

3 LON-Zugriff mit S7

Der Zugriff auf Daten des LON-Netzwerkes mittels der S7 Programmierungsumgebung wird durch die Verwendung der Information in verschiedenen Daten-Blöcken (DB) und mit einem 'System Function Call' (SFC) erreicht.

3.1 Data-Block Informationen

Der Data-Block kann ausschliesslich mit der SNET32-Software erzeugt, danach unter der S7 Programmierungsumgebung importiert und kompiliert werden.

Abhängig von der Konfigurierung, können bis 4 Data-Blocks (DB) in einem einzigen Projekt vorkommen. Diese DB haben aufeinanderfolgende Block-Adressen. Ist die Basis-Blockadresse = 500, wird

DB500 (Basisadresse)
Konfigurations-Daten

DB501 (Basisadresse +1)
Enthält die NV (Network Variable). Dieser DB existiert nur, wenn mehrere NV deklariert wurden.

DB502 (Basisadresse +2)
Enthält die 'Explicit message'. Dieser DB existiert nur, wenn mehrere 'Explicit messages' deklariert wurden.

DB503 (Basisadresse +3)
Enthält die Diagnose-Flags für die NV und die 'Explicit message'

3.1.1 Konfigurations Data-Block (Basisadresse)

Dieser DB enthält alle Daten, welche für die Initialisierung des LON-Interfaces (PCD7.F80x) notwendig sind. Die Daten, welche in diesem DB liegen, dürfen vom Anwender in keinem Fall verändert werden.

3.1.2 'Network Variable' DB (Basisadresse +1)

Dieser DB enthält alle NV (Network Variable).

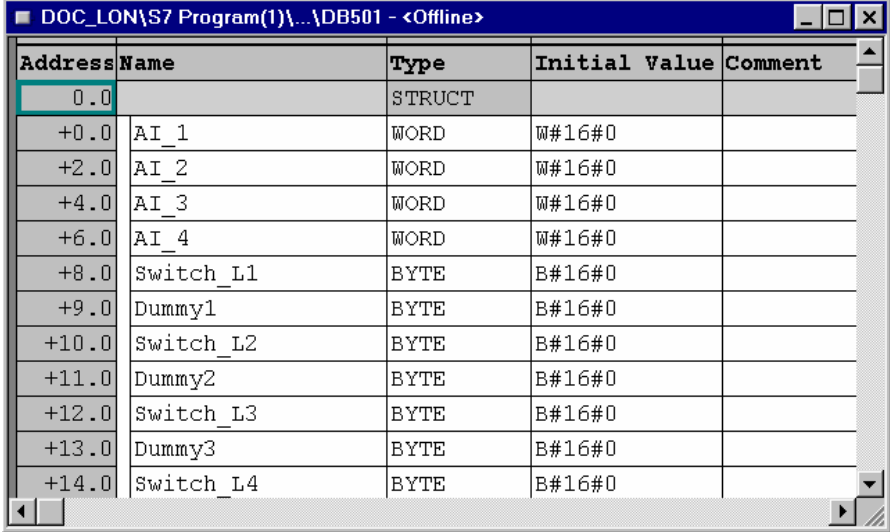
Die NV werden durch die Namen identifiziert, welche bei der Konfiguration mit den SNET32 Werkzeugen angegeben wurden.

Für die als INPUT deklarierten NVs wird ein neuer, durch den LON-Knoten aufgefrischten Wert direkt in einen DB geschrieben.

Für die als OUTPUT deklarierten NVs ist der zu einem LON-Knoten zu sendende Wert in die korrespondierende NV zu schreiben und danach der Send-SFC aufzurufen.

Der Zugriff auf die Daten erfolgt mit dem symbolischen Namen (siehe Abschnitt 3.1.5).

