

ALLGEMEINES RS-485 PROTOKOLL

V 1.0

1. SERIAL COMMUNICATIONS BASICS

Communication settings

1.1 Communication Protocol

Communication protocol summary

Like every communication method the serial communication uses a particular protocol (ANSI) which must be respected in order to allow communication to take place, following table gives a summary of the predefined communication terms.

Sync byte \0xff
Start byte \xfe
Module address
Command byte(s)
Data bytes1
Stop byte \xff

Start byte

The "Start byte" informs the SDK-U4-10-AD (in case of transmission) or the Master (in case of reception) that a new data transfer will take place

Module address

The maximum number of Modules that can be addressed by one RS-485-Port is 127.

Adress 0 (zero) is for transmit a command to all modules !

Es können 127 Dimmer an einem RS-485 BUS ohne Repeater angeschlossen werden. Es wird auf dem Dimmerkoppler ein RS-485 Treiber mit ¼ Load eingesetzt. (MAX487)

Auf Adr.0 reagieren alle Dimmer, es wird aber kein Acknowledge zurückgesendet !

Bei Adr. 0 können keine Parameter übertragen und kein Status abfragt werden. Dies geht ausschließlich bei der am DIP-Switch eingestellten Adresse.

Command byte(s)

There is at least one command byte to define the action to be performed. Commands that are not often used or complex commands can take more than one byte. All command bytes that are sent by the Master to get information out of the Module are repeated in the answer-data-transfer of the module.

Data bytes (optional)

Whether the command bytes are followed by one or more data bytes depends on the contents of the command bytes.

Stop byte

The "Stop byte" informs the SDK-U4-10-AD (in case of transmission) or the Master (in case of reception) that the data transfer is complete and that the interpretation of the command and data bytes can start.

1.2 Communication settings

The baud rate can set by DIP-Switch 3.

Baud rate

Defines the speed of the data transfer. The baud rate can be set using the send_command in the Software.

Data Bits

Eight data bits are used for each character of the data transfer.

Parity

There is no parity bit used to perform error checking.

Stop Bits

One stop bit is used to define the end of a character.

Übertragungsraten allg. RS-485 Protokoll

DIP-Switch 3=OFF 38400 Baud, 8 Databit, 1 Startbit, No Parity, 1 Stopbit

DIP-Switch 3=ON 9600 Baud, 8 Databit, 1 Startbit, No Parity, 1 Stopbit

Der Adressraum des Dimmers beträgt \x00.....\0x7f

DIP-Schalter = OFF Adressbit=0

DIP-Schalter = ON Adressbit=1

Der Dimmerkoppler beginnt frühestens sofort nach Empfang des Stop-Byts spätestens 250µs nach Empfang des Stop-Byts mit dem Senden des Receive (acknowledge) . (ausgenommen Parameter setzen)

Setze Level 8 Bit:

Command:

Command[0]	\x57
------------	------

Data:

- Data[0] = Levelvalue Dimmer1
- Data[1] = Dimmzeit in Sekunden Dimmer 1
- Data[2] = Levelvalue Dimmer2
- Data[3] = Dimmzeit in Sekunden Dimmer 2
- Data[4] = Levelvalue Dimmer3
- Data[5] = Dimmzeit in Sekunden Dimmer 3
- Data[6] = Levelvalue Dimmer4
- Data[7] = Dimmzeit in Sekunden Dimmer 4
- Data[8] = Pruefsumme Moduladresse+Command+Data[0]+Data[1]+ Data[2]+ Data[3]+ Data[4]+ Data[5]+ Data[6]+ Data[7]

Anmerkung: Die Prüfsumme wird durch 8 Bit Addition gebildet. Nach der Addition wird das Bit 7 der Prüfsumme auf 0 gesetzt. Bei der virtuellen Gruppenadresse wird nur der Wert des korrespondierenden Ausganges gesetzt, dies bedeutet, dass die Werte für die Level-Bytes Data[0], Data[2], Data[4], Data[6] gleich sein müssen. Bei der Adresse 0 oder der physikalischen Adresse werden die Level-Bytes Data[0], Data[2], Data[4], Data[6] berücksichtigt. Für das setzen von Szenen muss die physikalische bzw. virtuelle Adresse mit den entsprechenden Werten für die Level-Bytes gesendet werden. Soll ein entsprechender Ausgangswert nicht verändert werden, wird Wert 1 übergeben.

