FB61 - <offline>

"RECEIVE"

Nome: **Famiglia:**
Autore: **Versione:** 2.1
Versione blocco: 2
Data e ora Codice: 16/08/2011 12.40.33
Interfaccia: 30/03/2011 11.47.42
Lunghezze (blocco / codice / dati): 01864 01568 00042

Nome	Tipo di dati	Indirizzo	Valore iniziale	Commento
IN		0.0		
EN_R	Bool	0.0	FALSE	Freigabe für Daten lesen
R	Bool	0.1	FALSE	Synchron Reset auslösen
LADDR	Int	2.0	0	logische Basisadresse des CP - entspricht der Adresse aus der HW-Konfiguration
DB_NO	Int	4.0	0	Datenbausteinnummer - Nummer des Empfangs-DB (>0)
DBB_NO	Int	6.0	0	Datenbytenummer - Empfangsdaten ab Datenbyte
IO_SIZE	Word	8.0	W#16#0	parametrierte IO Größe des Moduls
OUT		0.0		
LEN	Int	10.0	0	Länge des empfangenen Telegramms in Byte (>0 und <1025)
NDR	Bool	12.0	FALSE	Auftrag fertig ohne Fehler, Daten empfangen, Parameter STATUS=00h
ERROR	Bool	12.1	FALSE	Auftrag fertig mit Fehler, Parameter STATUS enthält die Fehlerinformation
STATUS	Word	14.0	W#16#0	Spezifikation des Fehlers bei ERROR=1
IN_OUT		0.0		
CONTROL	Byte	16.0	B#16#0	geteilter Speicher: Sendebaustein Bit 0..3, Empfangsbaustein Bit 4-7
STAT		0.0		
bIntStatus	Byte	18.0	B#16#0	interner Bausteinstatus
xFpReset	Bool	19.0	FALSE	Hilfsbit zur Erkennung einer steigenden Flanke an R
xResetDone	Bool	19.1	FALSE	Fertigmeldung Synchron Reset (Statemachine Bit)
bFragNr	Byte	20.0	B#16#0	Fragmentnummer
iLen	Int	22.0	0	gelieferte Länge im Header bzw. Header/Trailer kombiniert
iLenRead	Int	24.0	0	Anzahl der bisher gelesenen Bytes
iTeleOffset	Int	26.0	0	geliefertes Telegramm Offset im Header bzw. Header/Trailer kombiniert
wReturnValue	Word	28.0	W#16#0	geliefertes Return Value im Header bzw. Header/Trailer kombiniert
xResetBecauseAbort	Bool	30.0	FALSE	Reset durch fallende Flanke an EN_R ausgelöst
iFragCounter	Int	32.0	0	Fragmentzähler
xWait	Bool	34.0	FALSE	warte nach fertigem Reset oder Auftrag mit Fehler auf FALS an EN_R und R
arInputData	Array [0..59] Of Byte	36.0		
arOutputData	Array [0..0] Of Byte	96.0		
TEMP		0.0		
wTemp1	Word	0.0		
wTemp2	Word	2.0		
dwTemp1	DWord	4.0		
dwTemp2	DWord	8.0		
anyScr	Any	12.0		
anyDst	Any	22.0		
iRetval	Int	32.0		
wLaddr	Word	34.0		

Blocco: FB61 RECEIVE

Histoy:
V1.0 erste Version
V1.2 Übergabe BLKMOV RetVal an Status
Reset auch bei fehlerhaften Eingangsparametern möglich
Ausgangsbits stehen nur für einen SPS-Zyklus an
V1.3 Korrektur Verarbeitung des empfangenen DataOffset
V2.0 Prozessdatenzugriff auf SFC14 und PAB umgestellt
V2.1 Korrektur im SEND-Baustein, keine Änderung RECEIVE-Baustein

Segmento: 1 INIT

```
Sprungmarken Definition:
   Ixxx INIT
   Rxxx SCHREIBE RESET
   E8xx Eingangsbyte=Fragmentnummer=0x08
   EAxx Eingangsbyte=Fragmentnummer=0x0A
   E9xx Eingangsbyte=Fragmentnummer=0x09
   Hxxx HEADER EMPFANGEN
   Exxx ENDE
Der Buchstabe kennzeichnet den Bausteinstatus/Netzwerk. Die x kennzeichnen die
Sprungmarkennummer in dezimaler Form.

