Schnittstellenprotokoll
OCR Reader BR40

Version 1.0

Inhaltsverzeichnis

1 Verwendetes Datenformat 2
1.1 Long/Short 2
1.2 Float 2
2 Steuerzeichen 2
3 Telegrammaufbau für die Datenübertragung 3
3.1 Telegrammkopf (Header) 3
3.2 Datenkopf (Daten-Header) 4
3.3 Nutzdaten 4
3.4 Telegrammende 4
3.4.1 Berechnung der Quersumme BCC (Block Check Character) 5
3.5 Beispiel Protokoll 6
4 Kommunikationsablauf 7
4.1 Acknowledge 7
5 Befehle 8
5.1 Laden von Projekten aus dem Festspeicher 9
5.2 Speichern von Projekten in den Festspeicher 9
5.3 Startprojekt festlegen 9
5.4 Startprojekt auslesen 9
5.5 Matchcode setzten 10
5.6 Matchcode auslesen 10
5.7 Timing setzten 10
5.8 Timing auslesen 11

1 Verwendetes Datenformat

1.1 Long/Short

In diesem Protokoll wird das Datenformat Little Endian verwendet.

Erklärung:

Unter Little-Endian versteht man die Anordnung des Least Significant Byte (LSB) einer Datenstruktur im Speicher. Liegt das LSB als erstes Byte der Struktur im Speicher, so spricht man von einer Little-Endian Darstellung

Beispiele:

16-Bit (Short) Wert: DE45

Adresse	n-1	n	n+1	n+2
Little-Endian	xx	45	DE	xx

32-Bit (Long) Wert: FF01DE45

Adresse	n-1	n	n+1	n+2	n+3	n+4
Little-Endian	xx	45	DE	01	FF	xx

1.2 Float

Die Float-Variablen sind nach dem IEEE Standard for Binary Floating Point Arithmetic (ANSI/IEEE Std 754-1985) aufgebaut

2 Steuerzeichen

Zeichen	**ASCII-Wert (dezimal)**	**ASCII-Wert (hexadezimal)**	**Bedeutung**
$	36	0x24	Startzeichen
.	46	0x2E	Stoppzeichen 1
;	59	0x3B	Stoppzeichen 2

3 Telegrammaufbau für die Datenübertragung

	Länge	Telegrammbereich
Startzeichen	1 Byte	Telegrammkopf (Header) 12 Byte
Frame Typ	1 Byte	
MSG_ID	1 Byte	
Repeat	1 Byte	
ProtocollLen	2 Byte	
MsgType	2 Byte	
Adress	4 Byte	
CMD0: Command Type 0	1 Byte	Datenkopf (Daten-Header) 16 Byte
CMD1: Command Type 1	1 Byte	
Parameter 1	2 Byte	
Parameter 2	2 Byte	
Parameter 3	2 Byte	
Parameter 4	4 Byte	
Datenlänge:	4 Byte	
1. Datenbyte	1 Byte	Nutzdaten
2. Datenbyte	1 Byte	
...		
n. Datenbyte	1 Byte	
Quersumme	2 Byte	Telegrammende 4 Byte
Stoppzeichen_0	1 Byte	
Stoppzeichen_1	1 Byte	

3.1 Telegrammkopf (Header)

- **Startzeichen (1 Byte):**
  Das Startzeichen <$> signalisiert den Beginn eines Protokolls.

- **Frame Typ (1 Byte):**
  Der Frame Typ wird immer auf 0 gestezt.

- **MSG_ID (1 Byte):**
  Die Message Identifikation ist eine fortlaufende Zahl, die beliebig gewählt werden kann. Der Sensor antwortet mit derselben Zahl im Antwort-Protokoll.

- **Repeat (1 Byte):**
  Repeat ist die Kennung für eine Protokoll-Wiederholung.

- **ProtocollLen (2 Byte):**
  ProtocollLen beschreibt die gesamte Länge des gesendeten Telegramms in Byte.

