



**Serielle Kommunikation  
&  
Protokolle**

**PCD Serie xx7**

ASCII

DK3964 (R)

RK512 (R)

Treiber im Transparent Modus

# Inhalt

---

## **1 Produktbeschreibung**

## **2 Serielle Kommunikations-Schnittstellen**

## **3 ASCII**

## **4 DK3964 (R)**

## **5 RK 512 (R)**

## **6 RK512 (R) Multi-Point nur zwischen PCD**

## **7 Transparent Modus (SAIA SFC & SFB )**

# 1 Produktbeschreibung

---

## 1.1 Einführung

Die implementierten Protokolle (ASCII, DK3964, RK512 und Transparent-Modus) erlauben es, Daten im Punkt-zu-Punkt Modus zwischen PCDs, Terminals und Computern auszutauschen. Weiterhin besteht die Möglichkeit, das RK512 Protokoll im Multi-Point Modus zu betreiben. Dies aber nur und mit RS485. Schnittstellen zwischen PCDs der Serie xx7

Weiterhin erlaubt der ASCII-Treiber, in einer einfachen Weise eigene Protokolle zu erstellen.

### Einsatz

Es können unterschiedliche Protokolle gleichzeitig auf unterschiedliche Ports mit unterschiedlichen Schnittstellenmodulen benutzt werden. Es werden alle Schnittstellenmodule unterstützt: RS232, RS422, RS485 und 20 mA Stromschleife.

## 1.2 Implementierung in der PCD Serie xx7

Diese Protokolle sind Standardfunktionalitäten. Sie sind auf allen Steuerungen der Serie xx7 vorhanden. Von welcher Firmware Version an, ist der folgenden Tabelle zu entnehmen:

	Punkt zu Punkt	Multi-Point
PCD1.M137	V1.310	V1.400
PCD2.M1x7	V1.300	V1.316
PCD2.M487	Von Beginn an	Von Beginn an
PCD3.Mxxx7	Von Beginn an	nicht integriert

## 1.3 Ausführen

In all den unterschiedlichen Protokollen und Treibern werden Operationen mit denselben Systemfunktionen ausgeführt. Natürlich haben dabei die Aufrufparameter der Funktion, abhängig vom Protokoll und Treiber, unterschiedliche Bedeutung.

Folgende Funktionen sind verfügbar:

Funktion	Protokoll	Beschreibung
SFC245	ASCII, DK3964, RK512, Transparent.	Auswahl des gewünschten Protokolls, Konfiguration und Initialisierung der seriellen Schnittstelle.
SFB12	ASCII, DK3964, RK512, Transparent.	Daten senden.
SFB13	ASCII, DK3964, Transparent.	Daten empfangen.
SFB14	RK512(R)	Daten auslesen

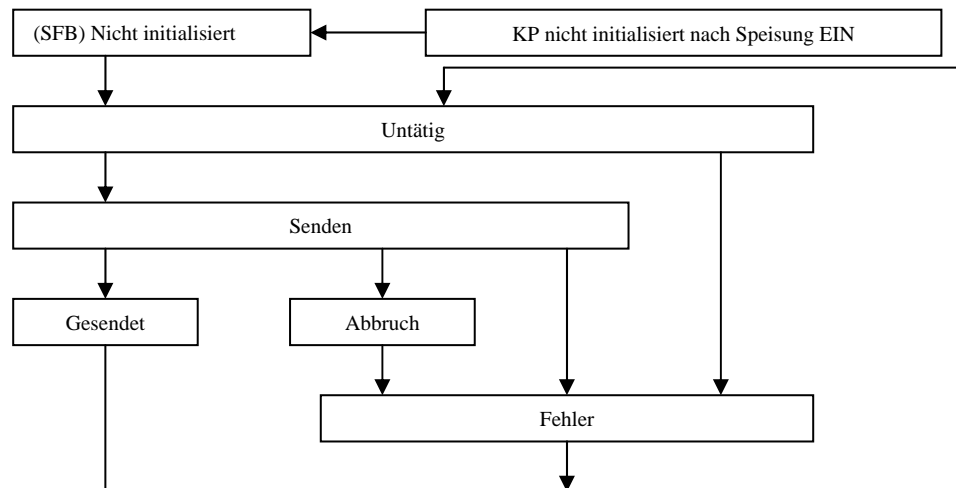
Alle Funktionen werden für das jeweilige Protokoll detailliert in den entsprechenden Kapiteln beschrieben.

## 1.4 Funktion Status Maschine (State Machine)

Alle in Kapitel 1.3 beschriebenen Funktionen, mit Ausnahme der SFC245, sind Status Maschinen (State Machine). Das heisst, die Funktion wird in Schritten ausgeführt, weil jedes Kommunikationsprotokoll (KP) bestimmten Regeln unterliegt und somit die Gesamtausführung der Funktion nicht in einem Zyklus erfolgen kann.

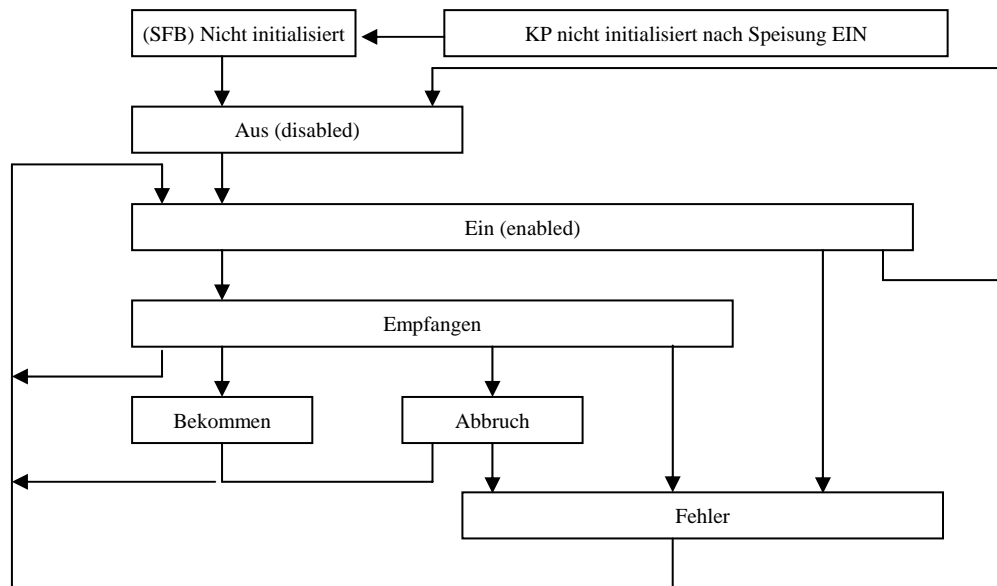
Der Status der Funktion kann, abhängig von Ereignissen, wechseln. Dabei gibt es zwei mögliche Ursachen für den Statuswechsel. Eine Ursache ist das S7-Programm, das den SFB mittels Kommandos aufruft, die andere Ursache kommt intern aus der Funktion. Nachfolgend sind die Ereignisse beschrieben, die für jeden möglichen Fall den Statuswechsel der Funktion verursachen.

### 1.4.1 Möglicher Status des SFB12



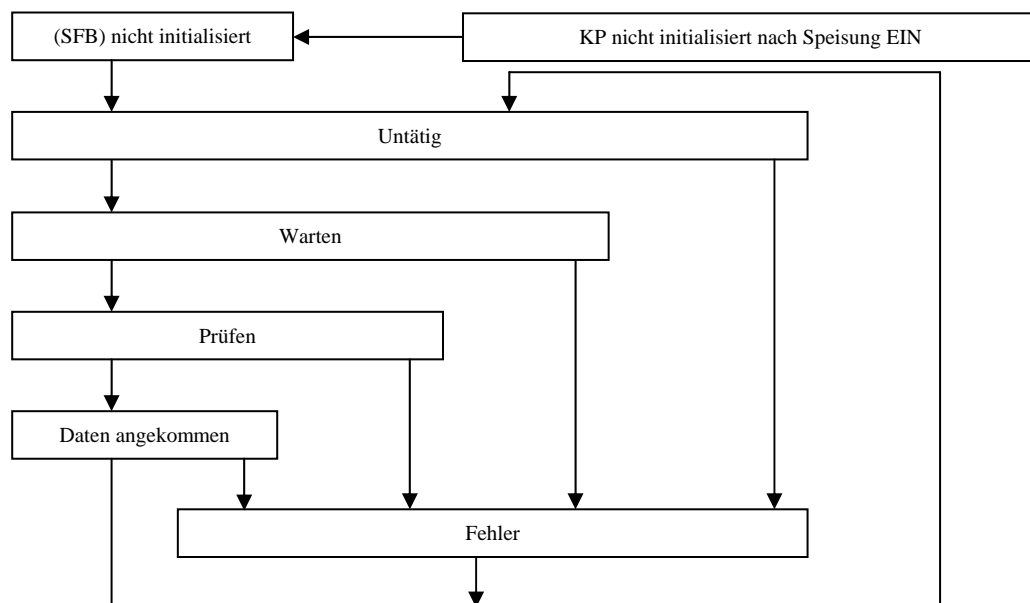
Von Status	Nach Status	Ereignis
Kein KP	Nicht initialisiert	Der Aufruf der SFC245 schaltet die KP-Funktionalität ein und setzt den Status des SFBs immer auf "Nicht initialisiert"
Nicht initialisiert	Untätig	Durch das erstmalige Aufrufen des SFB12
Untätig	Senden	Positive Flanke am Eingang REQ des SFB12
Untätig	Fehler	Problem während der Kommunikation
Senden	Gesendet	Operation wurde ausgeführt
	Abbruch	Positive Flanke am Eingang R, oder ein spezifisches S7-Objekt existiert nicht
	Fehler	Problem während der Kommunikation
Gesendet	Untätig	Nächster Aufruf des SFB12
Fehler	Untätig	Nächster Aufruf des SFB12

1.4.2 Möglicher Status des SFB13



Von Status	Nach Status	Ereignis
Kein KP	Nicht initialisiert	Der Aufruf der SFC245 schaltet die KP-Funktionalität ein und setzt den Status des SFBs immer in“ Nicht initialisiert”
Nicht initialisiert	Aus	Durch das erstmalige Aufrufen des SFB13
Aus	Ein	Aufruf des SFB13 mit EN_R = 1
Ein	Aus	Aufruf des SFB13 mit EN_R = 0
Ein	Empfangen	Eingehendes Telegramm kommt
Empfangen	Bekommen	Das ganze Telegramm ist angekommen
Bekommen	Ein	Nächster Aufruf des SFB13
Abbruch		
Fehler		
Empfangen	Ein	Aufruf des SFB13 mit EN_R = 0
Empfangen	Abbruch	Das spezifische S7-Objekt existiert nicht
Empfangen	Fehler	Fehler mit dem Empfangspuffer (Zu klein)

## 1.4.3 Möglicher Status des SFB14



Von Status	Nach Status	Ereignis
Kein KP	Nicht initialisiert	Der Aufruf der SFC245 schaltet die KP-Funktionalität ein und setzt den Status des SFBs immer in "Nicht initialisiert"
Nicht initialisiert	Untätig	Durch das erstmalige Aufrufen des SFB14
Untätig	Warten	Positive Flanke am Eingang REQ des SFB12
Warten	Prüfen	Internes Ereignis
Prüfen	Daten angekommen	Die empfangenen Daten sind korrekt
Daten angekommen	Untätig	Nächster Aufruf des SFB14
Untätig	Fehler	Problem mit der Kommunikation oder rein S7-spezifisches Objekt existiert nicht.
Warten		
Prüfen		
Have Data		
Error	Untätig	Nächster Aufruf des SFB14

## 2 Serielle Schnittstellenmodule

Detaillierte Beschreibungen zu den

- Steckplätzen A, B, B1 und B2
- Seriellen Schnittstellenmodulen
- Anschlussbelegungen
- Verdrahtung

können dem **Kapitel 8** des Handbuchs **26/757** entnommen werden.

### 2.1 Unterstützte Schnittstellen und Baudraten

Aus der folgenden Tabelle können die unterstützten Schnittstellen und Baudraten entnommen werden.

	Slot	Port	Baudraten
PCD1.M137	A	1	300,600,1200,2400,4800,9600,19200,38400
	B	2	300,600,1200,2400,4800,9600,19200,38400
	B	3	300,600,1200,2400,4800,9600,19200,38400
PCD2.M127/157	A	1	300,600,1200,2400,4800,9600,19200,38400
	B	2	300,600,1200,2400,4800,9600,19200 <sup>1</sup> ,38400 <sup>1</sup>
	B	3	300,600,1200,2400,4800,9600,19200 <sup>1</sup> ,38400 <sup>1</sup>
PCD2.M177	A	1	300,600,1200,2400,4800,9600,19200,38400
	B1	2	300,600,1200,2400,4800,9600,19200 <sup>1</sup> ,38400 <sup>1</sup>
	B1	3	300,600,1200,2400,4800,9600,19200 <sup>1</sup> ,38400 <sup>1</sup>
	B2	4	300,600,1200,2400,4800,9600,19200 <sup>1</sup> ,38400 <sup>1</sup>
	B2	5	300,600,1200,2400,4800,9600,19200 <sup>1</sup> ,38400 <sup>1</sup>
PCD2.M487	PGU	0	1200,2400,4800,9600,19200,38400,57600,115200
	A	1	1200,2400,4800,9600,19200,38400,57600,115200
	B1	2	1200,2400,4800,9600,19200 <sup>1</sup> ,38400 <sup>1</sup>
	B1	3	1200,2400,4800,9600,19200 <sup>1</sup> ,38400 <sup>1</sup>
	B2	4	1200,2400,4800,9600,19200 <sup>1</sup> ,38400 <sup>1</sup>
	B2 (27-29)	5 6	1200,2400,4800,9600,19200 <sup>1</sup> ,38400 <sup>1</sup> 1200,2400,4800,9600,19200,38400,57600,115200
PCD3.Mxxx7	PGU	0	wie PCD2.M487
	0	1	
	onboard	2	

#### Hinweis:

<sup>1</sup>Die maximale Baudrate von 19200, bzw. 38400 Baud der Schnittstellen 2...5 lässt sich nur mit Hilfe eines Stringeintrags im Konfigurationsdatenbausteins (CDB) umschalten. Folgender Stringeintrag stellt die maximale Baudrate für Steckplatz B oder B1 auf 38400.

Address	Name	Type	Initial value
0.0		STRUCT	
+0.0	Identificator	STRING[12]	'SAIA xx7 CDB'
+14.0	SLOT_B1	STRING[20]	'SLOT_B1:ENABLE_38400'
=36.0		END_STRUCT	

Ein Konfigurationsdatenbaustein lässt sich mit dem entsprechenden Eintrag mit dem I/O-Builder erstellen, der von der Internetadresse [www.sbc-support.ch](http://www.sbc-support.ch) herunter geladen werden kann.

**Einschränkung:**

- Als Voreinstellung stehen die Schnittstellen 2...5 auf 19200 Baud
- Wenn die eine Baudrate eingestellt ist, ist die andere nicht erlaubt, d.h. bei 38400 Baud wird 19200 nicht unterstützt und umgekehrt.



## 3 ASCII

### Einführung

Der ASCII Kommunikationstreiber führt nicht wirklich ein Protokoll aus, sondern er kontrolliert und steuert eher einen Kommunikationsrahmen. Es gibt 4 unterschiedliche Wege, diesen Rahmen zu steuern.

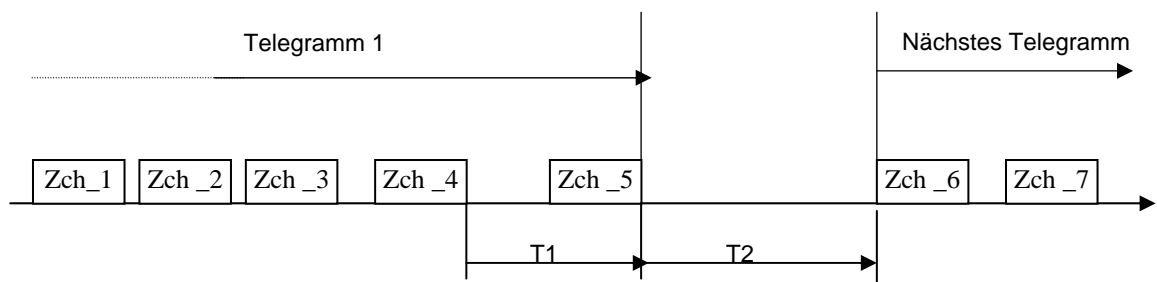
- Time out (Zeichenverzugszeit)
- Feste Länge
- Ein Endzeichen
- Zwei Endzeichen

Wenn das Ende eines Telegramms erkannt wurde, benachrichtigt das System den Anwender, dass dieses komplett empfangen im Empfangsspeicherbereich abgelegt wurde. Es ist nicht möglich, mehr als eine Telegramm im Empfangspuffer zu haben. Wenn sich ein Telegramm im Empfangspuffer befindet, gehen alle weiteren Zeichen verloren, bis das im Empfangspuffer wartende Telegramm aus dem Empfangspuffer in einen anderen Speicherbereich transferiert wurde.

Im Folgenden werden die 4 unterschiedlichen ASCII Modi detaillierter beschrieben.

### 3.1 Time out (Zeichenverzugszeit)

Der Time out Modus ist eine Möglichkeit, das Ende des Telegramms zu detektieren. Wenn nach einer vordefinierten Zeit (ZVZ = Zeichenverzugszeit) kein Zeichen empfangen wurde, wird dies von der Empfangsschnittstelle als Ende eines Telegramms interpretiert. Zeichen, die danach kommen, werden als neuer Telegramm oder gar nicht betrachtet.



