# Table of Contents

# Chapter 1

## Programming Information for Control FPWIN Pro

## 1.1  General Information

Make sure you use at least version 7.1 of FPWIN Pro, which includes the functions necessary for programming the FP7-FMUs. The function blocks described in this section are used to configure the various FMUs and to start communication with the specific network.

The functions and function blocks can be used for theFP7-FMU unit.

You can download the function blocks free of charge from the Panasonic Electric Works Europe AG Web site.

## 1.2  GetPointer Function

Inputs —— GetPointer ——InputsPointer
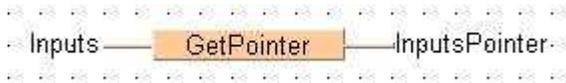
The GetPointer function outputs the size, area and offset of the input variable and writes it to the output variable of the type POINTER. Connect the output of this function directly to the respective input of the function block.

**◆REFERENCE**

**For more information about the GetPointer function, please refer to the FPWIN Pro online help.**

## 1.3 FMU_DataExchange
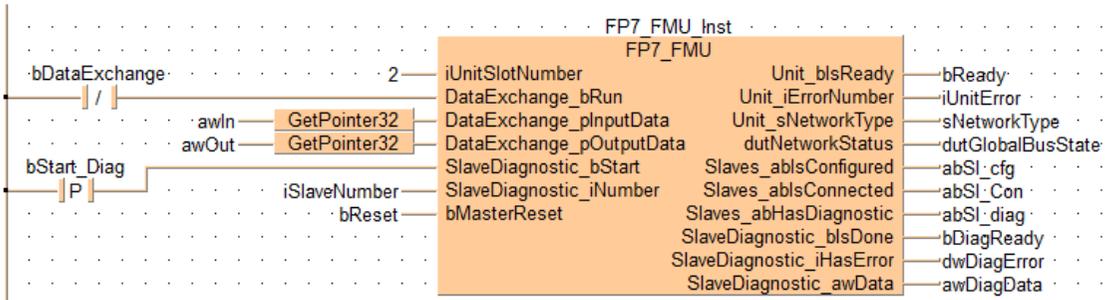


The FMU_DataExchange function block is used to start communication with the network. It has to be supplied with information about the input and output size, and network-specific data.

Just leave the pins unconnected that you do not require.

**PLC types: available for FP7**

Variables of this function block have to be of one of the following data types:

### Inputs

| Data type | Input | Function |
|---|---|---|
| INT | iUnitSlotNumber | Installation position of the FMU. |
| BOOL | DataExchange_bRun | If true, network communication runs. |
| POINTER | DataExchange_pInputData | Pointer to the input's process data table. |
| POINTER | DataExchange_pOutputData | Pointer to the output's process data table. |
| BOOL | SlaveDiagnostic_bStart | If rising edge, slave diagnostic starts on time. |
| INT | SlaveDiagnostic_iNumber | Node number for diagnostic. |
| BOOL | bMasterReset | If true, the unit is in reset stage. |

### Outputs

| Data type | Output | Function |
|---|---|---|
| BOOL | Unit_bIsReady | ON if unit is ready. |
| INT | Unit_ErrorNumber | Unit is in error. |
| String | sNetworkType | It shows the found Network Type |
| DUT | dutNetworkStatus | Depends on the network (see "dutNetworkStatus Output" on page 7). |
| BOOL[128] | Slaves_abIsConfigured | Depends on the network (see "Slaves_abIsConfigured Output" on page 17). |
| BOOL[128] | Slaves_abIsConnected | Depends on the network (see "Slaves_abIsConnected Output" on page 19). |
| BOOL[128] | Slaves_abHasDiagnostic | Depends on the network (see "Slaves_abHasDiagnostic Output" on page 20). |
| BOOL | SlaveDiagnostic_bIsDone | ON if done. |

| Data type | Output | Function |
|-----------|--------|----------|
| INT | SlaveDiagnostic_iHasError | Depends on the network (see "SlaveDiagnostic_iHasError Output" on page 23). |
| WORD[4] | SlaveDiagnostic_awData | Depends on the network (see "SlaveDiagnostic_awData Output" on page 24). |

**List of error codes for the function block FMU_DataExchange**

| Error code | Indication |
|------------|------------|
| 16#0000 | No error. |
| 16#0001 | Input variable is not in DT area. |
| 16#0002 | Output variable is not in DT area. |
| 16#0003 | 0 < Slot number  > 16 |
| 16#0004 | Unknown Network Module |
| 16#0005 | Not a Network Module |

### 1.3.1    dutNetworkStatus Output

Connected to the output pin **dutNetworkStatus** is the variable **GlobalBusState**, which in turn is part of the DUT **GlobalBusStateField** included in the FMU library for Control FPWIN Pro.
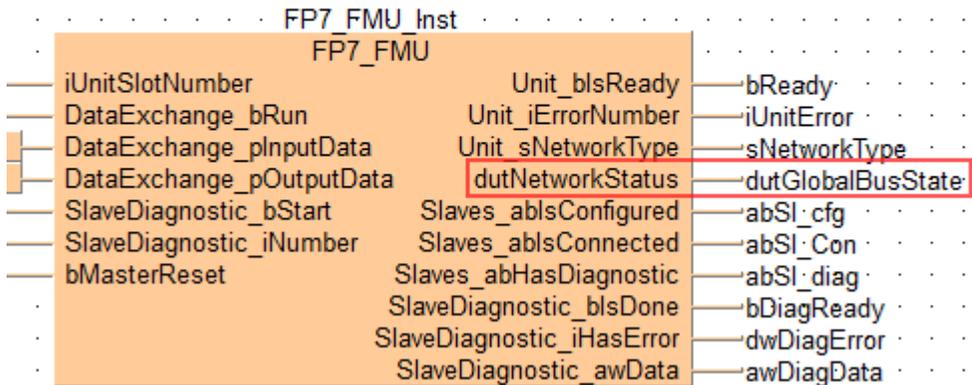


This DUT is used for all network types, but the information stored in the DUT is nevertheless network-specific:

- PROFIBUS (see page 8)
- DeviceNET (see page 11)
- CANopen (see page 15)