Data Block (Basisadresse +1)



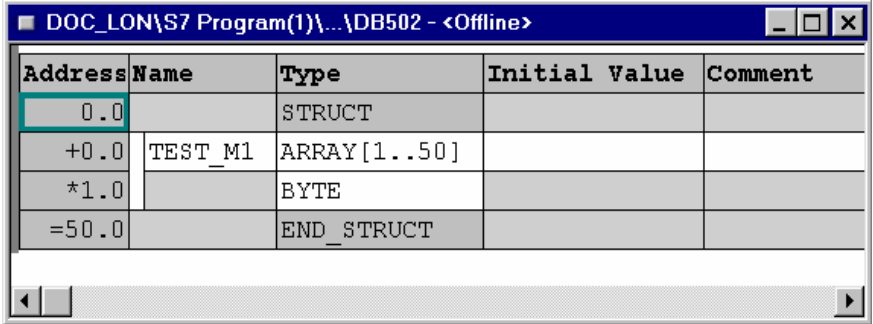
| Address | Name | Type | Initial Value | Comment |
|---------|-----------|--------|---------------|---------|
| 0.0 | | STRUCT | | |
| +0.0 | AI_1 | WORD | W#16#0 | |
| +2.0 | AI_2 | WORD | W#16#0 | |
| +4.0 | AI_3 | WORD | W#16#0 | |
| +6.0 | AI_4 | WORD | W#16#0 | |
| +8.0 | Switch_L1 | BYTE | B#16#0 | |
| +9.0 | Dummy1 | BYTE | B#16#0 | |
| +10.0 | Switch_L2 | BYTE | B#16#0 | |
| +11.0 | Dummy2 | BYTE | B#16#0 | |
| +12.0 | Switch_L3 | BYTE | B#16#0 | |
| +13.0 | Dummy3 | BYTE | B#16#0 | |
| +14.0 | Switch_L4 | BYTE | B#16#0 | |

3.1.3 'Explicit message' DB (Basisadresse +2)

Dieser DB enthält alle deklarierten 'Explicit messages'.

Der DB arbeitet in derselben Weise wie der NV-DB.

Data Block (Basisadresse +2)



| Address | Name | Type | Initial Value | Comment |
|---------|---------|--------------|---------------|---------|
| 0.0 | | STRUCT | | |
| +0.0 | TEST_M1 | ARRAY[1..50] | | |
| *1.0 | | BYTE | | |
| =50.0 | | END_STRUCT | | |

3.1.4 Diagnose-Flag DB (Basisadresse +3)

Dieser DB enthält die Diagnose für alle NVs und die 'Explicit messages'.

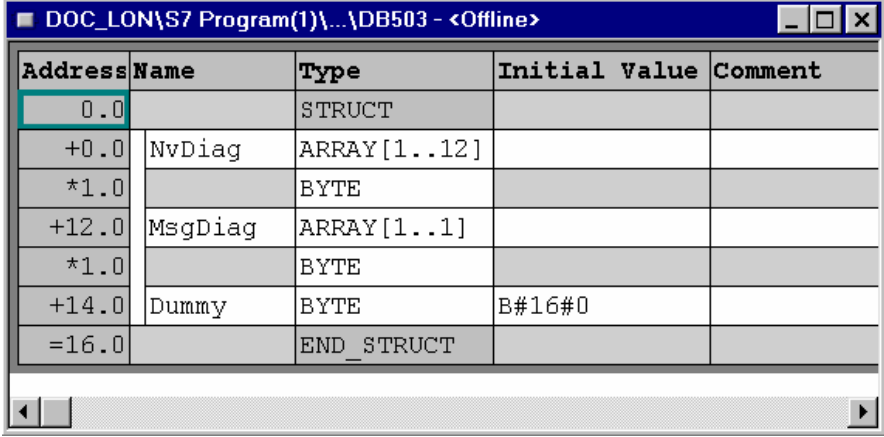
Das Feld '**NvDiag**' enthält je ein Diagnose-Byte für jede NV.

Die NV Diagnose-Bytes sind in derselben Reihenfolge wie im DB 'NV' aufgelistet (Basisadresse +1).

Beispiel: Die NV '**AI_3**' ist die 3. NV im DB 'NV', damit wird das Diagnose-Byte das 3. Byte im Feld '**NvDiag**', also **NvDiag[3]**.

