

# Connected Lighting 16011

## Pflichtenheft Connected Lighting Module

Projekt	Connected Lighting 16011
Dokument	Pflichtenheft Connected Lighting Module
Dateiname	pflichtenheft software connected lighting module.docx
Tags	

### Dokument Version

Version	Datum	Autor	Status
0.1	20.04.17	M. Borer	Draft
0.2	21.04.17	M. Feistle	Draft
0.3	03.05.17	M. Feistle	Draft
0.4	10.05.17	M. Feistle	Draft
0.5	17.05.17	M. Feistle	Draft
0.6	23.05.17	M. Feistle	Draft
0.7	24.05.17	M. Feistle	Draft
0.8	01.06.17	M. Feistle	Draft
0.9	06.06.17	M. Feistle	Draft
1.0	26.06.17	M. Feistle	Draft
1.1	04.07.17	M. Feistle /M.Borer	Draft
1.2	14.07.17	M. Borer	Review - Änderungen Interne Kommandos Zeit eliminiert
1.3	07.08.17	M. Feistle	Review - Befehlsstruktur LF <-> CR
1.4b	28.09.17	M. Borer	Released (M. Feistle / M. Borer) - Anpassungen Protokoll im Kapitel 3.3.2 <ul style="list-style-type: none"> <li>○ Beschreibung Verhalten Counter</li> <li>○ Beschreibung Kommandobefehle</li> <li>○ Beschreibung Notifications</li> <li>○ Kommandobefehle: Error codes</li> </ul>
1.5b	21.12.17	M. Feistle	Released - Anpassungen Protokoll im Kapitel 3.3.2 <ul style="list-style-type: none"> <li>○ Protokollaufbau Interne Kommunikation CLC – CLBTM</li> </ul>

1.5c	05.01.18	M. Feistle	Released <ul style="list-style-type: none"> <li>- Anpassungen Protokoll im Kapitel 3.3.2 <ul style="list-style-type: none"> <li>o Protokollaufbau Interne Kommunikation CLC – CLBTM</li> </ul> </li> </ul>
1.6	26.1.2018	M.Borer / M. Feistle	In Work <ul style="list-style-type: none"> <li>- Anpassungen für Downstream Protokoll im Kapitel Protokoll und Aufbau File Downstream / Firmware update (CLBTM zum CLC)</li> <li>- Notification errors updated</li> <li>- Überarbeitung Kapitel 3.4. Date-Upload-Config / Actions Definition</li> <li>- Internes Kommando hinzugefügt (head type, nr. of light decks)</li> <li>- Neudefinition Struktur Imagefile</li> <li>- FW Update Protokoll E1/E3 überarbeitet</li> </ul>

[Dokument-Status: In Work/Draft/In Review/Released - Final]



## Inhaltsverzeichnis

1.	Einleitung .....	6
1.1.	Zweck des Dokumentes .....	6
1.2.	Adressaten .....	6
2.	Übersicht.....	7
2.1.	Systemaufbau .....	7
3.	Anforderungen .....	8
3.1.	Schnittstellen.....	8
3.1.1.	Bluetooth Modul zu CLC (Connected Lighting Controller) .....	8
3.1.2.	CLC zum Modbus .....	8
3.1.3.	SWD Interface zum CLC .....	8
3.1.4.	JTAG (2-wire: SWDIO, SWDCLK) Interface zum Bluetooth Mikrocontroller .....	8
3.1.5.	Debug Interface am CLC.....	8
3.1.6.	Debug Interface am Bluetooth Modul .....	8
3.2.	Mikrocontroller.....	9
3.2.1.	Aufgaben CLC.....	9
3.2.2.	Aufgaben des Bluetooth Moduls.....	9
3.3.	Kommunikationsprotokoll .....	10
3.3.1.	Übersicht Datenframetransfer .....	10
3.3.2.	Protokoll zwischen CLC und CLBTM.....	11
3.3.3.	Protokoll zwischen CLM und Modbus.....	27
3.4.	Action File .....	32
3.4.1.	Speicherstruktur action file .....	33
3.5.	Startup Verhalten CLC - CLBTM .....	34
3.6.	Gemeinsame Holdingregister Steuermodul – CLM.....	35
3.6.1.	Gemeinsame Holdingregister Steuermodul – CLM.....	35
3.7.	Datentransparenz CLM Modul.....	37
3.7.1.	Datenherkunft aller Regent Modbus Modulen .....	37
3.7.2.	Datenherkunft Steuermodul.....	37
3.7.3.	Datenherkunft Sensormodul.....	38
3.7.4.	Datenherkunft Panel .....	38
3.7.5.	Datenherkunft ALONEatWORK.....	38
3.7.6.	Datenherkunft CLM .....	38
3.7.7.	Datenherkunft Human Centric Lighting.....	38
4.	Funktionen .....	41
4.1.	Bootup Abfrage des angeschlossenen Systems.....	41
4.2.	Funktionen für Tunable White .....	42
4.3.	Funktionen für Connected Lighting.....	43
A.	Anhang.....	44
A.1.	Abkürzungen.....	44

*Internes Dokument!*

A.2.	Device typen .....	44
A.3.	Referenzen .....	45

## 1. Einleitung

### 1.1. Zweck des Dokumentes

Dieses Dokument beschreibt die Anforderungen für die Software des Connected Lighting Modul.

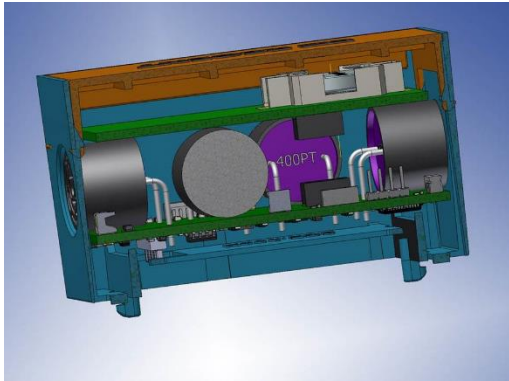


Abbildung 1 **Symbolbild**

### 1.2. Adressaten

Dieses Dokument richtet sich an alle Projektverantwortlichen, insbesondere an die Elektronik Entwicklung und an das PM.

## 2. Übersicht

Das Connected Lighting Modul ist grundsätzlich ein Elektronisches Modul mit einer Modbus-Schnittstelle und einer Bluetooth-Schnittstelle. In einer Ausprägung (für Lightpad Slim, TW, ev. Tweak) wird das Connected Lighting Module konstruktiv mit dem ALONEatWORK Modul zusammengefasst und im gleichen Gehäuse verbaut. Softwaretechnisch gesehen ist das Connected Lighting Modul aber ein eigenständiger Modbus-Teilnehmer.

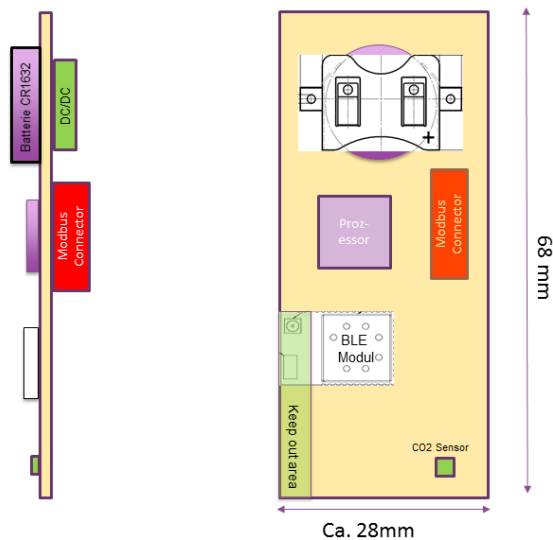
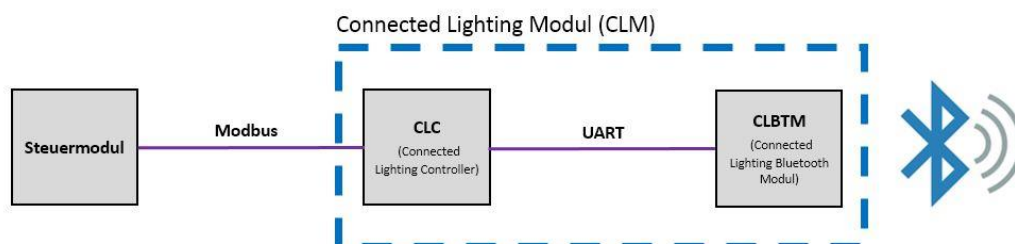


Abbildung 2 **Systemansicht**

### 2.1. Systemaufbau



Das CLM besteht im Wesentlichen aus einem CLC (Connected Lighting Controller, Cortex M0) und einem CLBTM (Connected Lighting Bluetooth Modul, Laird BL652).

Das CLBTM ist mit dem CLC über ein UART verbunden. Das Protokoll hierzu ist spezifisch und wird in diesem Dokument beschrieben.

Der CLC ist über Modbus mit dem Steuermodul und anderen Teilnehmern am Modbus verbunden. In seiner grundsätzlichen Funktion agiert der CLC als Modbus Slave. Die auszutauschenden Daten werden in diesem Dokument beschrieben.

Über den Pfad CLBTM – CLC – Modbus, können Daten in beide Richtungen ausgetauscht werden. Daten können Konfigurationen, Parameter, Firmwarefiles, Tabellen sein.

### 3. Anforderungen

#### 3.1. Schnittstellen

##### 3.1.1. Bluetooth Modul zu CLC (Connected Lighting Controller)

RQ	Schnittstelle Bluetooth Modul zu CLC
A	Die Schnittstelle ist eine UART-Schnittstelle und wird mit 115200 baud, 8 bits, even Parity, 1 Stop-Bit und Hardware Flusskontrolle RTS / CTS betrieben
B	Der CLC und CLBTM können beide unabhängig voneinander Daten austauschen. Die Schnittstelle ist full duplex fähig.
C	Die Daten über die serielle Schnittstelle sind im ASCII Datenformat und somit lesbar.
D	CLC Mikrocontroller bekommt die Informationen von Fehlern via serieller Schnittstelle.
E	Der CLC kann mittels GPIO Leitung das BLE Modul einem expliziten Hardwarereset unterziehen.

##### 3.1.2. CLC zum Modbus

RQ	Schnittstelle CLC zum Modbus
A	Die Schnittstelle ist eine Modbus Schnittstelle. Die control unit der Leuchte ist Modbus master und koordiniert den Datenaustausch in beide Richtungen
B	Der CLC Mikrocontroller verhält sich im Standard Betrieb wie ein Modbus Slave
C	Der CLC Mikrocontroller kann das control unit der Leuchte in den Modbus slave Zustand versetzen und selber modbus master werden. Z.b. bei einem Firmwareupdate.

##### 3.1.3. SWD Interface zum CLC

RQ	SWD Interface CLC
A	Das SWD Interface erlaubt auf den CLC Mikrocontroller eine Firmware zu laden
B	Das SWD Interface erlaubt Firmware auf dem CLC Mikrocontroller zu Debuggen

##### 3.1.4. JTAG (2-wire: SWDIO, SWDCLK) Interface zum Bluetooth Mikrocontroller

RQ	JTAG Interface zum CLBTM
A	Das JTAG Interface erlaubt auf das Bluetooth Module eine Firmware zu laden
B	Das JTAG Interface erlaubt Firmware auf dem Bluetooth Modul zu Debuggen

##### 3.1.5. Debug Interface am CLC

RQ	Debug Interface am CLC
A	Das Debug Interface ist eine bidirektionale UART Schnittstelle.
B	Das Debug Interface erlaubt ohne Debugger Ausgaben im Live-Betrieb auszugeben
C	Das Debug Interface erlaubt ohne Debugger Eingaben im Live-Betrieb zu akzeptieren.
D	Aktionen des Debuginterfaces sollen in einem Low-Prio Handler (main Loop) abgearbeitet werden, aber mittels Bufferung die zeitliche Abfolge erhalten werden.

##### 3.1.6. Debug Interface am Bluetooth Modul

RQ	Debug Interface am Bluetooth Modul
A	Das Debug Interface ist in die Verbindung zwischen CLC Mikrocontroller und Bluetooth Modul eingebettet und kann explizit gesniff werden.