Über das Byte "byIntStatus" wird der Bausteinstatus (Statemachine)
gekennzeichnet:
   0x00 LEERLAUF (lese Header/Trailer oder Header oder warte auf Daten)
   0x10 SCHREIBE RESET (Synchron Reset)
   0x20 HEADER EMPFANGEN (lese Fragment oder Trailer)
   0x30 AUFTRAG ABSCHLIEßEN
```

```
//*********************************************************************************************
// Eingangsdaten konsistent lesen
//*********************************************************************************************
      L     #LADDR                                                      #LADDR            -- logische Basis
                                                                        adresse des CP - entspricht der Adre
                                                                        sse aus der HW-Konfiguration
      T     #wLaddr                                                     #wLaddr

//  DSTBLK zusammenbauen
      LAR1  P##anyDst           // Any Pointer Anfangsadresse
      L     W#16#1002
      T     LW [AR1,P#0.0]      // Byte 0+1: Datentyp Byte
      L     #IO_SIZE                                                    #IO_SIZE          -- parametrierte
                                                                        IO Größe des Moduls
      T     LW [AR1,P#2.0]      // Byte 2+3: Länge
      L     DINO
      T     LW [AR1,P#4.0]      // Byte 4+5: DB-Nummer
      TAR2
      UD    DW#16#FFFFFF
      L     P##arInputData
      +D
      T     LD [AR1,P#6.0]      // DWord 6: Speicherbereich + Offset im Speicherbereich

// Eingangsdaten lesen
      CALL  "DPRD_DAT"                                                  SFC14             -- Read Consisten
                                                                        t Data of a Standard DP Slave
       LADDR  :=#wLaddr                                                 #wLaddr
       RET_VAL:=#iRetval                                                #iRetval
       RECORD :=#anyDst                                                 #anyDst

// RetVal auswerten
      L     #iRetval                                                    #iRetval
      L     0
      ==I
      SPB   S001
      L     #iRetval                                                    #iRetval
      T     #STATUS             // schreibe STATUS = iRetVal DPRD_DAT   #STATUS           -- Spezifikation
                                                                        des Fehlers bei ERROR=1
      SPA   E100
S001: NOP   0

//*********************************************************************************************
// CONTROL aktualisieren
//*********************************************************************************************

      L     #CONTROL                                                    #CONTROL          -- geteilter Spei
                                                                        cher: Sendebaustein Bit 0..3, Empfan
                                                                        gsbaustein Bit 4-7
      T     #arOutputData[0]                                            #arOutputData[0]


//*********************************************************************************************
// Auftrag/Reset mit Fehler oder abgeschlossener Reset -> Handling Ausgänge
//*********************************************************************************************

      O     #EN_R                                                       #EN_R             -- Freigabe für D
                                                                        aten lesen
      O     #R                                                          #R                -- Synchron Reset
                                                                         auslösen
      U     #ERROR                                                      #ERROR            -- Auftrag fertig
                                                                         mit Fehler, Parameter STATUS enthäl
                                                                        t die Fehlerinformation
      SPBN  FE00
      L     0
      T     #STATUS                                                     #STATUS           -- Spezifikation
                                                                        des Fehlers bei ERROR=1
```

```
      SET
      =     #xWait                                                  #xWait            -- warte nach fer
                                                                    tigem Reset oder Auftrag mit Fehler
                                                                    auf FALS an EN_R und R
FE00: NOP   0

      U     #xFpReset           // wenn Reset angestoßen            #xFpReset         -- Hilfsbit zur E
                                                                    rkennung einer steigenden Flanke an
                                                                    R
      U     #NDR                // und bearbeitet                   #NDR              -- Auftrag fertig
                                                                     ohne Fehler, Daten empfangen, Param
                                                                    eter STATUS=00h
      S     #xWait                                                  #xWait            -- warte nach fer
                                                                    tigem Reset oder Auftrag mit Fehler
                                                                    auf FALS an EN_R und R
      R     #NDR                                                    #NDR              -- Auftrag fertig
                                                                     ohne Fehler, Daten empfangen, Param
                                                                    eter STATUS=00h

      UN    #EN_R                                                   #EN_R             -- Freigabe für D
                                                                    aten lesen
      UN    #R                                                      #R                -- Synchron Reset
                                                                     auslösen
      R     #xWait                                                  #xWait            -- warte nach fer
                                                                    tigem Reset oder Auftrag mit Fehler
                                                                    auf FALS an EN_R und R

      U     #xWait                                                  #xWait            -- warte nach fer
                                                                    tigem Reset oder Auftrag mit Fehler
                                                                    auf FALS an EN_R und R
      SPB   E100


//*********************************************************************************************
// EN_R=0 während Datenempfang -> Synchron Reset anstoßen
//*********************************************************************************************

      O     #EN_R                                                   #EN_R             -- Freigabe für D
                                                                    aten lesen
      O     #NDR                                                    #NDR              -- Auftrag fertig
                                                                     ohne Fehler, Daten empfangen, Param
                                                                    eter STATUS=00h
      SPB   I000
      L     #bIntStatus                                             #bIntStatus       -- interner Baust
                                                                    einstatus
      L     B#16#0              // Int_Status = LEERLAUF?