### 1.3.1.1   GlobalBusStateField for PROFIBUS

| | Identifier | Type | Initial | |
|---|---|---|---|---|
| 0 | GlobBit_Status | WORD | 0 | |
| 1 | ErrRemAdr_ErrEvent | WORD | 0 | |
| 2 | BusErrorCnt | INT | 0 | |
| 3 | TimeOutCount | INT | 0 | |
| 4 | | | | |

| Identifier part | Identifier | Explanation |
|---|---|---|
| Global bits | GlobBit_Status | Lower byte of the WORD. Details follow the table. |
| Status | GlobBit_Status | Higher byte of the WORD. The master system can be in one of the following states: <br>• 16#00: OFFLINE <br>• 16#40: STOP <br>• 16#80: CLEAR <br>• 16#C0: OPERATE |
| Error remote address | ErrRemAddr_ErrEvent | Lower byte of the WORD. Some of the global bits indicate errors in the network or in the FMU itself that require a more detailed explanation. In these cases, "ErrRemAddr" represents the source of the error and can arise from either the FMU itself (value = 255), or be detected and reported by a network device. In this case, the lower byte represents the station address. Range: 0 to 125. |
| Error event | ErrRemAddr_ErrEvent | Higher byte of the WORD. Each error is assigned a number. The errors are explained following the table. |
| Bus error count | BusErrorCnt | Counts severe bus errors, e.g. bus short circuits. |
| Timeout count | TimeOutCount | Counts the number of rejected PROFIBUS telegrams due to severe bus errors. |

**Global bits**

The bit field serves as collective display of global notifications. Errors can either occur at the FMU itself or at the slaves. To distinguish between the different errors, the variable part ErrRemAddr contains the error location (bus address), while the variable part ErrEvent lists the corresponding error number. If more than one error is determined, the error location will always show the lowest faulty bus address.

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|

| Bit | Explanation |
|---|---|
| 0 | **Control error**. Parameterization error. |
| 1 | **Auto-clear error**. FMU stopped communication to all slaves and reached the auto-clear end state. |

| Bit | Explanation |
|---|---|
| 2 | **Non-exchange error**. At least one slave has not reached the data exchange state, hence no process data can be exchanged. |
| 3 | **Fatal error**. Due to severe bus error, no further bus communication is possible. |
| 4 | **Event error**. The FMU has detected bus short circuits. The number of detected events are fixed in the BusErrorCnt variable. The bit is set when the first event is detected. |
| 5 | **Host not ready notification**. If the bit is set, the host program ist NOT ready to communicate. |
| 6 | **Timeout error**. The FMU has detected an overstepped timout supervision time due to rejected PROFIBUS telegramms. It indicates bus short circuits while the master interrupts the communication.The number of detected timeouts are fixed in the TimeOutCount variable.The bit is set when the first timeout is detected. |
| 7 | Reserved. |

**Errors**

The following error numbers are valid for the error event if "Error remote address" is 255.

| ErrEvent | Explanation | Source of error | Recommendation |
|---|---|---|---|
| 0 | No errors | | |
| 50 | User interface task not found. | FMU | Contact technical support. |
| 51 | No global data field. | FMU | Contact technical support. |
| 52 | Fieldbus data link task not found. | FMU | Contact technical support. |
| 53 | PLC task not found. | FMU | Contact technical support. |
| 54 | Non-existent master parameters. | FMU | Download database again. |
| 55 | Faulty master parameter value. | Project planning | Contact technical support. |
| 56 | Non-existent slave parameters. | Project planning | Download database again. |
| 57 | Faulty slave parameter value in data file. | Project planning | Contact technical support. |
| 58 | Double slave address. | Project planning | Check projected addresses. |
| 59 | Projected send process data offset of participant's address outside allowable range of 0-255. | Project planning | Check projected addresses. |
| 60 | Projected receive process data offset of participant's address outside allowable range of 0-255. | Project planning | Check projected addresses. |
| 61 | Slaves' data areas overlapping for send process data. | Project planning | Check projected addresses. |
| 62 | Slaves' data areas overlapping for receive process data. | Project planning | Check projected addresses. |
| 63 | Unknown process data handshake. | Warm start | Check warm start parameters. |
| 64 | Free RAM exceeded. | FMU | Contact technical support. |
| 65 | Faulty slave parameter data sets. | Project planning | Contact technical support. |
| 202 | No segment for treatment free. | FMU | Contact technical support. |
| 212 | Faulty reading of database. | FMU | Download database again. |
| 213 | Structure surrender to operating system faulty. | FMU | Contact technical support. |
| 220 | Software watchdog error. | Host | Check host program. |

| ErrEvent | Explanation | Source of error | Recommendation |
|---|---|---|---|
| 221 | No data acknowledgement in process data handshake mode 0. | Host | Host program did not acknowledge the last handshake in time. |
| 222 | Master in auto-clear mode. | Slave | Auto-clear mode was activated because one slave was missing during runtime. |
| 225 | No further segments. | FMU | Contact technical support. |

The following error numbers are valid for the error event if "Error remote address" is not equal to 255.

| ErrEvent | Explanation | Source of error | Recommendation |
|---|---|---|---|
| 2 | Station reports overflow. | Master telegram | Check length of configured slave configuration or parameter data. |
| 3 | Request function of master not activated in the station. | Master telegram | Check slave if PROFIBUS DP norm compatible.. |
| 9 | No answer even though slave must respond. | Slave | Check configuration data of the station and compare it with the physical I/O data length. |
| 17 | No response from station. | Slave | Check bus cable. Check bus address of slave. |
| 18 | Master not in the logical token ring. | FMU | Check master's fieldbus data link address or the highest station address of other master systems. Check bus cable for short circuits. |
| 21 | Faulty parameter in request. | Master telegram | Contact technical support. |

### 1.3.1.2  GlobalBusStateField for DeviceNet

| | Identifier | Type | Initial | |
|---|---|---|---|---|
| 0 | GlobBit_Status | WORD | 0 | |
| 1 | ErrRemAdr_ErrEvent | WORD | 0 | |
| 2 | BusErrorCnt | INT | 0 | |
| 3 | TimeOutCount | INT | 0 | |
| 4 | | | | |

| Identifier part | Identifier | Explanation |
|---|---|---|
| Global bits | GlobBit_Status | Lower byte of the WORD. Details follow the table. |
| Status | GlobBit_Status | Higher byte of the WORD. The master system can be in one of the following states: <br>• 16#00: OFFLINE <br>• 16#40: STOP <br>• 16#80: CLEAR <br>• 16#C0: OPERATE |
| Error remote address | ErrRemAddr_ErrEvent | Lower byte of the WORD. <br>If the global bits "control error, auto-clear error or non-exchange error" are set, this variable indicates the address of the device. If the error occurs in the FMU, the value is 255. Otherwise the faulty device address = MAC ID is stored. |
| Error event | ErrRemAddr_ErrEvent | Higher byte of the WORD. <br>Each error is assigned a number. The errors are explained following the table. |
| Bus error count | BusErrorCnt | This variable is incremented whenever the error frame counter of the Philips CAN chip used has reached the warning limit due to disturbed bus communication. |
| Timeout count | TimeOutCount | This variable is incremented when the CAN chip reports that it is no longer involved in bus activities because the bus error frame counter has been exceeded. The chip must be reinitialized, which is done automatically by the FMU. |

**Global bits**

The bit field serves as collective display of global notifications. Errors can either occur at the FMU itself or at the slaves. To distinguish between the different errors, the variable part ErrRemAddr contains the error location (address = MAC ID), while the variable part ErrEvent lists the corresponding error number. If more than one error is detected, the error location will always show the device with the lowest MAC ID.