ASCII Bild 1

Zeit T1 ist kleiner als die vordefinierte **Time out** Zeit (ZVZ). Somit ist Zch\_5 innerhalb des Telegramms. Aber die Zeit T2 ist grösser als die vordefinierte **Time out** Zeit (ZVZ). Somit ist Zch\_5 das letzte Zeichen. Zch\_6 gehört zum nächsten Telegramm.

### 3.2 Feste Länge

Dieser Modus ist sehr einfach. Jedes Telegramm hat eine fest definierte Anzahl Zeichen. Wenn die Empfangsschnittstelle die vordefinierte Anzahl Zeichen empfangen hat, ist das Telegramm zu Ende. Die nachfolgenden Zeichen gehören zum nächsten Telegramm.

Es muss aber weiterhin die maximale Zeit zwischen den empfangenen Zeichen beachtet werden. Wenn diese länger als die vordefinierte Zeit zwischen den Zeichen (**ZVZ**) ist, wird eine Fehlermeldung ausgegeben.

### **3.3 Ein Endzeichen**

Das Ende eines Telegramms wird durch ein vordefiniertes Endzeichen identifiziert. Jedes mal, wenn ein solches Zeichen beim Empfänger eintrifft, wird dieser das Ende eines Telegrammrahmens erkennen.

Es muss aber weiterhin die maximale Zeit zwischen den empfangenen Zeichen beachtet werden. Wenn diese länger als die vordefinierte Zeit zwischen den Zeichen (**ZVZ**) ist, wird eine Fehlermeldung ausgegeben.

### **3.4 Zwei Endzeichen**

Hier ist das Verhalten identisch mit dem „Ein Endzeichen“ Modus in Kapitel 3.3. Aber in diesem Fall sind 2 vordefinierte, aufeinander folgende Zeichen für das Ende eines Telegramms nötig, damit dieses identifiziert wird.

### 3.5 Initialisierung der PCD.xx7 im ASCII Modus

Um den ASCII Modus zu benutzen, muss der COM Port der PCD initialisiert und konfiguriert werden. Dieses geschieht mit der SFC245. Um einen Modus einzuschalten, muss die SFC245 einmal aufgerufen werden.

Diese SFC wird auch benutzt, um den COM Port für andere Kommunikationsprotokolle wie RK512 und DK3964 zu konfigurieren. Dann werden einige Parameter des ASCII Treibers nicht benötigt. Die Spalte „Ausgewählter Modus“ zeigt an, welche Parameter, abhängig vom gewählten Modus, für den ASCII Treiber benötigt werden.

Name	Para	Typ	Möglicher Wert	Ausgew. Modus				Bemerkung
				5	6	7	8	
Port	IN0	INT	0...6					COM Port Nummer (siehe Kapitel 2.1)
Mode	IN1	INT	5...8					5 ASCII – Feste Länge 6 ASCII – 1 Endzeichen 7 ASCII – 2 Endzeichen 8 ASCII – Time Out (ZVZ)
Baud Rate	IN2	DINT						Baudrate (siehe Kapitel 2.1)
Data Bit	IN3	INT	7...8					Anzahl Datenbits
Stop Bit	IN4	INT	1...2					Anzahl Stoppbits
Parity	IN5	INT	0...4					0 Keine 1 Gerade 2 Ungerade 3 Force Low 4 Force High
Control	IN6	INT	0...3					Schnittstellentyp 0 RS 232 1 RS485 2 RS422 3 TTY
XON	IN7	BYTE	0...FFh					Nicht verwendet
XOFF	IN8	BYTE	0...FFh					Nicht verwendet
WaitSend	IN9	WORD	0...FFFEh					Nicht verwendet
WaitInactiv	IN10	WORD	0...FFFEh					Nicht verwendet
TelCount	IN11	INT	1					Anzahl Rahmen im Puffer
Overwrite	IN12	BOOL	FALSE / TRUE					FALSE: Telegramm überschreibbar TRUE: Telegramm überschreibbar im Puffer, aber nur wenn TelCount = 1.
DelRxPuffer	IN13	BOOL	FALSE / TRUE					Nicht verwendet
DKPriority	IN14	BOOL	FALSE / TRUE					Nicht verwendet
ZVZ	IN15	WORD	0, 1...FFFEh					Zeichenverzugszeit in ms, 0 = Vorgabewert von 4ms
QVZ	IN16	WORD	0, 1...FFFEh					Nicht verwendet
TryToConnect	IN17	INT	0...255					Nicht verwendet
TryToSend	IN18	INT	0...255					Nicht verwendet
FixedLen	IN19	INT	1...1024					Telegrammlänge beim Empfang
EndChar1	IN20	BYTE	0...255					Endzeichen 1
EndChar2	IN21	BYTE	0...255					Endzeichen 2
SENDBuffer	IN22	INT	0...4000					Sendepuffergröße, abhängig von der zu sendenden Telegrammlänge. (In Bytes)
RCVBuffer	IN23	INT	0...4000					Empfangspuffergröße, abhängig von der zu empfangenen Telegrammlänge. (In Bytes)
Dummy_I0	IN24	INT	0					Nicht verwendet
Dummy_W1	IN25	WORD	0					Nicht verwendet
Dummy_W2	IN26	WORD	0					Nicht verwendet
Dummy_DW1	IN27	DWORD	0					Nicht verwendet
RetVal	OUT	WORD						Operationsergebnis siehe Kapitel 3.5.2.

Nicht verwendet
  Erforderlich

### 3.5.1 Beispiel :

In diesem Beispiel wird der COM Port 1 der PCD xx7 für den ASCII Treiber im Time Out Modus (ZVZ) konfiguriert. Die Baudrate ist 9600, 1 Stoppbit, keine Parität. Die Schnittstelle ist RS232. Die Zeichenverzugszeit ist fest auf 20ms. Send- und Empfangspuffer sind 300 Bytes gross.

```
CALL  SFC  245
      IN0   :=1           // Seriell es Port N°1
      IN1   :=8           // Time out Modus (ZVZ)
      IN2   :=L#9600      // Baudrate
      IN3   :=8           // Datenbits
      IN4   :=1           // Stoppbit
      IN5   :=0           // Parität (Keine)
      IN6   :=0           // RS232 Schnittstelle
      IN7   :=B#16#0      // Nicht verwendet
      IN8   :=B#16#0      // Nicht verwendet
      IN9   :=W#16#0      // Nicht verwendet
      IN10  :=W#16#0      // Nicht verwendet
      IN11  :=1           // TelCount
      IN12  :=FALSE       // Rahmen nicht überschreibbar
      IN13  :=FALSE       // Nicht verwendet
      IN14  :=FALSE       // Nicht verwendet
      IN15  :=W#16#14     // ZVZ (Time Out) = 20 ms
      IN16  :=W#16#0      // Nicht verwendet
      IN17  :=0           // Nicht verwendet
      IN18  :=0           // Nicht verwendet
      IN19  :=0           // Hier nicht verwendet (Feste Länge)
      IN20  :=B#16#0      // Hier nicht verwendet (1Endzeichen)
      IN21  :=B#16#0      // Hier nicht verwendet (2Endzeichen)
      IN22  :=300         // 300 Bytes für den Sendepuffer
      IN23  :=300         // 300 bytes für den Empfangspuffer
      IN24  :=0           // Nicht verwendet
      IN25  :=W#16#0      // Nicht verwendet
      IN26  :=W#16#0      // Nicht verwendet
      IN27  :=DW#16#0     // Nicht verwendet
      RET_VAL:=#RetVal    // Rückgabewert
```

## 3.5.2 Rückgabewert der SFC245

Wert	Beschreibung
0	Initialisierung war erfolgreich
-1	Ungültige Schnittstellennummer
-2	Nicht genügend S7-Speicher, um die Puffer zu erzeugen. Evtl. Komprimierung durchführen.
-3	Nicht genügend S7-Speicher, um die Puffer zu erzeugen (nach durchgeführter Komprimierung).
-4	Ungültiger Wert für den Parameter MODE
-5	Ungültige Schnittstellenparameter (Baudrate, Datenbit, Stoppbit oder Parität)
-6	Ungültiger Wert für Parameter WAIT_SEND oder WAIT_INACTIVITY
-7	Ungültiger Wert für Parameter TEL_COUNT
-8	Ungültiger Wert für Parameter ZVZ oder QVZ
-9	Ungültiger Wert für Parameter TRY_CONNECT oder TRY_SEND
-10	ASCII – Feste Länge: Die Telegrammlänge ist länger als der Empfangspuffergrösse
-11	Ungültiger Wert für Parameter in SEND_BUFFER oder RCV_BUFFER
-12	Die Gesamtspeichergösse des Sende- und Empfangspuffer ist grösser als die maximal erlaubten 64kB.
-13	Die SFC wurde mit einer unterschiedlichen Summe aus Empfangs- und Sendepuffer aufgerufen als beim letzten Aufruf.

**Bemerkung:**

Es ist möglich, den Schnittstellentreiber oder das Protokoll während der Ausführung umzuschalten, aber folgendes muss beachtet werden:

- 1) Die Puffergrösse kann nicht verändert werden
- 2) Die Sende- und Empfangsbaustein benötigen eine Flanke am Enable-Eingang, um den neuen Modus zu erkennen.

### 3.7 SENDEN eines Telegrammrahmens

Um einen Telegrammrahmen zu senden, wird der SFB12 benötigt. Dieser SFB kann unter Berücksichtigung der unterschiedlichen Bedingungen des gewählten Modus für alle ASCII Modi benutzt werden.

#### 3.7.1 Parameter SFB 12

Parameter	Beschreibung
REQ	Mit einer positiven Flanke wird die SENDE Prozedur gestartet
R	Mit einer positiven Flanke wird abgebrochen und das Senden zurückgesetzt.
ID	Serielle COM Port Nummer..
R_ID	Nicht verwendet
DONE	SENDE Prozedur ist abgeschlossen (DONE). Das Flag <b>DONE</b> bleibt einen Zyklus lang EINS.
ERROR	Fehler (ERROR) in der Sende Prozedur. Das Flag <b>ERROR</b> bleibt einen Zyklus lang EINS.
STATUS	Fehlercode
SD_1	Datenbereich, in den gesendet wird. Dieser Parameter ist ein ANY Pointer, aber die Länge des ANY Pointer wird hier nicht berücksichtigt. Diese wird von Parameter LEN übernommen.
LEN	Es können maximal 4'000 Bytes auf einmal gesendet werden. Natürlich muss die Sendepuffergrösse gleich oder grosser deklariert sein.

#### 3.7.2 SENDE Bedingungen

ASCII Modus	Bedingung
Feste Länge (5)	Die Anzahl zu sendende Zeichen wird durch den Parameter <b>LEN</b> bestimmt.
Ein Endzeichen (6)	Das erste übertragene Zeichen ist das, auf welches durch SD_1 gezeigt wird. Alle weiteren Zeichen werden gesendet, bis das <b>Endzeichen</b> gefunden wird. Das Endzeichen muss innerhalb der vorgegebenen Länge des Parameters <b>LEN</b> liegen. Sonst wird nichts gesendet.
Zwei Endzeichen (7)	Gleich wie Modus 6, aber mit 2 Endzeichen.
Time Out (ZVZ) (8)	Die Anzahl zu sendende Zeichen ist durch den Parameter <b>LEN</b> festgelegt. Wenn das letzte Zeichen gesendet wurde, wartet die PCD die ZVZ ab, bevor erneut gesendet wird. (Wenn eine weitere Sendeanforderung vorliegt)

Es sind einige weitere Bedingungen in Bezug auf die Status Maschine der Funktion in Kapitel 1.4 beschrieben.

## 3.7.3 SENDE Beispiel

```

CALL SFB 12 , DB12
      REQ  :=M50.0           // Sende Anforderung
      R    :=M50.1           // Übertragung zurücksetzen
      ID   :=W#16#1          // COM Port Nummer
      R_ID :=                 // Nicht verwendet
      DONE :=M100.0         // Übertragung ohne Fehler abgeschlossen
      ERROR :=M100.1         // Fehler bei der Übertragung
      STATUS:=MW102          // Fehlercode
      SD_1 :=P#DB100.DBX 0.0 BYTE 1000 // Datenquelle
      LEN  :=MW104           // Anzahl zu sendenden Bytes

```

## 3.7.4 Rückgabewerte STATUS

Wert	Beschreibung
-9 (FFF7h)	Telegramm kann nicht kopiert werden
-8 (FFF8h)	Länge ist zu gross
-7 (FFF9h)	Unbekannter Typ
-6 (FFFAh)	Ungültiger Zielbereich
-5 (FFFBh)	Ungültige Länge
-4 (FFFCh)	DB ist nicht geladen
-3 (FFFDh)	Ungültige DB Nummer
11	Warnung: Die Funktion kann nicht ausgeführt werden, da sie bereits ausgeführt wird.



### 3.8 Telegrammrahmen EMPFANGEN

Um die eingehenden Daten auf der seriellen Schnittstelle mit dem ASCII Treiber zu lesen, wird der SFB 13 benötigt.

#### 3.8.1 Parameter SFB 13

Parameter	Beschreibung
EN_R	EINS = Empfang einschalten
ID	Serielle Port Nummer
R_ID	Nicht verwendet.
NDR	New Data Ready: Neue Daten sind angekommen und in den gewünschten Speicherbereich kopiert worden. <b>NDR</b> bleibt einen Zyklus lang EINS.
ERROR	Fehler beim Empfang. <b>ERROR</b> bleibt einen Zyklus lang EINS.
STATUS	Fehlercode
RD_1	Spezifiziert den Bereich, in den die eingehenden Daten gelegt werden sollen. Dieser Parameter ist ein ANY Pointer. Die Anzahl festgelegter Bytes in diesem Parameter muss gleich oder grösser als die Anzahl empfangener Bytes sein.
LEN	Anzahl der sich im Empfangspuffer befindlichen Bytes.

#### 3.8.2 EMPFANGS Bedingungen

ASCII Modus	Bedingung
Feste Länge (5)	Das Signal NDR wird EINS, wenn die Anzahl empfangener Zeichen gleich der Anzahl der spezifizierten Bytes im Parameter FixLen (IN19) der SFC245 ist
Ein Endzeichen (6)	Das Signal NDR wird EINS, wenn das Endzeichen im Empfangspuffer erkannt wird. Ausserdem muss die Telegrammlänge kleiner als die im Parameter RD_1 spezifizierten Anzahl Bytes sein.
Zwei Endzeichen (7)	Gleich wie Modus 6, aber mit 2 Endzeichen.
Time Out (ZVZ) (8)	Das Signal NDR wird EINS, wenn die Zeit zwischen zwei Zeichen die im Parameter ZVZ der SFC245(IN15) spezifizierte Zeit überschreitet.

Es sind einige weitere Bedingungen in Bezug auf die Status Maschine der Funktion in Kapitel 1.4 beschrieben.

## 3.8.3 Beispiel

```

CALL SFB 13 , DB13
  EN_R :=M200.0    // Empfangsfunktion einschalten
  ID   :=W#16#1    // COM Port 1
  R_ID :=          // Nicht verwendet
  NDR  :=M200.1    // New Data Ready flag (Neue Daten da)
  ERROR:=M200.2    // Fehler Flag
  STATUS:=MW202    // Fehlercode
  RD_1 :=P#DB200.DBX 0.0 BYTE 2000 // Data Zielbereich
  LEN  :=MW204     // Anzahl empfangener Bytes

```

## 3.8.4 Parameter STATUS

Wert	Beschreibung
-7 (FFF9h)	Ungültiger Typ
-6 (FFFAh)	Ungültiger Zielbereich
-5 (FFFBh)	Ungültige Länge
-4 (FFFCh)	DB nicht geladen
-3 (FFFDh)	Ungültige DB Nummer
-2 (FFFEh)	COM Port nicht initialisiert
-1 (FFFFh)	Ungültige COM Port Nummer
0	OK
4	Länge zu gross im Vergleich zur Puffergrösse

### 3.9 Beispiele

Es wurde ein Beispiel in Step7 geschrieben, um mit den 4 möglichen Treibern zu arbeiten. Dieses Beispiel kann unter der Internetadresse [www.sbc-support.ch](http://www.sbc-support.ch) heruntergeladen werden. Der Dateiname ist DOC\_CP44.zip.

Diese Datei muss dearchiviert werden. Für die Durchführung mit dem SIMATIC Manager benutzen Sie das Menü  
-> Datei -> Dearchivieren

Im Projekt sind zwei Steuerungen vorhanden:

**PCD\_SEND** : Diese PCD sendet ASCII Telegramme  
**PCD\_RECEIVE** : Diese PCD empfängt ASCII Telegramme.

Um korrekt zu arbeiten, müssen die beiden COM Port 1 der 2 PCDs mit Hilfe eines PCD2.F120 Moduls verbunden werden.

Eine PCD sendet den ASCII Telegrammrahmen im ausgewählten Modus und die andere PCD empfängt diese in demselben Modus.

Der entsprechende ASCII Modus wird im Step7 Programm im Netzwerk 1 des OB100 ausgewählt.

Weiterhin sind in beiden PCDs FCs für den jeweiligen Modus:

FC5 : läuft im Modus "Feste Länge"

FC6 : läuft im Modus "Ein Endzeichen"

FC7 : läuft im Modus "Zwei Endzeichen"

FC8 : läuft im Modus "Time Out" (ZVZ).

Alle FCs sind in der gleichen Art strukturiert.