      ==I
      SPB   I000
      SET
      =     #xResetBecauseAbort // fallende Flanke an EN_R erkannt  #xResetBecauseAbort -- Reset durch f
                                                                    allende Flanke an EN_R ausgelöst
I000: NOP   0


//*********************************************************************************************
// Aufruf mit EN_R=R=0 und keine fallende Flanke an EN_R -> Baustein verlassen
//*********************************************************************************************

      O     #EN_R                                                   #EN_R             -- Freigabe für D
                                                                    aten lesen
      O     #R                                                      #R                -- Synchron Reset
                                                                     auslösen
      O     #xResetBecauseAbort                                     #xResetBecauseAbort -- Reset durch f
                                                                    allende Flanke an EN_R ausgelöst
      SPB   I010
// Ausgangsparameter (Ausgangsdaten bleiben unverändert) und statische Variablen zurücksetzen
      CLR
      =     #NDR                // NDR                              #NDR              -- Auftrag fertig
                                                                     ohne Fehler, Daten empfangen, Param
                                                                    eter STATUS=00h
      =     #ERROR              // ERROR                            #ERROR            -- Auftrag fertig
                                                                     mit Fehler, Parameter STATUS enthäl
                                                                    t die Fehlerinformation
      =     #xFpReset                                               #xFpReset         -- Hilfsbit zur E
                                                                    rkennung einer steigenden Flanke an
                                                                    R
      =     #xResetDone                                             #xResetDone       -- Fertigmeldung
                                                                    Synchron Reset (Statemachine Bit)
      =     #xResetBecauseAbort                                     #xResetBecauseAbort -- Reset durch f
                                                                    allende Flanke an EN_R ausgelöst
      L     0
      T     #LEN                // LEN                              #LEN              -- Länge des empf
                                                                    angenen Telegramms in Byte (>0 und <
                                                                    1025)
      T     #STATUS             // STATUS                           #STATUS           -- Spezifikation
                                                                    des Fehlers bei ERROR=1
      T     #bIntStatus                                             #bIntStatus       -- interner Baust
                                                                    einstatus
```

```
      T     #bFragNr                                                  #bFragNr          -- Fragmentnummer
      T     #iLen                                                     #iLen             -- gelieferte Län
                                                                      ge im Header bzw. Header/Trailer kom
                                                                      biniert
      T     #iLenRead                                                 #iLenRead         -- Anzahl der bis
                                                                      her gelesenen Bytes
      T     #iTeleOffset                                              #iTeleOffset      -- geliefertes Te
                                                                      legramm Offset im Header bzw. Header
                                                                      /Trailer kombiniert
      T     #wReturnValue                                             #wReturnValue     -- geliefertes Re
                                                                      turn Value im Header bzw. Header/Tra
                                                                      iler kombiniert
      T     #iFragCounter                                             #iFragCounter     -- Fragmentzähler

//wenn kein Auftrag ansteht -> Idle Zustand an CP melden
      L     #CONTROL                                                  #CONTROL          -- geteilter Spei
                                                                      cher: Sendebaustein Bit 0..3, Empfan
                                                                      gsbaustein Bit 4-7
      L     B#16#F
      UW
      L     B#16#80
      OW
      T     #arOutputData[0]                                          #arOutputData[0]
      SPA   E200                  // Baustein verlassen
I010: NOP   0



//*********************************************************************************************
// Eingangsparameter prüfen
//*********************************************************************************************

// Reset auch bei ungültigen Eingangsparametern möglich
      U     #R                                                        #R                -- Synchron Reset
                                                                       auslösen
      SPB   I022

// DB_NO prüfen
      L     #DB_NO                                                    #DB_NO            -- Datenbausteinn
                                                                      ummer - Nummer des Empfangs-DB (>0)
      L     0
      <>I                         // Fehler bei DB_NO=0
      SPB   I020
      L     W#16#301
      T     #STATUS               // schreibe STATUS 0x0301           #STATUS           -- Spezifikation
                                                                      des Fehlers bei ERROR=1
      SET
      =     #ERROR                // ERROR setzen                     #ERROR            -- Auftrag fertig
                                                                       mit Fehler, Parameter STATUS enthäl
                                                                      t die Fehlerinformation
      SPA   E200                  // Baustein verlassen
I020: NOP   0
// IO_SIZE prüfen
      L     #IO_SIZE                                                  #IO_SIZE          -- parametrierte
                                                                      IO Größe des Moduls
      L     0
      ==I                         // Fehler bei IO_SIZE=0
      SPB   I021
      L     #IO_SIZE                                                  #IO_SIZE          -- parametrierte
                                                                      IO Größe des Moduls
      L     60
      >I                          // Fehler bei IO_SIZE>60
      SPBN  I022
I021: NOP   0
      L     W#16#202
      T     #STATUS               // schreibe STATUS 0x0202           #STATUS           -- Spezifikation
                                                                      des Fehlers bei ERROR=1
      SET
      =     #ERROR                // ERROR setzen                     #ERROR            -- Auftrag fertig
                                                                       mit Fehler, Parameter STATUS enthäl
                                                                      t die Fehlerinformation
      SPA   E200                  // Baustein verlassen
I022: NOP   0
// keine Fehler an den Eingangsparametern -> ERROR und STATUS zurücksetzen
      L     W#16#0
      T     #STATUS               // schreibe STATUS 0x0000           #STATUS           -- Spezifikation