Das Gleiche gilt für die 'Explicit message', jedoch im Feld '**MsgDiag**'.

Data Block (Basisadresse +3)



| Address | Name | Type | Initial Value | Comment |
|---------|---------|--------------|---------------|---------|
| 0.0 | | STRUCT | | |
| +0.0 | NvDiag | ARRAY[1..12] | | |
| *1.0 | | BYTE | | |
| +12.0 | MsgDiag | ARRAY[1..1] | | |
| *1.0 | | BYTE | | |
| +14.0 | Dummy | BYTE | B#16#0 | |
| =16.0 | | END_STRUCT | | |

Die einzelnen Bit haben die folgende Bedeutung:

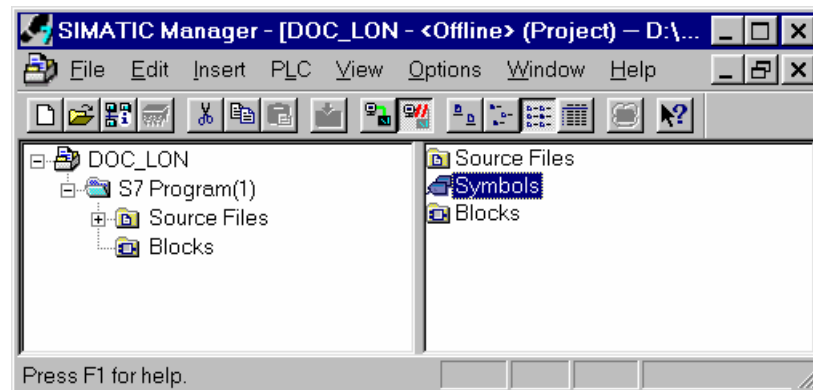
- **Bit 0:** Schreibt ein LON-Knoten in die betreffende NV (auffrischen des Wertes), wird das Bit = H (True) gesetzt. Dieses Bit ist durch die S7-Software nach der Auswertung zurück zu setzen.
- **Bit 1:** Liest ein LON-Knoten die betreffende NV, wird dieses Bit = H (True) gesetzt. Dieses Bit ist durch die S7-Software nach der Auswertung zurück zu setzen.

3.1.5 Das Arbeiten mit symbolischen Namen

Der einfachste Weg, mit der Diagnose der NVs umzugehen, führt über die symbolischen Namen. Bevor jedoch mit den symbolischen Namen der NVs, welche sich innerhalb der DBs befinden gearbeitet werden kann, muss dem DB selbst ein symbolischer Name zugewiesen werden.

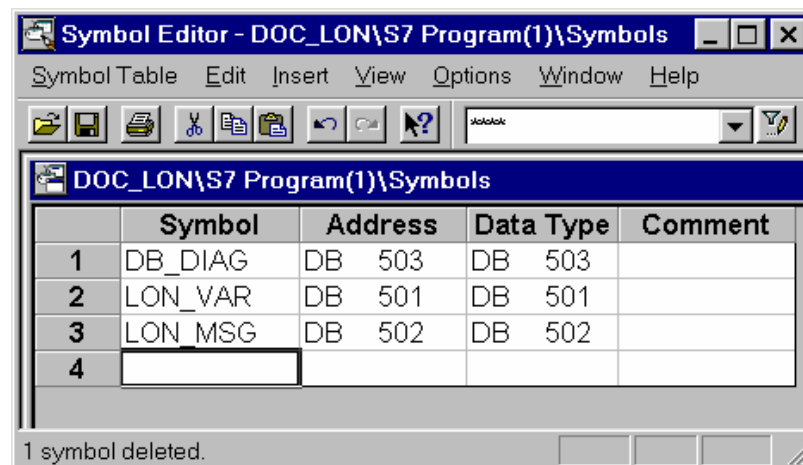
Um dies zu tun, ist die Symbol-Tabelle zu öffnen.