## 3.2. Mikrocontroller

### 3.2.1. Aufgaben CLC

RQ	Aufgaben des CLC
A	Der CLC Mikrocontroller bootet mit dem internen Taktgeber
B	Der CLC Mikrocontroller kann Modbusbefehle empfangen und verarbeiten.
C	Der CLC Mikrocontroller kann Daten von anderen ModbusTeilnehmern verwalten. Daten können sein: FW-Version, HW-Version, Lichtdaten, Aktivitätsdaten, Fehlermeldungen etc.
D	Der CLC Mikrocontroller kann Konfigurationen (Parameterdaten und Tabellen) an andere ModbusTeilnehmer übermitteln.
E	Der CLC Mikrocontroller kann Dateien (wie Firmware) an andere Modbus-Teilnehmer übermitteln. Die Art und Weise des Firmware-Übermittlung wird in <a href="#">3.3 Kommunikationsprotokoll</a> spezifiziert.
F	Der CLC Mikrocontroller kann eine Datums – und Zeitverwaltung mittels RTC unterhalten.
G	Der CLC Mikrocontroller kann Daten im Kapitel <a href="#">3.3.1. Protokoll zwischen CLC und Bluetooth Modul</a> spezifizierten Format an das Bluetooth Modul übermitteln.
H	Der CLC Mikrocontroller kann Daten im Kapitel <a href="#">3.3.1. Protokoll zwischen CLC und Bluetooth Modul</a> spezifizierten Format vom Bluetooth Modul empfangen.
I	Die Daten sind im Kapitel <a href="#">3.3. Kommunikationsprotokoll</a> spezifiziert.
J	Der CLC Mikrocontroller kann Fehlerzustände erkennen und darauf entsprechend reagieren. Fehlerzustände können sein: Bufferoverflow, Connection Loss, Unbekannte Kommandos oder Befehle, Übermittlungsfehler (CRC)

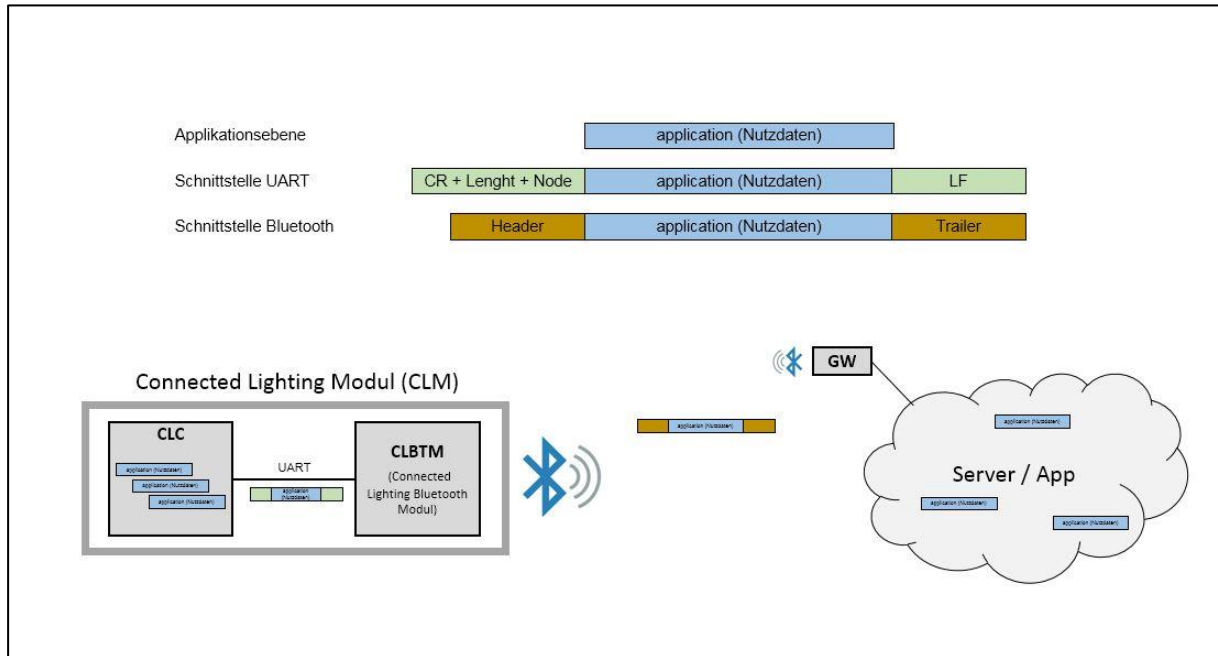
### 3.2.2. Aufgaben des Bluetooth Moduls

RQ	Aufgaben des Bluetooth Moduls in Richtung CLC
A	Das Bluetooth Modul kann Daten im spezifizierten Format (Kapitel <a href="#">3.3.1. Protokoll zwischen CLC und Bluetooth Modul</a> ) empfangen.
B	Das Bluetooth Modul kann empfangene Daten signalisieren und dadurch einen Abfragezyklus des CLC auslösen.
C	Das Bluetooth Modul liefert Daten an das CLC gemäss dem im Kapitel <a href="#">3.3.1. Protokoll zwischen CLC und Bluetooth Modul</a> spezifizierten Format.
D	Das Bluetooth Module stellt Fehlerzustände via serieller Schnittstelle zur Verfügung.

### 3.3. Kommunikationsprotokoll

#### 3.3.1. Übersicht Datenframetransfer

Um Nutzdaten zwischen den einzelnen Schnittstellen zu transferieren, werden header und footer hinzugefügt.



### 3.3.2. Protokoll zwischen CLC und CLBTM

#### Kommunikationsprinzip CLC und CLBTM

Die Kommunikation ist bidirektional, full duplex. CLC und CLBTM können gleichzeitig Daten senden und empfangen. Im ersten Schritt wird vorerst eine sequenzielle Befehlsabfolge angestrebt. Eine Antwort muss innerhalb 500ms ausgeführt werden.

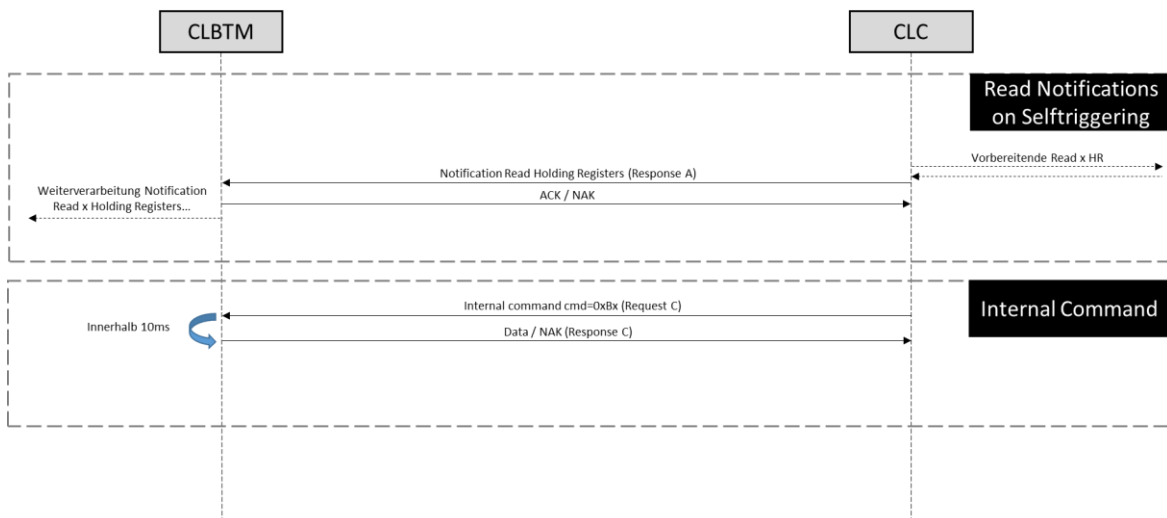
Protokoll: 115200 Baud, 8E1, Flusskontrolle RTS/CTS

Die Kommunikation ist bidirektional, full duplex. CLC und CLBTM können gleichzeitig Daten senden und empfangen. Im ersten Schritt wird vorerst eine sequenzielle Befehlsabfolge angestrebt. Die Verarbeitung einer Anfrage dauert ca. 200 ms (bedingt durch das Modbus Interface) bis eine Antwort versendet werden kann. Bei mehreren Anfragen hintereinander dauert es entsprechend länger bis die jeweilige Antwort versendet werden kann.

#### Sequenzablauf mit CLBTM als Initiator



#### Sequenzablauf mit CLC als Initiator



### ***Zeitliche Randbedingungen***

Antwort NAK (typischerweise  $t < 2\text{ms}$ ): Der Befehl wurde nicht erkannt.

Antwort ACK (typischerweise  $t < 2\text{ms}$ ): Der Befehl wurde erkannt und wird ausgeführt.

Antwort mit Daten (typischerweise  $t > 200\text{ms}$ ):

- wird asynchron zur Anfrage als Notification verschickt (mit ACK/NACK Bestätigung)
- Bsp: Write w/o Notification → nur ACK/NACK Bestätigung
- Bsp: Write with Notification → nur ACK/NACK Bestätigung, asynchron Notification write
- Bsp: Read → ACK/NACK Bestätigung synchron, asynchron Notification Read

### Prinzipieller Aufbau eines Kommunikationsframes

Ein Frame ist folgendermassen aufgebaut:

Zuerst kommt das Startzeichen line feed danach die Länge, Node-ID Source, Node-ID Destination und der Counter, dann die Nutzdaten und abgeschlossen wird das Frame mit einem carriage return.

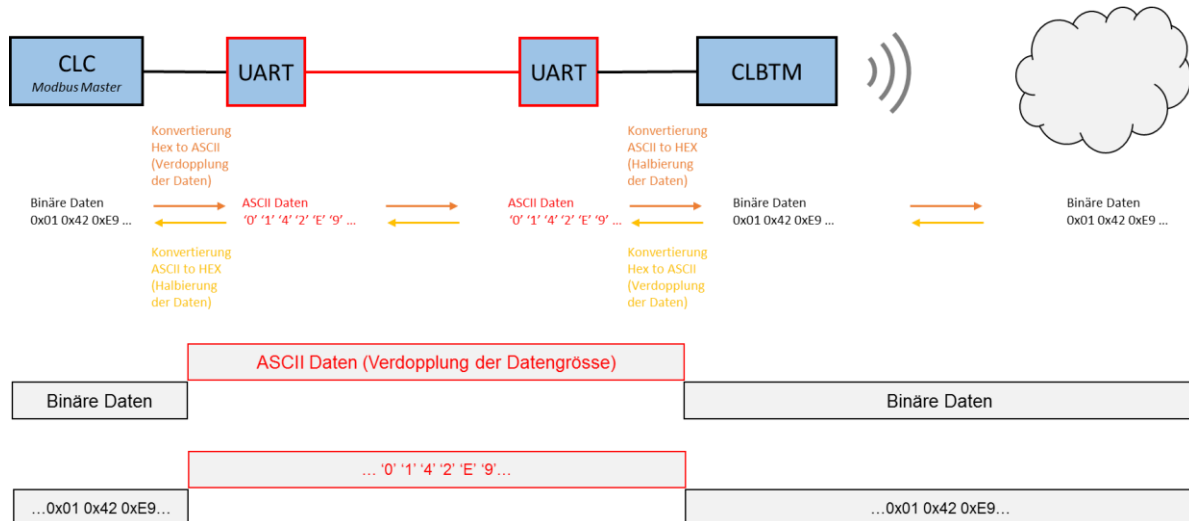
length = SourceID + DestinationID + Counter + Nutzdaten						
Carriage return (0x0D)	length	Source ID	Destination ID	Counter	Nutzdaten (Payload)	Line feed (0x0A)
1 Byte	2 Bytes	2 Bytes	2 Bytes	1 Byte	1-1024 Bytes	1 Byte

**Default Node ID Mesh Gateway:** 0x7918 (31000).

**Default Node ID App Tunable White:** 0x80E8 (33000)

## Ascii Konvertierung für Lesbarkeit

Für die UART Schnittstelle zwischen CLC und CLBTM werden die zu übermittelnden Daten in ASCII konvertiert (Verdoppelung der Daten) und nach Empfang wieder zurückkonvertiert (aus Gründen der Lesbarkeit)



Beispiel: Nutzdaten (Binäre Daten) = 0xAA 0x1B (2 Bytes Datengrösse)

Komplettes „ASCII“ Datenpaket via UART:

CR	Lenght				Node Source				Node Dest.				Nutzdaten				LF
'0'	'0'	'0'	'6'		'0'	'5'	'0'	'4'	'7'	'8'	'1'	'8'	'A'	'A'	'1'	'B'	
0x0D	0x30	0x30	0x36		0x30	0x35	0x30	0x34	0x37	0x38	0x31	0x38	0x41	0x41	0x31	0x42	0x0A

## Byte Ordering: Little Endian

Die verwendete Ordnung der Bytes im Speicher ist little endian (niederwertigstes Byte an tiefster Adresse) weil CLC und CLBTM standartmässig ihre Daten in diesem Format verwalten. Die Konsequenz davon ist, dass bei der seriellen Datenübertragen int16 Byte Werte vertauscht übertragen werden.