                                                                      des Fehlers bei ERROR=1
      CLR
      =     #ERROR                // ERROR zurücksetzen               #ERROR            -- Auftrag fertig
                                                                       mit Fehler, Parameter STATUS enthäl
                                                                      t die Fehlerinformation



//*********************************************************************************************
// Aufruf mit R=1 (Reset immer möglich, unabhängig von EN_R)
//*********************************************************************************************

      O     #R                    // R gesetzt?                       #R                -- Synchron Reset
                                                                       auslösen
```

```
      O     #xResetBecauseAbort  // oder Reset durch fallende Flanke an   #xResetBecauseAbort -- Reset durch f
                                  EN_R?                                   allende Flanke an EN_R ausgelöst
      SPBN  I030                 // nein -> Flankenmerker R zurücksetzen
      U     #xFpReset            // Synchon Reset bereits gesartet?       #xFpReset          -- Hilfsbit zur E
                                                                          rkennung einer steigenden Flanke an
                                                                          R
      SPB   I031                 // ja -> Reset bearbeiten
// neuen Reset anstoßen
      L     B#16#10              // Int_Status = SCHREIBE RESET
      T     #bIntStatus                                                  #bIntStatus        -- interner Baust
                                                                          einstatus
      SET
      =     #xFpReset                                                    #xFpReset          -- Hilfsbit zur E
                                                                          rkennung einer steigenden Flanke an
                                                                          R
      SPA   I031
I030: NOP   0
      CLR
      =     #xFpReset            // Flankenmerker R zurücksetzen         #xFpReset          -- Hilfsbit zur E
                                                                          rkennung einer steigenden Flanke an
                                                                          R
I031: NOP   0

      U     #xFpReset            // wenn Reset angestoßen                #xFpReset          -- Hilfsbit zur E
                                                                          rkennung einer steigenden Flanke an
                                                                          R
      U     #NDR                 // und bearbeitet                       #NDR               -- Auftrag fertig
                                                                           ohne Fehler, Daten empfangen, Param
                                                                          eter STATUS=00h
      SPB   E100                 // --> Bausteinende


//*********************************************************************************************
// Aufruf mit EN_R=1 (Empfang nur mit R=0 möglich)
//*********************************************************************************************

      U     #xFpReset            // Reset angestoßen?                    #xFpReset          -- Hilfsbit zur E
                                                                          rkennung einer steigenden Flanke an
                                                                          R
      SPB   I040                 // ja -> Synchron Reset bearbeiten
      UN    #EN_R                // EN_R gesetzt?                        #EN_R              -- Freigabe für D
                                                                          aten lesen
      SPB   E200                 // nein -> Baustein verlassen
      U     #NDR                 // Auftrag abgeschlossen?               #NDR               -- Auftrag fertig
                                                                           ohne Fehler, Daten empfangen, Param
                                                                          eter STATUS=00h
      SPBN  I040                 // nein -> Auftrag bearbeiten
// Auftrag bearbeitet -> Ausgangsparameter zurücksetzen
      L     0
      T     #LEN                 // LEN                                  #LEN               -- Länge des empf
                                                                          angenen Telegramms in Byte (>0 und <
                                                                          1025)
      T     #STATUS              // STATUS                               #STATUS            -- Spezifikation
                                                                          des Fehlers bei ERROR=1
      T     #iLen                                                        #iLen              -- gelieferte Läng
                                                                          e im Header bzw. Header/Trailer kom
                                                                          biniert
      T     #iLenRead                                                    #iLenRead          -- Anzahl der bis
                                                                          her gelesenen Bytes
      T     #bIntStatus          // Int_Status = LEERLAUF                #bIntStatus        -- interner Baust
                                                                          einstatus
      T     #iTeleOffset                                                 #iTeleOffset       -- geliefertes Te
                                                                          legramm Offset im Header bzw. Header
                                                                          /Trailer kombiniert
      T     #iFragCounter                                                #iFragCounter      -- Fragmentzähler
      CLR
      =     #NDR                                                         #NDR               -- Auftrag fertig
                                                                           ohne Fehler, Daten empfangen, Param
                                                                          eter STATUS=00h
I040: NOP   0


//*********************************************************************************************
// Fragmentnummer bestimmen
//*********************************************************************************************

// logische Basisadresse in das Adressregister schreiben
      L     #arInputData[0]                                              #arInputData[0]
      SLW   12
      SRW   12
      T     #bFragNr             // Fragmentnummer = Bit 0..3 von Einga  #bFragNr           -- Fragmentnummer
                                 ngsbyte 0


//*********************************************************************************************
// Bausteinstatus / Eingangsbyte=Fragmentnummer auswerten
//*********************************************************************************************
```

```
// Int_Status = SCHREIBE RESET
      L     B#16#10
      L     #bIntStatus                                                 #bIntStatus        -- interner Baust
                                                                        einstatus
      ==I
      SPB   R000
// Int_Status = Eingangsbyte=Fragmentnummer=0x08
      L     #bFragNr                                                    #bFragNr           -- Fragmentnummer
      L     B#16#8
      ==I
      SPB   E800
// Int_Status = Eingangsbyte=Fragmentnummer=0x0A