Im **SIMATIC Manager** ist das Unterverzeichnis '**S7 Program**' der betreffenden PCD zu wählen und ein Doppelklick auf die Ikone '**Symbols**' auszuführen.



Im Symbol-Editor sind die Symbole zu allen betroffenen DBs einzugeben und die Symbol-Tabelle abzuspeichern.

Die NVs können jetzt direkt mit dem Namen aufgerufen werden.



Beispiel:

Zum Lesen oder Schreiben eines Wertes des DBs, welcher die NV enthält

```
L      "lon_Var". switch_11
T      "lon_var". led_4
```

Lesen eines Diagnose-Bytes:

```
L      "db_diag". nvdiag[1]
```

3.2 System-Funktionen für LON

Es sind 3 System-Funktionen für den Zugriff auf das LON-Netzwerk im Betriebssystem implementiert.

Diese SFC sind: 'LON_INIT', 'LON_NV_SEND', 'LON_MSG_SEND'. Die Verwendung dieser SFCs wird nachfolgend im Detail beschrieben

3.2.1 Initialisierung mit SFC 220 "LON_INIT"

Der SFC 220 wird dazu verwendet, das LON-Interface zu initialisieren und auch zur Definition, welches die LON-DBs sind.

| Parameter | Deklaration | Typ | Beschreibung |
|-----------|-------------|------|--|
| REQ | INPUT | BOOL | REQ = 1 → Initialisierung, Reset |
| DB_NO | INPUT | WORD | Adresse des DB, welcher die Daten der LON-Konfiguration enthält (Basisadresse) |
| RET_VAL | OUTPUT | WORD | Error Information (siehe 3.2.4) |

Beispiel:

```
CALL  SFC  220
      IN0   :=TRUE
      IN1   :=500      // base address of the LON DB
      RET_VAL:=MW240
```

Bemerkung:

Es ist zu beachten, dass das Betriebssystem den OB 86 aufruft, wenn ein Fehler oder eine Neuinitialisierung erfolgt. Der OB 86 wird ebenfalls aufgerufen, wenn die Initialisierung erfolgreich durchgeführt wurde. (Für Details, siehe Abschnitt 3.3).

3.2.1.1 Option 'Self-Installation'

Wird eine PCDxx7 mit deren LON-Interface zum ersten Mal in einem LON-Netzwerk installiert, ist eine "Bindung" (engl. 'binding') zwischen den Knoten erforderlich. Dieser 'Binding'-Prozess generiert in der PCD einige Daten, welche im Konfigurations-DB gespeichert werden (Basisadresse siehe Abschnitt 3.1.1). Wurde dieser DB nicht überschrieben oder gelöscht, liegen die 'Binding'-Informationen in der PCD bereit.

Der DB in der PCD kann auch im SIMATIC-Manager-Projekt abgelegt werden, so dass die 'Binding'-Information auch innerhalb des Projekts verfügbar ist. Die andere Möglichkeit ist das Speichern des DB in einem Flash-EEPROM, dies beim Speichern der ganzen Software (Save RAM to ROM). Sollte es aus irgend welchen Gründen notwendig sein, die PCD oder das LON-Interface auszutauschen, kann zum Wiederfinden der 'Binding'-Information unter Vermeidung einer erneuten 'Binding'-Operation die 'Self-Installation' Option verwendet werden, vorausgesetzt der DB wurde korrekt gespeichert.

Diese 'Self-Installation'-Funktion stellt die 'Binding'-Information aus dem DB wieder her.

Die Verwendung dieser 'Self-Installation'-Option verlangt die Konfiguration des Feldes für den DB mit einem vordefinierten Wert. Das Feld im DB ist **'Header.SelfInstFlag'**, der einzusetzende Wert ist **'B#16#FF'**

Diese Konfigurierung hat vor dem SFC220-Aufruf "Init_LON" zu erfolgen.