Bsp:

Der int16 Wert 0x1234 wird über die serielle Datenleitung so übertragen: ... 0x34 0x12...

## Kommandobefehle

$$\text{Lenght} = \text{length} + \text{Scr/Dest/Cnt} + \text{Nutzdaten}$$

CR	length	Source	Dest.	Cnt	CMD	Modbus Adr.	HR Adr.	Nr. HR	Data	LF
1 Byte	2 Bytes	2 Bytes	2 Bytes	1 Byte	1 Byte	1 Byte	2 Byte	1 Byte	0-1024 Bytes	1 Byte

Das **CMD Feld** wird als Kommando (0x00 – 0xEF) und Bestätigungsfeld 0xF0 – 0xFF) verwendet:

<b>Test Mode</b>	0x00	Plaintext Mode (Testmode CLBTM)
<b>Internal Commands</b>	0xB0	Read Smartbeacon Node ID
	0xB1	Read Node Status
	0xB2	Reset Bluetooth Module
	0xB5	Ready to pair with smartphone
	0xB8	Commandos Testprocedures set
	0xB9	Commandos Testprocedures get
	0xBA	Commandos Testprocedures test bluetooth connection
<b>Notifications</b>	0xC0	Notification Error
	0xC1	Notification Read
	0xC2	Notification Write
<b>R/W</b>	0xD0	Write without Notification
	0xD1	Read
	0xD2	Write with Notification
<b>File Stream</b>	0xE0	Image Datei downstream
	0xE1	Image Datei gültig
	0xE2	Imagedatei aktivieren
	0xE3	Imagedatei erfolgreich
<b>ACK / NACK</b>	0xF0	ACK
	0xF1	Error general
	0xF2	Error command length
	0xF3	Error Holdingregister length
	0xF4	Error CRC
	0xF5	Error wrong command
	0xF6	Error ASCII
	0xF7	Error buffer busy
	0xF8	Error buffer full
	0xF9	Error wrong data length
	0xFA	Error modbus slave timeout
	0xFB	Error modbus exception

## Verhalten Counter

$$\text{Lenght} = \text{length} + \text{Scr/Dest/Cnt} + \text{Nutzdaten}$$

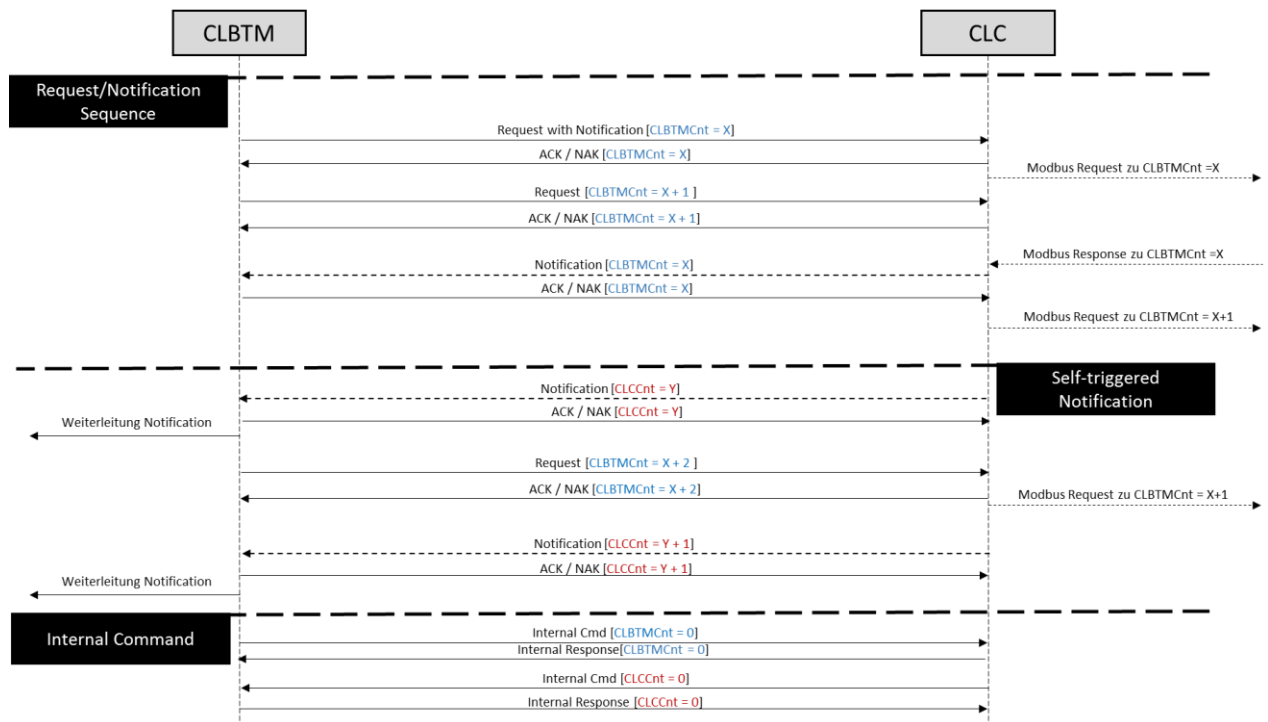
CR	length	Source	Dest.	Cnt	CMD	Modbus Adr.	HR Adr.	Nr. HR	Data	LF
1 Byte	2 Bytes	2 Bytes	2 Bytes	1 Byte	1 Byte	1 Byte	2 Byte	1 Byte	0-1024 Bytes	1 Byte

Ziel des Counters ist die Referenzierung von Messages, um asynchrone Command / Response Abläufe einander zuzuordnen.

- Ein ACK/NAK Befehl gibt immer die gleiche Counternummer zurück wie diejenige des Requests/Notification
- Eine Notification Read

Die Messages von CLC und CLBTM werden separat gezählt. Dazu wird der Counter (1 Byte) unterteilt in einen Low Nibble und High Nibble.

Messagecounter	Wertebereich	Bemerkung
CLBTM	Low Range: [1 – 127]	Ringzähler von 1,2,..127,1...
CLC	High Range: [128 – 254]	Ringzähler von 128,129,...254,128..
	0	Interl Commando
	255	Reserve







### Protokollaufbau Read / Write without Notification / Write with Notification

#### Command

Lenght = length + Scr/Dest/Cnt + Nutzdaten										
CR	length	Source	Dest.	Cnt	CMD	Modbus Adr.	HR Adr.	Nr. HR	Data	LF
1 Byte	2 Bytes	2 Bytes	2 Bytes	1 Byte	1 Byte	1 Byte	2 Byte	1 Byte	0-1024 Bytes	1 Byte

#### Response

Lenght = length + Scr+ Dest + Cnt + Nutzdaten								
CR	length	Source	Dest.	Cnt	CMD			LF
1 Byte	2 Bytes	2 Bytes	2 Bytes	1 Byte	1 Byte			1 Byte

length: length = length + Scr+ Dest + Cnt + Nutzdaten (CMD bis und mit Data)  
 Source: Data source. 0 = broadcast, 0x7918 (31000) = tunnel to gateway  
 Dest.: Data destination: 0x7918 (31000) = tunnel to gateway  
 cnt: Ringcounter, Beschreibung siehe Kapitel Verhalten Counter Seite 16  
 CMD: Kommandobefehl, Beschreibung siehe Kapitel Kommandobefehle Seite 15  
 ModbusAdr: Modbus Device Adresse 1..255  
 HR Adr.: Modbus Startadresse Holding Register 0..65535  
 Nr. HR: Anzahl Holding Register 1..255  
 Data: Datenbytes 0..1024

#### CMD Examples:

Beschreibung	Command / Response	Initator	CMD	Modbus Adr	Holding Register	Anzahl Hol.-Reg	Data	Info
<b>Write without Notification</b>	Command	CLBTM	<b>0xD0</b>	[MB Adr]	[HR-Reg]	[#HR]	Data [0..(x-1)]	x = 2* Nr HR
	Response	CLC	0xF0/ 0xFx					ACK / NACK
<b>Read</b>	Command	CLBTM	<b>0xD1</b>	[MB Adr]	[HR-Reg]	[#HR]	None	
	Response	CLC	0xF0/ 0xFx					ACK / NACK
<b>Write with Notification</b>	Command	CLBTM	<b>0xD2</b>	[MB Adr]	[HR-Reg]	[#HR]	Data [0..(x-1)]	x = 2* Nr HR
	Response	CLC	0xF0/ 0xFx					

### Protokollaufbau Notifikation Read / Notifikation Write (CLC zum CLBTM)

#### Command

Lenght = length + Scr + Dest + Cnt + Nutzdaten										
CR	length	Source	Dest.	Cnt	CMD	Modbus Adr.	HR Adr.	Nr. HR	Data	LF
1 Byte	2 Bytes	2 Bytes	2 Bytes	1 Byte	1 Byte	1 Byte	2 Byte	1 Byte	0-1024 Bytes	1 Byte

#### Response

Lenght = length + Scr+ Dest + Cnt + Nutzdaten								LF
CR	length	Source	Dest.	Cnt	CMD			LF
1 Byte	2 Bytes	2 Bytes	2 Bytes	1 Byte	1 Byte			1 Byte

length: length + Scr+ Dest + Cnt + Nutzdaten (CMD bis und mit Data)  
 Source: Data source. 0 = broadcast, 0x7918 (31000) = tunnel to gateway  
 Dest.: Sent to Node. 0x7918 (31000) = tunnel to gateway  
 cnt: Ringcounter, Beschreibung siehe Kapitel Verhalten Counter Seite 16  
 CMD: Kommandobefehl, Beschreibung siehe Kapitel Kommandobefehle Seite 15  
 ModbusAdr: Modbus Device Adresse 1..255  
 HR Adr.: Modbus Startadresse Holding Register 0..65535  
 Nr. HR: Anzahl Holding Register 1..255  
 Data: Daten 0..1024

#### CMD Beispiele

Beschreibung	Command / Response	Initator	CMD	Modbus Adr	Holding Register	Anzahl Hol.-Reg	Data	Info
Notification Read	Command	CLC	0xC1	[MB Adr]	[HR-Reg]	[#HR]	Data [0..(x-1)]	x = 2* Anzahl-Hol.Reg
	Response	CLBTM	0xF0/ 0xFx					ACK / NACK
Notification Write	Command	CLC	0xC2	[MB Adr]	[HR-Reg]	[#HR]	None	
	Response	CLBTM	0xF0/ 0xFx					ACK / NACK

### Protokollaufbau Notifikation Error (CLC zum CLCBTM)

#### Command

Lenght = length + Scr + Dest + Cnt + Nutzdaten										
CR	length	Source	Dest.	Cnt	CMD	Modbus Adr.	HR Adr.	Nr. HR	Data	LF
1 Byte	2 Bytes	2 Bytes	2 Bytes	1 Byte	1 Byte	1 Byte	2 Byte	1 Byte	2 Bytes	1 Byte

#### Response

Lenght = length + Scr+ Dest + Cnt + Nutzdaten								LF
CR	length	Source	Dest.	Cnt	CMD			LF
1 Byte	2 Bytes	2 Bytes	2 Bytes	1 Byte	1 Byte			1 Byte

length: length + Scr+ Dest + Cnt + Nutzdaten (CMD bis und mit Data)  
 Source: Data source. 0 = broadcast, 0x7918 (31000) = tunnel to gateway  
 Dest.: Sent to Node. 0x7918 (31000) = tunnel to gateway  
 cnt: Ringcounter, Beschreibung siehe Kapitel Verhalten Counter Seite 16  
 CMD: Kommandobefehl, Beschreibung siehe Kapitel Kommandobefehle Seite 15  
 ModbusAdr: Modbus Device Adresse 1..255  
 HR Adr.: Modbus Startadresse Holding Register 0..65535  
 Nr. HR: Anzahl Holding Register 1..255  
 Data: Beschreibung untenstehend

#### CMD Beispiel

Beschreibung	Command / Response	Initiator	CMD	Modbus Adr	Holding Register	Anzahl Hol.-Reg	Data	Info
Notification Error	Command	CLC	0xC0	[MB Adr]	[HR-Reg]	[#HR]	Errordata [0..(x-1)]	x = 2* Anzahl-Hol.Reg
	Response	CLBTM	0xF0/ 0xFx					

#### Beschreibung Errordata (Data)

Errordata Type	data[1] Extended Error Code	data[0] Error Code	Info
Sucess	0xXX	0x00	Sucess
Modbus Exception	Modbus exception	0x01	Modbus Excepcion (Data[1]): Gemäss MODBUS APPLICATION PROTOCOL SPECIFICATION 1.1
Modbus Timeout	0xXX	0x02	Modbus timeout / Slave timeout
Invalidde lenght	0xXX	0x03	Invalidde lenght

### Protokollaufbau Interne Kommunikation CLC - CLBTM

Die interne Kommunikation ist für Kommandos gedacht die nur zwischen CLC und CLBTM ausgetauscht werden und nicht über Holdingregister (und somit für die Cloud oder die Leuchte) verfügbar sein müssen. Hier wird die Response instantan zurückgeschickt (im CMD Feld das CMD Pattern wiederholt).