      L     #bFragNr                                                    #bFragNr           -- Fragmentnummer
      L     B#16#A
      ==I
      SPB   EA00
// Int_Status = Eingangsbyte=Fragmentnummer=0x09
      L     #bFragNr                                                    #bFragNr           -- Fragmentnummer
      L     B#16#9
      ==I
      SPB   E900
// Int_Status = HEADER EMPFANGEN
      L     B#16#20
      L     #bIntStatus                                                 #bIntStatus        -- interner Baust
                                                                        einstatus
      ==I
      SPB   H000
// Int_Status nicht definiert
      L     W#16#407
      T     #STATUS              // schreibe STATUS 0x0407             #STATUS            -- Spezifikation
                                                                        des Fehlers bei ERROR=1
      SPA   E200                 // Baustein verlassen
```

Segmento: 2 SCHREIBE RESET

```
R000: NOP   0


//*********************************************************************************************
// "Reset abschließen" prüfen
//*********************************************************************************************

      U     #xResetDone                                                 #xResetDone        -- Fertigmeldung
                                                                        Synchron Reset (Statemachine Bit)
      SPB   R010                 // Fragmentnummer 0x08 prüfen
      SPA   R020                 // Fragmentnummer 0x0C prüfen


R010: NOP   0


//*********************************************************************************************
// Fragmentnummer 0x08 prüfen
//*********************************************************************************************

      L     #bFragNr                                                    #bFragNr           -- Fragmentnummer
      L     B#16#8
      <>I
      SPB   R011
      L     B#16#0               // Int_Status = Leerlauf
      T     #bIntStatus                                                 #bIntStatus        -- interner Baust
                                                                        einstatus
      CLR
      =     #xResetDone                                                 #xResetDone        -- Fertigmeldung
                                                                        Synchron Reset (Statemachine Bit)
      =     #xResetBecauseAbort                                         #xResetBecauseAbort -- Reset durch f
                                                                        allende Flanke an EN_R ausgelöst
      SET
      =     #NDR                                                        #NDR               -- Auftrag fertig
                                                                         ohne Fehler, Daten empfangen, Param
                                                                        eter STATUS=00h
R011: NOP   0
      SPA   E200                 // Baustein verlassen


R020: NOP   0


//*********************************************************************************************
// Fragmentnummer 0x0C prüfen
```

```
//*****************************************************************************************

      L     #bFragNr                                                    #bFragNr           -- Fragmentnummer
      L     B#16#C
      <>I
      SPB   R021
// Ausgangsbyte 0 = CONTROL
      L     #CONTROL                                                    #CONTROL           -- geteilter Spei
                                                                        cher: Sendebaustein Bit 0..3, Empfan
                                                                        gsbaustein Bit 4-7
      L     B#16#F
      UW
      L     B#16#80             // Synchron Reset quittieren
      OW
      T     #arOutputData[0]                                            #arOutputData[0]
      SET
      =     #xResetDone         // "Reset abschließen" setzen           #xResetDone        -- Fertigmeldung
                                                                        Synchron Reset (Statemachine Bit)
      SPA   R022
R021: NOP   0
// Ausgangsbyte 0 = CONTROL
      L     #CONTROL                                                    #CONTROL           -- geteilter Spei
                                                                        cher: Sendebaustein Bit 0..3, Empfan
                                                                        gsbaustein Bit 4-7
      L     B#16#F
      UW
      L     B#16#B0             // Synchron Reset Quittierung anfordern
      OW
      T     #arOutputData[0]                                            #arOutputData[0]
R022: NOP   0
      SPA   E200                // Baustein verlassen
```

Segmento: 3 Eingangsbyte=Fragmentnummer=0x08

```
E800: NOP   0


//*****************************************************************************************
// Prüfung ob Leerlauf oder Auftrag abschließen
//*****************************************************************************************

      L     B#16#30         // Int_Status = AUFTRAG ABSCHLIEßEN?
      L     #bIntStatus                                               #bIntStatus        -- interner Baustein
                                                                      status
      ==I
      SPB   E810            // ja -> Auftrag abschließen
      SPA   E100            // nein -> Ausgangsbyte 0 schreiben



E810: NOP   0



//*****************************************************************************************
// Auftrag abschließen
//*****************************************************************************************

// Ausgangsparameter für einen SPS-Zyklus setzen
      L     #iLen                                                     #iLen              -- gelieferte Länge
                                                                      im Header bzw. Header/Trailer kombinier
                                                                      t
      T     #LEN            // LEN = Länge aus Header Daten           #LEN               -- Länge des empfang
                                                                      enen Telegramms in Byte (>0 und <1025)
      L     #wReturnValue                                             #wReturnValue      -- geliefertes Retur
                                                                      n Value im Header bzw. Header/Trailer k
                                                                      ombiniert
      T     #STATUS         // STATUS = Return Value aus Header Daten #STATUS            -- Spezifikation des
                                                                       Fehlers bei ERROR=1
      SET
      =     #NDR                                                      #NDR               -- Auftrag fertig oh
                                                                      ne Fehler, Daten empfangen, Parameter S
                                                                      TATUS=00h
      SPA   E100            // Ausgangsbyte 0 schreiben
```

```
Segmento: 4      Eingangsbyte=Fragmentnummer=0x0A

EA00: NOP   0



//*********************************************************************************************
// Prüfung ob Trailer oder Header/Trailer kombiniert oder bereits bearbeitet
//*********************************************************************************************

      L     B#16#30          // Int_Status = AUFTRAG ABSCHLIEßEN?
      L     #bIntStatus                                                #bIntStatus        -- interner Baustei
                                                                       nstatus
      ==I
      SPB   E200             // ja -> Baustein verlassen
      L     B#16#20          // Int_Status = HEADER EMPFANGEN?