- **MsgType (2 Byte):**
  MsgType beschreibt den Typ der Message (siehe Kapitel 4.1 Acknowledge)

- **Adress (4 Byte):**
  Adress beschreibt die Adresse des Device.

3.2 Datenkopf (Daten-Header)

- **CMD0 (1 Byte) und CMD1 (1Byte):**
  Der CMD0 fasst Befehle hierarchisch in Gruppen zusammen. Alle Datenpakete, die z.B. das Teachen betreffen beginnen mit dem gleichen CMD0 Identifier. Innerhalb dieser Gruppe unterscheidet dann der CMD1 eindeutig die Befehle voneinander. (siehe Kapitel 5 Befehle)

- **Parameter 1...4 (10 Byte):**
  Die (Übertragungs-) Parameter 1...4 können je nach Befehl- oder Daten-Kopf für die jeweils notwendigen Parameter verwendet werden.

- **Datenlänge (4 Byte):**
  Im Parameter Datenlänge wird die Länge der folgenden Nutzdaten (max.900 Byte) eingetragen.

3.3 Nutzdaten

Das Nutzdatenfeld beinhaltet gegebenenfalls für den jeweiligen Befehl erforderlichen Nutzdaten. Dieses Nutzdatenfeld kann Max 900 Byte lang sein. Es gibt auch Befehle die ohne Nutzdatenfeld arbeiten. Die Länge wird im Datenkopf in dem Parameter Datenlänge beschrieben

3.4 Telegrammende

- **Quersumme (2 Byte) :**
  Die Quersumme muss bei jedem Sendetelegramm neu berechnet werden (siehe Kapitel 3.4.1 Berechnung der Quersumme)

- **Stoppzeichen (2 Byte):**
  Die Stoppzeichen signalisieren das Ende des Protokolls.
  Stoppzeichen_0 <.>
  Stoppzeichen_1 <;>

3.4.1 Berechnung der Quersumme BCC (Block Check Character)

Die Quersumme wird berechnet, indem alle Zeichen des Telegrammkopf + Daten-Kopf + Nutzdaten bitweise miteinander XOR-verknüpft werden
Beispiel:
PC → Sensor
Darstellung als Stream:
24 00 01 00 20 00 00 00 00 00 00 00 0A 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 **0F 00** 2E 3B

Byte		Datum	Protokoll-Bereich
1		0x24	Protokoll-Header
2	XOR	0x00	
3	XOR	0x01	
4	XOR	0x00	
5	XOR	0x20	
6	XOR	0x00	
7	XOR	0x00	
8	XOR	0x20	
9	XOR	0x00	
10	XOR	0x00	
11	XOR	0x00	
12	XOR	0x00	
13	XOR	0x0A	Daten-Header
14	XOR	0x00	
15	XOR	0x00	
16	XOR	0x00	
17	XOR	0x00	
18	XOR	0x00	
19	XOR	0x00	
20	XOR	0x00	
21	XOR	0x00	
22	XOR	0x00	
23	XOR	0x00	
24	XOR	0x00	
25	XOR	0x00	
26	XOR	0x00	
27	XOR	0x00	
28	XOR	0x00	
29	=	0x0F	Telegrammende
30	Quers.	0x00	
31	Stop 1	0x2E	
32	Stop 2	0x3B	

3.5 Beispiel Protokoll

In diesem Kapitel wird ein Beispiel für ein Schnittstellen-Protokoll exemplarisch dargestellt und näher beschrieben. Dieses Beispiel ist grundlegend für die weiteren Kapitel, d.h. in den nächsten Kapitel werden nur die Prokoll-Bausteine beschrieben, die sich verändern und relevant für die jeweilige Funktion sind.