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|

| Bit | Explanation |
|---|---|
| 0 | **Control error**. Parameterization error. |
| 1 | **Auto-clear error**. FMU stopped communication to all slaves and reached the auto-clear end state. |
| 2 | **Non-exchange error**. At least one slave has not reached the data exchange state, hence no process data can be exchanged. |
| 3 | **Fatal error**. Due to severe bus error, no further bus communication is possible. |

| Bit | Explanation |
|-----|-------------|
| 4 | **Event error**. The FMU has detected bus short circuits. The number of detected events are fixed in the BusErrorCnt variable. The bit is set when the first event is detected. |
| 5 | **Host not ready notification**. If the bit is set, the host program ist NOT ready to communicate. |
| 6 | **Duplicate MAC ID detected**. The FMU has detected another device in the network with the same MAC ID. |
| 7 | **Duplicate MAC ID check in progress**. As long this bit is set, the FMU is checking for duplicate MAC IDs. |

## Errors

The following error numbers are valid for the error event if "Error remote address" is 255.

| ErrEvent | Explanation | Source of error | Recommendation |
|----------|-------------|-----------------|----------------|
| 52 | Unknown process data handshake mode configured. | Configuration. | |
| 53 | Baud rate out of range. | Configuration. | |
| 54 | FMU MAC ID address out of range. | Configuration. | |
| 57 | Duplicate FMU MAC ID address detected in the network | Configuration or network. | |
| 58 | No device entry found in the current configuration database. | Download error in the current database. | Contact technical support. |
| 210 | No database found in the system. | Configuration not downloaded. FMU is not configured by Control Configurator FM. | |
| 212 | Failure reading the database. | | Contact technical support. |
| 220 | User watchdog failed. | Application. | |
| 221 | No data acknowledgement from user. | Application. | |
| 223 | Master has stopped bus communication due to CAN-based bus off error. In the Control Configurator FM for the bus parameters under "Error handling", "Auto Clear" is activated. | Network error. | |
| 226 | Master firmware downloaded to slave EC1 device. | User error. | |

The following error numbers are valid for the error event if "Error remote address" is not equal to 255.

| ErrEvent | Explanation | Source of error | Recommendation |
|----------|-------------|-----------------|----------------|
| 0 | No errors. | | |
| 1 | Device guarding failed after device was operational. | Device | Check if device is still running. |
| 30 | Device access timeout. | Device | Device does not respond. Check the baud rate and MAC ID. |
| 32 | Device rejects access with unknown error code. | Device | Use single-device diagnostic to get reject code. |
| 35 | Device response in allocation phase with connection error. | Device | Use single-device diagnostic to get additional reject code. |

| ErrEvent | Explanation | Source of error | Recommendation |
|---|---|---|---|
| 36 | Connection produced (process data input length from the FMU's point-of-view) is different from the one configured. | Device/configuration | Use single-device diagnostic to get true produced connection size. |
| 37 | Size of connection consumed (process data output length from the FMU's point-of-view) is different from the one configured. | Device/configuration | Use single-device diagnostic to get true consumed connection size. |
| 38 | Device service response telegram unknown and not handled. | Device/configuration | Use single-device diagnostic to get true consumed connection size. |
| 39 | Connection already requested. | Device | Connection will be automatically released. |
| 40 | Number of CAN message data bytes read in the produced or consumed connection size response unequal to 4. | Device | Device cannot operate with the FMU and norm description. |
| 41 | Predefined master-slave connection already exits. | Device/FMU | Connection will be automatically released. |
| 42 | Length in polling device response unequal to produced connection size. | Device | |
| 43 | Sequence error in device polling response. | Device | Two initial segments in multiplexed transfer were received. |
| 44 | Fragment error in device polling response. | Device | Fragmentation counter during multiplexed transfer differs from the one expected. |
| 45 | Sequence error in device polling response. | Device | Middle or last segment was received before the first segment. |
| 46 | Length in bit strobe device response unequal to produced connection size. | Device | |
| 47 | Sequence error in device COS or cyclic response. | Device | Two initial segments in multiplexed transfer were received. |
| 48 | Fragment error in device COS or cyclic response. | Device | Fragmentation counter during multiplexed transfer differs from the one expected. |
| 49 | Sequence error in device COS or cyclic response. | Device | Middle or last segment was received before the first segment. |
| 50 | Length in COS or cyclic device response unequal to produced connection size. | Device | |
| 51 | UCMM group not supported. | Device | Change the UCMM group. |
| 52 | Device keying failed: vendor ID mismatch. | Device/configuration | Check vendor ID configured with device's vendor ID. |
| 53 | Device keying failed: device type mismatch. | Device/configuration | Check device type configured with device's device type. |
| 54 | Device keying failed: product code mismatch. | Device/configuration | Check product code configured with device's product code. |
| 55 | Device keying failed: revision mismatch. | Device/configuration | Check revision configured with device's revision. |