      L     #bIntStatus                                                #bIntStatus        -- interner Baustei
                                                                       nstatus
      ==I
      SPB   EA10             // lese Trailer
      SPA   EA20             // lese Header/Trailer kombiniert



EA10: NOP   0



//*********************************************************************************************
// lese Trailer
//*********************************************************************************************

// SRCBLK / DSTBLK zusammenbauen
      L     8                // 1 Byte Header
      T     #dwTemp1         // Byte 8+9: Offset im Eingangsdatenbere   #dwTemp1
                             ich
      L     #iLen                                                      #iLen              -- gelieferte Länge
                                                                        im Header bzw. Header/Trailer kombini
                                                                       ert
      L     #iLenRead                                                  #iLenRead          -- Anzahl der bishe
                                                                       r gelesenen Bytes
      -I
      T     #wTemp1          // Byte 2+3: Länge im Eingangsdatenberei   #wTemp1
                             ch
      T     #wTemp2          // Byte 2+3: Länge im Empfangs-DB          #wTemp2
      L     #DBB_NO                                                    #DBB_NO            -- Datenbytenummer
                                                                       - Empfangsdaten ab Datenbyte
      L     #iLenRead                                                  #iLenRead          -- Anzahl der bishe
                                                                       r gelesenen Bytes
      +D
      SLD   3
      T     #dwTemp2         // Byte 8+9: Offset im Empfangs-DB         #dwTemp2
      SPA   EA50             // Auftragsabschluss anstoßen



EA20: NOP   0



//*********************************************************************************************
// lese Header/Trailer kombiniert
//*********************************************************************************************

// Eingangsbyte 2+3 = Anzahl Datenbytes + 2
      L     #arInputData[2]                                            #arInputData[2]
      SLW   8
      L     #arInputData[3]                                            #arInputData[3]
      OW
      L     2
      -I
      T     #iLen                                                      #iLen              -- gelieferte Länge
                                                                        im Header bzw. Header/Trailer kombini
                                                                       ert
// Eingangsbyte 1 = prüfe ob Telegramm mit oder ohne Offset
      L     #arInputData[1]                                            #arInputData[1]
      L     B#16#4           // Offset vorhanden?
      ==I
      SPB   EA21             // ja
      L     #arInputData[1]                                            #arInputData[1]
      L     B#16#0           // Offset<>0?
      <>I
      SPB   E200             // ja -> ungültiger Offset -> Baustein verlassen
// Eingangsbyte 4+5 = Return Value
      L     #arInputData[4]                                            #arInputData[4]
      SLW   8
      L     #arInputData[5]                                            #arInputData[5]
```

```
      OW
      T     #wReturnValue                                          #wReturnValue      -- geliefertes Retu
                                                                   rn Value im Header bzw. Header/Trailer
                                                                    kombiniert
      SPA   EA22
EA21: NOP   0
// Eingangsbyte 4+5 = Telegramm Offset
      L     2               // Nutzdaten sind um 2 Byte nach hinten verschoben bei Offset<>0
      T     #iTeleOffset                                           #iTeleOffset       -- geliefertes Tele
                                                                   gramm Offset im Header bzw. Header/Tra
                                                                   iler kombiniert
// Eingangsbyte 6+7 = Return Value
      L     #arInputData[6]                                        #arInputData[6]
      SLW   8
      L     #arInputData[7]                                        #arInputData[7]
      OW
      T     #wReturnValue                                          #wReturnValue      -- geliefertes Retu
                                                                   rn Value im Header bzw. Header/Trailer
                                                                    kombiniert
EA22: NOP   0
// SRCBLK / DSTBLK zusammenbauen
      L     6               // 6 Byte Header
      L     #iTeleOffset    // + mögliches Offset definiert durch Ei   #iTeleOffset       -- geliefertes Tele
                            ngangsdaten                            gramm Offset im Header bzw. Header/Tra
                                                                   iler kombiniert
      +D
      SLD   3
      T     #dwTemp1        // Byte 8+9: Offset im Eingangsdatenbere   #dwTemp1
                            ich
      L     #iLen                                                  #iLen              -- gelieferte Länge
                                                                    im Header bzw. Header/Trailer kombini
                                                                   ert
      T     #wTemp1         // Byte 2+3: Länge im Eingangsdatenberei   #wTemp1
                            ch
      T     #wTemp2         // Byte 2+3: Länge im Empfangs-DB          #wTemp2
      L     #arInputData[4] // Telegrammoffset                       #arInputData[4]
      SLW   8
      L     #arInputData[5]                                        #arInputData[5]
      OW
      L     #DBB_NO                                                #DBB_NO            -- Datenbytenummer
                                                                   - Empfangsdaten ab Datenbyte
      +I
      L     #iLenRead                                              #iLenRead          -- Anzahl der bishe
                                                                   r gelesenen Bytes
      +D
      SLD   3
      T     #dwTemp2        // Byte 7+8+9: Offset im Empfangs-DB       #dwTemp2
      SPA   EA50            // Auftragsabschluss anstoßen


EA50: NOP   0


//*********************************************************************************************
// Auftragsabschluss anstoßen
//*********************************************************************************************

      L     B#16#30         // Int_Status = AUFTRAG ABSCHLIEßEN
      T     #bIntStatus                                            #bIntStatus        -- interner Baustei
                                                                   nstatus
      SPA   E000            // Daten kopieren
```

Segmento: 5 Eingangsbyte=Fragmentnummer=0x09

```
E900: NOP   0


//*********************************************************************************************
// Prüfung ob Header bereits gelesen wurde
//*********************************************************************************************

      L     B#16#20         // Int_Status = HEADER EMPFANGEN?