Data	Byte Value
Data[0]..... Data[7]	\x00-\xFD

Example:

Dimmermodul hat Adresse 12,
setze Dimmer 1+3 auf Wert 180 Zeit 5 Sec , Dimmer 2+4 auf Wert 128 Zeit 10s

Transmit	
Sync	\xFF
Start	\xFE
Moduladresse	\x0C
Command[0]	\x57
Data[0]	\xB4
Data[1]	\x5
Data[2]	\x80
Data[3]	\x0A
Data[4]	\xB4
Data[5]	\x05
Data[6]	\x80
Data[7]	\x0A
Data[8]	\xxx
Ende	\xFF

Receive (acknowledge)	
Start	\xFE
Moduladresse	\x0C
Command[0]	\x06
Ende	\xFF

Die Dimmzeit wird standardmäßig in 0,5s Steps ausgegeben.
Der Wertebereich beträgt 0...240.
Die Zeitkonstanten größer 240 sind für Spezialfälle reserviert.

- Zeitkonstante: #240 Der übertragene Analogwert wird mit einer Geschwindigkeit eingestellt, welche für eine Regelung von 0 auf 100% 120 s benötigen würde.
- Zeitkonstante: #241 SONDERFUNKTION (LINEARISIERUNG)
Für die Konstante #241 kann auch der String "SOFT" verwendet werden.
Das Analogmodul stellt den Analogwert sofort ein, führt aber eine Linearisierung der Analogwerte durch, um eine durch längere Zykluszeiten des Programms bedingte, sprunghafte Änderung der Analogwerte zu kompensieren.
- Zeitkonstante: #242 SONDERFUNKTION (STOP)
Mit dieser Funktion können Regelvorgänge beim Anfahren von Endwerten mit einer Zeitkonstanten durch das Analogmodul unterbrochen werden. Der Ausgang bleibt dann auf dem zu diesem Zeitpunkt eingestellten Wert stehen. Diese Funktion ist für einfache Dimmung mit 2 Tasten (Heller/Dunkler) sehr effektiv einsetzbar.
- Zeitkonstante: #243 SONDERFUNKTION (BLINKEN)
Bevor der übertragene Analogwert eingestellt wird, wird der Ausgang sechs mal zwischen 20 und 80% in einer Zeit von 4,5s getoggelt. Diese Funktion kann zur Erhöhung der Übersichtlichkeit beim Programmieren von Stimmungen genutzt werden, indem der ausgewählte Kanal durch das Blinken hervorgehoben wird.

Zeitkonstante 244...251

Hiermit kann die ausgegebene Dimmkurve für verschiedene Leuchtmittel online ohne Parametrierung geändert werden wenn im Parameterfile bei Parameter Time Mode Kx eine 4 eingetragen ist.
Achtung!!! Die Interpretation der Dimmzeit ist in diesem Mode fest. 0.5s / Digit.

- #244 -> linear
- #245 -> log
- #246 -> quadratisch
- #247 -> schalten
- #248 -> Userk. 1
- #249 -> Userk. 2
- #250 -> Userk. 3
- #251 -> Userk. 4

Dimmen:

Command:

Command[0]	\x64
------------	------

Data:

Data[0] = Dimmerausgang

Data[1] = Dimmart: 0= Stop, 1= Up, 2= Down

Data[2] = Dimmgeschwindigkeit in Sekunden (Wert von 0 auf 100%)

Data[3] = Pruefsumme Moduladresse+Command+Data[0]+Data[1]+Data[2]

Anmerkung: Die Prüfsumme wird durch 8 Bit Addition gebildet. Nach der Addition wird das Bit 7 der Prüfsumme auf 0 gesetzt.

Bei der virtuellen Gruppenadresse wird Data[0] nicht berücksichtigt, respektive auf 0 gesetzt.

Bei der Adresse 0 oder der physikalischen Adresse wird Data[0] berücksichtigt.