| ErrEvent | Explanation | Source of error | Recommendation |
|---|---|---|---|
| 59 | Double device address configured in current configuration. | Configuration | Each device in DeviceNet must have its own MAC ID. |
| 60 | Whole size indicator of one device data set is corrupt. | Configuration | Download error in the current database. Contact technical support. |
| 61 | Size of the additional table for predefined master-slave connections is corrupt. | Configuration | Download error in the current database. Contact technical support. |
| 62 | Size of predefined master-slave I/O configuration table is corrupt. | Configuration | Download error in the current database. Contact technical support. |
| 63 | Predefined master-slave I/O configuration does not correspond to the additional table. | Configuration | Number of I/O units and the number of configured offset addresses are different. |
| 64 | Size indicator of parameter data table is corrupt. | Configuration | Value of size indicator too small. |
| 65 | Number of inputs declared in the additional table does not correspond to the number in the I/O configuration table. | Configuration | Each entry in the I/O configuration must have only one entry in the additional table. |
| 66 | Number of outputs declared in the additional table does not correspond to the number in the I/O configuration table. | Configuration | Each entry in the I/O configuration must have only one entry in the additional table. |
| 67 | Unknown data type in I/O configuration detected. | Configuration | Data types supported: BOOL, BYTE, WORD, DWORD and STRING only. |
| 68 | Data type of a defined I/O unit in a connection does not correspond with the defined data size. | Configuration | The following types and size are valid:<br>• BOOLEAN = 1 byte<br>• Unit 8 = 1 byte<br>• Unit 16 = 2 bytes<br>• Unit 32 = 4 bytes |
| 69 | Output address configured for one unit exceeds the possible address range of 3584 bytes. | Configuration | The process data image is limited to 3584 bytes. |
| 70 | Input address configured for one unit exceeds the possible address range of 3584 bytes. | Configuration | The process data image is limited to 3584 bytes. |
| 71 | One predefined connection type is unknown. | Configuration | Support of cyclic, polled, change of state, bit strobed only. |
| 72 | Multiple connections defined in parallel. | Configuration | Supports only one type of connection to one device. |
| 73 | The value configured for the expected packet rate value configured is less than the value for production inhibit time. | Configuration | The value for the expected packet rate must be larger than the production inhibit time.<br>In Control Configurator FM, check the settings for the slave's |

| ErrEvent | Explanation | Source of error | Recommendation |
|----------|-------------|-----------------|----------------|
| | | | poll connection configuration. |
| | | | |

### 1.3.1.3  GlobalBusStateField for CANopen

| | Identifier | Type | Initial | |
|---|------------|------|---------|---|
| 0 | GlobBit_Status | WORD | 0 | |
| 1 | ErrRemAdr_ErrEvent | WORD | 0 | |
| 2 | BusErrorCnt | INT | 0 | |
| 3 | TimeOutCount | INT | 0 | |
| 4 | | | | |

GlobalBusStateField ×

| Identifier part | Identifier | Explanation |
|-----------------|-----------|-------------|
| Global bits | GlobBit_Status | Lower byte of the WORD. Details follow the table. |
| Status | GlobBit_Status | Higher byte of the WORD.<br>The master system can be in one of the following states:<br>• 16#00: OFFLINE<br>• 16#40: STOP<br>• 16#80: CLEAR<br>• 16#C0: OPERATE |
| Error remote address | ErrRemAddr_ErrEvent | Lower byte of the WORD.<br>If the global bits "control error, auto-clear error or non-exchange error" are set, indicates the address of the node. If the error occurs in the FMU, the value is 255. Otherwise the faulty node address is stored. |
| Error event | ErrRemAddr_ErrEvent | Higher byte of the WORD.<br>Each error is assigned a number, which are explained following the table. |
| Bus error count | BusErrorCnt | Counts how many times bus error limits are exceeded. |
| Timeout count | TimeOutCount | Counts the number of CAN chip reinitializations. |

**Global bits**

The bit field serves as collective display of global notifications. Errors can either occur at the FMU itself or at the nodes. To distinguish between the different errors, the variable part ErrRemAddr contains the error location (address), while the variable part ErrEvent lists the corresponding error number. If more than one error is determined, the error location will always show the lowest faulty bus address.

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|------|---|---|---|---|---|---|---|---|

| Bit | Explanation |
|-----|-------------|
| 0 | **Control error**. Parameterization error. |
| 1 | **Auto-clear error**. FMU stopped communication to all nodes and reached the auto-clear end state. |
| 2 | **Non-exchange error**. At least one node has not reached the data exchange state, hence no process |

| Bit | Explanation |
|---|---|
| | data can be exchanged. |
| 3 | **Fatal error**. Due to severe internal error, no further bus communication is possible. |
| 4 | **Event error**. The FMU has detected transmission errors. The number of detected events are fixed in the BusErrorCnt variable. The bit is set when the first event is detected. |
| 5 | **Host not ready notification**. If the bit is set, the host program ist NOT ready to communicate. |
| 6 | **Timeout error**. The FMU has detected an overstepped timeout supervision time of at least one CAN message to be sent. The transmission of this message was aborted. The data is lost. It indicated that no other CAN device was connected or could not acknowledge the message sent. The number of timeouts detected are stored in the message timeout variable. The bit is set when the first timeout is detected. |
| 7 | Reserved. |

**Errors**

The following error numbers are valid for the error event if "Error remote address" is 255.

| ErrEvent | Explanation | Source of error | Recommendation |
|---|---|---|---|
| 0 | No errors | | |
| 52 | Unknown handshake mode was configured. | Initialization | If you use the function block, please use the default handshake settings. |
| 56 | Baud rate out of range. | Project planning | Contact technical support. |
| 60 | Double node address was configured. | Project planning | Contact technical support. |
| 63 | Invalid parameter for SYNC mode. | Project planning | In Control Configurator FM under bus parameters, check the SYNC Master settings. |
| 210 | No database. | Project planning | Download database again. |
| 212 | Faulty reading of a database. | Device | Download database again. |
| 220 | Host watchdog error. | Host program | Check user program or watchdog timer. |

The following error numbers are valid for the error event if "Error remote address" is not equal to 255.

| ErrEvent | Explanation | Source of error | Recommendation |
|---|---|---|---|
| 30 | Guarding failed. | Node | Check whether node is connected. |
| 31 | Node has changed its state and is no longer operational. | Node | Reset node. |
| 32 | Sequence error in guarding protocol. | Node | Reset node. |
| 33 | No response to a configured remote frame PDO. | Node | Check whether node can handle remote frames. |
| 34 | No response of the node while being configured. | Node | Check whether node is connected and operational. |
| 35 | The node profile number configured in the master differs from the actual node profile number. | Project planning | Check the supported profile number of the node: I/O, encoder, etc. |

| ErrEvent | Explanation | Source of error | Recommendation |
|----------|-------------|-----------------|----------------|
| 36 | The device type configured in the master differs from the actual node device type. | Project planning | Check the supported services of the node. |
| 37 | Unknown SDO response received. | Node | Node not compatible with CiA protocol specification. |
| 38 | Length indicator of received SDO message does not equal 8. | Node | Node not compatible with CiA protocol specification. |
| 39 | Node not handled. Node stopped. | Device | Activated auto-clear mode or host is not ready. |

## 1.3.2   Slaves_abIsConfigured Output

Connected to the output pin **Slaves_abIsConfigured** is a variable of the type ARRAY [0...127] of BOOL.