      L     #bIntStatus                                            #bIntStatus        -- interner Baustei
                                                                   nstatus
      ==I
      SPB   E200            // ja -> Baustein verlassen


//*********************************************************************************************
// lese Header
//*********************************************************************************************
```

```
// Eingangsbyte 2+3 = Anzahl Datenbytes + 2
      L     #arInputData[2]                                          #arInputData[2]
      SLW   8
      L     #arInputData[3]                                          #arInputData[3]
      OW
      L     2
      -I
      T     #iLen                                                    #iLen              -- gelieferte Länge
                                                                      im Header bzw. Header/Trailer kombini
                                                                     ert
// Eingangsbyte 1 = prüfe ob Telegramm mit oder ohne Offset
      L     #arInputData[1]                                          #arInputData[1]
      L     B#16#4           // Offset vorhanden?
      ==I
      SPB   E921             // ja
      L     #arInputData[1]                                          #arInputData[1]
      L     B#16#0           // Offset<>0?
      <>I
      SPB   E900             // ja -> ungültiger Offset -> Baustein verlassen
// Eingangsbyte 4+5 = Return Value
      L     #arInputData[4]                                          #arInputData[4]
      SLW   8
      L     #arInputData[5]                                          #arInputData[5]
      OW
      T     #wReturnValue                                            #wReturnValue      -- geliefertes Retu
                                                                     rn Value im Header bzw. Header/Trailer
                                                                      kombiniert
      SPA   E922
E921: NOP   0
// Eingangsbyte 4+5 = Telegramm Offset
      L     2                // Nutzdaten sind um 2 Byte nach hinten verschoben bei Offset<>0
      T     #iTeleOffset                                             #iTeleOffset       -- geliefertes Tele
                                                                     gramm Offset im Header bzw. Header/Tra
                                                                     iler kombiniert
// Eingangsbyte 6+7 = Return Value
      L     #arInputData[6]                                          #arInputData[6]
      SLW   8
      L     #arInputData[7]                                          #arInputData[7]
      OW
      T     #wReturnValue                                            #wReturnValue      -- geliefertes Retu
                                                                     rn Value im Header bzw. Header/Trailer
                                                                      kombiniert
E922: NOP   0
// SRCBLK / DSTBLK zusammenbauen
      L     6                // 6 Byte Header
      L     #iTeleOffset     // + mögliches Offset definiert durch Ei   #iTeleOffset       -- geliefertes Tele
                             ngangsdaten                              gramm Offset im Header bzw. Header/Tra
                                                                     iler kombiniert
      +D
      T     #wTemp1          // = Offset                              #wTemp1
      SLD   3
      T     #dwTemp1         // Byte 8+9: Offset im Eingangsdatenbere   #dwTemp1
                             ich
      L     #IO_SIZE                                                 #IO_SIZE           -- parametrierte IO
                                                                      Größe des Moduls
      L     #wTemp1                                                  #wTemp1
      -I
      T     #wTemp1          // Byte 2+3: Länge im Eingangsdatenberei   #wTemp1
                             ch
      T     #wTemp2          // Byte 2+3: Länge im Empfangs-DB        #wTemp2
      L     #arInputData[4]  // Telegrammoffset                       #arInputData[4]
      SLW   8
      L     #arInputData[5]                                          #arInputData[5]
      OW
      L     #DBB_NO                                                  #DBB_NO            -- Datenbytenummer
                                                                     - Empfangsdaten ab Datenbyte
      +I
      L     #iLenRead                                                #iLenRead          -- Anzahl der bishe
                                                                     r gelesenen Bytes
      +D
      SLD   3
      T     #dwTemp2         // Byte 7+8+9: Offset im Empfangs-DB     #dwTemp2


//*****************************************************************************
// Auftragsabschluss anstoßen
//*****************************************************************************

      L     B#16#20          // Int_Status = HEADER EMPFANGEN
      T     #bIntStatus                                              #bIntStatus        -- interner Baustei
                                                                     nstatus
      SPA   E000             // Daten kopieren
```

Segmento: 6 HEADER EMPFANGEN

```
H000: NOP   0



//*********************************************************************************************
// Prüfung ob Fragment bereits gelesen wurde
//*********************************************************************************************

      L     #bFragNr                                                    #bFragNr           -- Fragmentnummer
      L     #iFragCounter                                               #iFragCounter      -- Fragmentzähler
      ==I
      SPB   E100           // Ausgangsbyte 0 schreiben

      L     #bFragNr                                                    #bFragNr           -- Fragmentnummer
      T     #iFragCounter                                               #iFragCounter      -- Fragmentzähler



//*********************************************************************************************
// lese Fragment
//*********************************************************************************************

// SRCBLK / DSTBLK zusammenbauen

      L     8              // 1 Byte Header
      T     #dwTemp1       // Byte 8+9: Offset im Eingangsdatenberei   #dwTemp1
                           ch
      L     #IO_SIZE       // Fragment hat immer die Größe IO_SIZE-1   #IO_SIZE           -- parametrierte IO
                                                                       Größe des Moduls
      L     1
      -I
      T     #wTemp1        // Byte 2+3: Länge im Eingangsdatenbereic   #wTemp1
                           h
      T     #wTemp2        // Byte 2+3: Länge im Empfangs-DB           #wTemp2
      L     #DBB_NO                                                    #DBB_NO            -- Datenbytenummer -