#### Sende PCD (SEND\_PCD):

Netzwerk 1: Lädt die Länge der zu sendenden Telegrammrahmen

Netzwerk 2: Aufruf des SFB12 (Senden)

Netzwerk 3: Warten von 2000 Zyklen zwischen jedem SENDEN (Nur für dieses Beispiel)

Netzwerk 4: Zählen der gesendeten Telegrammrahmen und der Fehler

#### Empfangs PCD (RECEIVE\_PCD):

Netzwerk 1: Aufruf des SFB13 (Empfangen)

Netzwerk 2: Zählen der empfangenen Telegrammrahmen und der Fehler

Die Variablen-tabelle VAT1 zeigt, wie viele Telegramme gesendet und empfangen wurden. Weiterhin wird angezeigt, wie viele Fehler aufgetreten sind.

## 4 DK3964

---

### Einführung

Die 3964(R) Prozedur steuert die Datenübertragung mittels Punkt-zu-Punkt Verbindung zwischen einer PCD Serie xx7 und einem Kommunikationspartner. In dieser Prozedur ist sowohl die physikalische Schicht 1 (ISO-Layer 1) als auch die Daten Verbindungsschicht(ISO-Layer-2) integriert.

### Steuerzeichen

Während der Datenübertragung fügt die 3964(R) Prozedur Steuerzeichen zu den Informationsdaten (Daten Verbindungsschicht) hinzu. Diese Steuerzeichen erlauben es dem Kommunikationspartner zu überprüfen, ob die Daten vollständig und ohne Fehler angekommen sind.

Folgenden Steuerzeichen werden von der 3964(R) Prozedur analysiert:

- **STX**        **Start of text** (Beginn der Daten)
- **DLE**        **Data link escape** (Datenfluss zuende)
- **ETX**        **End of text** (Datenende)
- **BCC**        **Block check character** (Blocksummenzeichen, Nur bei 3964(R))
- **NAK**        **Negative acknowledge** (Negative Quittierung)

### Priorität

Bei der 3964(R) Prozedur muss einer der beiden Teilnehmer eine höhere Priorität haben, als der andere. Dieses muss während der Konfiguration eingestellt werden.

D.h., wenn beide Partner zur gleichen Zeit beginnen, eine Verbindung aufzubauen, wird es dem Partner mit der höheren Priorität erlaubt sein, fortzufahren, während der mit der niedrigen Priorität seine Anforderung auf später verschiebt.

### Block Checksumme

Nur mit dem 3964(R) Übertragungsprotokoll wird die Datenintegrität durch das Senden eines zusätzlichen Blockprüfzeichens (BCC) erhöht  
Aber eine Blockprüfsummen-Berechnung (logische EXOR Operation) kann keine fehlenden Nullen erkennen, da der Wert Null das EXOR-Ergebnis nicht beeinflusst.

## 4.1 Initialisierung der PCD.xx7 für das 3964(R) Protokoll

Um eine serielle Schnittstelle der PCD Serie xx7 nutzen zu können, muss diese initialisiert und konfiguriert werden. Dies geschieht mit der SFC245. Um ein Protokoll auf einem seriellen Port einzuschalten, muss die SFC245 einmal aufgerufen werden. Diese SFC wird auch benutzt, um den COM Port für andere Kommunikationsprotokolle wie RK512 und ASCII zu konfigurieren. Dann werden einige Parameter des 3964 Protokolls nicht benötigt. Die Spalte „Ausgewählter Modus“ zeigt an, welche Parameter für das 3964 Protokoll werden.

Name	Para	Typ	Möglicher Wert	Ausgew. Modus		Bemerkung
				1	2	
Port	IN0	INT	0...6			COM Port Nummer (siehe Kapitel 2.1)
Mode	IN1	INT	1...2			1 DK 3964 2 DK 3964 R
Baud Rate	IN2	DINT				Baudrate (siehe Kapitel 2.1)
Data Bit	IN3	INT	7...8			Anzahl Datenbits
Stop Bit	IN4	INT	1...2			Anzahl Stoppbits
Parity	IN5	INT	0...4			0 Keine 1 Gerade 2 Ungerade 3 Force Low 4 Force High
Control	IN6	INT	0...3			Schnittstellentyp 0 RS 232 1 RS485 2 RS422 3 TTY
XON	IN7	BYTE	0...FFh			Nicht verwendet
XOFF	IN8	BYTE	0...FFh			Nicht verwendet
WaitSend	IN9	WORD	0...FFFEh			Nicht verwendet
WaitInactiv	IN10	WORD	0...FFFEh			Nicht verwendet
TelCount	IN11	INT	1			Anzahl Rahmen im Puffer
Overwrite	IN12	BOOL	FALSE / TRUE			FALSE: Telegramm überschreibbar TRUE: Telegramm überschreibbar im Puffer, aber nur wenn TelCount = 1.
DelRxPuffer	IN13	BOOL	FALSE / TRUE			Nicht verwendet
DKPriority	IN14	BOOL	FALSE / TRUE			EINS wenn diese PCD der Partner mit der höheren Priorität sein soll
ZVZ	IN15	WORD	0, 1...FFFEh			Zeichenverzugszeit (zwischen 2 Zeichen) in Schritten von 10ms, 0 = Vorgabewert von 220ms
QVZ	IN16	WORD	0, 1...FFFEh			Quittungsverzugszeit (warten auf Antwort) in Schritten von 10ms, 0 = Vorgabewert von 550 ms für das 3964 und 2000 ms für das 3964 R .
TryToConnect	IN17	INT	0...255			Anzahl der Wiederholungsversuche um erneut mit dem Partner wieder Verbindung aufzunehmen, 0 = Vorgabewert von 6.
TryToSend	IN18	INT	0...255			Anzahl der Wiederholungsversuche um erneut Daten zu senden, 0 = Vorgabewert von 6
FixedLen	IN19	INT	1...1024			Telegrammrahmenlänge beim Empfang
EndChar1	IN20	BYTE	0...255			Endzeichen 1
EndChar2	IN21	BYTE	0...255			Endzeichen 2
SENDBuffer	IN22	INT	0...4000			Sendepuffergrösse (in Bytes)
RCVBuffer	IN23	INT	0...4000			Empfangspuffergrösse (in Bytes)
Dummy_I0	IN24	INT	0			Nicht verwendet
Dummy_W1	IN25	WORD	0			Nicht verwendet
Dummy_W2	IN26	WORD	0			Nicht verwendet
Dummy_DW1	IN27	DWORD	0			Nicht verwendet
RetVal	OUT	WORD				Operationsergebnis, siehe Kapitel 4.1.2

Nicht verwendet
  Erforderlich

#### 4.1.1 Beispiel

In diesem Beispiel wird der COM Port 1 der PCD xx7 für das DK3964 Protokoll konfiguriert. Die Baudrate ist 9600, 1 Stoppbit, keine Parität. Die Schnittstelle ist RS232. Die Zeichenverzugszeit (ZVZ) ist fest auf 200ms. Die Quittungsverzugszeit (QVZ) steht auf dem Vorgabewert von 550ms. Sende- und Empfangspuffer sind 4000 Bytes gross. Diese Station hat höhere Priorität. Die Anzahl Wiederholungsversuche für erneuten Verbindungsaufbau und erneutes Senden steht auf dem Vorgabewert 6.

```
CALL SFC 245
      IN0      :=1           // Serielles Port Nr.1
      IN1      :=1           // DK3964
      IN2      :=L#9600     // Baudrate
      IN3      :=8           // Databits
      IN4      :=1           // Stoppbits
      IN5      :=0           // Parität (Keine)
      IN6      :=0           // RS232 Schnittstelle
      IN7      :=B#16#0     // Nicht verwendet
      IN8      :=B#16#0     // Nicht verwendet
      IN9      :=W#16#0     // Nicht verwendet
      IN10     :=W#16#0     // Nicht verwendet
      IN11     :=1           // TelCount
      IN12     :=FALSE      // Rahmen nicht überschreibbar
      IN13     :=FALSE      // Nicht verwendet
      IN14     :=TRUE       // DK Priorität ist hier hoch
      IN15     :=W#16#14    // ZVZ (zwischen Zeichen) = 200 ms
      IN16     :=W#16#0     // QVZ (für Antwort)Default =550ms
      IN17     :=0           // Verbindungsversuche, Default = 6
      IN18     :=0           // Sendeveruche, Default = 6
      IN19     :=0           // Nicht verwendet
      IN20     :=B#16#0     // Nicht verwendet
      IN21     :=B#16#0     // Nicht verwendet
      IN22     :=4000       // 4000 Bytes für den Sendepuffer
      IN23     :=4000       // 4000 Bytes für den Empfangspuffer
      IN24     :=0           // Nicht verwendet
      IN25     :=W#16#0     // Nicht verwendet
      IN26     :=W#16#0     // Nicht verwendet
      IN27     :=DW#16#0    // Nicht verwendet
      RET_VAL :=MW240       // Rückgabewert
```

### 4.1.2 Rückgabewert der SFC245

Wert	Beschreibung
0	Initialisierung war erfolgreich
-1	Ungültige Schnittstellennummer
-2	Nicht genügend S7-Speicher, um die Puffer zu erzeugen. Evtl. Komprimierung durchführen.
-3	Nicht genügend S7-Speicher, um die Puffer zu erzeugen (nach durchgeführter Komprimierung)
-4	Ungültiger Wert für den Parameter MODE
-5	Ungültige Schnittstellenparameter (Baudrate, Datenbit, Stoppbit oder Parität)
-6	Ungültiger Wert für Parameter WAIT_SEND oder WAIT_INACTIVITY
-7	Ungültiger Wert für Parameter TEL_COUNT
-8	Ungültiger Wert für Parameter ZVZ oder QVZ
-9	Ungültiger Wert für Parameter TRY_CONNECT oder TRY_SEND
-10	ASCII – Feste Länge: Die Telegrammlänge ist länger als der Empfangspuffergrösse
-11	Ungültiger Wert für Parameter in SEND_BUFFER oder RCV_BUFFER
-12	Die Gesamtspeichergrösse des Sende- und Empfangspuffer ist grösser als die maximal erlaubten 64kB.
-13	Die SFC wurde mit einer unterschiedlichen Summe aus Empfangs- und Sendepuffer aufgerufen als beim letzten Aufruf.

#### **Bemerkung:**

Es ist möglich, den Schnittstellentreiber oder das Protokoll während der Ausführung umzuschalten, aber folgendes muss beachtet werden:

- 1) Die Puffergrösse kann nicht verändert werden
- 2) Die Sende- und Empfangsbaustein benötigen eine Flanke am Enable-Eingang, um den neuen Modus zu erkennen.

## 4.2 Daten Senden mit DK3964(R)

Das Senden von Daten via DK3964(R) geschieht folgendermassen: Daten können mit dem System Funktionsbaustein BSEND(SFB12) gesendet werden. Der Kommunikationspartner kann diese Daten mit dem System Funktionsbaustein BRCV(SFB13) empfangen. Diese Art der Datenübertragung hat den Vorteil, dass vom Anwenderprogramm erkannt werden kann, wenn alle Daten übertragen wurden. Dies geschieht durch die Überprüfung des NDR Bit beim Empfänger und dem DONE Bit beim Sender.

### 4.2.1 Parameter SFB 12

Parameter	Beschreibung
REQ	Mit einer positiven Flanke wird die SENDE Prozedur gestartet
R	Mit einer positiven Flanke wird abgebrochen und das Senden zurückgesetzt.
ID	Serielle COM Port Nummer..
R_ID	Nicht verwendet
DONE	SENDE Prozedur ist abgeschlossen (DONE). Das Flag <b>DONE</b> bleibt einen Zyklus lang EINS.
ERROR	Fehler (ERROR) in der Sende Prozedur. Das Flag <b>ERROR</b> bleibt einen Zyklus lang EINS.
STATUS	Fehlercode
SD_1	Datenbereich, in den gesendet wird. Dieser Parameter ist ein ANY Pointer, aber die Länge des ANY Pointer wird hier nicht berücksichtigt. Diese wird von Parameter LEN übernommen.
LEN	Es können maximal 4'000 Bytes auf einmal gesendet werden. Natürlich muss die Sendepuffergrösse gleich oder grösser deklariert sein.

Es sind einige weitere Bedingungen in Bezug auf die Status Maschine der Funktion in Kapitel 1.4 beschrieben.

### 4.2.2 SENDE Beispiel

```
CALL SFB 12 , DB12
      REQ :=M50.0 // Sendeanforderung
      R :=M50.1 // Übertragung zurücksetzen
      ID :=W#16#1 // COM Port Nummer
      R_ID := // Nicht verwendet
      DONE :=M100.0 // Übertragung ohne Fehler abgeschlossen
      ERROR :=M100.1 // Fehler bei der Übertragung
      STATUS:=MW102 // Fehlercode
      SD_1 :=P#DB100.DBX 0.0 BYTE 1000 // Datenquelle
      LEN :=MW104 // Anzahl zu sendenden Bytes
```



**4.2.3 Rückgabewerte STATUS**

<b>Wert</b>	<b>Beschreibung</b>
-9 (FFF7h)	Telegramm kann nicht kopiert werden
-8 (FFF8h)	Länge ist zu gross
-7 (FFF9h)	Unbekannter Typ
-6 (FFFAh)	Ungültiger Zielbereich
-5 (FFFBh)	Ungültige Länge
-4 (FFFCh)	DB ist nicht geladen
-3 (FFFDh)	Ungültige DB Nummer
1	Kommunikationsprobleme (Verbindungsleitung überprüfen)
5	Rücksetzen der Anforderung empfangen
11	Warnung: Die Funktion kann nicht ausgeführt werden, da sie bereits ausgeführt wird.

**Bemerkung:**

Im Standard-DK3964R Protokoll gibt es eine zweite Möglichkeit, Daten zu einem Kommunikationspartner mit einer Empfangs-Mailbox zu übertragen. In der PCD Serie xx7 ist diese Funktion nicht im 3964(R) Protokoll, sondern im RK512 Protokoll implementiert. (Siehe Kapitel 5).

### 4.3 Daten Empfangen mit DK 3964(R)

Beim DK3964(R) werden Daten mittels dem SFB13 empfangen. Das COM Port muss vorher mit der SFC245 initialisiert und konfiguriert werden. Das Empfangen der Daten wird möglich, sobald der SFB13 ausgeführt wird. Dann kann der Anwender die Kontrolle über die Daten mittels **NDR** (New Data Ready) Bit übernehmen.

#### 4.3.1 Parameter SFB 13

Parameter	Beschreibung
EN_R	EINS = Empfang einschalten
ID	Serielle Port Nummer
R_ID	Nicht verwendet.
NDR	New Data Ready: Neue Daten sind angekommen und in den gewünschten Speicherbereich kopiert worden. <b>NDR</b> bleibt einen Zyklus lang EINS.
ERROR	Fehler beim Empfang. <b>ERROR</b> bleibt einen Zyklus lang EINS.
STATUS	Fehlercode
RD_1	Spezifiziert den Bereich, in den die eingehenden Daten gelegt werden sollen. Dieser Parameter ist ein ANY Pointer. Die Anzahl festgelegter Bytes in diesem Parameter muss gleich oder grösser als die Anzahl empfangener Bytes sein.
LEN	Anzahl der sich im Empfangspuffer befindlichen Bytes.

Es sind einige weitere Bedingungen in Bezug auf die Status Maschine der Funktion in Kapitel 1.4 beschrieben.

#### 4.3.2 Beispiel

```
CALL SFB 13 , DB13
  EN_R :=M200.0    // Empfangsfunktion Einschalten
  ID   :=W#16#1   // COM Port 1
  R_ID :=          // Nicht verwendet
  NDR  :=M200.1   // New Data Ready flag (Neue Daten angekommen)
  ERROR:=M200.2   // Fehler Flag
  STATUS:=MW202   // Fehlercode
  RD_1 :=P#DB200.DBX 0.0 BYTE 2000 // Daten Zielbereich
  LEN  :=MW204    // Anzahl empfangener Bytes
```

### 4.3.3 Parameter Status

<b>Wert</b>	<b>Beschreibung</b>
-7 (FFF9h)	Ungültiger Typ
-6 (FFFAh)	Ungültiger Zielbereich
-5 (FFFBh)	Ungültige Länge
-4 (FFFCh)	DB nicht geladen
-3 (FFFDh)	Ungültige DB Nummer
-2 (FFFEh)	COM Port nicht initialisiert
-1 (FFFFh)	Ungültige COM Port Nummer
0	OK
1	Kommunikationsprobleme (Verbindungsleitung überprüfen)
4	Länge zu gross im Vergleich zur Puffergrösse

## 4.4 Beispiele

Es wurde ein Beispiel in Step7 geschrieben, um zu zeigen, wie mit dem DK3964 zu arbeiten ist. Dieses Beispiel kann unter der Internetadresse [www.sbc-support.ch](http://www.sbc-support.ch) herunter geladen werden. Der Dateiname ist DOC\_CP44.zip.

Diese Datei muss dearchiviert werden. Für die Durchführung mit dem SIMATIC Manager benutzen Sie das Menü  
-> Datei -> Dearchivieren

Im Projekt sind zwei Steuerungen vorhanden:

**PCD\_SEND** : Diese PCD sendet mit 3964  
**PCD\_RECEIVE** : Diese PCD empfängt mit 3964

Um korrekt zu arbeiten, müssen die beiden COM Port 1 der 2 PCDs mit Hilfe eines PCD2.F120 Moduls verbunden werden.

Eine PCD sendet Daten im ausgewählten Modus und die andere PCD empfängt diese in demselben Modus.

Der 3964 Modus wird im Step7 Programm im Netzwerk 1 des OB100 ausgewählt.

Weiterhin sind in beiden PCDs FCs für den jeweiligen Modus:

FC1 : läuft im Modus "3964"  
FC2 : läuft im Modus "3964(R)"

Alle FCs sind in der gleichen Art strukturiert.