Beschreibung	Command / Response	Initiator	CMD	Data	Info
<b>Get Smartbeacon ID</b>	Command	CLC	0xB0	None	
	Response		0xB0/ 0xFx	Data 0 – 1: 0x0000 – 0xFFFF	SmartbeaconID / NACK
<b>Get Smartbeacon Status</b>	Command	CLC	0xB1	None	
	Response		0xB1/ 0xFx	Data 0 - 1: 0x0000 = offline 0x0001 = connected	Connection status / NACK
<b>Reset CLBTM</b>	Command	CLC	0xB2	None	
	Response		0xB2/ 0xFx	None	Nach Antwort wird Reset ausgeführt / NACK
<b>Set Ready to Pair</b>	Command	CLC	0xB5	Data 0 – 1: 0x0001 = Head 1 0x0002 = Head 2	HeadID
	Response		0xB5/ 0xFx	None	ACK / NACK
<b>Head type and number of light decks</b>	Command	CLC	0xB6	Data 0 – 1: 0x0000 = slim 0x0001 = tunable white 0x0002 = enec 0x0003 = smartmodul (standalone) Data 2 – 3: 0x0001 = 1 Light deck 0x0002 = 2 Light decks 0x0003 = 3 Light decks 0x0004 = 4 Light decks	Sent from CLC 2sec after boot up - head type - number of light decks
	Response		0xB6/ 0xFx	None	ACK / NACK
<b>Commandos Testprocedures set</b>	Command	CLC	0xB8	Data 0 – 1: 0x0000 = LED's are in control of CLBTM 0x0001 = LED 1 on (SIO_15) 0x0002 = LED 1 off (SIO_15) 0x0003 = LED 2 on (SIO_17) 0x0004 = LED 2 off (SIO_17)	Set LED's CLBTM / NACK
	Response		0xB8/ 0xFx	None	ACK / NACK
<b>Commandos Testprocedures get</b>	Command	CLC	0xB9	None	
	Response		0xB9/ 0xFx	Data 0-3: u32 bootloaderVersion Data 4-7: u32 softDeviceVersion Data 8-11: u32 applicationVersion Data 12-15	Firmware Versions, MeshID / NACK  Aufschlüsselung: bootloaderVersion = u32 softDeviceVersion = u32 applicationVersion = 100000000

				u32 serialIndex (MeshID)	* fw_major + 1000 * fw_minor + fw_patch (bsp 1.2.3) serialindex = generateBeaconSerialForIndex()
<b>Commandos Testprocedures test bluetooth connection</b>	Command	CLC	0xBA	Data 0-5: 6 x u8 (Bsp: FFCCAA332211)	MAC Adress Golden Device Bsp: 11-22-33-AA-CC-FF
	Response		0xBA/ 0xFx	Data 0 – 1: 0x0000 = connection test not sucessful 0x0001 = connection test sucessful	Status connection test / NACK

## Command

Lenght = length + Scr/Dest/Cnt + Nutzdaten							
CR	lenght	Source	Dest.	cnt	CMD	Data	LF
1 Bytes	2 Bytes	2 Bytes	2 Bytes	1 Byte	1 Byte	0-1024 Bytes	1 Byte

## Response

Lenght = length + Scr/Dest/Cnt + Nutzdaten							
CR	lenght	Source	Dest.	cnt	CMD	Data	LF
1 Bytes	2 Bytes	2 Bytes	2 Bytes	1 Byte	1 Byte	0-1024 Bytes	1 Byte

Source: immer 0

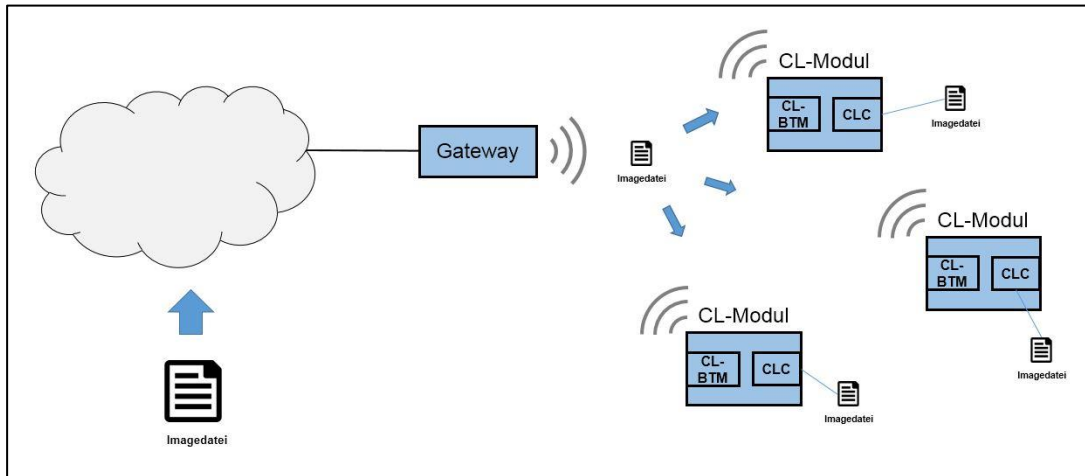
Dest.: immer 0

cnt: Ringcounter, Beschreibung siehe Kapitel Verhalten Counter Seite 16

CMD: interne Commandos, Beschreibung siehe Kapitel Kommandobefehle Seite 15

## Protokoll und Aufbau File Downstream / Firmware update (CLBTM zum CLC)

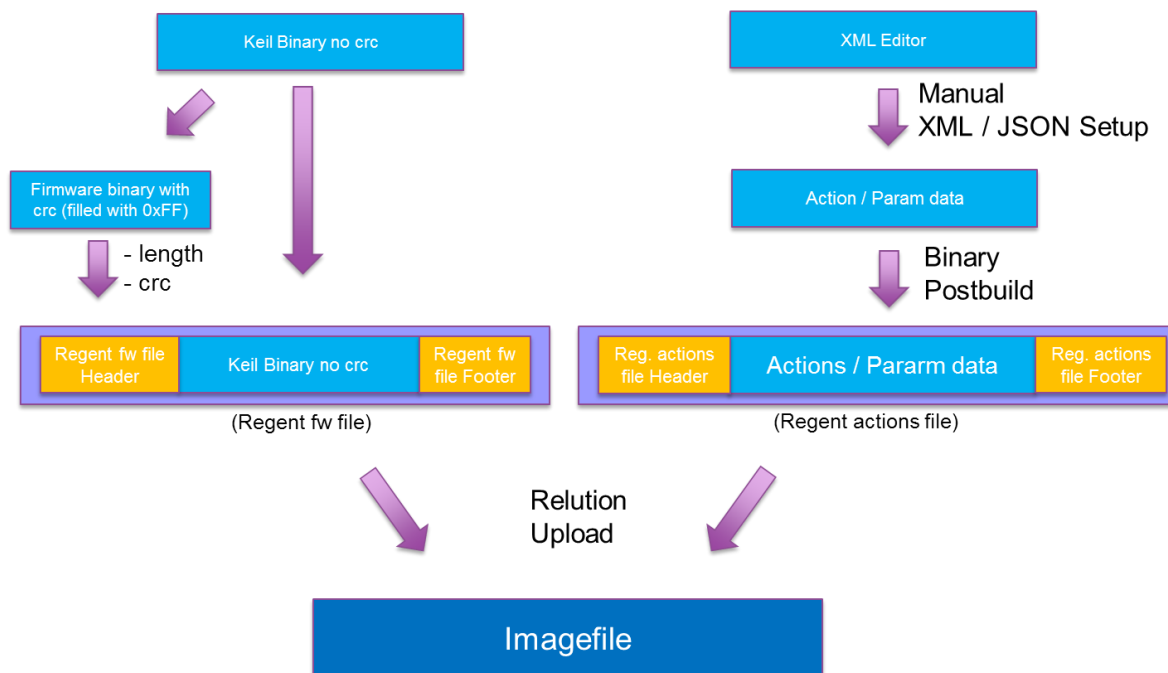
Protokollaufbau Downstream (CLBTM → CLC) bei einem Dateitransfer (Imagedatei.\*.img):

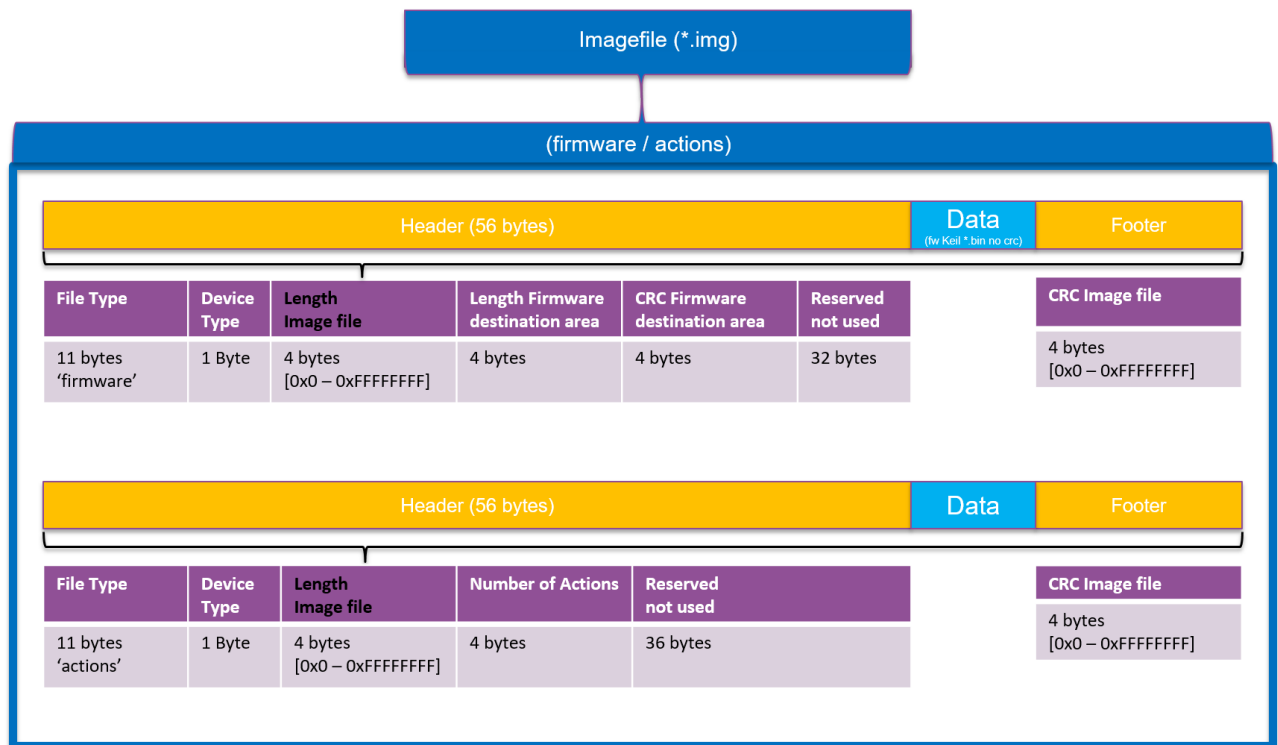


Grössere Datenmengen werden in Form einer Imagedatei (binäre Daten) von der Cloud zum CLM transferiert. Diese sind:

