gleich	Eingang	Vergleichswert	0
73	Nicht gleich	Eingang	Vergleichswert	0
74	Kleiner	Eingang	Vergleichswert	0
75	Kleiner gleich	Eingang	Vergleichswert	0
76	Größer	Eingang	Vergleichswert	0
77	Größer gleich	Eingang	Vergleichswert	0
80	Addition	Eingang	Summand 1	Summand 2
81	Subtraktion	Eingang	Subtrahend 1	Subtrahend 2

82	Multiplikation	Eingang	Faktor 1	Faktor 2
83	Division	Eingang	Divisor 1	Divisor 2
84	Multiplikation mit Offset	Eingang	Summand	Faktor
85	Division mit Offset	Eingang	Summand	Divisor
86	Multiplikation m. ration. Zahl	Eingang	Divisor	Faktor
96	Regelglied 0	Eingang	Referenzwert	0
100	Auf-/Ab-Steu. 1 Taster Lös.	Eingang	0	0
104	Weiterleitung (kleiner)	Eingang	Wert 1	Wert 2
105	Weiterleitung (mittlerer)	Eingang	Wert 1	Wert 2
106	Weiterleitung (größer)	Eingang	Wert 1	Wert 2
107	Weiterleitung (gesteuert)	Eingang	Wert 1	Wert 2
112	Ausschalt-verzögerung	Eingang	B0-2 Zeitbasis	Zeit
113	Einschalt-verzögerung	Eingang	B0-2 Zeitbasis	Zeit
114	Einschalt-begrenzung	Eingang	B0-2 Zeitbasis	Zeit
115	Weiterleitung (zeitgesteuert)	Eingang	B0-2 Zeitbasis	Zeit
120	Ausgang	Eingang	Modul-Adresse	Device-Adresse
121	Nativer Ausgang	Eingang	Modul-Adresse	Device-Adresse
122	Statusmeld.-ausgang	Eingang	0	Device-Adresse
127	Ausgangs-modifizierer	Eingang	B0-7 Del (1/10s)	B0-5 Del (256/10s) B6 fallende Flanke B7 steigende Fl.
128	UND (bitweise)	Eingang 1	Eingang 2	Wert
129	ODER (bitweise)	Eingang 1	Eingang 2	Wert
130	NICHT UND (bitweise)	Eingang 1	Eingang 2	Wert

131	NICHT ODER (bitweise)	Eingang 1	Eingang 2	Wert
132	Exklus. ODER (bitweise)	Eingang 1	Eingang 2	Wert
133	Bitweise Linksversch.	Eingang 1 (Wert)	Eingang 2 (Anzahl Bits)	0
134	Bitweise Rechtsversch.	Eingang 1 (Wert)	Eingang 2 (Anzahl Bits)	0
136	Gleich	Eingang 1	Eingang 2	0
137	Nicht gleich	Eingang 1	Eingang 2	0
138	Kleiner	Eingang 1	Eingang 2	0
139	Kleiner gleich	Eingang 1	Eingang 2	0
140	Größer	Eingang 1	Eingang 2	0
141	Größer gleich	Eingang 1	Eingang 2	0
144	Addition	Eingang 1	Eingang 2	Summand
145	Subtraktion	Eingang 1	Eingang 2	Subtrahend
146	Multiplikation	Eingang 1	Eingang 2	Faktor
147	Division	Eingang 1	Eingang 2	Divisor
152	Flipflop 0	Eingang Set	Eingang Reset	Wert
153	Flipflop 1	Eingang Set	Eingang Reset	Wert
154	Flipflop 2	Eingang Set	Eingang Reset	Wert
155	Flipflop 3	Eingang Set	Eingang Reset	Wert
156	Flipflop 4	Eingang Set	Eingang Reset	Wert
157	Flipflop 5	Eingang Set	Eingang Reset	Wert
164	Auf-/Ab-Steu. 2 Taster Lös.	Eingang Auf	Eingang Ab	0
165	Auf-/Ab-Steu. 2 Taster Lös. (Rollladen)	Eingang Auf	Eingang Ab	0
168	Weiterleitung (kleiner)	Eingang 1	Eingang 2	Wert
169	Weiterleitung (mittlerer)	Eingang 1	Eingang 2	Wert
170	Weiterleitung (größer)	Eingang 1	Eingang 2	Wert
171	Weiterleitung (gesteuert)	Eingang 1	Eingang 2	Wert
192	UND (bitweise)	Eingang 1	Eingang 2	Eingang 3

193	ODER (bitweise)	Eingang 1	Eingang 2	Eingang 3
194	NICHT UND (bitweise)	Eingang 1	Eingang 2	Eingang 3
195	NICHT ODER (bitweise)	Eingang 1	Eingang 2	Eingang 3
196	Exklus. ODER (bitweise)	Eingang 1	Eingang 2	Eingang 3
208	Addition	Eingang 1	Eingang 2	Eingang 3
209	Subtraktion	Eingang 1	Eingang 2	Eingang 3
210	Multiplikation	Eingang 1	Eingang 2	Eingang 3
211	Division	Eingang 1	Eingang 2	Eingang 3
216	Flipflop 0	Eingang Set	Eingang Reset	Eingang Wert
217	Flipflop 1	Eingang Set	Eingang Reset	Eingang Wert
218	Flipflop 2	Eingang Set	Eingang Reset	Eingang Wert
219	Flipflop 3	Eingang Set	Eingang Reset	Eingang Wert
220	Flipflop 4	Eingang Set	Eingang Reset	Eingang Wert
221	Flipflop 5	Eingang Set	Eingang Reset	Eingang Wert
232	Weiterleitung (kleiner)	Eingang 1	Eingang 2	Eingang 3
233	Weiterleitung (mittlerer)	Eingang 1	Eingang 2	Eingang 3
234	Weiterleitung (größer)	Eingang 1	Eingang 2	Eingang 3
235	Weiterleitung (gesteuert)	Eingang 1	Eingang 2	Eingang 3

*Tabelle 5.5: Objekttypen der autonomen Steuerung*



## Literaturverzeichnis

Atmel Corporation: *Datenblatt zum 8-bit AVR Microcontroller ATmega32*

STMicroelectronics: *Datenblatt zum 64 Kbit Serial I<sup>2</sup>C Bus EEPROM M24C64*

Atmel Corporation: *Datenblatt zum 256 Kbit Serial I<sup>2</sup>C Bus EEPROM AT24C256*

Microchip: *Datenblatt zum Stand-Alone CAN Controller with SPI MCP2515*

Philips Semiconductors: *Datenblatt zum CAN Controller Interface PCA82C250*

Dallas Semiconductor: *Datenblatt zum 1-Wire Digital Thermometer DS18B20*

Dallas Semiconductor: *Datenblatt zum 1-Wire Digital Thermometer DS18S20*

Sitronix: *Datenblatt zum Dot Matrix LCD Controller/Driver ST7036*

Electronic Assembly GmbH: *Datenblatt zu LCDs der DOG-Serie 3,3V EA DOG-M*

ALPS Electric Co., Ltd.: *Datenblatt zum Rotational Encoder STEC11B03*