Bedienungsanleitung

DD 55 IS

Displaydecoder mit InterBus-S

Inhalt

1 Einleitung

Der Decoder DD55IS ist ein InterBus-S-Teilnehmer mit implementierten Kommunikationsdiensten (PCP 1.5). Die Kommunikation mit der Anschaltbaugruppe erfolgt also vorwiegend über PCP-Dienste (Peripheral Communication Protocol), die ein Subset der MMS-Dienste darstellen.

Alle Daten vom Decoder zur Steuerung und umgekehrt, die beim DD55 über die serielle Schnittstelle übertragen werden, werden beim DD55IS über das Bussystem übertragen.

µ §

Bild 1: Decoder ohne/mit InterBus-S

Außerdem steht jeweils ein Prozeßdatenwort für Eingänge und ein Prozeßdatenwort für Ausgänge zur Verfügung. Damit wird z.B. dem InterBus-S-Master angezeigt, ob der Decoder Daten abzuliefern hat.

Die Identifikationsnummer des Decoders lautet F3 (Hex) bzw. 243 (dez). Im InterBus-S-Zyklus werden 3 Datenworte benötigt. Ein Datenwort für Eingänge, ein Datenwort für Ausgänge und ein Datenwort für den Kommunikationskanal (PCP).

2 Anschlußbelegung

µ **§**

Bild 2: Anschlußbelegung der InterBus-S-Schnittstelle

3 Prozeßdatenkanal

Im InterBus-S-Zyklus stehen zwei 16 Bit lange Datenworte zur Verfügung.

4 Eingangsdatenwort

Im Eingangsdatenwort (Process I/O: In Data) stehen Statusinformationen des Decoders (InterBus-S-Informationen) und die Signale der Schaltausgänge.

1 Wort Eingang

Bits 0..7 Statusinformationen (Steuerkanal)
Bits 8, 9 Schaltausgänge des Decoders
Bits 10..15 Reserviert

Die einzelnen Bits sind wie folgt belegt

µ §*µ*

§Bild 3: Statusinformationen: Eingangsdatenwort

5 Statusbits

Die Bits 0..7 des Eingangswortes sind folgendermaßen belegt:

Bit0 == 0 ==> Letzter Parameter OK, kein Paritätsfehler
Bit0 == 1 ==> Letzter Parameter fehlerhaft, Paritätsfehler

Bit1 == 0 ==> Empfangspuffer V24 Leer
Bit1 == 1 ==> Empfangspuffer V24 enthält Daten

Bit2 == 0 ==> Im Empfangspuffer sind mindestens 64 Bytes frei
Bit2 == 1 ==> Im Empfangspuffer sind weniger als 64 Bytes frei (Empfangspuffer voll)

Bit3 == 0 ==> Sendepuffer voll
Bit3 == 1 ==> Im Sendepuffer ist noch garantiert Platz für ein Telegramm (> 16 Bytes sind frei)

Bit4..7 ==> Reserviert

6 Ausgangsdatenwort

Im Ausgangsdatenwort wird dem Decoder die Sensoraktivierung übergeben (Rest: reserviert).

µ §

Bild 4: Ausgangsdatenwort

7 Kommunikation mit PCP

Außer über binäre Ein- und Ausgänge (jeweils ein Datenwort), kann mit dem InterBus-S-Decoder auch über Kommunikationsdienste (ISO/OSI, Schicht 1,2 und 7) in Verbindung getreten werden (1 Datenwort PCP). Von den insgesamt 39 FMS Diensten wie sie vom Profibus her bekannt sind, wurden beim InterBus-S-Decoder insgesamt 10 Dienste implementiert (PCP: Periperals Communication Protocol).

Von diesen 10 Diensten sind im InterBus-S-Decoder 8 Dienste verfügbar.

8 Implementierte Dienste

Initiate	Kommunikationsverbindung aufbauen
Abort	Kommunikationsverbindung beenden
Read	Lesen eines Wertes
Write	Schreiben eines Wertes
Status	Lesen von Geräte- und Anwenderstatus des Decoders
Identify	Lesen von Hersteller, Typ und Version des Moduls im Decoder
Get_OV	Objektverzeichnis des IBS-Moduls einlesen
Reject	Zurückweisen einer Dienstanforderung