Beispiel:

```
L B#16#FF
T DB500.DBB24

CALL SFC220
  IN0      : =TRUE
  IN1      : =500
  RET_VAL  : =MW240
```

3.2.1.2 Diagnose-Möglichkeiten der 'Binding'-Information

Muss aus irgend einem Grund (z.B. zufügen von NVs) eine neue 'Bindung' des PCD-Systems durchgeführt werden, wird die LED am LON-Interface blinken, um die Änderung anzuzeigen.

Die neue 'Bindung' ändert auch die Information im DB, so dass diese, im Flash-EEPROM oder in der Datei abgespeichert, nicht mehr aktuell ist. Um allen diesen Problemen auszuweichen, meldet das System die Änderung der 'Bindung' durch das Setzen der Feldes **'Header.LonCfgChanged'** (DBB25) des Konfigurations-DB auf den Wert **'b#16#FF'**. Wird der Wert wieder auf **'b#16#00'** gebracht, hört die LED auf zu blinken.

3.2.2 SEND NV mit SFC221 "LON_NV_SEND"

Dieser SFC (System Function Call) sendet eine NV (Network Variable). Die NV wird zu einem 'gebundenen' Knoten gesendet. Der parallele Aufruf ist für diese Funktion erlaubt, jedoch nur für zwei verschiedene NVs.

| Parameter | Deklaration | Typ | Beschreibung |
|-----------|-------------|------|--|
| REQ | INPUT | BOOL | REQ = 1 → senden |
| VAR_NAME | INPUT | ANY | symbolischer Name für die NV (z.B. LON_NV.variable1) |
| RET_VAL | OUTPUT | WORD | Fehler Information |
| BUSY | OUTPUT | BIT | BUSY = 1 → Funktion noch nicht abgeschlossen. |
| DONE | OUTPUT | BIT | DONE = 1 → Funktion ohne Fehler abgeschlossen |
| ERROR | OUTPUT | BIT | ERROR = 1 → Funktion fehlerhaft abgeschlossen |

REQ

Ist dieses Bit = H (true) (nicht flankenempfindlich) und die Funktion nicht "Busy", wird das Senden der NV freigegeben. Dies jedes Mal, wenn der SFC aufgerufen wird, bis das Bit nicht mehr rückgestellt wird.

VAR_NAME

Der symbolische Name der zu sendenden NV.

RET_VAL

Dieser Parameter gibt Information über das Verhalten zurück, wenn eine Funktion nicht korrekt ausgeführt werden konnte. Die Codes der Rückmeldungen sind im Abschnitt 3.2.4 beschrieben.

BUSY

Ist dieses Bit = H (true), wird signalisiert, dass eine Sendefunktion läuft und dass nicht gleichzeitig eine zweite Sendefunktion auf die gleiche NV durchgeführt werden kann.

DONE

Dieses Bit signalisiert, dass die Sendefunktion erfolgreich verlaufen ist. Das Bit bleibt nur für einen einzigen Zyklus = H (true).

ERROR

Dieses Bit wird = H (true), wenn das Resultat der Prozess-Funktion nicht erfüllt wurde. Siehe Abschnitt 3.2.4.

Bemerkung:

Das binäre Resultat (BR) des CPU-Statusregisters wird zurückgestellt, wenn die Funktion korrekt ausgeführt werden konnte. Wenn nicht, bleibt das BR = H und 'RET_VAL' enthält den Error-Code.

3.2.3 SEND Messages mit SFC 223 "LON_MSG_SEN"

Dieser SFC (System Function Call) sendet eine 'Explicit Message'. Die 'Explicit Message' wird zu einem 'gebundenen' Knoten gesendet. Der parallele Aufruf ist für diese Funktion erlaubt, jedoch nur für zwei verschiedene 'Explicit Messages'.

| Parameter | Deklaration | Typ | Beschreibung |
|-----------|-------------|------|---|
| REQ | IN | BOOL | REQ = 1 → senden |
| MSG | IN | ANY | Name der Message |
| LEN | IN | INT | Länge der Message |
| RET_VAL | OUT | WORD | Error Information |
| BUSY | OUT | BIT | BUSY = 1 → Funktion noch nicht abgeschlossen. |
| DONE | OUT | BIT | DONE = 1 → Funktion ohne Fehler abgeschlossen |
| ERROR | OUT | BIT | ERROR = 1 → Funktion fehlerhaft abgeschlossen |

REQ

Ist dieses Bit = H (true) (nicht flankenempfindlich) und die Funktion nicht "Busy", wird das Senden der 'Explicit Message' freigegeben. Dies jedes Mal, wenn der SFC aufgerufen wird, bis das Bit nicht mehr rückgestellt wird.