- Firmware file (Steuermodul, Sensormodul, Panel, ALONEatWORK, CLM, etc)
- Actions file (regelt und definiert Daten welche von der Leuchte zur cloud gesendet werden)

## Aufbau der Imagedatei





File Type: „firmware“ (string, Null terminiert)

„actions“ (string, Null terminiert)

Device Type:

- 1 = Panel
- 2 = Control unit m3
- 3 = ALONEatWORK 2.0
- 4 = Sensormodule Slim
- 6 = Control unit m4
- 10 = CLM
- 11 = Smartmodule
- 240 = Actions file

Length Firmware: Länge der Regent modifizierten firmware (aufgefüllt mit 0xFF und CRC32)

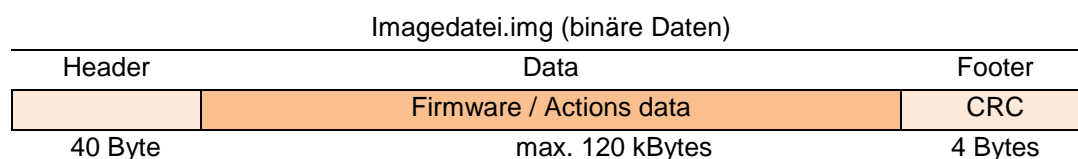
CRC Firmware: CRC der Regent modifizierten firmware. CRC-32 polynomial: 0x4C11DB7

Length Regent fw file: Totale Länge der Imagedatei

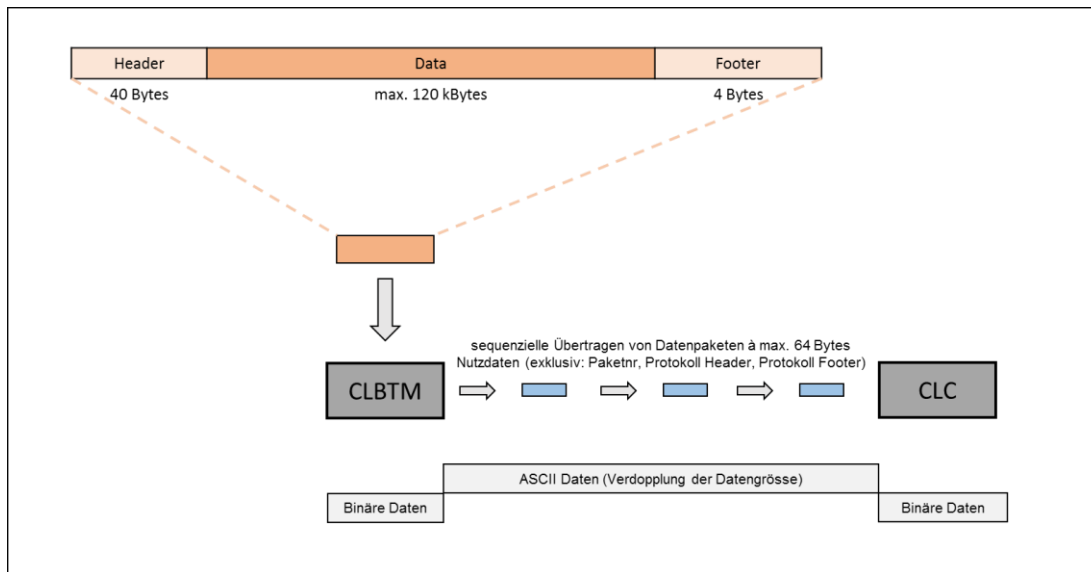
Data: Nutzdaten (Regent firmware file / Regent actions file) max. 122'880 bytes

CRC Regent fw file: CRC-32 der Imagedatei

### Transfer







Beschreibung	Command / Response	Initiator	CMD	Data	Info
<b>Imagedatei Interesse?</b>	Command	CLBTM	0xE0	Data 0-1: Devicetype	
	Response	CLC	0xE0	Data 0-1: Interesse	0 = YES 1 = NO 0xFx = NACK Befehl
<b>Imagedatei downstream</b>	Command	CLBTM	0xE1	Data 0-1: Paketnr Data 2-65: Data (Binär max. 64 Bytes, Datenpaket muss immer durch 4 teilbar sein)	Paketnr: 0x0 – 0xFFFF Paketnr = 0 löscht den reservierten Flashbereich des CLC/
	Response		0xE1/ 0xFx	Data 0-1: Status	0 = Paket ok 1 = not used → Retry/Repeat 2 = no data or not Modulo 4 → Restart 3 = Flash error → Restart 4 = too much transferred data → Restart 5 = wrong counter number → Restart ≥6 = failed → abort 0xFx = NACK
<b>Imagedatei aktivieren</b>	Command	CLBTM	0xE2	None	
	Response	CLBTM	0xE2/ 0xFx	Data 0-1: Status (nach kürzerer Zeit)	0 = ok 1 = error CRC 0xFx = NACK
<b>Imagedatei erfolgreich geflasht</b>	Command	CLC	0xE3	Data 0: Status	Status: 0 = ok 1-65635 = nicht erfolgreich (wiederholen bei NACK oder timeout)
	Response		0xE3/ 0xFx		0 = ACK 0xFx = NACK

## Command

$$\text{lenght} = \text{length} + \text{Src/Dest/cnt} + \text{Nutzdaten}$$

CR	lenght	Source	Dest.	cnt	CMD	Data	LF
1 Bytes	2 Bytes	2 Bytes	2 Bytes	1 Byte	1 Byte	0-x Bytes	1 Byte

## Response

$$\text{lenght} = \text{length} + \text{Src/Dest/cnt} + \text{Nutzdaten}$$

CR	lenght	Source	Dest.	cnt	CMD	Data	LF
1 Bytes	2 Bytes	2 Bytes	2 Bytes	1 Byte	1 Byte	0-x Bytes	1 Byte

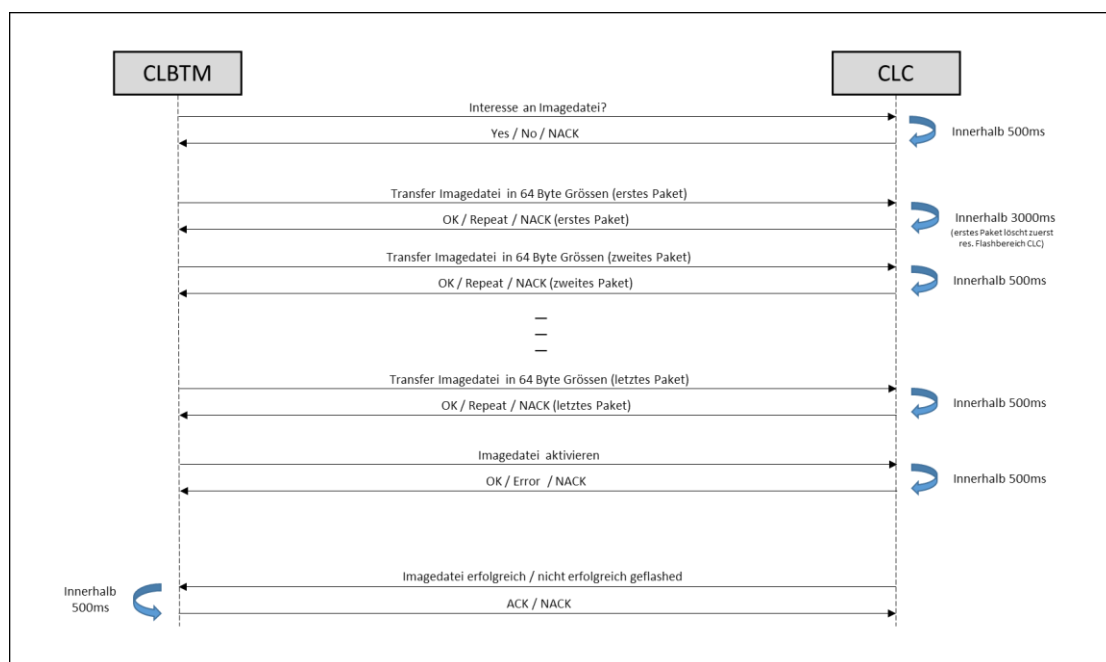
length: length+Src + Dest + cnt + CMD + Data

Source: Command = 0x7918 (31000) = tunnel to gateway / Response = SmartbeaconID

Dest.: Command = SmartbeaconID / Response: 0x7918 (31000) = tunnel to gateway

cnt: Ringcounter, Beschreibung siehe Kapitel Verhalten Counter Seite 16

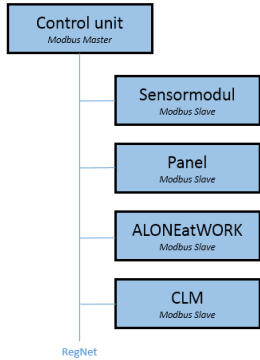
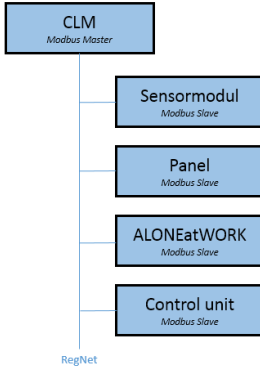
CMD: Kommandobefehl, Beschreibung siehe Kapitel Kommandobefehle Seite 15



Zwischen dem Downstreams Datenpaketen können auch andere Befehle dazwischen kommen.

Eine Imagedatei wird sequenziell transferiert bis diese komplett übertragen worden ist. Erst nach beenden der Transfers kann eine neue Imagedatei transferiert werden.

### 3.3.3. Protokoll zwischen CLM und Modbus

RQ	Protokoll CLM – Modbus
A	Protokoll: 115200 Baud, 8N1
B	Neue Modbus Register müssen eingeführt werden: Definiert im <a href="#">Abschnitt 3.6.2. Datenherkunft Steuermodul</a>
C	<p>Control unit ist Modbus Master bei Parameter Transfer</p>  <pre> graph TD     CU[Control unit Modbus Master] --- Bus[RegNet]     Bus --- SM[Sensormodul Modbus Slave]     Bus --- P[Panel Modbus Slave]     Bus --- AW[ALONEatWORK Modbus Slave]     Bus --- CLM[CLM Modbus Slave]           </pre>
D	<p>Im Falle eines Firmwareupdates kann das CPM das Control unit in den Modbus Slave Zustand versetzen und selber Modbus Master werden um einen Firmwareupdate an den Modbusteilnehmern vorzunehmen.</p>  <pre> graph TD     CLM[CLM Modbus Master] --- Bus[RegNet]     Bus --- SM[Sensormodul Modbus Slave]     Bus --- P[Panel Modbus Slave]     Bus --- AW[ALONEatWORK Modbus Slave]     Bus --- CU[Control unit Modbus Slave]           </pre>

E

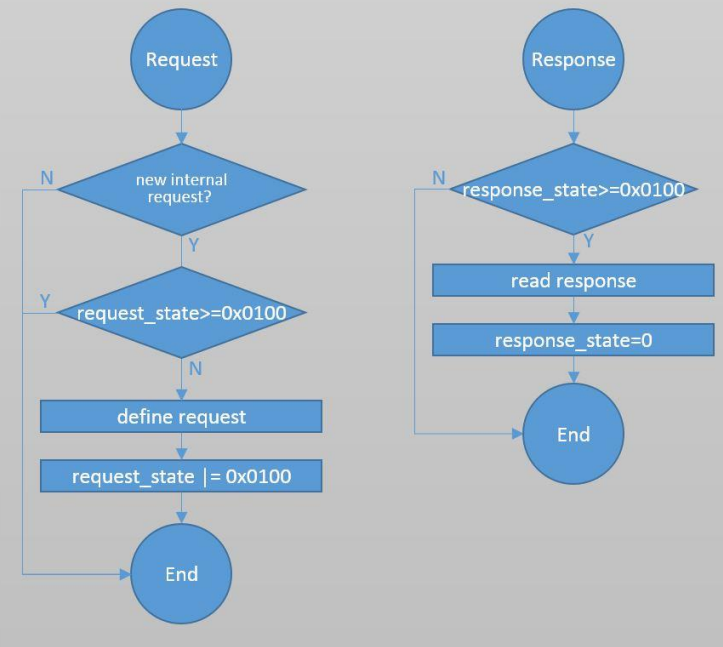
Weil das CLM den Datenaustausch regelt aber ein Modbus Slave Teilnehmer ist, werden Exchange Holdingregister eingeführt. Über diese Exchange Holdingregister kann das CLM Daten bei dem control unit anfragen.