Data	Byte Value
Data[0]	\x00-\x0F
Data[1]	\x00-\x02
Data[2]	\x00-\xf0

Example:

Dimmermodul hat Adresse 4, Dimm-Up alle Dimmerausgänge in 7 Sekunden

Transmit	
Sync	\xFF
Start	\xFE
Moduladresse	\x04
Command[0]	\x64
Data[0]	\x0F
Data[1]	\x01
Data[2]	\x07
Data[3]	\xx
Ende	\xFF

Receive (acknowledge)	
Start	\xFE
Moduladresse	\x0C
Command[0]	\x06
Ende	\xFF

Bei Erreichen des Endwertes [0%]; [100%] wird das Dimmen automatisch gestoppt und der Endwert auf den Dimmer ausgegeben. Das Receive (acknowledge) wird unmittelbar nach Empfang des Kommandos zurückgesendet.

Havarielevel setzen:

Command:

Command[0]	\x68
------------	------

Data:

Data[0] = Levelvalue Dimmer 1 bei Busunterbruch	default 0
Data[1] = Levelvalue Dimmer 1 bei Eingang Notbetrieb	default 3
Data[2] = Levelvalue Dimmer 2 bei Busunterbruch	default 0
Data[3] = Levelvalue Dimmer 2 bei Eingang Notbetrieb	default 3
Data[4] = Levelvalue Dimmer 3 bei Busunterbruch	default 0
Data[5] = Levelvalue Dimmer 3 bei Eingang Notbetrieb	default 3
Data[6] = Levelvalue Dimmer 4 bei Busunterbruch	default 0
Data[7] = Levelvalue Dimmer 4 bei Eingang Notbetrieb	default 3
Data[8] = Timeout bei Busunterbruch in Sekunden	default 50
Data[9] = Pruefsumme Moduladresse+Command+Data[0]+Data[1]+Data[2]+Data[3]+Data[4]+Data[5]+Data[6]+Data[7]+Data[8]	

Anmerkung: Die Prüfsumme wird durch 8 Bit Addition gebildet. Nach der Addition wird das Bit 7 der Prüfsumme auf 0 gesetzt.
Data[0]..Data[7] : 0 (inaktiv); 1 (0 %); 2 (20 %); 3 (50 %) ; 4 (80 %) ; 5 (100 %)

Data	Byte Value
Data[0]	\x00-\x05
Data[1]	\x00-\x05
Data[2]	\x00-\x05
Data[3]	\x00-\x05
Data[4]	\x00-\x05
Data[5]	\x00-\x05
Data[6]	\x00-\x05
Data[7]	\x00-\x05
Data[8]	\x00-\xFD

Example:

Dimmermodul hat Adresse 23, Havariewert bei Busunterbruch Ausgang 1-4 keine Änderung, bei Notbetrieb Ausgang 1-3 Wert 50%, Ausgang 4 keine Änderung. Reaktionszeit bei Busunterbruch 60 Sekunden. Bei Notbetrieb kann keine Verzögerung eingestellt werden, die Reaktionszeit ist sofort.

Transmit	
Sync	\xFF
Start	\Xfe
Moduladresse	\x17
Command[0]	\x68
Data[0]	\x00
Data[1]	\x03
Data[2]	\x00
Data[3]	\x03
Data[4]	\x00
Data[5]	\x03
Data[6]	\x00
Data[7]	\x00
Data[8]	\x3C
Data[9]	\xxx
Ende	\xFF

Receive (acknowledge)	
Start	\xFE
Moduladresse	\x17
Command[0]	\x06
Ende	\xFF

Der mit diesem Kommando ausgegebene Wert wird im Flash gespeichert und bei BUS-Ausfall bzw. Aktivierung des Eingangs Notbetrieb auf dem entsprechenden Dimmkanal ausgegeben.

Hinweis: Dieser Wert darf nicht zyklisch geändert werden da max. 10000 Schreibzugriffe auf das Flash garantiert sind !!!

Das Receive (acknowledge) wird erst nach Speichern der Daten in den Flash gesendet. Dies kann bis zu 50ms dauern. Der Master muß so lange warten und darf kein neues Protokoll auf den BUS senden, da der Dimmer nach Speichern der Daten das Receive (acknowledge) auf den BUS sendet.

