



PROGRAMMABLE CONTROLLER

FP Σ /FP2 Fieldbus Slave Units

Technical Manual

BEFORE BEGINNING

Liability and Copyright for the Hardware

This manual and everything described in it are copyrighted. You may not copy this manual, in whole or part, without written consent of Panasonic Electric Works Europe AG (PEWEU).

PEWEU pursues a policy of continuous improvement of the design and performance of its products. Therefore we reserve the right to change the manual/product without notice. In no event will PEWEU be liable for direct, special, incidental, or consequential damage resulting from any defect in the product or its documentation, even if advised of the possibility of such damages.

We invite your comments on this manual. Please e-mail us at:

tech-doc@eu.pewg.panasonic.com.

Please direct support matters and technical questions to your local Panasonic representative.

LIMITED WARRANTY

If physical defects caused by distribution are found, PEWEU will replace/repair the product free of charge. Exceptions include:

- When physical defects are due to different usage/treatment of the product other than described in the manual.
- When physical defects are due to defective equipment other than the distributed product.
- When physical defects are due to modifications/repairs by someone other than PEWEU.
- When physical defects are due to natural disasters.

Important Symbols

One or more of the following symbols may be used in this documentation:



DANGER!

The warning triangle indicates especially important safety instructions. If they are not adhered to, the results could be fatal or critical injury.



Indicates that you should proceed with caution. Failure to do so may result in injury or significant damage to instruments or their contents, e.g. data.



Contains important additional information.



Contains an illustrative example of the previous text section.



Indicates that a step-by-step procedure follows.



Indicates where you can find additional information on the subject at hand.



◆ KEY POINTS

Summarizes key points in a concise manner.



◆ SHORTCUTS

Provides helpful keyboard shortcuts.



◆ EXPLANATION

Provides a brief explanation of a function, e.g. why or when you should use it.

➡ next page

Indicates that the text will be continued on the next page.

Table of Contents

| | |
|---|-----------|
| 1. Features and Restrictions..... | 1 |
| 1.1 Fieldbus Slave Units | 2 |
| 1.2 Flexible Network Slave (FNS) Units and Blocks | 3 |
| 1.3 Expansion Restrictions and Current Limitations | 4 |
| 1.3.1 Expansion Restrictions for the FP2-FNS Unit | 4 |
| 1.3.2 Expansion Restrictions for the FPΣ FNS Unit | 4 |
| 1.3.3 Limitations on Current Consumption | 4 |
| 2. Parts and Functions | 5 |
| 2.1 Fieldbus Slave Units | 6 |
| 2.2 FP2 FNS Unit..... | 7 |
| 2.3 FPΣ FNS Unit..... | 8 |
| 2.4 FP-FNS Blocks | 9 |
| 2.4.1 FP-FNS Block PROFIBUS DP | 9 |
| 2.4.2 FP-FNS Block DeviceNet | 10 |
| 2.4.3 FP-FNS Block CANopen | 11 |
| 2.4.4 FP-FNS Block PROFINET IO..... | 13 |
| 3. Specifications | 15 |
| 3.1 FNS Unit General Specifications | 16 |
| 3.2 FP-FNS Block General Specifications | 17 |
| 3.2.1 FP-FNS Block PROFIBUS DP General Specifications | 17 |
| 3.2.2 FP-FNS Block DeviceNet General Specifications..... | 17 |

| | | |
|-----------|---|-----------|
| 3.2.3 | FP-FNS Block CANopen General Specifications | 18 |
| 3.2.4 | FP-FNS Block PROFINET IO General Specifications | 18 |
| 3.3 | FP-FNS Block Communication Specifications..... | 19 |
| 4. | Installation and Wiring..... | 21 |
| 4.1 | Fastening the FP-FNS Block | 22 |
| 4.2 | Removing the FP-FNS Block..... | 24 |
| 4.3 | Installation of the FP2/FPΣ Unit..... | 25 |
| 4.4 | Mounting Methods | 29 |
| 4.5 | Cable Selection | 30 |
| 4.6 | Wiring of the FP-FNS Blocks | 31 |
| 4.6.1 | FP-FNS Block PROFIBUS DP Wiring | 31 |
| 4.6.2 | DeviceNet Wiring | 31 |
| 4.6.3 | FP-FNS Block CANopen Wiring | 33 |
| 4.6.4 | FP-FNS Block PROFINET IO Wiring | 33 |
| 4.7 | Wiring of the FPΣ-FNS Unit..... | 34 |
| 5. | Programming Examples for FPWIN Pro | 37 |
| 5.1 | General Information | 38 |
| 5.2 | FNS_InitConfigDataTable Function..... | 39 |
| 5.3 | GetPointer Function..... | 40 |
| 5.4 | Programming Example, FP-FNS Block ProfibusDP | 41 |
| 5.4.1 | FNS_ProfibusDP Function Block | 44 |
| 5.5 | Programming Example, FP-FNS Block DeviceNet..... | 46 |
| 5.5.1 | FNS_DeviceNet Function Block | 48 |
| 5.6 | Programming Example, FP-FNS Block CANopen..... | 50 |
| 5.6.1 | FNS_CANopen Function Block | 53 |