Sende PCD (SEND\_PCD):

Netzwerk 1: Lädt die Länge der zu sendenden Telegrammrahmen

Netzwerk 2: Aufruf des SFB12 (Senden)

Netzwerk 3: Warten von 2000 Zyklen zwischen jedem SENDEN (Nur für dieses Beispiel)

Netzwerk 4: Zählen der gesendeten Telegrammrahmen und der Fehler

Empfangs PCD (RECEIVE\_PCD):

Netzwerk 1: Aufruf des SFB13 (Empfangen)

Netzwerk 2: Zählen der empfangenen Telegrammrahmen und der Fehler

Die Variablen-tabelle VAT1 zeigt, wie viele Telegramme gesendet und empfangen wurden. Weiterhin wird angezeigt, wie viele Fehler aufgetreten sind.

## 5 RK512

---

### Einführung

Die RK512(R) Computer Verbindung steuert die Datenübertragung mittels Punkt-zu-Punkt Verbindung zwischen einer PCD Serie xx7 und einem Kommunikationspartner. In dieser Prozedur sind die physikalische Schicht 1 (ISO-Layer 1), die Daten Verbindungsschicht(ISO-Layer-2) und die Transportschicht (ISO Layer 4) integriert.

Die RK512 Computer Verbindung bietet weiterhin eine hohe Datenintegrität und bessere Adressierung, weil das RK512 das 3964(R) Protokoll für den Datentransport nutzt.

Weitergehende Abhandlung beim Kommunikationspartner ist gewährleistet, da der RK512 Interpreter die zusätzliche Längenspezifikation im Header überprüft. Ausserdem wird eine Bestätigung (message frame acknowledging) über Erfolg oder Misserfolg der Datenübertragung erzeugt, nachdem diese im Zieldatenbaustein abgespeichert wurden.

### Funktionalität

Es gibt zwei spezifische Funktionalitäten beim RK512(R) Protokoll. Diese Funktionalitäten sind die Befehle SEND und GET.

- Die SEND Funktionalität sendet einen Telegramm mit Nutzdaten und der Kommunikationspartner antwortet mit einem Antworttelegramm ohne Nutzdaten (Bestätigung)
- Die GET Funktionalität sendet einen telegramm ohne Nutzdaten (Anforderung) und der Kommunikationspartner antwortet mit einem Telegramm inklusive Nutzdaten.

Wenn die Länge der Daten 128Bytes überschreitet wird dem SEND und GET Telegramm automatisch ein Folgetelegramm hinzugefügt.

Jedes Telegramm des RK512(R) beginnt mit einem Header. Er kann die Telegramm-Identifikation, Information über den Datenziel- und Quellbereich und eine Fehlernummer enthalten.

## 5.1 Initialisierung der PCD.xx7 für das RK512(R) Protokoll

Um eine serielle Schnittstelle der PCD Serie xx7 nutzen zu können, muss diese initialisiert und konfiguriert werden. Dies geschieht mit der SFC245. Um ein Protokoll auf einem seriellen Port einzuschalten, muss die SFC245 einmal aufgerufen werden.

Diese SFC wird auch benutzt, um den COM Port für andere Kommunikationsprotokolle wie 3964 und ASCII zu konfigurieren. Dann werden einige Parameter des RK512 Protokolls nicht benötigt. Die Spalte „Ausgewählter Modus“ zeigt an, welche Parameter für das RK512 Protokoll werden.

Name	Para	Typ	Möglicher Wert	Ausgew. Modus		Bemerkung
				3	4	
Port	IN0	INT	0...6			COM Port Nummer (siehe Kapitel 2.1)
Mode	IN1	INT	3...4			3 RK 512 4 RK 512 R
Baud Rate	IN2	DINT				Baudrate (siehe Kapitel 2.1)
Data Bit	IN3	INT	7...8			Anzahl Datenbits
Stop Bit	IN4	INT	1...2			Anzahl Stoppbits
Parity	IN5	INT	0...4			0 Keine 1 Gerade 2 Ungerade 3 Force Low 4 Force High
Control	IN6	INT	0...3			Schnittstellentyp 0 RS 232 1 RS485 2 RS422 3 TTY
XON	IN7	BYTE	0...FFh			Nicht verwendet
XOFF	IN8	BYTE	0...FFh			Nicht verwendet
WaitSend	IN9	WORD	0...FFFEh			Nicht verwendet
WaitInactiv	IN10	WORD	0...FFFEh			Nicht verwendet
TelCount	IN11	INT	1			Anzahl Rahmen im Puffer
Overwrite	IN12	BOOL	FALSE / TRUE			FALSE: Telegramm überschreibbar TRUE: Telegramm überschreibbar im Puffer, aber nur wenn TelCount = 1.
DelRxPuffer	IN13	BOOL	FALSE / TRUE			Nicht verwendet
DKPriority	IN14	BOOL	FALSE / TRUE			EINS, wenn diese PCD der Partner mit der höheren Priorität sein soll
ZVZ	IN15	WORD	0, 1...FFFEh			Zeichenverzugszeit (zwischen 2 Zeichen) in Schritten von 10ms, 0 = Vorgabewert von 220ms
QVZ	IN16	WORD	0, 1...FFFEh			Quittungsverzugszeit (warten auf Antwort) in Schritten von 10ms, 0 = Vorgabewert von 550 ms für das RK512 und 2000 ms für das RK512 R .
TryToConnect	IN17	INT	0...255			Anzahl der Wiederholungsversuche um erneut mit dem Partner wieder Verbindung aufzunehmen, 0 = Vorgabewert von 6.
TryToSend	IN18	INT	0...255			Anzahl der Wiederholungsversuche um erneut Daten zu senden, 0 = Vorgabewert von 6
FixedLen	IN19	INT	1...1024			Telegrammrahmenlänge beim Empfang
EndChar1	IN20	BYTE	0...255			Endzeichen 1
EndChar2	IN21	BYTE	0...255			Endzeichen 2
SENDBuffer	IN22	INT	0...4000			Sendepuffergröße (in Bytes)
RCVBuffer	IN23	INT	0...4000			Empfangspuffergröße (in Bytes)
Dummy_I0	IN24	INT	0			Nicht verwendet
Dummy_W1	IN25	WORD	0			Nicht verwendet
Dummy_W2	IN26	WORD	0			Nicht verwendet
Dummy_DW1	IN27	DWORD	0			Nicht verwendet
RetVal	OUT	WORD				Operationsergebnis, siehe Kapitel 5.1.2

■ Nicht verwendet ■ Erforderlich

### 5.1.1 Beispiel

In diesem Beispiel wird der COM Port 1 der PCD xx7 für das RK512R Protokoll konfiguriert. Die Baudrate ist 9600, 1 Stoppbit, keine Parität. Die Schnittstelle ist RS232. Die Zeichenverzugszeit (ZVZ) ist fest auf 200ms. Die Quittungsverzugszeit (QVZ) steht auf dem Vorgabewert von 2000ms. Sende- und Empfangspuffer sind 4000 Bytes gross. Diese Station hat höhere Priorität. Die Anzahl Wiederholungsversuche für erneuten Verbindungsaufbau und erneutes Senden steht auf dem Vorgabewert 6.

```
CALL SFC 245
      IN0      :=1           // Seriell Port Nr.1
      IN1      :=4           // RK512
      IN2      :=L#9600     // Baudrate
      IN3      :=8           // Datenbits
      IN4      :=1           // Stoppbits
      IN5      :=0           // Parität (Keine)
      IN6      :=0           // RS232 Schnittstelle
      IN7      :=B#16#0     // Nicht verwendet
      IN8      :=B#16#0     // Nicht verwendet
      IN9      :=W#16#0     // Nicht verwendet
      IN10     :=W#16#0     // Nicht verwendet
      IN11     :=1           // TelCount
      IN12     :=FALSE      // Rahmen nicht überschreibbar
      IN13     :=FALSE      // Nicht verwendet
      IN14     :=TRUE       // DK Priorität ist hier hoch
      IN15     :=W#16#14    // ZVZ (zwischen Zeichen) = 200 ms
      IN16     :=W#16#0     // QVZ (für Antwort)Default =2000ms
      IN17     :=0           // Verbindungsversuche, Default = 6
      IN18     :=0           // Sendeveruche, Default = 6
      IN19     :=0           // Nicht verwendet
      IN20     :=B#16#0     // Nicht verwendet
      IN21     :=B#16#0     // Nicht verwendet
      IN22     :=4000       // 4000 Bytes für den Sendepuffer
      IN23     :=4000       // 4000 Bytes für den Empfangspuffer
      IN24     :=0           // Nicht verwendet
      IN25     :=W#16#0     // Nicht verwendet
      IN26     :=W#16#0     // Nicht verwendet
      IN27     :=DW#16#0    // Nicht verwendet
      RET_VAL:=MW240        // Rückgabewert
```

### 5.1.2 Rückgabewert der SFC245

Wert	Beschreibung
0	Initialisierung war erfolgreich
-1	Ungültige Schnittstellennummer
-2	Nicht genügend S7-Speicher, um die Puffer zu erzeugen. Evtl. Komprimierung durchführen.
-3	Nicht genügend S7-Speicher, um die Puffer zu erzeugen (nach durchgeführter Komprimierung).
-4	Ungültiger Wert für den Parameter MODE
-5	Ungültige Schnittstellenparameter (Baudrate, Datenbit, Stoppbit oder Parität)
-6	Ungültiger Wert für Parameter WAIT_SEND oder WAIT_INACTIVITY
-7	Ungültiger Wert für Parameter TEL_COUNT
-8	Ungültiger Wert für Parameter ZVZ oder QVZ
-9	Ungültiger Wert für Parameter TRY_CONNECT oder TRY_SEND
-10	ASCII – Feste Länge: Die Telegrammlänge ist länger als der Empfangspuffergrösse
-11	Ungültiger Wert für Parameter in SEND_BUFFER oder RCV_BUFFER
-12	Die Gesamtspeichergrösse des Sende- und Empfangspuffer ist grösser als die maximal erlaubten 64kB.
-13	Die SFC wurde mit einer unterschiedlichen Summe aus Empfangs- und Sendepuffer aufgerufen als beim letzten Aufruf.

#### **Bemerkung:**

Es ist möglich, den Schnittstellentreiber oder das Protokoll während der Ausführung umzuschalten, aber folgendes muss beachtet werden:

- 1) Die Puffergrösse kann nicht verändert werden
- 2) Die Sende- und Empfangsbaustein benötigen eine Flanke am Enable-Eingang, um den neuen Modus zu erkennen.



## 5.2 Daten senden mit RK512(R)

Wenn der COM Port im RK512 Modus konfiguriert ist, gibt es zwei Möglichkeiten, Daten zwischen der PCD und dem Kommunikationspartner zu übertragen

- 1) Auf der Senderseite wird der SFB12(SEND) und auf der Empfängerseite der SFB13(RECEIVE) benutzt. (Es ist wie beim DK3964, siehe Kapitel 4.2 „Daten senden mit DK3964“)
- 2) Auf der Senderseite wird der SFB12(SEND) benutzt, aber er sendet zu einer Mailbox des Kommunikationspartners (RECEIVER). Dann werden die eingehenden Daten beim Empfänger interpretiert. Wenn der Empfänger eine PCD.xx7 ist, benötigt diese kein weiteres Programm, ausser dass die SFC245 einmalig aufgerufen wird, um den seriellen Port zu konfigurieren. Das folgende Kapitel beschreibt diese Möglichkeit.

### 5.2.1 Synchronisierung

Da die Empfangs-PCD nicht programmiert werden muss, sind alle Schreiboperationen der Daten absolut transparent für dieses System. Dieses kann zu Problemen führen, wenn eine Synchronisation der beiden Systeme erforderlich ist. Um dieses Problem zu vermeiden, ist auf der Empfängerseite eine Funktion im Protokoll verfügbar. Diese Funktion ist das **IPC Flag** (Inter-Processor Communication Flag), welches die neu eingetroffenen Daten anzeigt.

Dieses IPC-Flag befindet sich auf der Empfängerseite. Es muss aber auf der Senderseite durch den Parameter R\_ID konfiguriert werden. Das IPC-Flag hat zwei Funktionen.

- 1) Anzuzeigen, dass sich neue Daten in der Mailbox befinden. (Das Bit ist gesetzt)
- 2) Zu vermeiden, dass Daten in der Mailbox überschrieben werden.

Ist das IPC-Flag beim Empfänger einmal durch das Senden von Daten an ihn gesetzt, ist es nicht möglich, erneut Daten dahin zu senden, bis das vorherige Flag zurück gesetzt wurde.

### 5.2.2 Parameter SFB 12

Parameter	Beschreibung
REQ	Mit einer positiven Flanke wird die SENDE Prozedur gestartet
R	Mit einer positiven Flanke wird abgebrochen und das Senden zurückgesetzt.
ID	Serielle COM Port Nummer..
R_ID	Nicht verwendet
DONE	SENDE Prozedur ist abgeschlossen (DONE). Das Flag <b>DONE</b> bleibt einen Zyklus lang EINS.
ERROR	Fehler (ERROR) in der Sende Prozedur. Das Flag <b>ERROR</b> bleibt einen Zyklus lang EINS.
STATUS	Fehlercode
SD_1	Datenbereich, in den gesendet wird. Dieser Parameter ist ein ANY Pointer, aber die Länge des ANY Pointer wird hier nicht berücksichtigt. Diese wird von Parameter LEN übernommen.
LEN	Es können maximal 4'000 Bytes auf einmal gesendet werden. Natürlich muss die Sendepuffergrösse gleich oder grösser deklariert sein.

Es sind einige weitere Bedingungen in Bezug auf die Status Maschine der Funktion in Kapitel 1.4 beschrieben.

### 5.2.3 R\_ID Parameter

Byte	Beschreibung	
<b>Byte 0</b> { Bit 0...3 }	0 → Ziel DX	1 → Ziel DB
<b>Byte 0</b> { Bit 4...7 }	Nicht verwendet (0H)	Bit-Nummer des IPC flag
<b>Byte 1</b>	Nicht verwendet (00H)	Byte-Nummer des IPC flag (zwischen 1..254)
<b>Byte 2</b>	Nicht verwendet (00H)	Offset 0..255 (Wort)
<b>Byte 3</b>	Nicht verwendet (00H)	DB-Nummer (1..255)

#### Byte 0: DX oder DB

Der DX-Modus wird für Kommunikationspartner benutzt, die das Mailbox-System nicht unterstützen. Der gesamte R\_ID = 0h.

Der DB-Modus benötigt einen Kommunikationspartner, der das Mailbox-System unterstützt.

#### Byte 2 and 3: Ziel-Offset und DB-Nummer

Hier wird der festgelegt, wo sich die Mailbox des Kommunikationspartners befindet. Die Mailbox muss ein Datenbaustein sein.

Byte 2 spezifiziert den Offset innerhalb des Datenbausteins in Worten (Wert zwischen 1...255)

Byte 3 spezifiziert die DB-Nummer (Wert zwischen 1...255)

### 5.2.4 Beispiele

In diesem Beispiel ist der R\_ID Parameter wie folgt konfiguriert:

- Sende zu einer Mailbox in DB33 ( 21h), mit dem Offset von 2 Worten (02h)
- Benutze als IPC-Flag den Merker M30.3, d.h. dies ist M30 (1Eh), Bit 3 des Kommunikationspartners.

```
CALL  SFB  12 , DB12
      REQ   :=M300.0
      R     :=M300.1
      ID    :=W#16#1
      R_ID  :=DW#16#311E0221
      DONE  :=M300.2
      ERROR :=M300.3
      STATUS:=MW302
      SD_1  :=P#DB100.DBX 0.0 BYTE 200
      LEN   :=MW304
```

**5.2.5 Rückgabewerte STATUS**

<b>Wert</b>	<b>Beschreibung</b>
-9 (FFF7h)	Telegramm kann nicht kopiert werden
-8 (FFF8h)	Länge ist zu gross
-7 (FFF9h)	Unbekannter Typ
-6 (FFFAh)	Ungültiger Zielbereich
-5 (FFFBh)	Ungültige Länge
-4 (FFFCh)	DB ist nicht geladen
-3 (FFFDh)	Ungültige DB Nummer
1	Kommunikationsprobleme (Verbindungsleitung überprüfen)
2	Funktion kann aufgrund des Kommunikationspartners nicht ausgeführt werden
5	Rücksetzen der Anforderung empfangen
9	Kann Daten nicht zum Partner schreiben, da das IPC-Flag dieses blockiert
10	Daten konnten wegen fehlendem Mailbox-Bereich nicht zum Partner geschrieben werden.
11	Warnung: Die Funktion kann nicht ausgeführt werden, da sie bereits ausgeführt wird.

**Bemerkung:**

Mit dieser Funktion können Sie Daten eines Kommunikationspartners direkt lesen und schreiben. Dies ist aber nur von DB zu DB möglich.

### 5.3 Daten empfangen mit RK512(R)

Es gibt zwei Möglichkeiten, Daten im RK512 Modus zu empfangen.

- Der aktive Modus, bei dem der SFB13(BRCV) aufgerufen wird und auf der Senderseite der SFB12 im richtigen Modus. Dies ist wie beim DK3964(R) Modus. (Siehe Kapitel 4.3)
- Ein passiver Modus, bei dem die PCD Daten empfängt, ohne irgendetwas zu tun. Es muss lediglich beim Aufstarten der COM-Port richtig konfiguriert werden. Abhängig durch die Auswahl beim Sender, kann beim Empfänger durch das IPC-Flag(Inter-Processor Communication flag) festgestellt werden, ob neue Daten eingetroffen sind.

#### **Vorgehensweise:**

- 1) SFC245 aufrufen und den COM Port konfigurieren
- 2) Wenn das IPC-Flag benutzt wird, bedeutet eine steigende Flanke, dass neue Daten im Datenbaustein eingetroffen sind.
- 3) Wenn das IPC-Flag genutzt wird, muss dieses zurückgesetzt werden, um neue Daten erhalten zu können.