In this example, the identifier is **abSI_cfg**. This variable is used for all network types, but the information stored in the variable is nevertheless network-specific:

- abSI_cfg for PROFIBUS (see page 17)
- abSI_cfg for DeviceNet (see page 18)
- abSI_cfg for CANopen (see page 18)

### 1.3.2.1   abSI_cfg for PROFIBUS

This variable is an ARRAY [0...127] of BOOL, i.e. a field of 8 words, and contains the parametrization state of each FMU slave.

The bit for the corresponding slave is logical:

- 1 = the slave is configured in the FMU.
- 0 = the slave is not configured in the FMU.

### 1.3.2.2 abSI_cfg for DeviceNet

This variable is an ARRAY [0...127] of BOOL, i.e. a field of 8 words, and contains the parametrization state of each FMU slave. For DeviceNet, bits 0 to 63 are available.

The bit for the corresponding slave is logical:

- 1 = the slave is configured in the FMU.
- 0 = the slave is not configured in the FMU.

### 1.3.2.3 abSI_cfg for CANopen

This variable is an ARRAY [0...127] of BOOL, i.e. a field of 8 words, and contains the parametrization state of each FMU node.

The bit for the corresponding node is logical:

- 1 = the node is configured in the FMU.
- 0 = the node is not configured in the FMU.

### 1.3.3   Slaves_abIsConnected Output

Connected to the output pin **Slaves_abIsConnected** is a variable of the type ARRAY [0...127] of BOOL.

| | Class | Identifier | Type | Initial |
|---|---|---|---|---|
| 0 | VAR | abSI_cfg | ARRAY [0..127] OF BOOL | [128(FALSE)] |
| 1 | VAR | abSI_Con | ARRAY [0..127] OF BOOL | [128(FALSE)] |
| 2 | VAR | abSI_diag | ARRAY [0..127] OF BOOL | [128(FALSE)] |

```
                        FP7_FMU_Inst
                           FP7_FMU
 ─── iUnitSlotNumber                    Unit_bIsReady ─── bReady
 ─── DataExchange_bRun                Unit_iErrorNumber ─── iUnitError
 ─── DataExchange_pInputData          Unit_sNetworkType ─── sNetworkType
 ─── DataExchange_pOutputData          dutNetworkStatus ─── dutGlobalBusState
 ─── SlaveDiagnostic_bStart        Slaves_abIsConfigured ─── abSI·cfg
 ─── SlaveDiagnostic_iNumber        Slaves_abIsConnected ─── abSI·Con
 ─── bMasterReset                  Slaves_abHasDiagnostic ─── abSI·diag
                                  SlaveDiagnostic_bIsDone ─── bDiagReady
                                 SlaveDiagnostic_iHasError ─── dwDiagError
                                    SlaveDiagnostic_awData ─── awDiagData
```

In this example, the identifier is ab**SI_Con**. This variable is used for all network types, but the information stored in the variable is nevertheless network-specific:

- abSI_Con for PROFIBUS (see page 19)
- abSI_Con for DeviceNET (see page 19)
- abSI_Con for CANopen (see page 20)

### 1.3.3.1   abSI_Con for PROFIBUS

This variable is an ARRAY [0...127] of BOOL, i.e. a field of 8 words, and contains the parametrization state of each FMU slave.

The bit of the corresponding slave is logical:

- 1 = the slave and the master are exchanging I/O data.
- 0 = the slave and the master are not exchanging I/O data.

The values in the variable abSI_Con are only valid if the master is in the OPERATE state.

### 1.3.3.2   abSI_Con for DeviceNet

This variable is an ARRAY [0...127] of BOOL, i.e. a field of 8 words, and contains the parametrization state of each FMU slave.

The lower 4 words, ARRAY [0...63] of BOOL, store the state for the device's explicit connection. The bit of the corresponding device is logical:

- 1 = the device's explicit connection is in the established state
- 0 = the device's explicit connection is not in the established state

The values in the variable abSl_Con are only valid if the master is in the OPERATE state.

The higher 4 words, ARRAY [64...127] of BOOL, store the state for the device's I/O connection. The bit of the corresponding device is logical:

- 1 = the device's I/O connection is in the established state
- 0 = the device's I/O connection is not in the established state

### 1.3.3.3  abSl_Con for CANopen

This variable is an ARRAY [0...127] of BOOL, i.e. a field of 8 words, and contains the parametrization state of each FMU node.

The bit of the corresponding node is logical:

- 1 = node is operating; node guarding reports no error
- 0 = node is not operating because it is not configured or an error has occurred

The values in the variable abSl_Con are only valid if the master is in the OPERATE state.

## 1.3.4   Slaves_abHasDiagnostic Output

The inputs and outputs found on the lower half of the function block help you diagnose problems.

Connected to the output pin **Slaves_abHasDiagnostic** is a variable of the type ARRAY [0...127] of BOOL.

| | Class | Identifier | Type | Initial |
|---|---|---|---|---|
| 0 | VAR | abSl_cfg | ARRAY [0..127] OF BOOL | [128(FALSE)] |
| 1 | VAR | abSl_Con | ARRAY [0..127] OF BOOL | [128(FALSE)] |
| 2 | VAR | abSl_diag | ARRAY [0..127] OF BOOL | [128(FALSE)] |
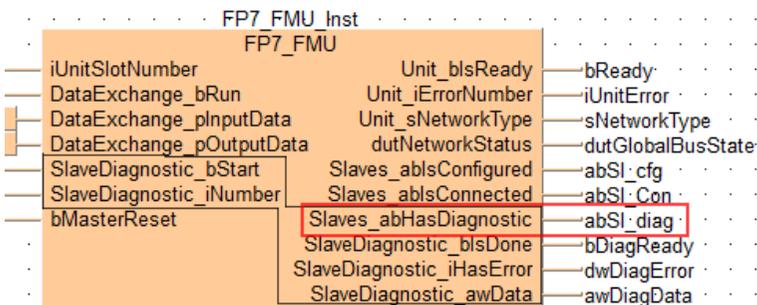
In this example, the identifier is ab**Sl_diag**. This variable is used for all network types, but the information stored in the variable is nevertheless network-specific:

- abSl_diag for PROFIBUS (see page 21)

- abSl_diag for DeviceNET (see page 22)

- abSl_diag for CANopen (see page 21)

### 1.3.4.1  abSl_diag for PROFIBUS

This variable is an ARRAY [0...127] of BOOL, i.e. a field of 8 words, and contains the diagnostic bit for each FMU slave.