Folgende Holdingregister Struktur regelt den Datenaustausch:

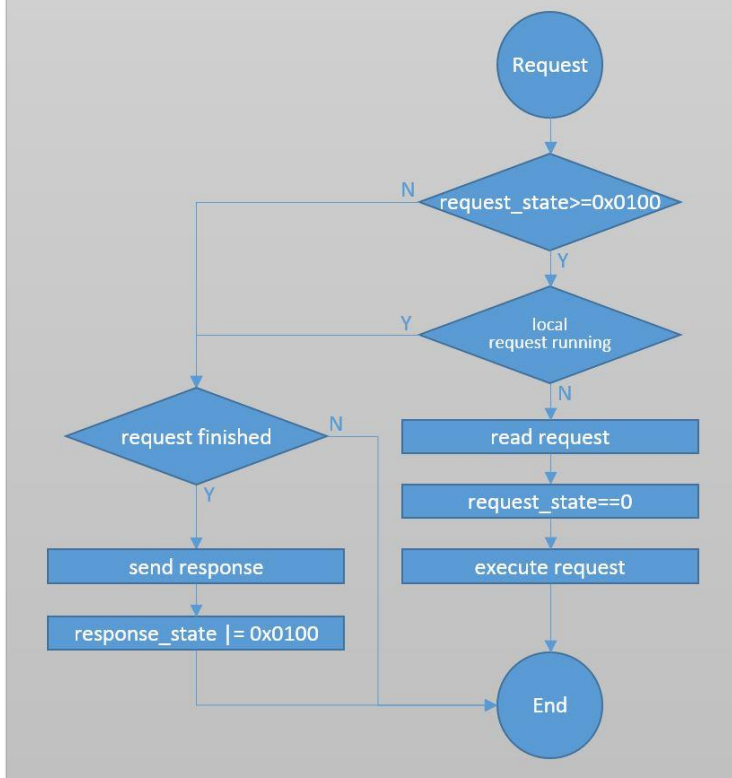
	B15	B8 B7	B0
exchange_request_state	<b>State</b> 0 = empty (data processed) 1 = new data ready		<b>Access</b> 0 =read 1=write
exchange_request_address	<b>Address</b> Device address modbus slave		<b>NumberHR</b> Number of holding registers to read/write
exchange_request_hr	<b>HR</b> Start address holding register		
exchange_request_data0	<b>data 0</b>		
...	...		
exchange_request_data31	<b>data 31</b>		
	...		
exchange_response_state	<b>State</b> 0 = empty (data processed) 1 = new data ready		<b>Access</b> 0 =read 1=write
exchange_response_error	<b>Exception</b> Modbus exeption code		<b>error</b> 0 = success 1 = modbus exception 2 = modbus slave timeout (not available)
Exchange_response_address	<b>Address</b> Modbus slave device		<b>NumberHR</b> Number of holding registers to read/write
Exchange_response_hr	<b>HR</b> Start address holding register		
exchange_response_data0	<b>data 0</b>		
...	...		
exchange_response_data31	<b>data 31</b>		

Flussdiagramme:

## CLM

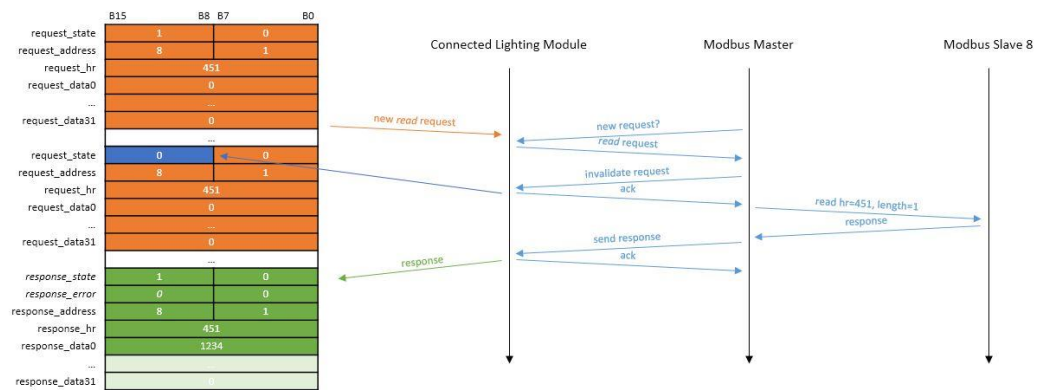


## Control Unit

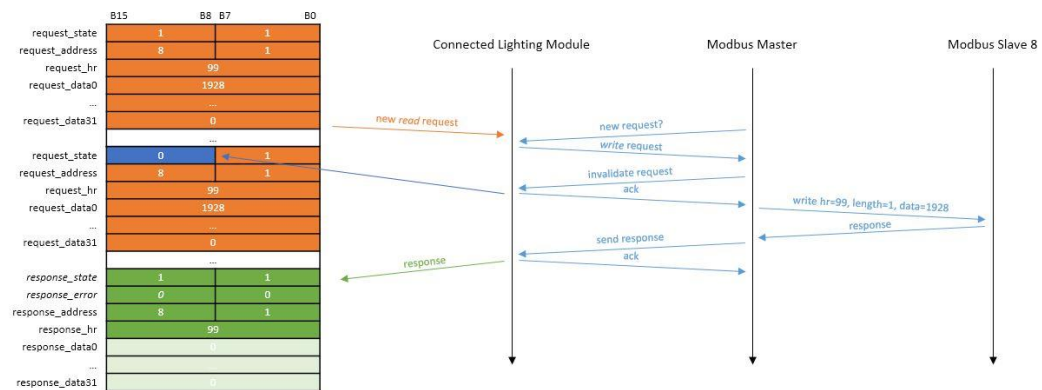


## Beispiele:

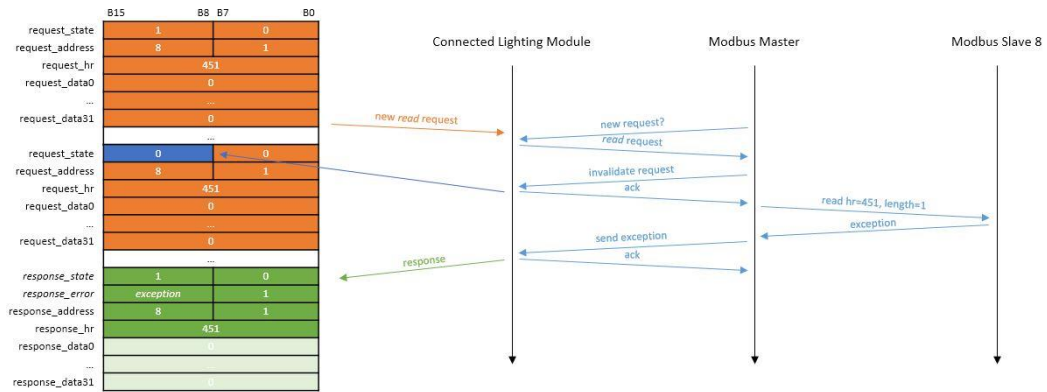
## Read Access



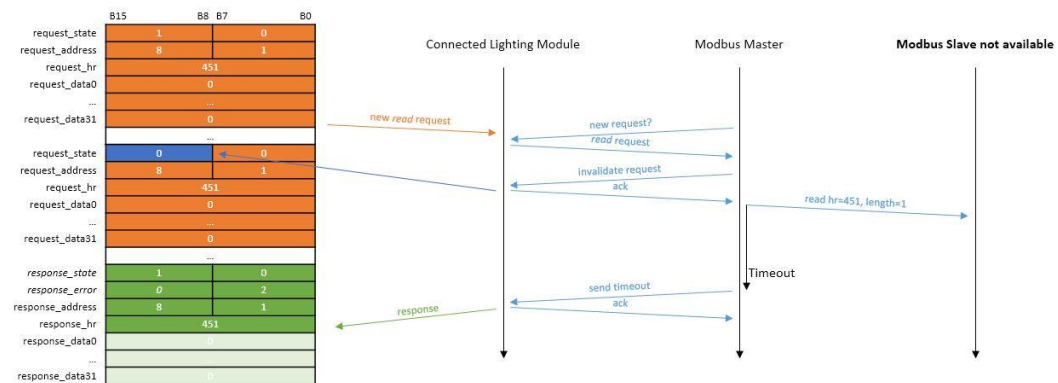
## Write Access



## Modbus Exception



## Modbus Slave Timeout



### 3.4. Action File

Das action file konfiguriert und regelt den Upstream von Daten. Das action file ist eine \*.bin Datei, verpackt in einer \*.img Datei.

Daten können so konfiguriert werden dass alle zur Verfügung stehenden Daten der Holdingregister hochgesendet werden oder beschrieben werden:

- Einmalig
- Ereignisgesteuert (mit / ohne Betrachtung Änderung absoluter / relativer Hysteresenwert, zum Zeitpunkt des abgelaufenen periodischen Timers!)
- Periodisch (mit / ohne Betrachtung Änderung absoluter / relativer Hysteresenwert)

Remote Add.	Periodic rate [s]	CNT	CMD	HR Adr.	Nr of HR	Device Adr.	absolute change	relative change [%]	Event triggert
(2 Bytes)	(2 Bytes)	(1 Byte)	(1 Byte)	(2 Bytes)	(1 Bytes)	(1 Byte)	(2 Byte)	(1 Byte)	(1 Byte)
0x7918	0x1E	0x00	0xD1	0x130A	0x01	0x017	0x14	0x00	0x00
0x7918	0x01	0x00	0xD1	0x1307	0x01	0x017	0x00	0x05	0x01
...									

Remote address:	Adresse Gateway: 0x7918, Adresse für timesync: 0xFFFF
Periodic rate:	Zeit in [s] um periodisch Daten zu senden (0 = deactivated)
CNT:	Default = 0x00. Interner counter (Wert wird in der firmware dynamisch generiert)
CMD:	0xD1: read Data 0xD0: write Data
HR Adr:	Holding Register Adresse
Nr. of HR:	Anzahl zu sender Holding Register
Device Adr:	Modbus Device Adresse
Absolute change:	Sende Daten (periodic rate / Event triggert) falls sich der Wert um <i>Absolute change</i> ändert
Relative change:	Sende Daten (periodic rate / Event triggert) falls sich der Wert um <i>Relative change</i> ändert
Event triggert:	0 = deactivated, 1 = Send on change (benötigt einen Wert "Periodic rate" > 0), 2 = only once at Startup

Bsp :

Sende HR 130A von Device 23 alle 30 s falls sich der Wert um 0x14 ändert

Sende HR 1307 von Device 23 falls sich der Wert um 5 % ändert (sampling rate 1 s)



### 3.4.1. Speicherstruktur action file

1 Flashpage (2048 Bytes) steht zu Verfügung, unmittelbar nach dem Mirror Flash Bereich.

Remote Address	(2 Bytes)
Timer [s]	(2 Bytes)
CNT	(1 Byte)
CMD	(1 Byte)
Reserved	(2 Bytes)
HRAdr	(2 Bytes)
nrHR	(1 Byte)
ModbusAdr	(1 Byte)
Ads. change	(2 Bytes)
Rel. change [%]	(1 Byte)
Event trigger	(1 Byte)
Data	(32 x 2 Bytes)
<hr/>	
Total grösse Struktur Actions	(80 Bytes)

$2048 / 80 = 25$  Actions

⇒ Maximal 25 Actions zur Verfügung!