Beispiel: Abstandswert auslesen aus Kapitel 5.4.1 Prozessdaten anfordern

Zeichenfolge PC => Sensor:

	Telegrammkopf (Header)						
	Start-zeichen	Frame Typ	MSG _ID	Repeat	Protocoll Len	Msg type	Adress
Wert (Hexa-dezimale Darstellung)	24	00	01	00	20 00	00 00	00 00 00 00

Daten-Kopf (Daten-Header)							Nutz-daten	Telegrammende		
CMD 0	CMD1	Param 1	Param 2	Param 3	Param 4	Daten -länge	Daten	Quer-summe	Stopp-zeichen_0	Stopp-zeichen_1
0A	00	00 00	00 00	00 00	00 00 00 00	00 00 00 00		0F 00	2E	3B

Sensor-Antwort:

	Telegrammkopf (Header)						
	Start-zeichen	Frame Typ	MSG _ID	Repeat	Protocoll Len	Msg type	Adress
Wert (Hexadezimale Darstellung)	24	00	01	00	40 00	01 00	00 00 00 00

Daten-Kopf (Daten-Header)						
CMD0	CMD1	Param 1	Param 2	Param3	Param4	Datenlänge
0A	00	00 00	00 00	00 00	00 00 00 00	20 00 00 00

Nutzdaten	Telegrammende		
Daten	Quersumme	Stopp-zeichen_0	Stopp-zeichen_1
92 05 00 00 10 27 00 00 **F6 05 00 00** E 02 00 00 E 02 00 00 E 02 00 00 00 00 00 00 00 00 00	11 00	2E	3B

Die absolute Position des aktuellen Abstandswert im Stream ist 36 und hat eine Länge von 4 Byte.
Der Abstandswert ist im Stream unterstrichen dargestellt und entspricht aufgrund des verwendeten Datenformats Little Endian der hexadezimalen Zahl 00 00 05 F6. Dies entspricht dem Abstandswert von 1526 mm.

4 Kommunikationsablauf

4.1 Acknowledge

Jeder Befehl, der vom PC an den Sensor geschickt wird, wird durch ein Acknowledge Paket bestätigt

Bei jedem Datenpaket, das der PC vom Sensor empfängt ist das ACK-Flag des Msg_Type- Parameters gesetzt.

Header-Beispiel:

Anforderung (PC → Sensor)

Telegrammkopf (Header):

Start	Frame Typ	MSG_ID	Repeat	ProtocollLEN	MsgType	Gap	Adress
$	0	0x10	0	0x20(+Datenlänge)	0	0	0

Bleibt gleich

Ändert sich

Antwort (Sensor → PC)

Telegrammkopf (Header):

Start	Frame Typ	MSG_ID	Repeat	ProtocollLEN	MsgType	Gap	Adress
$	0	0x10	0	0x20(+Datenlänge)	0x01	0	0

5 Befehle

Im folgenden ist eine Übersicht über die Befehls-Gruppen (CMD0). Die Unterscheidung dieser Befehls-Gruppen geschieht durch CMD1 und ist in den jeweiligen Kapiteln beschrieben.

CMD0	CMD1	Param1	Param2	Param3	Param4	Daten-länge
0x01 CMD_SETTINGS	0x05 GET_SETTINGS_FROM_FLASH	Speicherplatz				0
	0x04 SET_SETTINGS_TO_FLASH	Speicherplatz				0
	0x0A SET_START_SETTINGS	Speicherplatz				0
	0x20 GET_START_SETTINGS	Speicherplatz				0
	0x21 SET_MATCHCODE	ObjectNr	MatchcodeNr			Dynamisch
	0x22 GET_MATCHCODE	ObjectNr	MatchcodeNr			Dynamisch
	0x23 SET_TRIGGER					
	0x24 GET_TRIGGER					0
	0x25 SET_TIMING	T_TriggerDelay	T_Pulse	T_Switch		0
	0x26 GET_TIMING	T_TriggerDelay	T_Pulse	T_Switch		0

5.1 Laden von Projekten aus dem Festspeicher

CMD0	CMD1	Param1	Param2	Param3	Param4	DataLen
0x01	0x05	Speicherplatz				0

Mit diesem Befehl können Einstellungen aus dem Flash ins Ram des Readers geladen werden. In Param1 wird der Speicherplatz (0-19) übergeben. Die an die Einstellungen gebundenen Fonts werden mitgeladen.