## 5.4 Daten Holen mit RK512(R)

Die GET Funktionalität erlaubt es, Daten von einem Kommunikationspartner (Server) zu holen, ohne diesen programmieren zu müssen. Er muss lediglich konfiguriert werden.

Das Protokoll handelt alle notwendigen Operationen ab. Es muss nur der SFB14 ausgeführt werden, um Daten zu bekommen.

### 5.4.1 Synchronisation

Da die Partner-PCD (Server) nicht programmiert werden muss, sind alle Leseoperationen der Daten absolut transparent für dieses System. Dieses kann zu Problemen führen, wenn eine Synchronisation der beiden Systeme erforderlich ist. Um dieses Problem zu vermeiden, ist auf der Serverseite eine Funktion im Protokoll verfügbar. Diese Funktion ist das **IPC Flag** (Inter-Processor Communication Flag), welches anzeigt, dass Daten gelesen wurden.

Dieses IPC-Flag befindet sich auf der Serverseite. Es muss aber auf der Leserseite (der die Daten holt) durch den Parameter R\_ID konfiguriert werden. Das IPC-Flag hat zwei Funktionen.

- 1) Anzuzeigen, dass Daten gelesen wurden. (Das Bit ist gesetzt)
- 2) Zu vermeiden, dass alten Daten gelesen werden.

Ist das IPC-Flag beim Server einmal durch das Lesen von Daten gesetzt, ist es nicht möglich, erneut Daten zu lesen, bis das vorherige Flag zurückgesetzt wurde.

### 5.4.2 Parameter SFB 14

Parameter	Beschreibung
REQ	Positive Flanke startet das Holen von Daten
ID	Serielle COM Port Nummer der xx7 und Informationen über das IPC Flag (Siehe Kapitel 5.5.2)
NDR	Neue Daten fertig (New Data Ready). Wenn das Holen von Daten abgeschlossen ist, bleibt dieses Bit für einen Zyklus EINS.
ERROR	Fehler (ERROR) bei der Operation. Das Flag <b>ERROR</b> bleibt einen Zyklus lang EINS Error
STATUS	Fehlercode
ADDR_1	Quellbereich der internen Daten des Kommunikationspartners.
ADDR_2	Nicht verwendet
ADDR_3	Nicht verwendet
ADDR_4	Nicht verwendet
RD_1	Zielbereich der Daten. Diese Variable ist vom Typ ANY Pointer. Die Länge der Daten muss der des Quellbereichs entsprechen.
RD_2	Nicht verwendet
RD_3	Nicht verwendet
RD_4	Nicht verwendet

### 5.4.3 ID Parameter

Bit	Beschreibung
0...3	Serial COM Port der PCD.xx7
4...7	Bit Nummer des IPC Flags
8...15	Byte Nummer des IPC Flags (1..254)

### 5.4.4 Aufrufbeispiel

In diesem Beispiel wird der Merker M40.4 als IPC-Flag benutzt. Dazu wird der Parameter ID wie folgt konfiguriert: M40 => 28h, .4 => 4h, COM1 => 1h, daraus folgt zusammen => 2841h.

```
CALL SFB 14, DB14
  REQ :=M400.0           // Anfrage um neue Daten zu bekommen (GET)
  ID  :=W#16#2841       // COM Port =1 ; IPC Flag ist M40.4
  NDR :=M400.2         // Neue Daten fertig. (New Data Ready)
  ERROR :=M400.3       // Fehler während der Operation
  STATUS:=MW402        // Fehlercode
  ADDR_1:=P#DB200.DBX 0.0 BYTE 100 // Quellbereich der Daten
  ADDR_2:=
  ADDR_3:=
  ADDR_4:=
  RD_1 :=P#DB100.DBX 0.0 BYTE 100 // Zielbereich Daten
  RD_2 :=
  RD_3 :=
  RD_4 :=
```

### 5.4.5 Parameter Status

Wert	Fehler	Beschreibung
-11 (FFF5h)	1	Der aktuelle Modus des seriellen Ports ist nicht RK512
-10 (FFF6h)	1	Ungültiger Parameter ADDR_1
-2 (FFFEh)	1	COM Port ist nicht initialisiert
-1 (FFFFh)	1	Ungültige COM Port Nummer
0	0	OK
1	1	Kommunikationsprobleme (Verbindungsleitung überprüfen)
2	1	Negative Antwort des Kommunikationspartners
9	1	Kann auf Daten nicht zugreifen, da das IPC-Flag dieses blockiert
10	1	Ungültiger Parameter RD_1

## 5.5 Beispiele

Es wurde ein Beispiel in Step7 geschrieben, um zu zeigen, wie mit dem RK512 zu arbeiten ist. Dieses Beispiel kann unter der Internetadresse [www.sbc-support.ch](http://www.sbc-support.ch) herunter geladen werden. Der Dateiname ist DOC\_CP44.zip.

Diese Datei muss dearchiviert werden. Für die Durchführung mit dem SIMATIC Manager benutzen Sie das Menü  
-> Datei -> Dearchivieren

Im Projekt sind zwei Steuerungen vorhanden:

### **PCD\_SEND**

FC3: Dieser Funktion sendet Daten zur Mailbox des Kommunikationspartners (PCD\_RECEIVE) mittels RK512 Protokoll.

FC4 : Dieser FC4 holt Daten aus einem Db des Kommunikationspartners mittels RK512 Protokoll.

### **PCD\_RECEIVE**

FC3 : Diese Funktion wertet das IPC-Flag aus, indem sie die Anzahl Datenübertragungen zählt und das Flag zurücksetzt.

FC4 : Diese Funktion wertet das IPC-Flag aus, indem sie die Anzahl geholter Datenpakete zählt und das Flag zurücksetzt.

Um korrekt zu arbeiten, müssen die beiden COM Port 1 der 2 PCDs mit Hilfe eines PCD2.F120 Moduls verbunden werden.

Eine PCD sendet Daten im ausgewählten Modus und die andere PCD empfängt diese in demselben Modus.

Der RK512(R) Modus wird im Step7 Programm im Netzwerk 1 des OB100 ausgewählt.

Weiterhin sind in beiden PCDs FCs für den jeweiligen Modus:

FC1 : läuft im Modus "3964"

FC2 : läuft im Modus "3964(R)"

Alle FCs sind in der gleichen Art strukturiert.

#### Sende PCD (SEND\_PCD):

Netzwerk 1: Lädt die Länge der zu sendenden Telegrammrahmen

Netzwerk 2: Zusammensetzung des Parameters R\_ID mit dem IPC-Flag

Netzwerk 3: Aufruf des SFB12 (Senden)

Netzwerk 4: Warten von 2000 Zyklen zwischen jedem SENDEN (Nur für dieses Beispiel)

Network 5: Zählen der gesendeten Telegrammrahmen und der Fehler

#### Empfangs PCD (RECEIVE\_PCD):

Netzwerk 1: Zählen der Übertragungen und Rücksetzen des IPC-Flags

Der einzige Unterschied zwischen dem RK512 und dem RK512(R) ist in der Konfiguration. Dies wird im OB100 gemacht.

Die Variablen-tabelle VAT1 zeigt, wie viele Telegramme gesendet und empfangen wurden. Weiterhin wird angezeigt, wie viele Fehler aufgetreten sind.



## 6 RK512 (Multi-Point Kommunikation)

---

### Einführung

Die RK512(R) steuert die Datenübertragung normalerweise mittels Punkt zu Punkt Verbindung zwischen einer PCD Serie xx7 und einem Kommunikationspartner. Aber es ist auch möglich, eine Multi-Point Kommunikation (Mehrpunkt Kommunikation) mit mehr als 2 PCD.xx7 Steuerungen via RS485 Schnittstellen zu realisieren.

Dieses Multi-Point Netzwerk basiert auf Master-Slave Kommunikation, in dem nur ein Master erlaubt ist. Der Master schreibt Daten zum Slave-System, oder fordert diese von ihm an.

### Wie arbeitet die Multi-Point Kommunikation?

Sie benutzt eine spezielle Funktion des DUARTs, Multi-DROP Modus genannt. Der Multi-DROP Modus ist eine andere Art, das Paritätsbit zu nutzen. Genau genommen wird das Paritätsbit nicht mehr als solches benutzt, sondern jetzt neu als Flag für die Slaveadresse. Das heisst, wenn der Master eine Kommunikation zu einem Slave öffnet, sendet es die Adresse des jeweiligen Slave und setzt das Paritätsbit auf NULL. Für alle anderen Daten wird das Paritätsbit auf EINS gesetzt.

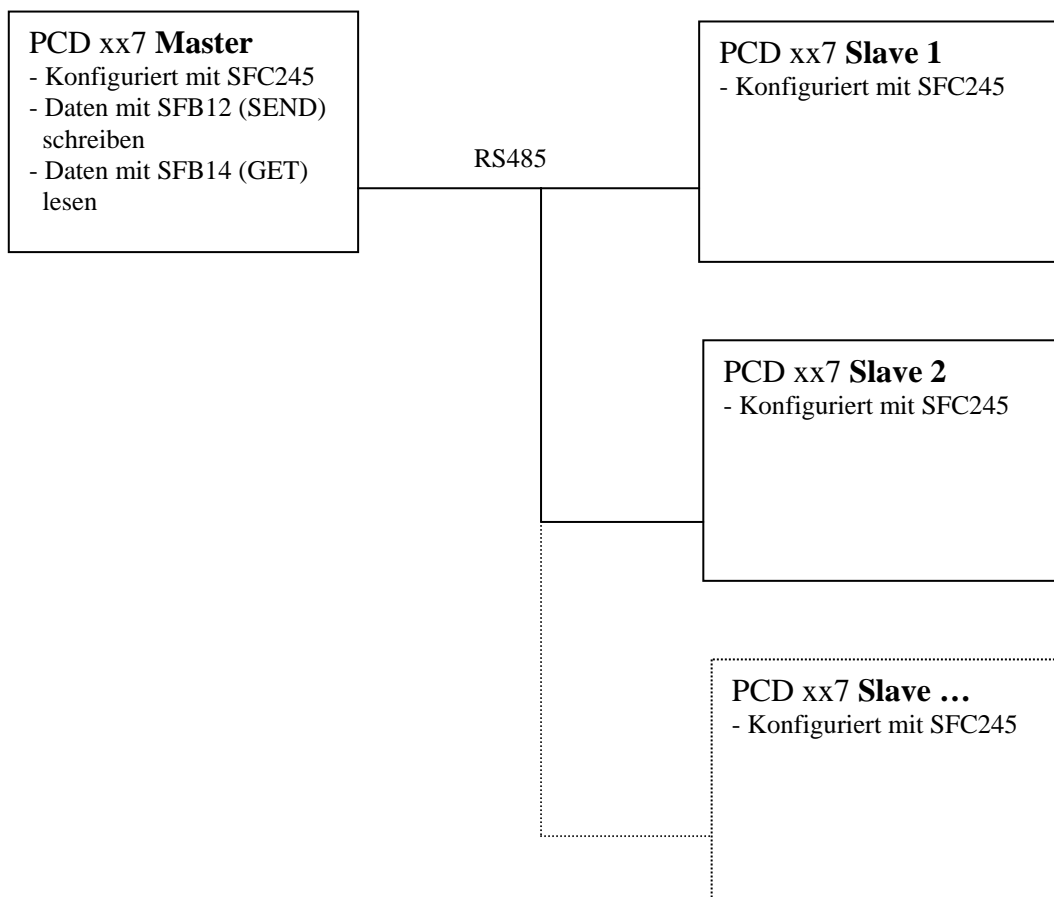
### Funktionalität

Im RK512 Multi-Point Modus können vom Master nur 2 Funktionen genutzt werden. Diese Funktionen sind die SFBs SEND and GET.

- SEND (SFB12): erlaubt es dem Master, in einen Datenbaustein eines beliebigen Slaves zu schreiben.
- GET (SFB14): erlaubt es dem Master, aus einen Datenbaustein eines beliebigen Slaves zu lesen.

Auf der PCD2.M487 und der PCD3.Mxxx7 wird dieser Mode nicht unterstützt!

## Struktur



Dieses Multi-Point Netzwerk basiert auf RS485 Topologie. (Details bezüglich korrekter Verdrahtung können dem Handbuch 26/740 entnommen werden. Es ist nur **ein** Master im System erlaubt.

Die Master-Steuerung wird dadurch als Master konfiguriert, in dem der SFC245 mit dem richtigen Parameter (DKPriority=EINS) ausgeführt wird.  
Die Slave-Steuerungen werden dadurch als Slave konfiguriert, in dem der SFC245 mit dem richtigen Parameter (DKPriority=NULL) und der richtigen Adresse (IN25= Eigene Adresse) ausgeführt werden.

Ist dies geschehen, ist der Master die einzige Steuerung, die für den Datenaustausch programmiert werden muss. Das heisst, der Master schreibt (SEND) und liest (GET) Daten von einem Slave nach dem anderen. Es kann immer nur auf einen Slave zur Zeit zugegriffen werden und auch nur ein Lese- oder Schreibzugriff zur Zeit erfolgen.

## 6.1 Initialisierung der PCD.xx7 für das RK512(R) Multi-Point

Um eine serielle Schnittstelle der PCD Serie xx7 nutzen zu können, muss diese initialisiert und konfiguriert werden. Dies geschieht mit der SFC245. Um ein Protokoll auf einem seriellen Port einzuschalten, muss die SFC245 einmal im OB100 aufgerufen werden.

Diese SFC wird auch benutzt, um den COM Port für andere Kommunikationsprotokolle wie 3964 und ASCII zu konfigurieren. Dann werden einige Parameter des RK512 Protokolls nicht benötigt. Die Spalte „Ausgewählter Modus“ zeigt an, welche Parameter für das RK512 Protokoll benötigt werden.

Name	Para	Typ	Möglicher Wert	Ausgew. Modus		Bemerkung
				3	4	
Port	IN0	INT	1,3,5,6			COM Port Nummer (siehe Kapitel 2.1)
Mode	IN1	INT	9			4 RK 512 R (Multi-Point)
Baud Rate	IN2	DINT				Baudrate (siehe Kapitel 2.1)
Data Bit	IN3	INT	7...8			Anzahl Datenbits
Stop Bit	IN4	INT	1...2			Anzahl Stoppbits
Parity	IN5	INT	5			5 Multi-Point
Control	IN6	INT	1			Schnittstellentyp 1 RS485
XON	IN7	BYTE	0...FFh			Nicht verwendet
XOFF	IN8	BYTE	0...FFh			Nicht verwendet
WaitSend	IN9	WORD	0...FFFEh			Nicht verwendet
WaitInactiv	IN10	WORD	0...FFFEh			Nicht verwendet
TelCount	IN11	INT	1			Anzahl Rahmen im Puffer
Overwrite	IN12	BOOL	FALSE / TRUE			FALSE: Telegramm überschreibbar TRUE: Telegramm überschreibbar im Puffer, aber nur wenn TelCount = 1.
DelRxPuffer	IN13	BOOL	FALSE / TRUE			Nicht verwendet
DKPriority	IN14	BOOL	FALSE / TRUE			EINS, wenn dies der Master ist
ZVZ	IN15	WORD	0, 1...FFFEh			Zeichenverzugszeit (zwischen 2 Zeichen) in Schritten von 10ms, 0 = Vorgabewert von 220ms
QVZ	IN16	WORD	0, 1...FFFEh			Quittungsverzugszeit (warten auf Antwort) in Schritten von 10ms, 0 = Vorgabewert von 550 ms für das RK512 und 2000 ms für das RK512 R .
TryToConnect	IN17	INT	0...255			Anzahl der Wiederholungsversuche um erneut mit dem Partner wieder Verbindung aufzunehmen, 0 = Vorgabewert von 6.
TryToSend	IN18	INT	0...255			Anzahl der Wiederholungsversuche um erneut Daten zu senden, 0 = Vorgabewert von 6
FixedLen	IN19	INT	1...1024			Telegrammrahmenlänge beim Empfang
EndChar1	IN20	BYTE	0...255			Endzeichen 1
EndChar2	IN21	BYTE	0...255			Endzeichen 2
SENDBuffer	IN22	INT	0...32768			Sendepuffergröße (in Bytes)
RCVBuffer	IN23	INT	0...32768			Empfangspuffergröße (in Bytes)
Dummy_I0	IN24	INT	0			Nicht verwendet
Dummy_W1	IN25	WORD	0			Eigene Netzwerkadresse
Dummy_W2	IN26	WORD	0			Wird nur beim Master verwendet. Wenn die Slave-Adresse sich in einem DB befindet, ist dieser Parameter die DB Nummer
Dummy_DW1	IN27	DWORD	0			Pointer(Zeiger zu) der Slave-Adresse
RetVal	OUT	WORD				Operationsergebnis, siehe Kapitel 6.1.2

Nicht  
 verwendet
  Erforderlich
  Nur Master

### 6.1.1 Beispiel für die Master Konfiguration

In diesem Beispiel wird der COM Port 1 der PCD xx7 für das RK512R Protokoll konfiguriert. Die Baudrate ist 9600, 8 Datenbits 1 Stoppbit, keine Parität. Die Schnittstelle ist RS232. Die Zeichenverzugszeit (ZVZ) ist fest auf 200ms. Die Quittungsverzugszeit (QVZ) steht auf dem Vorgabewert von 2000ms. Sende- und Empfangspuffer sind 4000 Bytes gross. Diese Station ist der Master im Netzwerk. Die Anzahl Wiederholungsversuche für erneuten Verbindungsaufbau und erneutes Senden steht auf dem Vorgabewert 6. Dies muss nur einmalig, z.B. im OB100, ausgeführt werden.

```

L      P#M 20.0           // Slave-Adresse (Ziel=Destination)
T      #DestPointer      // ist das Merkerbyte MB20

L      0                 // Slave-Adresse ist in MB20 nicht
T      #DBNumber         // in einem DB, also DB-Nummer=0

CALL   SFC 245
      IN0  :=1           // Serielles Port Nr.1
      IN1  :=4           // RK512
      IN2  :=L#9600      // Baudrate
      IN3  :=8           // Datenbits
      IN4  :=1           // Stoppbits
      IN5  :=5           // Multi-Point Kommunikation(mit IPC)
      IN6  :=1           // RS485 Schnittstelle
      IN7  :=B#16#0      // Nicht verwendet
      IN8  :=B#16#0      // Nicht verwendet
      IN9  :=W#16#0      // Nicht verwendet
      IN10 :=W#16#0      // Nicht verwendet
      IN11 :=1           // TelCount
      IN12 :=FALSE       // Rahmen nicht überschreibbar
      IN13 :=FALSE       // Nicht verwendet
      IN14 :=TRUE        // Diese Steuerung ist Master
      IN15 :=W#16#14     // ZVZ (zwischen Zeichen) = 200 ms
      IN16 :=W#16#0      // QVZ (für Antwort)Default =2000ms
      IN17 :=0           // Verbindungsversuche, Default = 6
      IN18 :=0           // Sendeversuche, Default = 6
      IN19 :=0           // Nicht verwendet
      IN20 :=B#16#0      // Nicht verwendet
      IN21 :=B#16#0      // Nicht verwendet
      IN22 :=4000        // 4000 Bytes für den Sendepuffer
      IN23 :=4000        // 4000 Bytes für den Empfangspuffer
      IN24 :=0           // Nicht verwendet
      IN25 :=W#16#0      // Nicht verwendet
      IN26 :=#DBnumber   // DB Nummer, mit Slave-Adresse
      IN27 :=#destpointer // Pointer der Slave-Adresse
RET_VAL:=MW240          // Rückgabewert

```

### 6.1.2 Beispiel für die Slave Konfiguration

Die Kommunikationseinstellungen sind die gleichen, wie bei Master (Kapitel 6.1.3), ausser dass er als Slave mit der Adresse **4** konfiguriert wird.

