

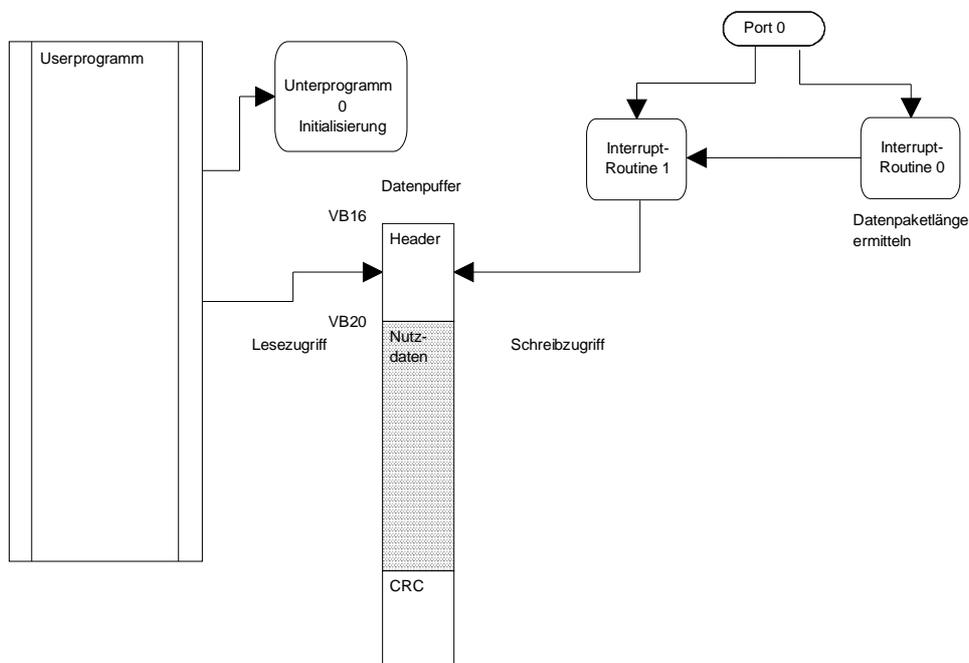


Softwaremodul zur Datenübernahme der Strahlenden Lichtgitter in eine SPS vom Typ Simatic S7/200

Das als Interrupt-Modul realisierte Treiberprogramm übernimmt die Meßdaten als seriellen Datenstrom von SCAN über Port 0 und legt sie in einem Datenpuffer ab. Die einzelnen Lichtachsen werden dem Anwender bitweise ab Speicherplatz VB20 zur Verfügung gestellt (1=Licht, 0=kein Licht), fortlaufend von der ersten Lichtachse (am Anschluß des SCAN) zur letzten Lichtachse (am freien Ende des SCAN).

Bei erfolgreichem Empfang eines Datenpaketes wird vom Treiberprogramm der Merker „M_Userbuffer_ready“ gesetzt. Dieser kann vom Anwenderprogramm nach dem Einlesen der Meßwerte gelöscht und somit als Empfangskontrolle verwendet werden.

Dem Anwenderprogramm steht der gesamte Speicherbereich der SPS mit Ausnahme des VariablenSpeichers VB0...VB50 und des Merkerbits M0.0 zur Verfügung.



Projekt: Scan Driver for S7/200
Dokument: scandrv.doc
Version: 1.0
Verfasser: Gerhard Bran



Programmlisting STEP7/Micro:

```
// *****  
// Datenübernahme Lichtgitter Typ SCAN Fa. Leuze lumiflex  
//  
// Um eine intelligente Weiterverarbeitung zu ermöglichen, werden die  
// bei jedem Scan ermittelten Zustände der einzelnen Lichtachsen als  
// serieller Datenstrom an ein RS-485 Interface übermittelt. Die  
// Übertragung erfolgt halbduplex mit 19.200 Baud.  
// Die vollständige Beschreibung der gesendeten Daten entnehmen Sie  
// bitte dem Handbuch des verwendeten Lichtgitters.  
// Dem User stehen die Lichtachsen bitweise ab VB20 zur Verfügung (0 =  
// kein Licht, 1 = Licht).  
// Diese können vom Userprogramm weiterverarbeitet werden, ein Beispiel  
// dazu sehen Sie weiter unten im Hauptprogramm.  
// Bei erfolgreichem Empfang eines Datenpaketes vom SCAN-Lichtgitter  
// wird der Merker M_Userbuffer_ready gesetzt. Dieser kann vom User-  
// programm gelöscht und somit als Empfangskontrolle verwendet werden.  
// Dem Userprogramm steht der gesamte Speicherbereich der CPU zur  
// Verfügung, mit Ausnahme des Variablenspeichers VB0...VB50 und  
// des Merkerbits M0.0  
// *****  
  
NETWORK 1          //NETZWERKTITEL (eine Zeile)  
//  
//Platz für Userprogramm  
//  
  
NETWORK 2          // Initialisierung  
LD      "M_Erster_Zyklus"  
CALL    0  
  
NETWORK 3          // Anzeige Strahldaten (nur zu Testzwecken)  
LD      "M_Userbuffer_ready"  
MOVB    VB20, AB0  
MOVB    VB21, AB1  
R      "M_Userbuffer_ready", 1  
  
NETWORK 4          // Programmende  
// MEND  
  
// *****  
// Initialisierung Variablen + Schnittstelle 0  
// *****  
  
NETWORK 1          // Unterprogramm 0 (Initialisierung)  
// SBR      0  
  
NETWORK 2  
LD      "M_Immer_Eins"  
MOVD    +0, VD0  
MOVW    +0, VW4  
MOVB    5, "K_S0_Init_Param"          // 19200Bd, Frei prog. Komm.
```