### **3.5. Startup Verhalten CLC - CLBTM**

Beim Lightpad TW Leuchte fragt das CLBTM beim Bootup (nach 2 s) zwei Register beim CLC ab. Dies um das Bild in der App richtig darzustellen. (neuste fw fragt nur an, falls noch nicht bekannt)

- Register 0x2001 (Device 8) Number of Lightdecks (1 – 4)
- Register 0x2717 (Device 8) Luminaire Type (0 = Normal, 1 = Tunable, 2 = ENEC)

CLC sendet nach 1.5 s die Daten der beiden Register 0x2001 und 0x2717 ohne Aufforderung zum CLBTM. Falls keine Kommunikation zum Steuermodul existiert, wird ein default Wert (2 head, version tw) an das CLBTM gesendet.

### 3.6. Gemeinsame Holdingregister Steuermodul – CLM

#### 3.6.1. Gemeinsame Holdingregister Steuermodul – CLM

Data	HR	Grösse	Zugriffsrechte	EEPROM	Default
exchange_request_state	0x0000	2 Bytes	R / W	0	0
exchange_request_address	0x0001	2 Bytes	R / W	0	0
exchange_request_hr	0x0002	2 Bytes	R / W	0	0
exchange_request_data_0	0x0003	2 Bytes	R / W	0	0
exchange_request_data_1	0x0004	2 Bytes	R / W	0	0
exchange_request_data_2	0x0005	2 Bytes	R / W	0	0
exchange_request_data_3	0x0006	2 Bytes	R / W	0	0
exchange_request_data_4	0x0007	2 Bytes	R / W	0	0
exchange_request_data_5	0x0008	2 Bytes	R / W	0	0
exchange_request_data_6	0x0009	2 Bytes	R / W	0	0
exchange_request_data_7	0x000A	2 Bytes	R / W	0	0
exchange_request_data_8	0x000B	2 Bytes	R / W	0	0
exchange_request_data_9	0x000C	2 Bytes	R / W	0	0
exchange_request_data_10	0x000D	2 Bytes	R / W	0	0
exchange_request_data_11	0x000E	2 Bytes	R / W	0	0
exchange_request_data_12	0x000F	2 Bytes	R / W	0	0
exchange_request_data_13	0x0010	2 Bytes	R / W	0	0
exchange_request_data_14	0x0011	2 Bytes	R / W	0	0
exchange_request_data_15	0x0012	2 Bytes	R / W	0	0
exchange_request_data_16	0x0013	2 Bytes	R / W	0	0
exchange_request_data_17	0x0014	2 Bytes	R / W	0	0
exchange_request_data_18	0x0015	2 Bytes	R / W	0	0
exchange_request_data_19	0x0016	2 Bytes	R / W	0	0
exchange_request_data_20	0x0017	2 Bytes	R / W	0	0
exchange_request_data_21	0x0018	2 Bytes	R / W	0	0
exchange_request_data_22	0x0019	2 Bytes	R / W	0	0
exchange_request_data_23	0x001A	2 Bytes	R / W	0	0
exchange_request_data_24	0x001B	2 Bytes	R / W	0	0
exchange_request_data_25	0x001C	2 Bytes	R / W	0	0
exchange_request_data_26	0x001D	2 Bytes	R / W	0	0
exchange_request_data_27	0x001E	2 Bytes	R / W	0	0
exchange_request_data_28	0x001F	2 Bytes	R / W	0	0
exchange_request_data_29	0x0020	2 Bytes	R / W	0	0
exchange_request_data_30	0x0021	2 Bytes	R / W	0	0
exchange_request_data_31	0x0022	2 Bytes	R / W	0	0
exchange_response_state	0x0030	2 Bytes	R / W	0	0
exchange_response_error	0x0031	2 Bytes	R / W	0	0
Exchange_response_address	0x0032	2 Bytes	R / W	0	0
Exchange_response_hr	0x0033	2 Bytes	R / W	0	0
exchange_response_data_0	0x0034	2 Bytes	R / W	0	0
exchange_response_data_1	0x0035	2 Bytes	R / W	0	0
exchange_response_data_2	0x0036	2 Bytes	R / W	0	0
exchange_response_data_3	0x0037	2 Bytes	R / W	0	0

exchange_response_data_4	0x0038	2 Bytes	R / W	0	0
exchange_response_data_5	0x0039	2 Bytes	R / W	0	0
exchange_response_data_6	0x003A	2 Bytes	R / W	0	0
exchange_response_data_7	0x003B	2 Bytes	R / W	0	0
exchange_response_data_8	0x003C	2 Bytes	R / W	0	0
exchange_response_data_9	0x003D	2 Bytes	R / W	0	0
exchange_response_data_10	0x003E	2 Bytes	R / W	0	0
exchange_response_data_11	0x003F	2 Bytes	R / W	0	0
exchange_response_data_12	0x0040	2 Bytes	R / W	0	0
exchange_response_data_13	0x0041	2 Bytes	R / W	0	0
exchange_response_data_14	0x0042	2 Bytes	R / W	0	0
exchange_response_data_15	0x0043	2 Bytes	R / W	0	0
exchange_response_data_16	0x0044	2 Bytes	R / W	0	0
exchange_response_data_17	0x0045	2 Bytes	R / W	0	0
exchange_response_data_18	0x0046	2 Bytes	R / W	0	0
exchange_response_data_19	0x0047	2 Bytes	R / W	0	0
exchange_response_data_20	0x0048	2 Bytes	R / W	0	0
exchange_response_data_21	0x0049	2 Bytes	R / W	0	0
exchange_response_data_22	0x004A	2 Bytes	R / W	0	0
exchange_response_data_23	0x004B	2 Bytes	R / W	0	0
exchange_response_data_24	0x004C	2 Bytes	R / W	0	0
exchange_response_data_25	0x004D	2 Bytes	R / W	0	0
exchange_response_data_26	0x004E	2 Bytes	R / W	0	0
exchange_response_data_27	0x004F	2 Bytes	R / W	0	0
exchange_response_data_28	0x0050	2 Bytes	R / W	0	0
exchange_response_data_29	0x0051	2 Bytes	R / W	0	0
exchange_response_data_30	0x0052	2 Bytes	R / W	0	0
exchange_response_data_31	0x0053	2 Bytes	R / W	0	0
Aktuelle Zeit [min] Bsp: 00:02 → 2 Bsp: 12:00 → 720	0x0060	2 Bytes	R / W	0	0
Aktuelles Datum Jahr [YY] Bsp: 2021 → 21	0x0061	2 Bytes	R / W	0	0
Aktuelles Datum Monat [MM] Bsp: 1 → Jan Bsp: 12 → Dez	0x0062	2 Bytes	R / W	0	0
Aktueller Wochtag [W] W = Wochentag, 1=Mo...7=So	0x0063	2 Bytes	R / W	0	0

On request by CLM through exchange registers:

- FW-Version (von control unit / Sensormodule / Panel / ALONEatWORK)
- FW-BL-Version (von control unit / Sensormodule / Panel / ALONEatWORK)
- HW/BOM-Vers. (von control unit / Sensormodule / Panel / ALONEatWORK)
- Serial\_number (von control unit / Sensormodule / Panel / ALONEatWORK)
- HCL\_T\_TABLE\_x
- HCL\_C\_TABLE\_x
- MODE\_HEAD\_T
- HEAD\_TW\_PROTECTED

- PIR delay time
- Security level delay
- Powersafe en./disable
- Powersafe treshhold
- Powersafe duration
- ALONEatWORK en./disable
- Status CLM (mesh connected / not connected)

### 3.7. Datentransparenz CLM Modul

#### 3.7.1. Datenherkunft aller Regent Modbus Modulen

Data	HR	Grösse	einmalig	Event	periodisch
FW-Version	0x0000	32 Bytes	x		
FWBL-Version	0x0035	32 Bytes	x		
HW-Version	0x0218	8 Bytes	x		
Serial_number	0x0210	8 Bytes	x		

#### 3.7.2. Datenherkunft Steuermodul

Data	HR	Grösse	Zugriff	einmalig	Event	periodisch
Number of light decks	0x2001	2 Bytes	R	x		
Leuchte tunable white?	New	2 Bytes	R	x		
Status Leuchtenkopf 1 (Nachlaufzeit aktiv)	No	2 Bytes	R		x	
Status Leuchtenkopf 2 (Nachlaufzeit aktiv)	No	2 Bytes	R		x	
Status Leuchtenkopf 3 (Nachlaufzeit aktiv)	No	2 Bytes	R		x	
Status Leuchtenkopf 4 (Nachlaufzeit aktiv)	No	2 Bytes	R		x	
Nachlaufzeit aktiv Kopf 1	No	2 Bytes	R		x	
Nachlaufzeit aktiv Kopf 2	No	2 Bytes	R		x	
Nachlaufzeit aktiv Kopf 3	No	2 Bytes	R		x	
Nachlaufzeit aktiv Kopf 4	No	2 Bytes	R		x	
Daliwert Leuchtenkopf kalt 1 (oder Kopf 1)	No	2 Bytes	R		x	x
Daliwert Leuchtenk. warm 1	No	2 Bytes	R		x	x
Daliwert Leuchtenkopf kalt 2 (oder Kopf 2)	No	2 Bytes	R		x	x
Daliwert Leuchtenk. warm 2	No	2 Bytes	R		x	x
Daliwert Leuchtenkopf kalt 3 (oder Kopf 3)	No	2 Bytes	R		x	x
Daliwert Leuchtenk. warm 3	No	2 Bytes	R		x	x
Daliwert Leuchtenkopf kalt 4 (oder Kopf 4)	No	2 Bytes	R		x	x
Daliwert Leuchtenk. warm 4	No	2 Bytes	R		x	x
PIR delay time	0x2003	2 Bytes	R / W		x	
Security level delay	0x2004	2 Bytes	R / W		x	
Powersafe en./disable	0x2018	2 Bytes	R / W		x	
Powersafe treshhold	0x2019	2 Bytes	R / W		x	
Powersafe duration	0x201A	2 Bytes	R / W		x	

ALONEatWORK en./disable	0x2022	2 Bytes	R / W		x	
-------------------------	--------	---------	-------	--	---	--

### 3.7.3. Datenherkunft Sensormodul

Data	Adr	HR	Grösse	einmalig	Event	periodisch
Helligkeitswert Leuchte 1	23	0x1307	2 Bytes			x
Helligkeitswert Leuchte 2	24	0x1307	2 Bytes			x
Helligkeitswert Leuchte 3	25	0x1307	2 Bytes			x
Helligkeitswert Leuchte 4	26	0x1307	2 Bytes			x
Arbeitsplatz besetzt 1	23	0x130A	2 Bytes		x	
Arbeitsplatz besetzt 2	24	0x130A	2 Bytes		x	
Arbeitsplatz besetzt 3	25	0x130A	2 Bytes		x	
Arbeitsplatz besetzt 4	26	0x130A	2 Bytes		x	

### 3.7.4. Datenherkunft Panel

Data	HR	Grösse	einmalig	Event	periodisch

### 3.7.5. Datenherkunft ALONEatWORK

Data	HR	Grösse	einmalig	Event	periodisch

### 3.7.6. Datenherkunft CLM

Data	HR	Grösse	einmalig	Event	periodisch
Status CLM (mesh connected / not connected)		2 Bytes		x	
Date year		2 Bytes		x	
Date month		2 Bytes		x	
Weekday		2 Bytes		x	
Time in minutes		2 Bytes			x

### 3.7.7. Datenherkunft Human Centric Lighting

Data	HR	Grösse	einmalig	Event	periodisch
HCL_T_TABLE_0 HCL_T_TABLE_0 – HCL_T_TABLE_47 = Timestamp				x	
HCL_T_TABLE_1				x	
HCL_T_TABLE_2				x	
HCL_T_TABLE_3				x	
HCL_T_TABLE_4				x	
HCL_T_TABLE_5				x	
HCL_T_TABLE_6				x	
HCL_T_TABLE_7				x	
HCL_T_TABLE_8				x	
HCL_T_TABLE_9				x	