                                                                        Empfangsdaten ab Datenbyte
      L     #iLenRead                                                  #iLenRead          -- Anzahl der bisher
                                                                        gelesenen Bytes
      +D
      SLD   3
      T     #dwTemp2       // Byte 8+9: Offset im Empfangs-DB          #dwTemp2
      SPA   E000           // Daten kopieren
```

Segmento: 7 ENDE

```
E000: NOP   0



//*********************************************************************************************
// Daten kopieren
//*********************************************************************************************

//  SRCBLK zusammenbauen
      LAR1  P##anyScr         // Any Pointer Anfangsadresse
      L     W#16#1002
      T     LW [AR1,P#0.0]    // Byte 0+1: Datentyp Byte
      L     #wTemp1                                                    #wTemp1
      T     LW [AR1,P#2.0]    // Byte 2+3: Länge
      L     DINO
      T     LW [AR1,P#4.0]    // Byte 4+5: DB-Nummer
      TAR2
      UD    DW#16#FFFFFF
      L     P##arInputData
      +D
      L     #dwTemp1                                                   #dwTemp1
      +D
      T     LD [AR1,P#6.0]    // DWord 6: Speicherbereich + Offset im Speicherbereich

//  DSTBLK zusammenbauen
      LAR1  P##anyDst         // Any Pointer Anfangsadresse
      L     W#16#1002
      T     LW [AR1,P#0.0]    // Byte 0+1: Datentyp Byte
      L     #wTemp2                                                    #wTemp2
      T     LW [AR1,P#2.0]    // Byte 2+3: Länge
      L     #DB_NO                                                     #DB_NO             -- Datenbausteinnum
                                                                       mer - Nummer des Empfangs-DB (>0)
      T     LW [AR1,P#4.0]    // Byte 4+5: Empfangs-DB
      L     #dwTemp2                                                   #dwTemp2
      T     LD [AR1,P#6.0]    // Byte 7+8+9: Offset im Empfangs-DB
```

```
      L     B#16#84
      T     LB [AR1,P#6.0]    // Byte 6: Speicherbereich DB
// Eingangsdaten in Empfangs-DB umkopieren
      CALL  "BLKMOV"                                                SFC20             -- Copy Variables
       SRCBLK :=#anyScr                                             #anyScr
       RET_VAL:=#iRetval                                            #iRetval
       DSTBLK :=#anyDst                                             #anyDst
// RetVal auswerten
      L     #iRetval                                                #iRetval
      L     0
      ==I
      SPB   E001
      L     #iRetval                                                #iRetval
      T     #STATUS           // schreibe STATUS = iRetVal BLKMOV   #STATUS           -- Spezifikation de
                                                                    s Fehlers bei ERROR=1
E001: NOP   0

// Anzahl der bereits kopierten/empfangenen Datenbytes aktualisieren
      L     #iLenRead                                               #iLenRead         -- Anzahl der bishe
                                                                    r gelesenen Bytes
      L     #wTemp2           // Anzahl der in den DB geschriebenen Da #wTemp2
                              tenbytes
      +I
      T     #iLenRead                                               #iLenRead         -- Anzahl der bishe
                                                                    r gelesenen Bytes

E100: NOP   0



//*********************************************************************************************
// schreibe Ausgangsbyte 0
//*********************************************************************************************

//verknüpfe Controlbyte mit Fragmentnummer
      L     #CONTROL                                                #CONTROL          -- geteilter Speich
                                                                    er: Sendebaustein Bit 0..3, Empfangsba
                                                                    ustein Bit 4-7
      L     B#16#F
      UW
      L     #bFragNr                                                #bFragNr          -- Fragmentnummer
      SLW   4
      OW
      T     #arOutputData[0]  // Ausgangsbyte 0 (Bit 4..7)         #arOutputData[0]

E200: NOP   0



//*********************************************************************************************
// schreibe CONTROL
//*********************************************************************************************
      L     #LADDR                                                  #LADDR            -- logische Basisad
                                                                    resse des CP - entspricht der Adresse
                                                                    aus der HW-Konfiguration
      SLD   3
      LAR1

// schreibe CONTROL
      L     #arOutputData[0]  // CONTROL (Bit 4..7)                #arOutputData[0]
      T     #CONTROL                                                #CONTROL          -- geteilter Speich
                                                                    er: Sendebaustein Bit 0..3, Empfangsba
                                                                    ustein Bit 4-7
      T     PAB [AR1,P#0.0]

//*********************************************************************************************
// ERROR Bit steuern und auf BIE (Statuswort, Bit 8) abbilden
//*********************************************************************************************

// ERROR steuern
      L     #STATUS                                                 #STATUS           -- Spezifikation de
                                                                    s Fehlers bei ERROR=1
      L     B#16#0            // STATUS = 0x0000 -> kein Fehler
      ==I
      SPB   E110
      SET
      =     #ERROR            // STATUS = Fehler -> ERROR setzen   #ERROR            -- Auftrag fertig m
                                                                    it Fehler, Parameter STATUS enthält di
                                                                    e Fehlerinformation
      SPA   E111
E110: NOP   0
      CLR
      =     #ERROR            // STATUS = kein Fehler -> ERROR zurücks #ERROR        -- Auftrag fertig m
                              etzen                                 it Fehler, Parameter STATUS enthält di
                                                                    e Fehlerinformation
E111: NOP   0
// BIE steuern
      UN    #ERROR                                                  #ERROR            -- Auftrag fertig m
                                                                    it Fehler, Parameter STATUS enthält di
                                                                    e Fehlerinformation
```

```
      SAVE                    // BIE=VKE
```