The bit of the corresponding slave station is logical:

- 1 = latest received slave diagnostic data is available in the internal diagnostic buffer. If the bit is set to 1, you can enter the corresponding slave station number at the input **Slave_Diagnostic_iNumber**, and FPWIN Pro will tell you what the error is.

- 0 = since the last diagnostic buffer read access of the host, no values have changed in this buffer.

The values in the variable abSl_state are only valid if the master is in the OPERATE state.

The following table shows the relationship between the **SI_state** bit and the **SI_diag** bit.

|  | **SI_state = 0** | **SI_state = 1** |
|---|---|---|
| SI_diag = 0 | • No data I/O exchange between master and slave.<br>Perhaps this slave is not configured or not responsive. | • Slave is present on the bus.<br><br>• Data I/O exchange between master and slave. |
| SI_diag = 1 | • The master and the corresponding slave are not exchanging I/O data.<br><br>• The master holds newly received diagnostic data in the internal diagnostic buffer. | • Slave is present on the bus.<br><br>• The master and the corresponding slave are exchanging I/O data.<br><br>• The master holds newly received diagnostic data in the internal diagnostic buffer. |

## 1.3.4.2  abSI_diag for DeviceNet

This variable is an ARRAY [0...127] of BOOL, i.e. a field of 8 words, and contains the diagnostic bit of each device. For DeviceNet, bits 0 to 63 are available.

The of the corresponding device is logical:

- 1 = newly received diagnostic values are available in the internal diagnostic buffer or one of the diagnostics bit of the device has changed. If the bit is set to 1, you can enter the corresponding slave station number at the input **Slave_Diagnostic_iNumber**, and FPWIN Pro will tell you what the error is.

- 0 = since the last diagnostic buffer read access of the host, no values have changed in the internal diagnostic buffer.

The values in the variable abSI_Con are only valid if the master is in the OPERATE state.

The following table shows the relationship between the **SI_Con** bit and the **SI_diag** bit.

|  | **abSI_Con = 0** | **abSI_Con = 1** |
|---|---|---|
| abSI_diag = 0 | • Device not operative, no process data exchange between FMU and device.<br><br>• Device is not configured. | • Device is present on the network, device guarding active.<br><br>• Process data exchange between FMU and device happening as configured. |
| abSI_diag = 1 | • Device is not operating, device guarding failed or configuration fault detected.<br><br>• New diagnostic data provided by the FMU in the internal diagnostic buffer to be read by host. | • Device is present on the bus, device guarding is active, process data exchange.<br><br>• New diagnostic data provided by the FMU in the internal diagnostic buffer to be read by host. |

## 1.3.4.3  abSI_diag for CANopen

This variable is an ARRAY [0...127] of BOOL, i.e. a field of 8 words, and contains the diagnostic bit of each node. The bit of the corresponding node is logical:

- 1 = newly received emergency message are available in the internal diagnostic buffer or one of the diagnostics bit of the node has changed. If the bit is set to 1, you can enter the corresponding node number at the input **Slave_Diagnostic_iNumber**, and FPWIN Pro

will tell you what the error is.

- 0 = since the last diagnostic buffer read access of the host, no values have changed in this buffer.

The values in the variable SI_state are only valid if the master is in the OPERATE state.

The following table shows the relationship between the **SI_state** bit and the **SI_diag** bit.

| | abSI_Con = 0 | abSI_Con = 1 |
|---|---|---|
| abSI_diag = 0 | • Node not in operation, no data I/O exchange between master and node. Perhaps this slave is not configured. | • Node is present on the bus; node guarding is active.<br>• PDO exchange between master and node is happening as configured. |
| abSI_diag = 1 | • Node is not operating; node guarding failed.<br>• The master holds newly received diagnostic data in the internal diagnostic buffer. | • Node is present on the bus, node guarding is active, PDO exchange.<br>• The master holds newly received diagnostic data in the internal diagnostic buffer. |

## 1.3.5   SlaveDiagnostic_bIsDone Output

The inputs and outputs found on the lower half of the function block help you diagnose problems.

Connected to the output pin **SlavesDiagnostic_bIsDone** is a variable of the type BOOL.

- 1 = diagnosis of the slave (node) complete.
- 0 = diagnosis of the slave (node) not complete.
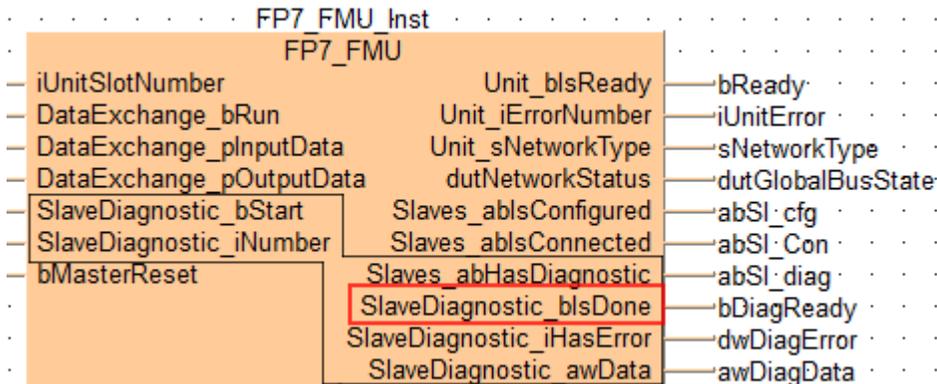


As long as the bit is set to 0, check at **SlaveDiagnostic_iHasError** (see "SlaveDiagnostic_iHasError Output" on page 23) to find out where the error is.

## 1.3.6   SlaveDiagnostic_iHasError Output

The inputs and outputs found on the lower half of the function block help you diagnose problems.