Dies muss nur einmalig, z.B. im OB100, ausgeführt werden.

```

L      4                // Vorgabe
T      #myadr           // Adresse 4

L      P#M 20.0        // Nicht relevant für den Slave
T      #DestPointer    // Muss nur ein Pointer sein

L      0                // Nicht relevant für den Slave
T      #DBNumber

CALL  SFC  245
      IN0   :=1          // Serielles Port Nr.1
      IN1   :=4          // RK512
      IN2   :=L#9600    // Baudrate
      IN3   :=8          // Datenbits
      IN4   :=1          // Stoppbits
      IN5   :=5          // Multi-Point Kommunikation(mit IPC)
      IN6   :=1          // RS485 Schnittstelle
      IN7   :=B#16#0    // Nicht verwendet
      IN8   :=B#16#0    // Nicht verwendet
      IN9   :=W#16#0    // Nicht verwendet
      IN10  :=W#16#0    // Nicht verwendet
      IN11  :=1          // TelCount
      IN12  :=FALSE     // Rahmen nicht überschreibbar
      IN13  :=FALSE     // Nicht verwendet
      IN14  :=FALSE     // Diese Steuerung ist SLAVE
      IN15  :=W#16#14   // ZVZ (zwischen Zeichen) = 200 ms
      IN16  :=W#16#0    // QVZ (für Antwort)Default =2000ms
      IN17  :=0          // Verbindungsversuche, Default = 6
      IN18  :=0          // Sendeversuche, Default = 6
      IN19  :=0          // Nicht verwendet
      IN20  :=B#16#0    // Nicht verwendet
      IN21  :=B#16#0    // Nicht verwendet
      IN22  :=4000       // 4000 Bytes für den Sendepuffer
      IN23  :=4000       // 4000 Bytes für den Empfangspuffer
      IN24  :=0          // Nicht verwendet
      IN25  :=#myadr     // Eigene Adresse im Netzwerk
      IN26  :=#DBnumber  // DB Nummer, mit Slave-Adresse
      IN27  :=#destpointer // Pointer der Slave-Adresse
      RET_VAL:=MW240     // Rückgabewert

```

### 6.1.3 Rückgabewerte der SFC245

Wert	Beschreibung
0	Initialisierung war erfolgreich
-1	Ungültige Schnittstellennummer
-2	Nicht genügend S7-Speicher, um die Puffer zu erzeugen. Evtl. Komprimierung durchführen.
-3	Nicht genügend S7-Speicher, um die Puffer zu erzeugen (nach durchgeführter Komprimierung).
-4	Ungültiger Wert für den Parameter MODE
-5	Ungültige Schnittstellenparameter (Baudrate, Datenbit, Stoppbit oder Parität)
-6	Ungültiger Wert für Parameter WAIT_SEND oder WAIT_INACTIVITY
-7	Ungültiger Wert für Parameter TEL_COUNT
-8	Ungültiger Wert für Parameter ZVZ oder QVZ
-9	Ungültiger Wert für Parameter TRY_CONNECT oder TRY_SEND
-10	ASCII – Feste Länge: Die Telegrammlänge ist länger als der Empfangspuffergrösse
-11	Ungültiger Wert für Parameter in SEND_BUFFER oder RCV_BUFFER
-12	Die Gesamtspeichergrösse des Sende- und Empfangspuffer ist grösser als die maximal erlaubten 64kB.
-13	Die SFC wurde mit einer unterschiedlichen Summe aus Empfangs- und Sendepuffer aufgerufen als beim letzten Aufruf.
-14	Eigene Adresse ist ungültig
-15	DB Nummer für den Slave ist ungültig
-16	Ungültiger Pointer Typ
-17	Ungültiger Offset im Pointer

### Bemerkung

## 6.2 Daten Senden mit RK512(R) Multi-Point

Der SFB12(SEND) sendet in der gleichen Art Daten, wie im RK512 Protokoll. Da dies hier eine Multi-Point Konfiguration ist, muss festgelegt werden, mit welcher Steuerung eine Verbindung aufgebaut werden soll. Um dies zu tun, muss nur der Integerwert der Slave-Adresse in das für diesen Zweck ausgewählte Speicherbyte (durch die SFC245, Parameter IN27) geschrieben werden. Diese Adresse muss in Betracht gezogen werden, sobald der SFB12 aufgerufen wird. Der Master sendet dann die Daten direkt in den Zielspeicherbereich des Slaves. Dort ist keine weitere Programmierung erforderlich. Es muss lediglich das COM Port mit der SFC245 initialisiert werden.

### 6.2.1 Parameter SFB 12

Parameter	Beschreibung
REQ	Mit einer positiven Flanke wird die SENDE Prozedur gestartet
R	Mit einer positiven Flanke wird abgebrochen und das Senden zurückgesetzt.
ID	Serielle COM Port Nummer..
R_ID	Datenzielbereich (Mailbox) und IPC-Flag Spezifikation
DONE	SENDE Prozedur ist abgeschlossen (DONE). Das Flag <b>DONE</b> bleibt einen Zyklus lang EINS.
ERROR	Fehler (ERROR) in der Sende Prozedur. Das Flag <b>ERROR</b> bleibt einen Zyklus lang EINS.
STATUS	Fehlercode
SD_1	Datenbereich, in den gesendet wird. Dieser Parameter ist ein ANY Pointer, aber die Länge des ANY Pointer wird hier nicht berücksichtigt. Diese wird von Parameter LEN übernommen.
LEN	Es können maximal 4'000 Bytes auf einmal gesendet werden. Natürlich muss die Sendepuffergrösse gleich oder grösser deklariert sein.

### 6.2.2 R\_ID Parameter:

Byte	Beschreibung	
<b>Byte 0</b> { Bit 0...3 }	0 → Ziel DX	1 → Ziel DB
<b>Byte 0</b> { Bit 4...7 }	Nicht verwendet (0H)	Bit-Nummer des IPC-Flag
<b>Byte 1</b>	Nicht verwendet (00H)	Byte-Nummer des IPC-Flag (zwischen 1..254)
<b>Byte 2</b>	Nicht verwendet (00H)	Offset 0..255 (Wort)
<b>Byte 3</b>	Nicht verwendet (00H)	DB-Nummer (1..255)

#### Byte 0: DX oder DB

Der DX-Modus wird für Kommunikationspartner benutzt, die das Mailbox-System nicht unterstützen. Der gesamte R\_ID = 0h.

Der DB-Modus benötigt einen Kommunikationspartner, der das Mailbox-System unterstützt.

#### IPC-Flag (Inter-Processor Communication Flag)

Dieses IPC-Flag befindet sich beim Kommunikationspartner. Das IPC-Flag hat zwei Funktionen.

- 1) Anzuzeigen, dass sich neue Daten in der Mailbox befinden. (Das Bit ist gesetzt)

2) Zu vermeiden, dass Daten in der Mailbox überschrieben werden.

Ist das IPC-Flag beim Empfänger einmal durch das Senden von Daten an ihn gesetzt, ist es nicht möglich, erneut Daten dahin zu senden, bis das vorherige Flag zurück gesetzt wurde. Weitere Information sind in Kapitel 5.2.1 beschrieben.

### Byte 2 and 3: Ziel-Offset und DB-Nummer

Hier wird der festgelegt, wo sich die Mailbox des Kommunikationspartners befindet. Die Mailbox muss ein Datenbaustein sein.

Byte 2 spezifiziert den Offset innerhalb des Datenbausteins in Worten (Wert zwischen 1...255)

Byte 3 spezifiziert die DB-Nummer (Wert zwischen 1...255)

### 6.2.3 Beispiel

Verbindung aufbauen und Daten zum Slave Nr. 4(=> MB20) senden.

In diesem Beispiel ist der R\_ID Parameter wie folgt konfiguriert:

- Sende zu einer Mailbox in DB33 ( 21h), mit dem Offset von 2 Worten (02h)
- Benutze als IPC-Flag den Merker M30.3, d.h. dies ist M30 (1Eh) , Bit 3 des Kommunikationspartners.

```

L      4           // SLAVE Adresse ; Ziel
T      MB      20  // wird ins Adressbyte geladen

CALL  SFB      12 , DB12
      REQ      :=M300.0
      R        :=M300.1
      ID       :=W#16#1
      R_ID     :=DW#16#311E0221
      DONE     :=M300.2
      ERROR    :=M300.3
      STATUS   :=MW302
      SD_1     :=P#DB100.DBX 0.0 BYTE 200
      LEN      :=MW304

```

### 6.2.4 Rückgabewerte STATUS

Wert	Beschreibung
-9 (FFF7h)	Telegramm kann nicht kopiert werden
-8 (FFF8h)	Länge ist zu gross
-7 (FFF9h)	Unbekannter Typ
-6 (FFFAh)	Ungültiger Zielbereich
-5 (FFFBh)	Ungültige Länge
-4 (FFFC h)	DB ist nicht geladen
-3 (FFFDh)	Ungültige DB Nummer
1	Kommunikationsprobleme (Verbindungsleitung überprüfen)
2	Funktion kann aufgrund des Kommunikationspartners nicht ausgeführt werden
5	Rücksetzen der Anforderung empfangen
9	Kann Daten nicht zum Partner schreiben, da das IPC-Flag dieses blockiert
10	Daten konnten wegen fehlendem Mailbox-Bereich nicht zum Partner geschrieben werden.
11	Warnung: Die Funktion kann nicht ausgeführt werden, da sie bereits ausgeführt wird.

#### **Bemerkung:**

Mit dieser Funktion können Sie Daten eines Kommunikationspartners direkt lesen und schreiben. Dies ist aber nur von DB zu DB möglich.



### 6.3 Daten Lesen (GET) mit RK512(R) Multi-Point

In dieser Multi-Point Konfiguration kann ein Slave keine Daten senden. Der Master holt die Daten vom Slave.

Zum Holen der Daten benutzt der Master den SFB14 (GET). Das Protokoll handelt alle notwendigen Operationen ab. Es muss nur der COM Port konfiguriert werden, damit das RK512 Multi-Point (Slave) Protokoll unterstützt wird. Bevor Daten vom Slave geholt werden können, muss die Verbindung geöffnet werden, d.h. es muss festgelegt werden, mit welchem Slave die Verbindung aufgebaut werden soll. Dies geschieht, in dem der Integerwert der Slave-Adresse in das für diesen Zweck ausgewählte Speicherbyte (durch die SFC245, Parameter IN27) geschrieben wird. Diese Adresse muss beachtet werden, sobald der SFB14 aufgerufen wird.

#### 6.3.1 Parameter SFB 14

Parameter	Beschreibung
REQ	Positive Flanke startet das Holen von Daten
ID	Serielle COM Port Nummer der xx7 und Informationen über das IPC Flag (Siehe Kapitel 5.5.2)
NDR	Neue Daten fertig (New Data Ready). Wenn das Holen von Daten abgeschlossen ist, bleibt dieses Bit für einen Zyklus EINS.
ERROR	Fehler (ERROR) bei der Operation. Das Flag <b>ERROR</b> bleibt einen Zyklus lang EINS Error
STATUS	Fehlercode
ADDR_1	Quellbereich der internen Daten des Kommunikationspartners.
ADDR_2	Nicht verwendet
ADDR_3	Nicht verwendet
ADDR_4	Nicht verwendet
RD_1	Zielbereich der Daten. Diese Variable ist vom Typ ANY Pointer. Die Länge der Daten muss der des Quellbereichs entsprechen.
RD_2	Nicht verwendet
RD_3	Nicht verwendet
RD_4	Nicht verwendet

#### 6.3.2 ID Parameter

Bit	Beschreibung
0...3	Serieller COM Port der PCD.xx7
4...7	Bit Nummer des IPC Flags
8...15	Byte Nummer des IPC Flags (1..254)

Weitere Details können dem Kapitel 5.4.1 entnommen werden.

### 6.3.3 Aufrufbeispiel

Dieses Beispiel zeigt eine Verbindung zur Slave-Adresse **3** (=> MB20) und das Lesen von Daten vom Slave. Merker M40.4 als IPC-Flag benutzt. Dazu wird der Parameter ID wie folgt konfiguriert: M40 => 28h, .4 => 4h, COM1 => 1h, daraus folgt zusammen => 2841h.

```
L      3           // Connect to SLAVE address = 3
T      MB      20  // MB20 ist durch SFC245 konfiguriert

CALL SFB 14, DB14
  REQ :=M400.0      // Anfrage um neue Daten zu bekommen (GET)
  ID  :=W#16#2841   // COM Port =1 ; IPC Flag ist M40.4
  NDR :=M400.2      // Neue Daten bekommen. (New Data Ready)
  ERROR :=M400.3    // Fehler während der Operation
  STATUS:=MW402     // Fehlercode
  ADDR_1:=P#DB200.DBX 0.0 BYTE 100 // Quellbereich der Daten
  ADDR_2:=
  ADDR_3:=
  ADDR_4:=
  RD_1 :=P#DB100.DBX 0.0 BYTE 100 // Zielbereich Daten
  RD_2 :=
  RD_3 :=
  RD_4 :=
```

### 6.3.4 Parameter Status

Wert	Fehler	Beschreibung
-11 (FFF5h)	1	Der aktuelle Modus des seriellen Ports ist nicht RK512
-10 (FFF6h)	1	Ungültiger Parameter ADDR_1
-2 (FFFEh)	1	COM Port ist nicht initialisiert
-1 (FFFFh)	1	Ungültige COM Port Nummer
0	0	OK
1	1	Kommunikationsprobleme (Verbindungsleitung überprüfen)
2	1	Negative Antwort des Kommunikationspartners
9	1	Kann auf Daten nicht zugreifen, da das IPC-Flag dieses blockiert
10	1	Ungültiger Parameter RD_1

## 6.4 Beispiele

Es wurde ein Beispiel in Step7 geschrieben, um zu zeigen, wie mit dem RK512 Protokoll im Multi-Point Modus zu arbeiten ist. Dieses Beispiel kann unter der Adresse [www.sbc-support.ch](http://www.sbc-support.ch) herunter geladen werden. Der Dateiname ist DOC\_CP44.zip.

Diese Datei muss dearchiviert werden. Für die Durchführung mit dem SIMATIC Manager benutzen Sie das Menü  
-> Datei -> Dearchivieren

Im Projekt sind drei Steuerungen vorhanden: Ein Master und zwei Slaves

### **PCD\_SEND (Master)**

FC9: Dieser Funktion holt (GET) Daten von den beiden Slave-Steuerungen (PCD\_RECEIVE) mittels RK512 Protokoll im Multi-Point Modus.

D.h. Daten von Slave-Adresse 2 holen(GET),dann 2000 Zyklen warten (optional), dann Daten von der Slave-Adresse 1 holen (GET), dann 200 Zyklen warten (optional) und dann wieder wie Slave-Adresse 1 beginnen.

FC10:Diese Funktion macht dasselbe wie FC9, aber sie benutzt das IPC-Flag nicht.

### **PCD\_RECEIVE\_or\_SLAVE1 (address =1)**

FC9 : Diese Funktion wertet das IPC-Flag aus, indem sie die Anzahl Datenübertragungen zählt und das Flag zurücksetzt.

### **SLAVE2 (address =2)**

FC9 : Diese Funktion wertet das IPC-Flag aus, indem sie die Anzahl Datenübertragungen zählt und das Flag zurücksetzt.

Die Master-Steuerung (PCD\_SEND) wählt einen Slave aus (in diesem Beispiel in dem die Adresse in MB20 geschrieben wird) und holt Daten von ihm.Auf der Gegenseite überprüft und zählt der Slave mit dem IPC-Flag, wenn die Daten von ihm geholt werden.