Der Decoder unterstützt das Schreiben und Lesen von einfachen Variablen. Der funktionsfähig KBL-Eintrag lautet:

```
.kbl_entry
com_ref   = 2;
*---- LLI - part ----------------------------------------
rem_addr  = 1;
conn_type = 0;
```

```
max_scc   = 1;
max_rcc   = 1;
max_sac   = 1;
max_rac   = 1;
max_aci   = 0;
conn_attr = D;
*---- PMS - part ----------------------------------------
req_len_h      = 0;
req_len_l      = 33;              * send buffer size
ind_len_h      = 0;
ind_len_l      = F9;              * receive buffer size
serv_sup[]     = 80 30 00   80 b0 80; * request/indication
out_serv_client = 1;
out_serv_server = 1;
com_typ        = 1;
*-- USER - part (used by ALI ) -------------------------
symbol[]       = ;
```

9 Beschreibung der implementierten Dienste

Es werden vier Variablen zur Steuerung des Decoders verwendet:

Name	Index	Typ	Zugriffsrechte
SER_MODE	0x4000	UNSIGNED16	ACC_WRITE_ALL | ACC_READ_ALL
SER_BAUD	0x4001	UNSIGNED16	ACC_WRITE_ALL | ACC_READ_ALL
SIO_SEND	0x4002	OCTET_STRING [17]	ACC_WRITE_ALL
SIO_RECV	0x4003	OCTET_STRING [17]	ACC_READ_ALL

Read (Lesen eines Wertes)

Index: 4000 (Schnittstelle: Parameter)
Index: 4001 (Schnittstelle: Baudrate)

Index: 4003 (serieller Port)

Beim seriellen Port werden jeweils 17 Byte blockweise übertragen. Dabei bestimmt das erste Byte die Länge der Nutzdaten in diesem Block.

Write (Schreiben eines Wertes)

(Index: 4000 (Schnittstelle: Parameter))
(Index: 4001 (Schnittstelle: Baudrate))
Diese beiden Variablen sind beim DD55IS nur lesbar.

Index: 4002 (serieller Port)

Beim seriellen Port müssen jeweils 17 Byte blockweise übertragen werden. Dabei bestimmt das erste Byte die Länge der Nutzdaten in diesem Block.

Identify (Lesen von Hersteller, Typ und Version des IBS-Moduls)

ISK AUTOMATION
SERIA ID-Modul
Revision 1.5

Get_OV (Objektverzeichnis einlesen)

First_index_S_OV 4000

4000	simple Variable	unsigned 16	02 length
4001	simple Variable	unsigned 16	02 length
4002	simple Variable	octet string	11 length
4003	simple Variable	octet string	11 length

10 Verwendete Variablen

11 SER_MODE

Diese Variable stellt die Betriebsart des seriellen Kanals ein und ist nicht veränderbar

Einstellung der Schnittstelle: ASCII Protokoll, 8 Bit, 1 Stoppbit, NONE Parity

12 SER_BAUD

Diese Variable stellt die Baudrate ein und ist ebenfalls fest eingestellt.

Einstellung der Schnittstelle: 9600 Baud

13 SIO_SEND

Sendepuffer, zum Schreiben von Daten auf den InterBus-S

sio_send [0] ==> Länge Nettodaten (0..16)
sio_send [1 .. 16] ==> Nettodaten, bzw. Füllbytes

Beim Übertragen von Daten zum Decoder mittels des write-Dienstes muß die Länge der Nettodaten vom Master angegeben werden.

14 SIO_RECV

Empfangspuffer, zum Lesen von Daten vom Interbus-S

sio_recv [0] ==> Länge Nettodaten (0..16)
sio_recv [1..16] ==> Nettodaten, bzw. Füllbytes

Der Benutzer hat das Längenbyte auszuwerten, da die Füllbytes keine Nutzdaten erhalten!

15 Nicht Implementierte Dienste

Start	Programm starten
Stop	Programm stoppen

16 Beispiel: PC-Anschaltbaugruppe

17 PCPM (Monitorprogramm)

18 Initialisierung der PCP-Kommunikation: initiate request

Die Initialisierung der PCP-Kommunikation wird mit dem Dienst initiate request gestartet.

1.) Communication
2.) PMS services
3.) Context management
4.) Initiate
5.) Request
    Initiate request
    "Communication Reference: 02"
    ok

--> Initiate confirmation

Nach der korrekt erfolgten Initialisierung der Karte können Daten des Empfangspuffers mittels Dienst read request ausgelesen oder Daten mittels write request in den Sendepuffer geschrieben werden.

19 Lesen des Empfangspuffers: read request

Nachdem ein Label gelesen wurde, steht im Eingangswort die Zahl $0f0a. "0a" ist das Zeichen dafür, daß der Empfangspuffer gefüllt ist (siehe Kap 3.1) und ausgelesen werden kann. Das Auslesen des Empfangspuffers erfolgt durch den Kommunikationsdienst read request.