Empfehlung: Timeout auf 60ms einstellen beim Absender

Betriebsart setzen:
entspricht Parameterblock zum Dimmer übertragen

Command:

Command[0]	\x42
------------	------

Example:

Dimmermodul hat Adresse 23 Parameterblock 250 Byte

Transmit	
Sync	\xFF
Start	\xFE
Moduladresse	\x17
Command[0]	\x42
Command[1]	Command[0] negiert \xbd
Pruefsumme 8 Bit nach Addition auf 7 Bit kürzen (Bit7=0 setzen)	Start+ Moduladresse+ Command[0]+ Command[1]
Die nachfolgend grün gekennzeichneten Datenbyte werden Nibbleweise übertragen da sonst die Steuerbytes 0xfe und 0xff vorkommen können. Dabei wird zuerst das H-Nibble und danach das L-Nibble übertragen. Die Kennung des H-Nibbles beträgt \0b0101xxxx, die Kennung des L-Nibbles beträgt \0b0011xxxx.	
Beispiel Data[0] = \xfd Es werden folgende zwei Byte gesendet: \x5f \x3d Es werden also 508 Datenbyte gesendet. Zum Schluß wird die Endekennung \xff gesendet.	
Data[0]	Parameterblock Datenbyte 0
....	
Data[249]	Bis Parameterblock Datenbyte 249
Pruefsumme 16Bit High	16 Bit Addition von Datenbyte 0...249
Pruefsumme 16Bit Low	
Pruefsumme 16Bit High negiert	
Pruefsumme 16Bit LOW negiert	
Ende	\xFF

Receive (acknowledge)	
Start	\xFE
Moduladresse	\x17
Command[0]	\x06
Ende	\xFF

Hinweis: Dieser Block darf nicht zyklisch geändert werden da max.10000 Schreibzugriffe auf das Flash garantiert sind !!!

Das Receive (acknowledge) wird erst nach Speichern der Daten in den Flash gesendet. Dies kann bis zu 50ms dauern. Der Master muß so lange warten und darf kein neues Protokoll auf den BUS senden, da der Dimmer nach Speichern der Daten das Receive (acknowledge) auf den BUS sendet.

Empfehlung: Timeout auf 60ms einstellen beim Absender

Statusabfrage

Command:

Command[0]	\x53
------------	------

Data:

Data[0] = Dimmerausgang nur alle pro Modul möglich
 Data[1] = Reserve
 Data[2] = Pruefsumme Moduladresse+Command+Data[0]+Data[1]

Data	Byte Value
Data[0]	\x0F
Data[1]	\x00

Returned Data:

Data[0] = Level Dimmerausgang 1
 Data[1] = Status Dimmerausgang 1 ¹
 Data[2] = Level Dimmerausgang 2
 Data[3] = Status Dimmerausgang 2
 Data[4] = Level Dimmerausgang 3
 Data[5] = Status Dimmerausgang 3
 Data[6] = Level Dimmerausgang 4
 Data[7] = Status Dimmerausgang 4

Data	Byte Value
Data[0] Data[7]	\x00-\xFD

¹ * Bedeutung der Bits im Statuswort:

- Bit 0: Notaus
 - 0 bedeutet kein NOT-AUS und
 - 1 bedeutet NOT-AUS
- Bit 1: Übertemperatur
 - 0 bedeutet keine Übertemperatur und
 - 1 bedeutet Übertemperatur
- Bit 2: Betriebsspannung Dimmer
 - 0 Dimmer online
 - 1 Dimmer offline (keine Antwort vom Dimmer empfangen)
- Bit 3: Überstrom
 - 0 bedeutet kein Überstrom und
 - 1 bedeutet Überstrom detektiert
- Bit 4: Schaltart
 - 0 bedeutet Abschnitt und
 - 1 bedeutet Abschnitt
- Bit 5: Netzfrequenz
 - 0 bedeutet 50Hz Spannungsnetz und
 - 1 bedeutet 60Hz Spannungsnetz
- Bit 6: Fehler falsche Lastart
 - 0 bedeutet Last korrekt erkannt
 - 1 bedeutet Lastart konnte nicht korrekt erkannt werden
- Bit7: immer 0

Example:

Statusabfrage Dimmermodul 16

Transmit	
Sync	\xFF
Start	\xFE
Moduladresse	\x10
Command[0]	\x53
Data[0]	\x0F
Data[1]	\x00
Data[2]	\x72
Ende	\xFF

Received Data	
Start	\xFE
Moduladresse	\x10
Command[0]	\x53
Data[0]	\x00-\xFD
Data[1]	\x00-\xFD
Data[2]	\x00-\xFD
Data[3]	\x00-\xFD
Data[4]	\x00-\xFD
Data[5]	\x00-\xFD
Data[6]	\x00-\xFD
Data[7]	\x00-\xFD
Ende	\xFF