### Sende PCD (SEND\_PCD):

Netzwerk 1: Zusammensetzung des Parameters R\_ID mit IPC-Flag und COM-Nr.

Netzwerk 2: Aufruf des SFB14(GET)

Netzwerk 3: Warten von 2000 Zyklen zwischen jedem GET und dann die Slave Adresse wechseln.

Network 4: Zählen der gesendeten Telegrammrahmen und der Fehler

### Empfangs PCD (RECEIVE\_PCD\_SLAVE):

Netzwerk 1: Zählen der Übertragungen und Rücksetzen des IPC-Flags

Um korrekt zu arbeiten, müssen die COM Port 1 der 3 PCDs mit Hilfe eines PCD2.F110 Moduls verbunden werden.

Der RK512(R) Multi-Point Modus wird im Step7 Programm jeder Steuerung im Netzwerk 1 des OB100 ausgewählt. Für Multi-Point mit IPC-Flag muss der Modus 9 gewählt werden.

## 7 Transparent Modus

---

### Einführung

Der Transparent Modus verhält sich sehr ähnlich wie der Modus ASCII (Feste Länge), aber er besitzt mehr Flexibilität in Bezug auf die Länge der Telegramme. Das heisst, für jede Übertragung, oder den Empfang von Daten kann die Telegrammlänge neu definiert werden.

Eine weitere Eigenschaft ist, dass der Puffer transparenter zum Anwenderprogramm ist. Besonders der Empfangspuffer, der die Möglichkeit bietet, jederzeit die in ihm vorhandene Anzahl Bytes auszulesen.

Es gibt zwei Möglichkeiten, im Transparent Modus zu arbeiten. Beide haben ihre eigene Charakteristik.

◆ **SAIA-SFC (Nicht für PCD2.M487 und PCD3.Mxxx7):**

Dieser Modus ist der Default-Modus und kann mittels SFC240, 241, 242 und 243 genutzt werden. Es ist möglich, den Status und Überlauf des Empfangs- oder Sendepuffers abzufragen. Aber die maximale Länge eines Telegramms ist 128 Bytes, welches die Grösse des Empfangs, bzw. Sendepuffers ist.

◆ **CP-SFB:**

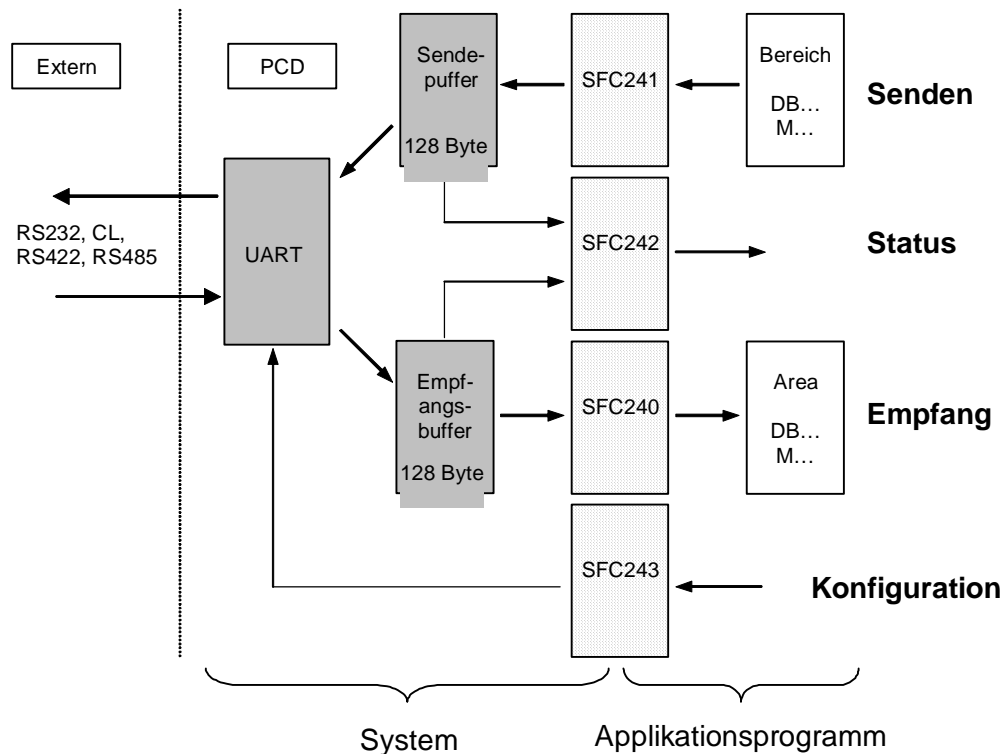
Dieser Modus muss mit der SFC245 konfiguriert werden. Es ist nur möglich den Status des Empfangspuffers abzufragen. Die maximale Länge eines Telegramms hängt von der konfigurierten Puffergrösse ab (bis zu 4kB).

### Warning

Es ist möglich, vom SAIA-SFC Modus zum CP-SFB Modus umzuschalten, aber das Zurückschalten nicht. D.h. nach dem Einschalten des CP-SFB Modus auf einem seriellen Port ist es nicht erlaubt, den SAIA-SFC Modus zu benutzen.

## 7.1 SAIA-SFC für den Ttransparent Modus

### Aufbau



Der UART (Universal Asynchronous Receiver Transmitter) ist die Schnittstelle zwischen der Übertragungsleitung und den Empfangs- oder Sendepuffern der PCDs. Daten werden zwischen UART und dem Empfangs- oder Sendepuffern mittels dem Systemprogramm übertragen. Datenübertragungen zwischen Empfangs- oder Sendepuffern und dem S7-Speicherbereich der PCDs wird innerhalb des Anwenderprogramms mit Hilfe spezieller Systemfunktionen (SFC) ausgeführt. Dieser Aufbau ist für jede Schnittstelle vorhanden.

Um dieses System dem Anwenderprogramm zugänglich zu machen, stehen 4 SFCs zu Verfügung:

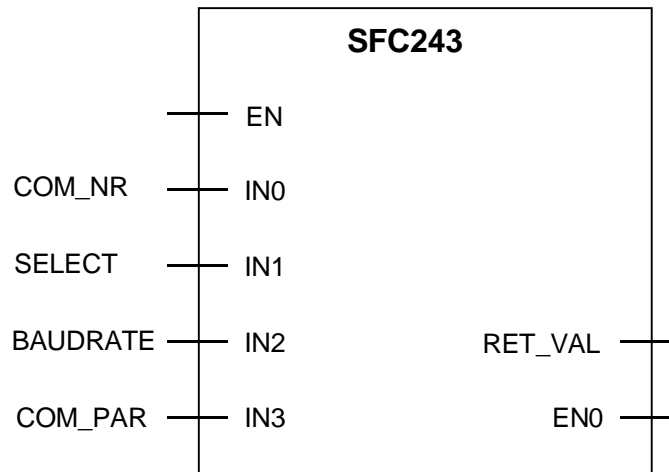
- SFC243      Konfiguration des seriellen Ports
- SFC242      Statusabfrage des seriellen Ports
- SFC241      Daten Senden
- SFC240      Daten Empfangen

Alle SFCs werden in den folgenden Kapiteln genauer beschrieben.

Diese 4 SFC werden auf der PCD2.M487 und den PCD3.Mxxx7 nicht unterstützt!

### 7.1.1 Konfigurieren und Initialisieren der seriellen Schnittstelle mit SFC 243 "COM\_INIT"

Wenn der "COM\_INIT" SFC aufgerufen wird, initialisiert dieser die ausgewählte serielle Schnittstelle.



#### Parameters

Parameter	Deklaration	Adressbereich	Beschreibung
COM_NR	IN / BYTE	E,A,M,D,L,Const.	Schnittstellenummer 1...5
SELECT	IN / BYTE	E,A,M,D,L,Const.	Schnittstellenmodus RS232 = 0 RS485 = 1 RS422 = 2 CL 20 mA = 3
BAUDRATE	IN / DINT	E,A,M,D,L,Const.	Baudrate
COM_PAR	IN / WORD	E,A,M,D,L,Const.	Initialisierungsparameter der Schnittstelle (siehe unten)
RET_VAL	OUT / WORD	E,A,M,D,L	Fehlerinformation

#### Fehlerinformation

Fehlercode (W#16#...)	Beschreibung
0000	Kein Fehler
00FE	Falsche Schnittstellenummer oder falsche Initialisierungswerte.

#### Bedeutung von "COM\_PAR"

Bits 1..0	Anzahl Datenbits (00 = 5, 01 = 6, 10 = 7, 11 = 8)
Bits 4..2	Parität (000 = gerade, 001 = ungerade, 010 = force low, 011 = force high, 10x =keine)
Bit 5	Stop bits, 0 => 1 stop bit, 1 => 2 stop bits

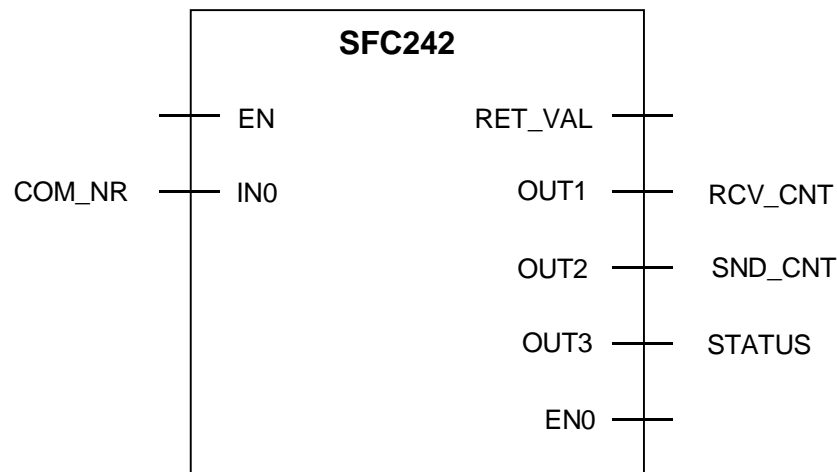
**Erlaubte Baudraten :** 300<sup>1</sup>, 600<sup>1</sup>, 1200, 2400, 4800, 9600, 19200<sup>2</sup>, 38400<sup>2</sup>)

<sup>1</sup>) Nicht auf PCD2.M487

<sup>2</sup>) 38400 nur für COM1, oder mit CDB für COM2..5 (Kapitel 2.1)

### 7.1.2 Status der seriellen Schnittstelle abfragen mit SFC 242 "COM\_STAT"

Wenn die SFC "COM\_STAT" aufgerufen wird, übergibt diese den aktuellen Status der seriellen Schnittstelle.



#### Parameter

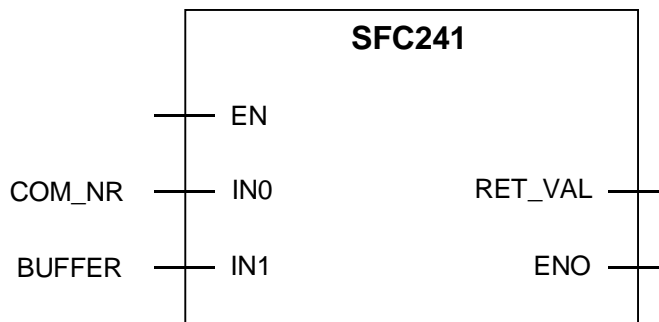
Parameter	Deklaration	Adressbereich	Beschreibung
COM_NR	IN / BYTE	E,A,M,D,L,Const.	Schnittstellenummer 1...5
RET_VAL	OUT / WORD	E,A,M,D,L	Fehlerinformation
RCV_CNT	OUT / WORD	E,A,M,D,L	Anzahl Bytes im Empfangspuffer
SND_CNT	OUT / WORD	E,A,M,D,L	Anzahl Bytes im Sendepuffer
STATUS	OUT / WORD	E;A;M;D;L	Status Bit 0 = 1 → Überlauf Empfangspuffer Bit 1 = 1 → Schnittstellenfehler

#### Fehlerinformation

Fehlercode (W#16#....)	Beschreibung
0000	Kein Fehler
00FE	Falsche Schnittstellenummer

### 7.1.3 Senden mit SFC 241 "COM\_SEND"

Beim Aufruf von "COM\_SEND" SFC wird die angegebene Anzahl Bytes vom ausgewählten Puffer in den Sendepuffer transferiert. Die eigentliche Übertragung findet im Hintergrund statt.



#### Parameters

Parameter	Deklaration	Adressbereich	Beschreibung
COM_NR	IN / BYTE	E,A,M,D,L,Const	Schnittstellenummer 1...5
BUFFER	IN / ANY		Zeiger zur Datenquellbereich (128 Byte)
RET_VAL	OUT / WORD	E,A,M,D,L	Fehlerinformation

#### Fehlerinformation

Fehlercode (W#16#....)	Beschreibung
0000	Kein Fehler
00FE	Falsche Schnittstellenummer
0001	Nicht genügend Platz im Sendepuffer

#### Beispiel:

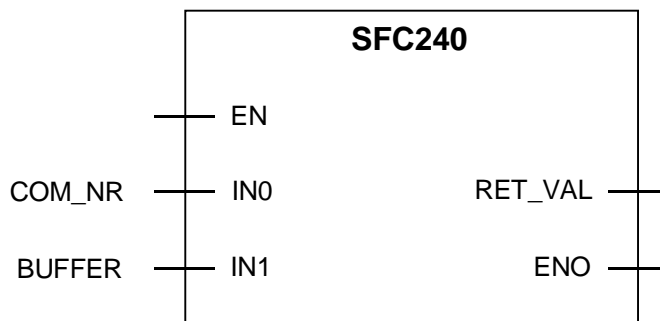
Senden von 50 Bytes des DB100 von DBB0 aus über COM 1. Rückgabewert in MW1040.

```
CALL SFC 241
  IN0      :=B#16#1
  IN1      :=P#DB100.DBX 0.0 BYTE 50
  RET_VAL :=MW1040
```



### 7.1.4 Empfangen mit SFC 240 "RCV\_COM"

Beim Aufruf von "RCV\_COM" SFC wird die angegebene Anzahl Bytes vom Empfangspuffer in den ausgewählten Puffer transferiert.



#### Parameters

Parameter	Deklaration	Adressbereich	Beschreibung
COM_NR	IN / BYTE	E,A,M,D,L,Const	Schnittstellenummer 1...5
BUFFER	IN / ANY		Zeiger zum Datenzielbereich (128 Byte)
RET_VAL	OUT / WORD	E,A,M,D,L	Fehlerinformation

#### Fehlerinformation

Fehlercode (W#16#....)	Beschreibung
0000	Kein Fehler
00FE	Falsche Schnittstellenummer
0001	Nicht genügend Platz im Empfangspuffer

#### Beispiel:

Transferiere 50 Bytes des Empfangspuffers der seriellen COM1 zum DB33 beginnend ab DBB0. Rückgabewert in MW240.

```
CALL SFC 240
  IN0      :=B#16#1
  IN1      :=P#DB33.DBX 0.0 BYTE 50
  RET_VAL :=MW240
```

### 7.1.5 Beispiel zu Kommunikation mit SAIA-SFC

Es wurde ein Beispiel in Step7 geschrieben, um zu zeigen, wie mit dem Transparent Modus zu arbeiten ist. Dieses Beispiel kann unter der Internetadresse [www.sbc-support.ch](http://www.sbc-support.ch) herunter geladen werden. Der Dateiname ist DOC\_CP44.zip.

Diese Datei muss dearchiviert werden. Für die Durchführung mit dem SIMATIC Manager benutzen Sie das Menü  
-> Datei -> Dearchivieren

Im Projekt sind zwei Steuerungen vorhanden:

**PCD\_SEND** : Diese PCD sendet 50 Zeichen auf einmal.  
**PCD\_RECEIVE** : Diese PCD empfängt diese Zeichen.

Um korrekt zu arbeiten, müssen die beiden COM Port 1 der 2 PCDs mit Hilfe eines PCD2.F120 Moduls verbunden werden.

Eine PCD sendet das Telegramm von 50 Zeichen und die andere PCD empfängt diese in ihrem Puffer und liest sie in einem mal.

Der Transparent Modus wird im Beispiel im Netzwerk 1 des OB100 ausgewählt.

Dann arbeiten beide Steuerungen durch Aufruf des FC240 im Transparent Modus.

Der FC240 ist wie folgt aufgebaut:

Sende PCD (SEND\_PCD):

Netzwerk 1: Bearbeitet den aktuellen Schrittstatus, SEND oder WAIT\_TOSEND.