5.2 Speichern von Projekten in den Festspeicher

CMD0	CMD1	Param1	Param2	Param3	Param4	DataLen
0x01	0x04	Speicherplatz				0

Mit diesem Befehl können Einstellungen aus dem Ram ins Flash des Readers gespeichert werden. In Param1 wird der Speicherplatz (0-19) übergeben. Die an die Einstellungen gebundenen Fonts werden mitgespeichert.

5.3 Startprojekt festlegen

CMD0	CMD1	Param1	Param2	Param3	Param4	DataLen
0x01	0x0A	Speicherplatz				0

Mit diesem Befehl wird festgelegt, welches Projekt beim nächsten Bootvorgang des Readers geladen wird. In Param1 wird der Speicherplatz (0-9) übergeben.

5.4 Startprojekt auslesen

CMD0	CMD1	Param1	Param2	Param3	Param4	DataLen
0x01	0x20	Speicherplatz				0

Mit diesem Befehl kann ausgelesen werden, welches Projekt beim nächsten Bootvorgang des Readers geladen wird. In Param1 wird der Speicherplatz (0-9) übergeben.

5.5 Matchcode setzten

CMD0	CMD1	Param1	Param2	Param3	Param4	DataLen
0x01	0x21	ObjectNr	MatchcodeNr			Dynamisch

Mit diesem Befehl kann ein Matchcode-Eintrag geändert werden. In Param1 wird die Objekt-Nummer (0-3) übergeben, in Param2 wird die Matchcode-Nummer (0-15) übergeben.

Das Muster selbst wird als Zeichenkette im Datenbereich übergeben. In das Feld DataLen ist die Länge (0-63) der Zeichenkette einzutragen.

!!! Dieser Befehl wirkt sich nur auf die Einstellungen im RAM des Readers aus. Sollen die neuen Einstellungen dauerhaft gespeichert werden, muss der Befeh SET_SETTINGS_TO_FLASH ausgeführt werden.

5.6 Matchcode auslesen

CMD0	CMD1	Param1	Param2	Param3	Param4	DataLen
0x01	0x22	ObjectNr	MatchcodeNr			Dynamisch

Mit diesem Befehl kann ein Matchcode-Eintrag ausgelesen werden. In Param1 wird die Objekt-Nummer (0-3) übergeben, in Param2 wird die Matchcode-Nummer (0-15) übergeben.

Das Muster selbst wird als Zeichenkette im Datenbereich übergeben. Im Feld DataLen übergibt der Reader die Länge des Musters.

5.7 Timing setzten

CMD0	CMD1	Param1	Param2	Param3	Param4	DataLen
0x01	0x25	T_TriggerDelay	T_Pulse	T_Switch		0

Mit diesem Befehl kann das Timing eingestellt werden. In den Param1 bis Param3 werden die Zeiten übertragen. Die Zeiten werden in Mirkosekunden übertragen.

!!! Dieser Befehl wirkt sich nur auf die Einstellungen im RAM des Readers aus. Sollen die neuen Einstellungen dauerhaft gespeichert werden, muss der Befeh SET_SETTINGS_TO_FLASH ausgeführt werden.

5.8 Timing auslesen

CMD0	CMD1	Param1	Param2	Param3	Param4	DataLen
0x01	0x25	T_TriggerDelay	T_Pulse	T_Switch		0

Mit diesem Befehl kann das Timing ausgelesen werden. In den Param1 bis Param3 werden die Zeiten übertragen. Die Zeiten werden in Mirkosekunden übertragen.

Technische Änderungen vorbehalten
Stand 05.09.2008