1.) Communication
2.) PMS services
3.) Variable access
4.) Read
5.) Request
    Read Request
    "Comunication Reference: 02
    "Index : 4003"
    ok
--> Read confirmation
    "Result (+)"
    "Data
     10 02 6d 69 74 20 4c 65 53 49
     53 2d 49 0d 0a 02 61"

Bemerkung: Alle Zahlen sind hexadezimal dargestellt $10 --> 16^{dez}.

Insgesamt werden pro Zyklus 17 Datenworte übertragen. Dabei bestimmt das erste Datenwort, wieviele nachfolgende Datenwörter relevant sind: hier $10 --> 16 Datenwörter.

Steht im Eingangswort immer noch die Zahl $0f0a, so ist das ein Zeichen dafür, daß der Empfangspuffer immer noch Daten enthält. Um diese auch auszulesen, muß obige Prozedur wiederholt werden. Als Ergebnis bekommt man z.B.

"Data
0e 6d 20 49 4e 54 45 52 42 55
53 2d 53 0d 0a 02 61"

mit $0e --> 14 dez Nutzdaten

Nun ist der Empfangspuffer leer, und im Eingangswort steht wieder $0f08.

Adresse des Empfangspuffers: $4003

20 Beschreiben des Sendepuffers: write request

Sollen Daten zum Gerät geschickt werden, so erfolgt die Kommunikation über den PCP-Dienst write request. Die über InterBus-S übertragenen Daten werden sofort über die RS232-Schnittstelle weitergereicht.

1.) Communication
2.) PMS services
3.) Variable access
4.) Write
5.) Request

Write Request
"Comunication Reference: 02
"Index : 4002"
"Data
04 02 61 0d 0a 00 00 00 00 00
00 00 00 00 00 00 00"

Bemerkung: Alle Zahlen sind hexadezimal dargestellt. Es müssen immer 17 Daten übertragen werden. Dabei bestimmt das erste Wort die Anzahl der relevanten Daten (hier : 4). Wurde die Übertragung erfolgreich abgeschlossen, so wird dies mit

Write confirmation
"Result (+)"

gemeldet.

Die Daten wurden dann über die RS232-Schnittstelle weitergereicht.

Adresse des Sendepuffers: $4002

21 Anwenderprogramm in "C"

Um eine automatisch ablaufende Kommunikation des Decoders mit der PC-Anschaltbaugruppe zu erhalten, kann das Anwenderprogramm selbst geschrieben werden (z.B. in der Programmiersprache C).

22 Empfangen von Decoderdaten

Im laufenden Zyklus wird das Statusbit "Empfangspuffer enthält Daten" vom Programm ausgewertet, und ggf. der Dienst read request abgeschickt.

Eingangsdatenwort =	$0f08:	keine Daten im Empfangspuffer
	$0f0a:	Empfangspuffer enthält Daten

Dieser Vorgang wird solange wiederholt bis der Empfangspuffer im Decoder vollständig geleert ist (Eingangsdatenwort = $0f08).

µ §

Bild 5: Struktogramm: Daten vom Decoder holen

23 Senden von Daten

Sollen Daten zum Decoder gesendet werden (z.B. Parametersatz oder Steuerbefehle), so werden diese Daten in Blöcke von jeweils max. 16 Byte verpackt, das Längenbyte entsprechend berechnet, und diesen Datenblock mittels write request-Dienst abgeschickt. Nach der Bestätigung des Decoders (write confirmation) kann dann der nächste Block gesendet werden.

µ §

Bild 6: Struktogramm: Daten zum Decoder senden

Werden viele Datenblöcke nacheinander an den Decoder gesendet, so ist es sinnvoll auch das Statusbit "Sendepuffer voll" abzufragen, um ein Überlaufen des Puffers zu vermeiden.

Literatur zur Programmierung der PC-Anschaltbaugruppe:

[1] Phoenix Contact, InterBus-S Anwenderhandbuch für InterBus-S, PC-Interfacekarte IBS PC AT UM Best.-Nr.: 2754477

24 Beispiel: SPS-Anschaltbaugruppe

Es muß eine Anschaltbaugruppe verwendet werden, die für die Datenübertragung mittels PCP-Kommunikation ausgelegt ist.

Ablauf einer Kommunikation mit dem Decoder siehe Kap 5.2

Literatur zur Programmierung von SPS-Anschaltbaugruppen: siehe Anwenderhandbücher