Connected to the output pin **SlavesDiagnostic_iHasError** is a variable of the type INTEGER.
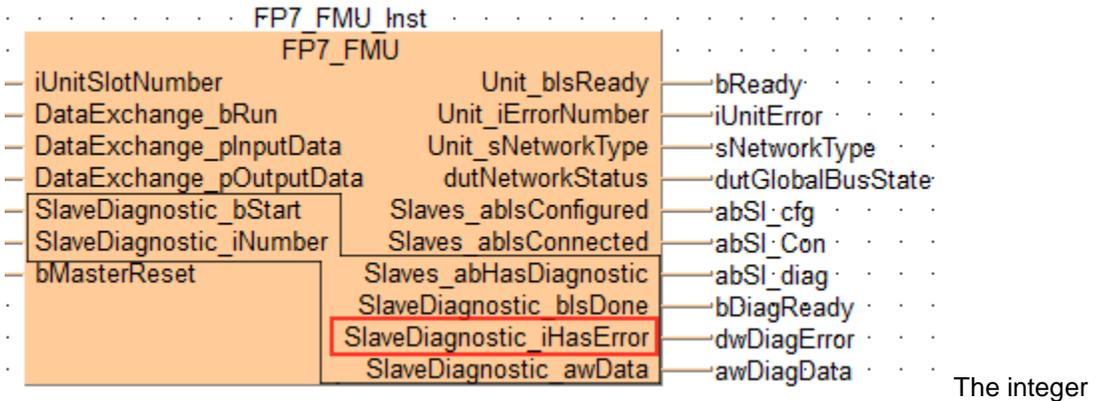


The integer stores the error code.

| Error | Indication | Network | Source | Recommendation |
|-------|-----------|---------|--------|----------------|
| **0** | No error. | All | | |
| **17** | No response from slave. | PROFIBUS | Slave | • Check network wiring.<br>• Check bus address of slave.<br>• Check baud rate setting. |
| **18** | Master not in logical token ring. | PROFIBUS | Network in general | • Check master DP address or highest-station address of other masters.<br>• Examine bus wiring for short circuits. |
| **161** | Remote address requested out of range. | All | Master | Check slave/node address in request message. |

## 1.3.7   SlaveDiagnostic_awData Output

The inputs and outputs found on the lower half of the function block help you diagnose problems.

Connected to the output pin **SlaveDiagnostic_awData** is a variable of the type ARRAY [0...3] of WORD.
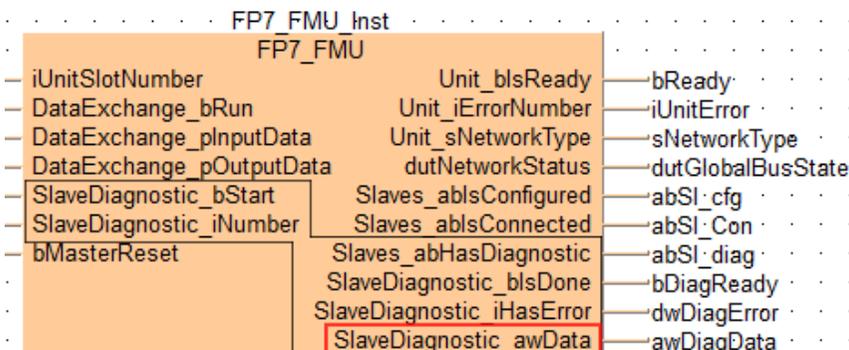


In this example, the identifier is **DiagData**. This variable is used for all network types, but the information stored in the variable is nevertheless network-specific:

- awDiagData for PROFIBUS (see page 25)

- awDiagData for DeviceNET (see page 26)

- awDiagData for CANopen (see page 28)

### 1.3.7.1  awDiagData for PROFIBUS

The ARRAYs of WORD contain the following diagnostic information.

#### ARRAY [0]: slave status

For ARRAY [0], slave status is stored as logical bits.
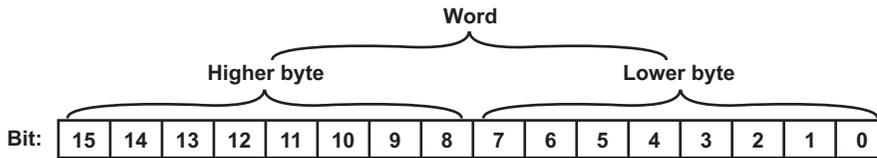
| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | Higher byte | | | | | | | | Lower byte | | | | | |
| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

| Bit | Description |
|---|---|
| 0 | Slave not responding. |
| 1 | Slave not ready. |
| 2 | Slave parameterized improperly. |
| 3 | Extended diagnostic area in ARRAY [3] of WORD used. |
| 4 | Unknown command detected by slave. |
| 5 | Inplausible response from slave. |
| 6 | Last parameter telegram faulty. |
| 7 | Slave parameterized by another master. |
| 8 | Slave must be parameterized. |
| 9 | Get diagnostic from slave until the bit is set to OFF (16#0000). |
| 10 | 1 |
| 11 | Watchdog activated. |
| 12 | Freeze command activated. |
| 13 | Sync command activated. |
| 14 | Reserved by system. |
| 15 | Slave not projected. |

#### ARRAY [1]: addtional slave status

For ARRAY [1], the lower byte contains additional slave status information. The higher byte contains the master address for the master that parameterized the slave.

| Bit | Description |
|---|---|
| 0-6 | Reserved by system. |
| 7 | The slave has more doagnostic data available than it can send. |
| 8-15 | The higher byte contains the master address for the master that parameterized the slave. If a slave is not parameterized, the value is 255. |

**ARRAY [2]: slave ID number**

In ARRAY [2], the slave reports its ID number.

**ARRAY [3]: extended diagnostic buffer**

ARRAY [3] is an extended diagnostic buffer. The values therein are fixed in the manual of the slave station or can be found in the PROFIBUS specifications.

## 1.3.7.2   awDiagData for DeviceNet

The ARRAYs of WORD contain the following diagnostic information.