Netzwerk 2: Aufruf der SFC241 (Senden)

Netzwerk 3: Warten von 2000 Zyklen bevor erneut gesendet wird (Nur für dieses Beispiel)

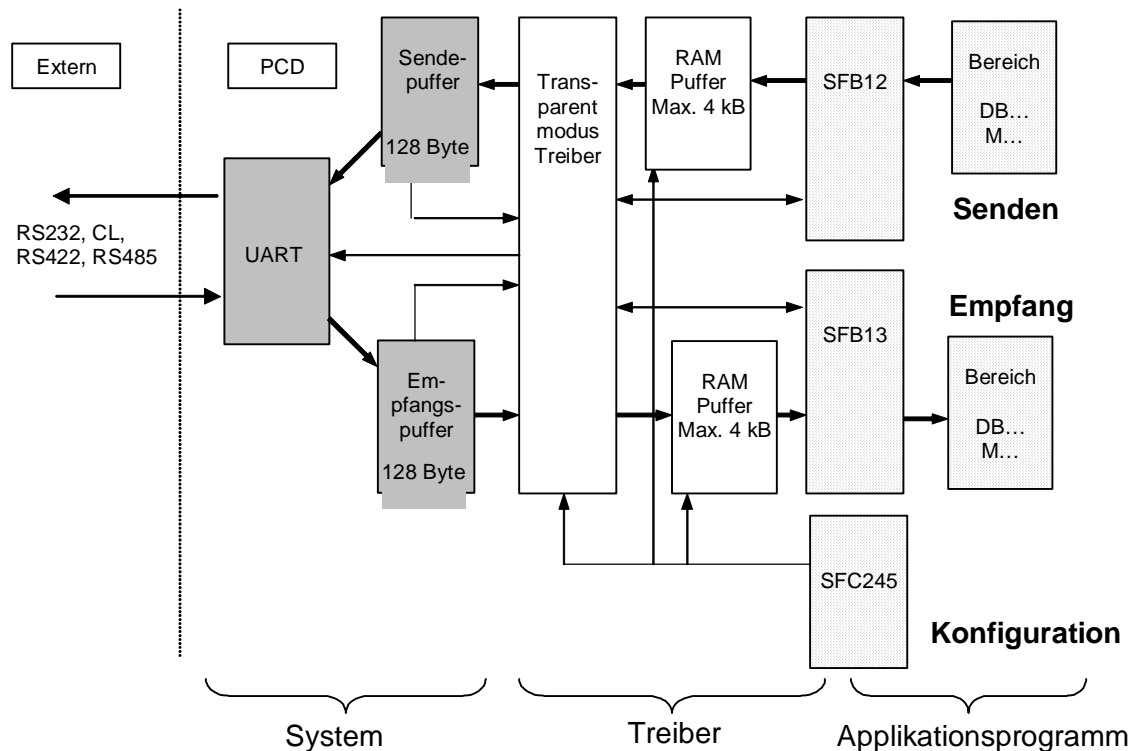
Empfangs PCD (RECEIVE\_PCD):

Netzwerk 1: Warten, bis 50 Bytes im Empfangspuffer eingetroffen sind und Auslesen dieser 50 Bytes. Weiterhin Zählen, wie oft der Empfangspuffer 50 Bytes enthält.

Dieses Beispiel zeigt lediglich wie die SFC benutzt werden kann. Es gibt weitere Möglichkeiten innerhalb des Kommunikationsaufbaus. Dies hängt von der jeweiligen Anwendung ab.

## 7.2 CP-SFB für den Transparent Modus

### Aufbau



Der UART (Universal Asynchronous Receiver Transmitter) ist die Schnittstelle zwischen der Übertragungsleitung und den Empfangs- oder Sendepuffern der PCDs. Daten werden zwischen UART und dem Empfangs- oder Sendepuffern mittels dem Systemprogramm übertragen. In diesem Transparent Modus ist eine zweite Schicht von Empfangs- und Sendepuffer vorhanden, deren Größe auf maximal 4kB eingestellt werden kann. Der Treiber des Transparent Modus sorgt für den Datentransfer zwischen den Puffern. Weiterhin gibt es zwei SFBs, die die Applikationssoftware und den Treiber verbinden. Dieser Aufbau ist für jede Schnittstelle vorhanden.

Um dieses System dem Anwenderprogramm zugänglich zu machen, stehen 2 Systemfunktionsbausteine und eine Systemfunktion (Konfiguration) zu Verfügung:

- SFC245 Konfiguration und Initialisierung des seriellen Ports
- SFB12 Daten senden
- SFB13 Daten empfangen

Alle Bausteine werden in den folgenden Kapiteln genauer beschrieben.

### 7.2.1 SFC245 Konfiguration und Initialisierung des seriellen COM Ports und des Treibers.

Um Transparent Modus nutzen zu können, muss die serielle Schnittstelle initialisiert und konfiguriert werden. Dies geschieht mit der SFC245. Um ein Protokoll auf einem seriellen Port einzuschalten, muss die SFC245 einmal aufgerufen werden.

Diese SFC wird auch benutzt, um den COM Port für andere Kommunikationsprotokolle wie RK512 und DK3964 zu konfigurieren. Dann werden einige Parameter des RK512 Protokolls nicht benötigt. Die Spalte „Ausgewählter Modus“ zeigt an, welche Parameter für den Transparent Modus benötigt werden.

Name	Para	Typ	Möglicher Wert	Bemerkung
Port	IN0	INT	0...6	COM Port Nummer (siehe Kapitel 2.1)
Mode	IN1	INT	0	0: Transparent Modus
Baud Rate	IN2	DINT		Baudrate (siehe Kapitel 2.1)
Data Bit	IN3	INT	7...8	Anzahl Datenbits
Stop Bit	IN4	INT	1...2	Anzahl Stoppbits
Parity	IN5	INT	0...4	0 Keine 1 Gerade 2 Ungerade 3 Force Low 4 Force High
Control	IN6	INT	0...3	Schnittstellentyp 0 RS 232 1 RS485 2 RS422 3 TTY
XON	IN7	BYTE	0...FFh	Nicht verwendet
XOFF	IN8	BYTE	0...FFh	Nicht verwendet
WaitSend	IN9	WORD	0...FFFEh	Nicht verwendet
WaitInactiv	IN10	WORD	0...FFFEh	Nicht verwendet
TelCount	IN11	INT	1	Anzahl Rahmen im Puffer
Overwrite	IN12	BOOL	FALSE / TRUE	FALSE: Telegramm überschreibbar TRUE: Telegramm überschreibbar im Puffer, aber nur wenn TelCount = 1.
DelRxPuffer	IN13	BOOL	FALSE / TRUE	Nicht verwendet
DKPriority	IN14	BOOL	FALSE / TRUE	Nicht verwendet
ZVZ	IN15	WORD	0, 1...FFFEh	Zeichenverzugszeit (zwischen 2 Zeichen) in Schritten von 10ms, 0 = Vorgabewert von 4ms
QVZ	IN16	WORD	0, 1...FFFEh	Nicht verwendet
TryToConnect	IN17	INT	0...255	Nicht verwendet
TryToSend	IN18	INT	0...255	Nicht verwendet
FixedLen	IN19	INT	1...1024	Telegrammrahmenlänge beim Empfang
EndChar1	IN20	BYTE	0...255	Endzeichen 1
EndChar2	IN21	BYTE	0...255	Endzeichen 2
SENDBuffer	IN22	INT	0...4000	Sendepuffergröße (in Bytes), abhängig vom der Telegrammlänge, die gesendet werden soll.
RCVBuffer	IN23	INT	0...4000	Empfangspuffergröße (in Bytes), abhängig vom der Telegrammlänge, die empfangen werden soll.
Dummy_I0	IN24	INT	0	Nicht verwendet
Dummy_W1	IN25	WORD	0	Nicht verwendet
Dummy_W2	IN26	WORD	0	Nicht verwendet
Dummy_DW1	IN27	DWORD	0	Nicht verwendet
RetVal	OUT	WORD		Operationsergebnis, siehe Kapitel 7.2.3

Nicht verwendet     Erforderlich

**Erlaubte Baudraten :** 300<sup>1</sup>, 600<sup>1</sup>, 1200, 2400, 4800, 9600, 19200<sup>2</sup>, 38400<sup>2</sup>)

<sup>1</sup>) Nicht auf PCD2.M487

<sup>2</sup>) 38400 nur für COM1, oder mit CDB für COM2..5 (Kapitel 2.1)

### 7.2.2 Beispiel

In diesem Beispiel wird der COM Port 1 der PCD xx7 für den Transparent Modus konfiguriert. Die Baudrate ist 9600, 8 Datenbits, 1 Stoppbit, keine Parität. Die Schnittstelle ist RS232. Die Grösse des Empfangs- und Sendepuffers ist 300 Bytes.

```
CALL SFC 245
      IN0      :=1          // Serielles Port Nr.1
      IN1      :=0          // Transparent
      IN2      :=L#9600    // Baudrate
      IN3      :=8          // Datenbits
      IN4      :=1          // Stoppbits
      IN5      :=0          // Keine Parität
      IN6      :=0          // RS232 Schnittstelle
      IN7      :=B#16#0    // Nicht verwendet
      IN8      :=B#16#0    // Nicht verwendet
      IN9      :=W#16#0    // Nicht verwendet
      IN10     :=W#16#0    // Nicht verwendet
      IN11     :=1          // TelCount
      IN12     :=FALSE     // Rahmen nicht überschreibbar
      IN13     :=FALSE     // Nicht verwendet
      IN14     :=FALSE     // Nicht verwendet
      IN15     :=W#16#0    // Nicht verwendet
      IN16     :=W#16#0    // Nicht verwendet
      IN17     :=0          // Nicht verwendet
      IN18     :=0          // Nicht verwendet
      IN19     :=0          // Nicht verwendet
      IN20     :=B#16#0    // Nicht verwendet
      IN21     :=B#16#0    // Nicht verwendet
      IN22     :=300       // 300 Bytes für den Sendepuffer
      IN23     :=300       // 300 Bytes für den Empfangspuffer
      IN24     :=0          // Nicht verwendet
      IN25     :=W#16#0    // Nicht verwendet
      IN26     :=W#16#0    // Nicht verwendet
      IN27     :=DW#16#0   // Nicht verwendet
      RET_VAL :=MW240      // Rückgabewert
```

### 7.2.3 Rückgabewerte der SFC245

Wert	Beschreibung
0	Initialisierung war erfolgreich
-1	Ungültige Schnittstellennummer
-2	Nicht genügend S7-Speicher, um die Puffer zu erzeugen. Evtl. Komprimierung durchführen.
-3	Nicht genügend S7-Speicher, um die Puffer zu erzeugen (nach durchgeführter Komprimierung)
-4	Ungültiger Wert für den Parameter MODE
-5	Ungültige Schnittstellenparameter (Baudrate, Datenbit, Stoppbit oder Parität)
-6	Ungültiger Wert für Parameter WAIT_SEND oder WAIT_INACTIVITY
-7	Ungültiger Wert für Parameter TEL_COUNT
-8	Ungültiger Wert für Parameter ZVZ oder QVZ
-9	Ungültiger Wert für Parameter TRY_CONNECT oder TRY_SEND
-10	ASCII – Feste Länge: Die Telegrammlänge ist länger als der Empfangspuffergrösse
-11	Ungültiger Wert für Parameter in SEND_BUFFER oder RCV_BUFFER
-12	Die Gesamtspeichergrösse des Sende- und Empfangspuffer ist grösser als die maximal erlaubten 64kB.
-13	Die SFC wurde mit einer unterschiedlichen Summe aus Empfangs- und Sendepuffer aufgerufen als beim letzten Aufruf.

#### **Bemerkung:**

Es ist möglich, den Schnittstellentreiber oder das Protokoll während der Ausführung umzuschalten, aber folgendes muss beachtet werden:

- 1) Die Puffergrösse kann nicht verändert werden
- 2) Die Sende- und Empfangsbaustein benötigen eine Flanke am Enable-Eingang, um den neuen Modus zu erkennen.

## 7.3 Daten senden

Um ein Datenpaket (Byte, Zeichen) zu senden, wird der SFB12 benötigt.

### 7.3.1 Parameter SFB 12

Parameter	Beschreibung
REQ	Mit einer positiven Flanke wird die SENDE Prozedur gestartet
R	Mit einer positiven Flanke wird abgebrochen und das Senden zurückgesetzt.
ID	Serielle COM Port Nummer..
R_ID	Nicht verwendet
DONE	SENDE Prozedur ist abgeschlossen (DONE). Das Flag <b>DONE</b> bleibt einen Zyklus lang EINS.
ERROR	Fehler (ERROR) in der Sende Prozedur. Das Flag <b>ERROR</b> bleibt einen Zyklus lang EINS.
STATUS	Fehlercode
SD_1	Datenbereich, in den gesendet wird. Dieser Parameter ist ein ANY Pointer, aber die Länge des ANY Pointer wird hier nicht berücksichtigt. Diese wird von Parameter LEN übernommen.
LEN	Es können maximal 4'000 Bytes auf einmal gesendet werden. Natürlich muss die Sendepuffergrösse gleich oder grösser deklariert sein.

Es sind einige weitere Bedingungen in Bezug auf die Status Maschine der Funktion in Kapitel 1.4 beschrieben.

### 7.3.2 SENDE Beispiel

```
CALL SFB 12 , DB12
      REQ :=M50.0           // Sendeanforderung
      R   :=M50.1           // Übertragung zurücksetzen
      ID  :=W#16#1          // COM Port Nummer
      R_ID :=               // Nicht verwendet
      DONE :=M100.0        // Übertragung ohne Fehler abgeschlossen
      ERROR :=M100.1        // Fehler bei der Übertragung
      STATUS:=MW102         // Fehlercode
      SD_1 :=P#DB100.DBX 0.0 BYTE 1000 // Datenquelle
      LEN  :=MW104          // Anzahl zu sendenden Bytes
```

### 7.3.3 Rückgabewerte STATUS

Wert	Beschreibung
-9 (FFF7h)	Telegramm kann nicht kopiert werden
-8 (FFF8h)	Länge ist zu gross
-7 (FFF9h)	Unbekannter Typ
-6 (FFFAh)	Ungültiger Zielbereich
-5 (FFFBh)	Ungültige Länge
-4 (FFFC h)	DB ist nicht geladen
-3 (FFFDh)	Ungültige DB Nummer
11	Warnung: Die Funktion kann nicht ausgeführt werden, da sie bereits ausgeführt wird.

## 7.4 Daten Empfangen

Um eingehende Daten vom seriellen Port mit dem Treiber des Transparent Modus zu lesen, wird der SFB13 benötigt.

### 7.4.1 Parameter SFB 13

Parameter	Beschreibung
EN_R	EINS = Empfang einschalten
ID	Serielle Port Nummer
R_ID	Nicht verwendet.
NDR	New Data Ready: Neue Daten sind angekommen und in den gewünschten Speicherbereich kopiert worden. <b>NDR</b> bleibt einen Zyklus lang EINS.
ERROR	Fehler beim Empfang. <b>ERROR</b> bleibt einen Zyklus lang EINS.
STATUS	Fehlercode
RD_1	Spezifiziert den Bereich, in den die eingehenden Daten gelegt werden sollen. Dieser Parameter ist ein ANY Pointer. Die Anzahl festgelegter Bytes in diesem Parameter muss gleich oder grösser als die Anzahl empfangener Bytes sein.
LEN	Dies ist ein IN/OUT Parameter, d.h. sein Wert kann auch durch den SFB modifiziert werden. Beim Lesen der Daten werden so viele Bytes ausgelesen, wie im Parameter stehen. Nach dem Vorgang steht dort die Anzahl verbleibender Bytes im Empfangspuffer.

Es sind einige weitere Bedingungen in Bezug auf die Status Maschine der Funktion in Kapitel 1.4 beschrieben.

### 7.4.2 Beispiel

```

L      200          // Anzahl Bytes, die gelesen werden sollen
T      MW204

CALL  SFB  13 , DB13
      EN_R  :=M200.0 // Empfangsfunktion Einschalten
      ID    :=W#16#1 // COM Port 1
      R_ID  :=      // Nicht verwendet
      NDR   :=M200.1 // New Data Ready flag (Neue Daten da)
      ERROR :=M200.2 // Fehler Flag
      STATUS:=MW202 // Fehlercode
      RD_1  :=P#DB200.DBX 0.0 BYTE 2000 // Daten Zielbereich
      LEN   :=MW204 // Anzahl empfangener Bytes

```

### 7.4.3 Parameter Status

Wert	Beschreibung
-7 (FFF9h)	Ungültiger Typ
-6 (FFFAh)	Ungültiger Zielbereich
-5 (FFFBh)	Ungültige Länge
-4 (FFFC h)	DB nicht geladen
-3 (FFFDh)	Ungültige DB Nummer
-2 (FFFEh)	COM Port nicht initialisiert
-1 (FFFFh)	Ungültige COM Port Nummer
0	OK
4	Länge zu gross im Vergleich zur Puffergrösse



## 7.5 Beispiele

Es wurde ein Beispiel in Step7 geschrieben, um zu zeigen, wie mit den 4 möglichen Treibern zu arbeiten ist. Dieses Beispiel kann unter der Internetadresse [www.sbc-support.ch](http://www.sbc-support.ch) herunter geladen werden. Der Dateiname ist DOC\_CP44.zip.

Diese Datei muss dearchiviert werden. Für die Durchführung mit dem SIMATIC Manager benutzen Sie das Menü  
-> Datei -> Dearchivieren

Im Projekt sind zwei Steuerungen vorhanden:

**PCD\_SEND** : Diese PCD sendet im Transparent-Modus  
**PCD\_RECEIVE** : Diese PCD empfängt diese Daten

Um korrekt zu arbeiten, müssen die beiden COM Port 1 der 2 PCDs mit Hilfe eines PCD2.F120 Moduls verbunden werden.

Eine PCD sendet Daten im Transparent-Modus und die andere PCD empfängt diese in demselben Modus.

Der ASCII-Modus wird im Step7 Programm im Netzwerk 1 des OB100 ausgewählt.

Weiterhin befindet sich in jeder PCD ein FC0. Dieser ist wie folgt aufgebaut:

### Sende PCD (SEND\_PCD):

Netzwerk 1: Lädt die Länge der zu sendenden Telegrammrahmen

Netzwerk 2: Aufruf des SFB12 (Senden)

Netzwerk 3: Warten von 2000 Zyklen zwischen jedem SENDEN (Nur für dieses Beispiel)

Netzwerk 4: Zählen der gesendeten Telegrammrahmen und der Fehler

### Empfangs PCD (RECEIVE\_PCD):

Netzwerk 1: Aufruf des SFB13 (Empfangen)

Netzwerk 2: Zählen der empfangen Telegrammrahmen und der Fehler

Die Variablentabelle VAT1 zeigt, wie viele Telegramme gesendet und empfangen wurden. Weiterhin wird angezeigt, wie viele Fehler aufgetreten sind.