HCL_T_TABLE_10				x	
HCL_T_TABLE_11				x	
HCL_T_TABLE_12				x	
HCL_T_TABLE_13				x	
HCL_T_TABLE_14				x	
HCL_T_TABLE_15				x	
HCL_T_TABLE_16				x	
HCL_T_TABLE_17				x	
HCL_T_TABLE_18				x	
HCL_T_TABLE_19				x	
HCL_T_TABLE_20				x	
HCL_T_TABLE_21				x	
HCL_T_TABLE_22				x	
HCL_T_TABLE_23				x	
HCL_T_TABLE_24				x	
HCL_T_TABLE_25				x	
HCL_T_TABLE_26				x	
HCL_T_TABLE_27				x	
HCL_T_TABLE_28				x	
HCL_T_TABLE_29				x	
HCL_T_TABLE_30				x	
HCL_T_TABLE_31				x	
HCL_T_TABLE_32				x	
HCL_T_TABLE_33				x	
HCL_T_TABLE_34				x	
HCL_T_TABLE_35				x	
HCL_T_TABLE_36				x	
HCL_T_TABLE_37				x	
HCL_T_TABLE_38				x	
HCL_T_TABLE_39				x	
HCL_T_TABLE_40				x	
HCL_T_TABLE_41				x	
HCL_T_TABLE_42				x	
HCL_T_TABLE_43				x	
HCL_T_TABLE_44				x	
HCL_T_TABLE_45				x	
HCL_T_TABLE_46				x	
HCL_T_TABLE_47				x	
HCL_T_TABLE_48				x	
HCL_C_TABLE_0 HCL_C_TABLE_0 – HCL_C_TABLE_47 = Color temp.				x	
HCL_C_TABLE_1				x	
HCL_C_TABLE_2				x	
HCL_C_TABLE_3				x	
HCL_C_TABLE_4				x	

HCL_C_TABLE_5				x	
HCL_C_TABLE_6				x	
HCL_C_TABLE_7				x	
HCL_C_TABLE_8				x	
HCL_C_TABLE_9				x	
HCL_C_TABLE_10				x	
HCL_C_TABLE_11				x	
HCL_C_TABLE_12				x	
HCL_C_TABLE_13				x	
HCL_C_TABLE_14				x	
HCL_C_TABLE_15				x	
HCL_C_TABLE_16				x	
HCL_C_TABLE_17				x	
HCL_C_TABLE_18				x	
HCL_C_TABLE_19				x	
HCL_C_TABLE_20				x	
HCL_C_TABLE_21				x	
HCL_C_TABLE_22				x	
HCL_C_TABLE_23				x	
HCL_C_TABLE_24				x	
HCL_C_TABLE_25				x	
HCL_C_TABLE_26				x	
HCL_C_TABLE_27				x	
HCL_C_TABLE_28				x	
HCL_C_TABLE_29				x	
HCL_C_TABLE_30				x	
HCL_C_TABLE_31				x	
HCL_C_TABLE_32				x	
HCL_C_TABLE_33				x	
HCL_C_TABLE_34				x	
HCL_C_TABLE_35				x	
HCL_C_TABLE_36				x	
HCL_C_TABLE_37				x	
HCL_C_TABLE_38				x	
HCL_C_TABLE_39				x	
HCL_C_TABLE_40				x	
HCL_C_TABLE_41				x	
HCL_C_TABLE_42				x	
HCL_C_TABLE_43				x	
HCL_C_TABLE_44				x	
HCL_C_TABLE_45				x	
HCL_C_TABLE_46				x	
HCL_C_TABLE_47				x	
HCL_C_TABLE_48				x	
MODE_HEAD_TW			x		



High Byte : head 2 Low Byte : head 1 0 = not tunable 1 = Natural 2 = Personal 3 = Manuel					
HEAD_TW_PROTECTED High Byte : head 2 Low Byte : head 1 0 = not proteted 1 = protected				x	
MANUAL_BRIGHTNESS_0 Brightness head 1 : 0-100%					x
MANUAL_BRIGHTNESS_1 Brightness head 2 : 0-100%					x
MANUAL_COLOR_0 Color head 1 : 2700 – 6500					x
MANUAL_COLOR_1 Color head 2 : 2700 – 6500					x

## 4. Funktionen

### 4.1. Bootup Abfrage des angeschlossenen Systems

Beim Lightpad TW Leuchte fragt das CLBTM beim Bootup zwei Register beim CLC ab. Dies um das Bild in der App richtig darzustellen.

- Register 0x2001 (Device 8) Number of Lightdecks (1 – 4)
- Register 0x2717 (Device 8) Luminaire Type (0 = Normal, 1 = Tunable, 2 = ENEC....)

Lightpad TW Antworten:

- Number of Lightdecks = 1 oder 2

- Luminaire Type = 1
- 

Bei Siemens hatten wir eine normale Lightpad in Betrieb. Deshalb waren die Antworten:

- Number of Lightdecks: 1, 2,3, oder 4
- Luminaire Type: 0
- Dabei haben wir bemerkt, dass der Luminaire Type immer wieder angefragt wurde vom CLBTM obwohl das CLC mit 0 geantwortet hat. Das CLBTM Mühe mit dem Luminaire Typ 0.
- Als Workaround hat das CLC dem CLBTM den Luminaire Type 1 zurückgeantwortet für den Siemensfall.

Wir haben noch einen weiteren Fall, das sogenannte Standalone Modul:

- Dieses Modul hat kein Device 8 und antwortet somit nicht auf die Anfragen von Number of Lightdecks und Luminaire Type.

Nach Besprechung mit Mike schlagen wir folgendes vor:

Gültige Antworten	Nr of Lightdecks	Luminaire Type	Bemerkung
Lightpad TW	1 oder 2	1	App Verbindung erlaubt
Lightpad normal (Slim)	1 oder 2 oder 3 oder 4	0	KEINE App Verbindung erlaubt
Standalone Module	Antwortet nicht	Antwortet Nicht	Nach 5 Versuchen abbrechen KEINE App Verbindung

## 4.2. Funktionen für Tunable White

- CLM meldet an Bluetooth Modul folgende Werte
  - o Leuchtentyp und -ausprägung (1 Kopf, 2 Kopf)
  - o Protected Mode Status (siehe Spezifikation App)
  - o ~~Protected Mode Password~~
- Bluetooth Modul meldet an CL folgende Werte
  - o Verbindungsstatus Bluetooth Modul
  - o Aktivierung Protected Mode (siehe Spezifikation App)
  - o **Abspeicherung Protected Mode Passwort**
  - o Zeit / Farbtemperatur Werte (insgesamt 64)
  - o Datum / Zeit Werte
- CLM unterhält:
  - o RTC Funktionalität und stellt diese interessierten Busteilnehmern zur Verfügung
  - o Übermittelt Zeit / Farbtemperatur Werte
  - o Unterstützt alle Standardfunktionen (Firmwareupdate über Modbus, Seriennummern etc.)

Im Dokumentenordner [Mobile App Beschreibung](#) wird genauer beschrieben, welche Daten zwischen App und Leuchte ausgetauscht werden müssen.

### 4.3. Funktionen für Connected Lighting

- CLM verwaltet folgende Punkte:
  - Unterhalten eines zeitlich „genauen“ Persistenzabbilds von relevanten Parametern (gemäss Kapitel **Fehler! Verweisquelle konnte nicht gefunden werden.**) über die gesamte Leuchte (bspw. PIR Zustände aller Leuchtenköpfe)
  - Verwaltung der aktivierten Upstream-Services und dadurch auch die Verwaltung, welche Daten über Bluetooth Modul an die Analytics geliefert werden müssen (Service Desksharing verlangt die Änderungen des PIR Sensors eventuell mit Zeitstempel)
  - Verwaltung der aktivierten Downstream-Services und dadurch auch die Verwaltung, welche Daten vom Bluetooth Modul akzeptiert werden zur Weiterleitung an das Steuermodul (Service HCL verlangt die Veränderung von Zeit/Farbtemperatur Tupeln)
  - Verteilung und Abarbeiten von Firmware-Updates, Konfigurationen und Parameterupdates für alle Modbus-Teilnehmer im Zusammenspiel mit dem Steuermodul.
  - Verwaltung von Administrationsmessages im Zusammenhang mit dem Bluetooth Modul, resp. der Bluerange Firmware auf dem Bluetooth Modul (Prüfung der Konnektivität, ev. Fehlerzustände signalisieren, Seriennummern und Identifikationsdaten verwalten und zur Verfügung stellen).

## A. Anhang

### A.1. Abkürzungen

CLM      Connected Lighting Modul  
 CLE      Connected Lighting Controller  
 CLBTM   Connected Lighting Bluetooth Modul

### A.2. Device typen

```
<modbus_devices>
  <device name="broadcast" type="" address="0"/>
  <device name="slyder/easytouch/panel standard" type="1" address="1"/>
  <device name="aloneatwork 1.0" type="" address="2"/>
  <device name="knx" type="" address="3"/>
  <device name="mylights finder" type="" address="4"/>
  <device name="mylights adapt" type="" address="5"/>
  <device name="sensormodul standard" type="4" address="6"/>
  <device name="FREE" type="" address="7"/>
  <device name="controlunit" type="2" address="8"/>
  <device name="controlunit_m4" type="6" address="8"/>
  <device name="controlunit TESTING TCI" type="2" address="9"/>
  <device name="FREE" type="" address="10"/>
  <device name="FREE" type="" address="11"/>
  <device name="FREE" type="" address="12"/>
  <device name="FREE" type="" address="13"/>
  <device name="FREE" type="" address="14"/>
  <device name="pc-bridge" type="5" address="15"/>
  <device name="aloneatwork 2.0" type="3" address="16"/>
  <device name="panel left" type="1" address="17"/>
  <device name="panel right" type="1" address="18"/>
  <device name="FREE" type="" address="19"/>
  <device name="FREE" type="" address="20"/>
  <device name="clm" type="10" address="21"/>
  <device name="smartmodul" type="11" address="22"/>
  <device name="sensormodul head 1" type="4" address="23"/>
  <device name="sensormodul head 2" type="4" address="24"/>
  <device name="sensormodul head 3" type="4" address="25"/>
  <device name="sensormodul head 4" type="4" address="26"/>
  <device name="FREE" type="" address="27"/>
  <device name="FREE" type="" address="28"/>
  <device name="FREE" type="" address="29"/>
  <device name="FREE" type="" address="30"/>
  <device name="FREE" type="" address="31"/>
  <device name="FREE" type="" address="32"/>
  <device name="FREE" type="" address="33"/>
  <device name="FREE" type="" address="34"/>
  <device name="FREE" type="" address="35"/>
  <device name="FREE" type="" address="36"/>
  <device name="RANDOM ADDRESS SPACE" type="" address="37-247"/>
  <device name="PANEL/SENSOR TESTSYSTEM SPACE" type="20" address="248-251"/>
  <device name="TESTSYSTEM SECOND MODBUS ADDRESS" type="" address="252"/>
  <device name="FREE" type="" address="253"/>
  <device name="FREE" type="" address="254"/>
  <device name="FREE" type="" address="255"/>
</modbus_devices>
```

### A.3. Referenzen

- /1/ Bedienung Lightpad Slim  
V1.1  
[pflichtenheft bedienung lightpad slim v1.1.docx](#)
- /2/ Pflichtenheft Hard-und Software Bedienelement  
V1.3  
[bedienelement hard- software pflichtenheft v1.3.docx](#)
- /3/ Pflichtenheft Hardware Software  
V1.5  
[Pflichtenheft hardware software Sensormodul Rev 1.5.docx](#)
- /4/ Pflichtenheft ALONEatWORK  
v1.8  
[pflichtenheft ALONEatWORK v1.8.docx](#)
- /5/ Spezifikation LED Blinkmuster  
V9  
[Spezifikation LED Blinkmuster V9.docx](#)
- /6/ Steuergerät 2.0 Design Spezifikation  
V1.4  
[steuergerät 2.0 design spezifikation V1.4.docx](#)
- /7/ Modbus Adressen Übersicht  
[Modbus Addresses.pptx](#)