**ARRAY [0], lower byte: device status**

For ARRAY [0], the lower byte stores device status as logical bits.
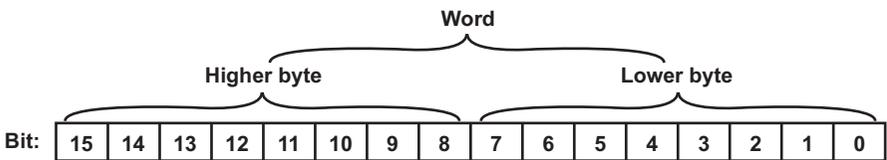


| Bit | Description |
|-----|-------------|
| 0 | Device not responding. |
| 1 | Reserved by system. |
| 2 | Device had denied access to at least one configured attribute to write in. |
| 3 | Difference between device produced and consumed connection size to the resulting configured ones. |
| 4-6 | Reserved by system. |
| 7 | Device is deactivated in current configuration and not handled. |

**ARRAY [0], higher byte: state of device**

For each device, the FMU has a "state machine handler". For ARRAY [0], the higher byte stores values that correspond to a state in the device.

| Value | Description |
|-------|-------------|
| 0 | State machine enter. |
| 1 | Device inactive, not handled. |
| 2 | Own MAC ID, state waiting for all incoming duplicate MAC ID requests. |
| 3 | Initialize internal predefined master slaves structures. |
| 4 | Allocated predefined master slave connection set request. |
| 5 | Wait for predefined master slave allocation connection response. |
| 6 | Release predefined master slave connection set request. |
| 7 | Wait for predefined master slave release connection response. |
| 8 | Initialize internal I/O configured structures. |
| 9 | Allocate configured I/O connection request. |
| 10 | Wait for I/O allocation response. |
| 11 | Release I/O connection request. |

| Value | Description |
|-------|-------------|
| 12 | Wait for I/O connection release response. |
| 13 | Read consumed connection size. |
| 14 | Wait for read consumed connection size response. |
| 15 | Compare consumed connection size with internal configured one. |
| 16 | Read produced connection size. |
| 17 | Wait for read produced connection size response. |
| 18 | Compare produced connection size with internal configured one. |
| 19 | Configure the I/O connection structures and register it. |
| 20 | Set expected packet rate. |
| 21 | Wait for set expected packet rate response. |
| 22 | I/O poll request 1st time. |
| 23 | Wait for I/O poll response. |
| 24 | I/O poll request 2nd time. |
| 25 | Wait for I/O poll response. |
| 26 | I/O poll request 3rd time. |
| 27 | Wait for I/O poll response. |
| 28 | Heart beat timeout to the device. |
| 30 | Open unconnected explicit connection request 1st time. |
| 31 | Wait for unconnected explicit connection response. |
| 32 | Open unconnected explicit connection request 2nd time. |
| 33 | Wait for unconnected explicit connection response. |
| 34 | Close unconnected connection request. |
| 35 | Wait for close unconnected connection response |
| 36 | Release all established connections request. |
| 37 | Wait for connection release response. |
| 38 | Open user unconnected explicit connection request. |
| 39 | Wait for user explicit connection response. |
| 40 | User predefined master slave allocate connection request |
| 41 | Wait for user allocation response. |
| 42 | User close unconnected connection request. |
| 43 | Wait for user close unconnected response. |
| 44 | Get or set user defined attribute request. |
| 45 | Wait for user defined get or set attribute response. |
| 46 | Send or wait fragmented get or set attribute. |

**ARRAY [1], lower byte: online error**

The lower byte of ARRAY [1] contains the actual online error of the device station is stored. See the table Err_Event of the global bus status field for possible entries (see page 11).

**ARRAY [1], higher byte: general error codes**

The higher byte ARRAY [1] provides more detailed general error codes if the value for Err_Event (see page 11) is 35.

| Value | Description |
|---|---|
| 2 | Resources unavailable. |
| 8 | Service not supported. |
| 9 | Invalid attribute value. |
| 11 | Already in request mode. |
| 12 | Object state conflict. |
| 14 | Attribute cannot be set. |
| 15 | Privilege violation. |
| 16 | Device state conflict. |
| 17 | Reply data too large. |
| 19 | Not enough data. |
| 20 | Attribute not supported. |
| 21 | Too much data. |
| 22 | Object does not exist. |

### ARRAY [2], lower byte: additional code

The lower byte of ARRAY [2] contains additional code.

This additional error information is only valid if value listed for general error codes in the higher byte of ARRAY [1] is not equal to 0. The value for the "additional code is filled transparently, just as with the general error codes, with the additional error code of each incoming error response message of the device.

### ARRAY [2], higher byte and ARRAY [3], lower byte: timeout information

The higher byte of ARRAY [2] and the lower byte of ARRAY [3] contain timeout information.

If a device is supervised by the expected packet rate of a connection and times out, the timer will be incremented. The actual value gives an overview of how good the transmission quality to this device is and how often a timeout has happened. After a device times out, the FMU always tries to reestablish the connection immediately.

## 1.3.7.3   awDiagData for CANopen

The ARRAYs of WORD contain the following diagnostic information.

### ARRAY [0], lower byte: node status

For ARRAY [0], the lower byte stores node status as logical bits.
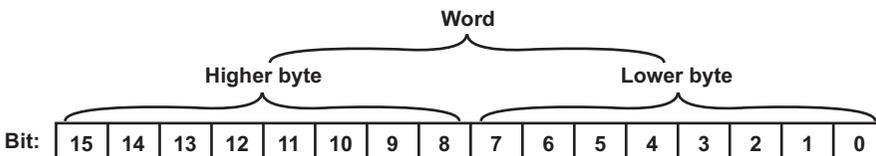


| Bit | Description |
|---|---|
| 0 | Node not responding. |
| 1 | Emergency buffer overflow. |
| 2 | Difference between master and node configuration data. |

| Bit | Description |
|-----|-------------|
| 3 | Node guarding protocol for this node is active. |
| 4-6 | Reserved by system. |
| 7 | Node is deactivated and not handled by the master. |

## ARRAY [0], higher byte and ARRAY [1], lower byte: extended information

These two bytes are read out from the node during startup. In the draft CiA specification, this word is declared as extended information of the node type. For example, whether the node supports digital input or outputs, etc. is fixed in this word.

## ARRAY [1], higher byte and ARRAY [2], lower byte: profile number

These two bytes are read out from the node during startup. Several predefined profile numbers exist, each described in its own specification manual. Here is an extract:

- Device profile for I/O modules: 401, 16#0191.
- Device profile for drives and motion control: 402, 16#0192.
- Device profile for encode: 406, 16#0196.

## ARRAY [2], higher byte: node state

If the node guarding protocol is active for this node, node status register read is written into this variable. The following values are defined in the CANopen specification.

| Value | Description |
|-------|-------------|
| 1 | Disconnected. |
| 2 | Connecting. |
| 3 | Preparing. |
| 4 | Prepared. |
| 5 | Operational. |
| 127 | Pre-operational. |

## ARRAY [3], lower byte: current error

In this byte the actual online error of this node station is held down. See the table containing "Err_Event" of the global bus status field (see page 15) for possible entries.

## ARRAY [3], higher byte: number of emergency messages saved

This byte contains the number of emergency messages saved in the following data area.

## ARRAY [4...23]: emergency messages

In this area the emergency messages are saved.