MSG

Der symbolische Name der zu sendenden 'Explicit Message'.

LEN

Charakterlänge der 'Explicit Message'.

RET_VAL

Dieser Parameter gibt Information über das Verhalten zurück, wenn eine Funktion nicht korrekt ausgeführt werden konnte. Die Codes der Rückmeldungen sind im Abschnitt 3.2.4 beschrieben.

BUSY

Ist dieses Bit = H (true), wird signalisiert, dass eine Sendefunktion läuft und dass nicht gleichzeitig eine zweite Sendefunktion auf die gleiche NV durchgeführt werden kann.

DONE

Dieses Bit signalisiert, dass die Sendefunktion erfolgreich verlaufen ist. Das Bit bleibt nur für einen einzigen Zyklus = H (true).

ERROR

Dieses Bit wird = H (true), wenn das Resultat der Prozess-Funktion nicht erfüllt wurde. Siehe Abschnitt 3.2.4.

Bemerkung:

Das binäre Resultat (BR) des CPU-Statusregisters wird zurückgestellt, wenn die Funktion korrekt ausgeführt werden konnte. Wenn nicht, bleibt das BR = H und 'RET_VAL' enthält den Error-Code.

3.2.4 Error Code (Fehlercode)

Alle bisher beschriebenen SFCs setzen das "BR"-Bit (Binary Result) des CPU-Statusregisters zurück, wenn die Funktion fehlerfrei durchgeführt werden konnte. Trat ein Fehler auf, wurde das "BR"-Bit gesetzt und 'RET_VAL' enthielt einen Wert, welcher den Fehler beschrieb. Einige dieser Meldungen sind Standard-S7-Fehlercodes, andere wurden für xx7-LON zugefügt.

| BUSY | DONE | ERROR | BR | RET_VAL | Beschreibung |
|------|------|-------|----|---------|---|
| 0 | 0 | 0 | 0 | 0x7000 | Erster Aufruf mit REQ = 0, → kein Datentransfer |
| 1 | 0 | 0 | 0 | 0x7001 | Erster Aufruf mit REQ = 1, → Datentransfer startet |
| 1 | 0 | 0 | 0 | 0x7002 | Interim Call (REQ irrelevant) → Datentransfer bereits aktiv |
| 0 | 1 | 0 | 0 | 0x0000 | Datentransfer erfolgreich durchgeführt |
| 0 | 0 | 1 | 0 | 0x0001 | LON-Interface funktioniert nicht korrekt |
| 0 | 0 | 1 | 0 | 0x0002 | Der Auftrag kann im Moment nicht ausgeführt werden |
| 0 | 0 | 1 | 0 | 0x4000 | Datenübertragung nicht erfolgreich |
| 0 | 0 | 1 | 1 | 0xFFFF | Der Auftrag konnte nicht erfüllt werden, da zu wenig (interner) Speicher verfügbar ist. Neuer Versuch später! |
| 0 | 0 | 1 | 1 | 0xFFFE | Die spezifizierte NV konnte nicht gefunden werden. |
| 0 | 0 | 1 | 1 | 0xFFFD | Fehler im LON-Konfigurations-DB |
| 0 | 0 | 1 | 1 | 0xFFFC | Der LON-Treiber ist noch nicht initialisiert. |
| 0 | 0 | 1 | 1 | 0xFFFB | Die DB-Adresse der Variablen stimmt nicht mit der Initialisierung überein |
| 0 | 0 | 1 | 1 | 0xFFFA | Message zu lang! |
| 0 | 0 | 1 | 1 | 0x803A | Der spezifizierte DB (z.B. der SNVT-DB) konnte nicht gefunden werden |
| 0 | 0 | 1 | 1 | 0x8022 | Der DB enthält nicht die SNVT oder die Message (DB-Länge Fehler) |