Projekt: Scan Driver for S7/200
Dokument: scandrv.doc
Version: 1.0
Verfasser: Gerhard Bran



```
MOVB 16#B, "K_S0_STK" // Startkennung des Datensatzes
MOVB 16#C0, "K_S0_Steuerbyte" // Empfang bereit, Anfangszeichen in
SMB88
ATCH 0, 8 // Ereigniss 8: von S0 Zeichen
empfangen
ENI // Freigabe aller Interruptereignisse

NETWORK 3 // Ende Initialisierung
// RET

// *****
// Interrupt-Routine 0: Empfang der ersten vier Zeichen über Interrupt
//
// Die Routine versucht die ersten vier Zeichen eines SCAN-Datenpaketes
// einzulesen und daraus die Datenpaketlänge zu bestimmen. Diese ist
// abhängig von der Anzahl der Lichtstrahlen des eingesetzten SCAN-Licht
// gitters. Danach wird dieser Interrupt gesperrt und die nachfolgenden
// Datenpakete mittels RCV-Anweisung empfangen.
// *****

NETWORK 1 // Interrupt-Routine 0
// INT 0

NETWORK 2
LDB= 0, "M_Schrittzaehler" // Erstes Zeichen wird empfangen
UB= "V_Empf_Zeichen", "K_S0_STK" // == Startzeichen?
INCB "M_Schrittzaehler" // Schrittzähler erhöhen
CRETI

NETWORK 3
LDB= 1, "M_Schrittzaehler" // Zweites Zeichen wird empfangen
MOVB "V_Empf_Zeichen", VB4 // LDP in VB4 ablegen
INCB "M_Schrittzaehler" // Schrittzähler erhöhen
CRETI

NETWORK 4
LDB= 2, "M_Schrittzaehler" // Drittes Zeichen wird empfangen
INCB "M_Schrittzaehler" // Schrittzähler erhöhen
CRETI

NETWORK 5
LDB= 3, "M_Schrittzaehler" // Viertes Zeichen wird empfangen
LPS
MOVD +0, VD0 // Überprüfung LDP - STZ
MOVB "V_Empf_Zeichen", VB3 // STZ nach VB3 verschieben
DTCH 8 // Interrupt sperren
DIV +8, VD0 // STZ durch 8 teilen
+I +6, VW2 // Header addieren + 1
UW= +0, VW0 // Teilungsrest == 0?
DECW VW2 // Dekrementieren
LPP
UB= VB3, VB4 // Vergl. LDP empf. - LDP berechnet
MOVB VB4, "K_S0_Ende_LDP" // LDP ist Ende-Kriterium
ATCH 1, 23 // Ende des Meldungsempfangs
RCV VB15, 0 // RCV von S0 nach Userbuffer
```

Projekt: Scan Driver for S7/200
Dokument: scandrv.doc
Version: 1.0
Verfasser: Gerhard Bran



```
CRETI
NOT                               // LDP empf. - LDP berechnet nicht
gleich
UB=    3, "M_Schrittzaehler"
MOVB   0, "M_Schrittzaehler"      // von vorne anfangen (Schrittz. = 0)
ATCH   0, 8                       // von S0 Zeichen empfangen

NETWORK 6      // Ende Interrupt-Routine 0
// RETI

// *****
// Interrupt-Routine 1: Ende des Meldungsempfangs über RCV-Anweisung
//
// Am Ende des Empfangs eines Datenpaketes wird ein Interrupt ausgelöst,
// der Empfangsstatus und das Datentelegramm werden überprüft.
// Bei Erfolg wird der Merker M_Userbuffer_ready gesetzt und ein neuer
// Empfang mittels RCV-Anweisung gestartet.
// *****

NETWORK 1      // Interrupt-Routine 1
// INT    1

NETWORK 2
LDB=   "V_S0_Empfstatus", 2      // Max. Zeichenzahl erreicht?
UB=    VB17, "K_S0_Ende_LDP"     // Zweites Zeichen = LDP?
S      "M_Userbuffer_ready", 1   // Merker Userbuffer_ready setzen
RCV    VB15, 0                   // Neuen Empfang beginnen
CRETI
NOT
RCV    VB15, 0                   // Neuen Empfang beginnen

NETWORK 3      // Ende Interrupt-Routine 1
// RETI
```

Projekt: Scan Driver for S7/200
Dokument: scandrv.doc
Version: 1.0
Verfasser: Gerhard Bran



Symboltabelle:

M_Immer_Ein	BOOL	Merker ist immer 1
M_Erster_Zyklus	BOOL	Merker ist 1 im ersten Zyklus
V_Empf_Zeichen	BYTE	Empfangenes Zeichen
K_S0_STK	BYTE	Startkennung (16#B)
K_S0_Init_Param	BYTE	Initialisierung Schnittstelle 0
K_S0_Steuerbyte	BYTE	Steuerbyte zum Empfangen von Meldungen
K_S0_Ende_LDP	BYTE	Länge des Datenpaketes
V_S0_Empfstatus	BYTE	Empfangsstatus Schnittstelle 0
M_Userbuffer_ready	BOOL	
M_Schritzaehler	BYTE	

Datenbaustein:

```
// Datenpuffer
VB16           //Start-Kennung
VB17           //Länge des Datenpaketes
VB18           //Status
VB19           //Strahlenzahl
VD20           //Nutzdaten
VD24           //Nutzdaten
VD28           //Nutzdaten
VD32           //Nutzdaten
VD36           //Nutzdaten
VD40           //Nutzdaten
VD44           //Nutzdaten
VW48           //Nutzdaten
VB50           //CRC
```

Projekt: Scan Driver for S7/200
Dokument: scandrv.doc
Version: 1.0
Verfasser: Gerhard Bran