3.3 Diagnose und Fehler-OB

Das Betriebssystem ruft den Diagnose-OB 82 oder den Fehler-OB 86 derart auf, dass der S7-Programmierer die Möglichkeit zum Reagieren auf gewisse Ereignisse behält. In jedem Fall wird eine Information zum spezifischen Ereignis in den lokalen Daten des entsprechenden OBs abgelegt. Die nachfolgende Tabelle zeigt, welches Ereignis welchen Block aufruft und wo die Information zu diesem Ereignis gefunden werden kann.

| Ereignis | Bedeutung | OB | Diagnose-Daten |
|------------------------------|---|-----------|---|
| ERR_OK (0x00) | Funktion erfolgreich ausgeführt | 86 | LW 4 (RESERVED) = event LB 1 (FLT_ID) = 0xC4 LB 0 (EVENT_CLASS) = 0x38 |
| ERR_INT (0x01) | LON-Interface funktioniert nicht korrekt | 86 | LW 4 (RESERVED) = event LB 1 (FLT_ID) = 0xC4 LB 0 (EVENT_CLASS) = 0x39 |
| ERR_RESET (0x04) | NEURON-Chip wurde rückgesetzt | 86 | LW 4 (RESERVED) = event LB 1 (FLT_ID) = 0xC4 LB 0 (EVENT_CLASS) = 0x39 |
| ERR_NEURON_CONF (0x08) | NEURON-Chip ist nicht konfiguriert | 86 | LW 4 (RESERVED) = event LB 1 (FLT_ID) = 0xC4 LB 0 (EVENT_CLASS) = 0x39 |
| ERR_PARAM (0x10) | Parameter-Fehler | 86 | LW 4 (RESERVED) = event LB 1 (FLT_ID) = 0xC4 LB 0 (EVENT_CLASS) = 0x39 |
| ERR_NEURON_RE_CONF (0x20) | Zugriff eines Servicetools zu einem selbst installierten Tool | 82 | LB 4 (RESERVED_1) = event LB 1 (FLT_ID) = 0xC4 LB 0 (EVENT_CLASS) = 0x39 |
| ERR_NOT_SYNCED (0x40) | Das LON-Interface muss synchronisiert werden | 86 | LW 4 (RESERVED) = event LB 1 (FLT_ID) = 0xC4 LB 0 (EVENT_CLASS) = 0x39 |
| STAT_WINK (0x80) | Wink is carried out | 82 | LB 4 (RESERVED_1) = event LB 1 (FLT_ID) = 0xC4 LB 0 (EVENT_CLASS) = 0x39 |
| COMM_OVERLOAD (0xFF) | Zu viele ankommende Jobs (write / poll) | 82 | LB 4 (RESERVED_1) = event LB 1 (FLT_ID) = 0xC4 LB 0 (EVENT_CLASS) = 0x39 |
| NEW_LON_DB | Neuer Konfiguration-DB wurde geladen | 86 | LB 1 (FLT_ID) = 0xC3 LD 8 (RACKS_FLTD) = 0xFFFFFFFF LB 0 (EVENT_CLASS) = 0x39 |

Es ist zu beachten, dass bei nicht programmierten OBs die PCD in 'Stop' geht, wenn eines der genannten Ereignisse auftreten sollte.

3.4 Einschränkungen

Es sind einige Einschränkungen bei der Verwendung des LON-Interfaces, herrührend von Systemgrenzen, zu beachten.

- Sobald das LON-Interface mit dem SFC 220 initialisiert wurde, kann der S7-Speicher mit der PCD in RUN nicht komprimiert werden.
- Die Anzahl simultaner Funktionen für die SENDING NV bei Verwendung des SFC 221, ist auf 15 beschränkt.