5.7 Programming Example, FP-FNS Block Profinet IO 55

5.7.1 FNS_ProfinetIO Function Block 58

6. Outline Dimensions 61

6.1 Outline Dimensions of FP2-FNS Unit 62

6.2 Outline Dimensions of FPΣ FNS Unit 63

6.3 Dimensions of the FP-FNS Blocks..... 64

6.4 Dimensions with FNS Blocks and Cables..... 65

7. Index 67

Chapter 1

Features and Restrictions

1.1 Fieldbus Slave Units

FP2 and FPΣ (Sigma) Fieldbus Slave Units are preassembled to include a Flexible Network Slave (FNS) unit and the corresponding FP-FNS block. Panasonic decided to offer customers these preassembled products to save them time and to prevent damage to the pins in the FNS units, which bend if the FP-FNS blocks are inserted improperly.

You can still order the FNS units and FP-FNS blocks separately (see page 3). Please contact your local sales office.

You can download convenient function blocks for Control FPWIN Pro to help you program the FP-FNS blocks free of charge from the Panasonic Electric Works Europe AG Web site:
<http://www.panasonic-electric-works.com>.

FP2 Fieldbus Slave Units

| Name | Specifications | Part no. |
|-----------------------------|----------------|------------|
| FP2 PROFIBUS DP Slave Unit | PROFIBUS DP | FP2-DPV1-S |
| FP2 DeviceNet Slave Unit | DeviceNet | FP2-DEV-S |
| FP2 CANopen Slave Unit | CANopen | FP2-CAN-S |
| FP2 PROFINET IO Device Unit | PROFINET IO | FP2-PRT-S |

FPΣ Fieldbus Slave Units

| Name | Specifications | Part no. |
|-----------------------------|----------------|------------|
| FPΣ PROFIBUS DP Slave Unit | PROFIBUS DP | FPG-DPV1-S |
| FPΣ DeviceNet Slave Unit | DeviceNet | FPG-DEV-S |
| FPΣ CANopen Slave Unit | CANopen | FPG-CAN-S |
| FPΣ PROFINET IO Device Unit | PROFINET IO | FPG-PRT-S |

1.2 Flexible Network Slave (FNS) Units and Blocks

Flexible Network Slave (FNS) units are used together with the programmable controllers FP2/FP2SH and FPΣ. By exchanging compact FP-FNS blocks, you can connect to various networking systems without having to modify your entire hardware platform.

FP-FNS blocks are currently available for four bus systems: PROFIBUS, DeviceNet, CANopen and PROFINET IO. Others are planned for the future.

You can download convenient function blocks for Control FPDWIN Pro to help you program the FP-FNS blocks free of charge from the Panasonic Electric Works Europe AG Web site:
<http://www.panasonic-electric-works.com>.

FNS Units

| Name | Specifications | Part no. |
|---------------------------------|-------------------|----------|
| FP2 Flexible Network Slave Unit | Expansion for FP2 | FP2-FNS |
| FPΣ Flexible Network Slave Unit | Expansion for FPΣ | FPG-FNS |



◆ NOTE

The FNS unit cannot be used alone but must be used in combination with an FP-FNS block.

FP-FNS Blocks

| Name | Specifications | Part no. |
|--|----------------|-------------|
| FP-FNS Block (PROFIBUS DP) | PROFIBUS DP | AFPN-AB6200 |
| FP-FNS Block (DeviceNet) | DeviceNet | AFPN-AB6201 |
| FP-FNS Block (CANopen) (9-pin Sub-D male) | CANopen | AFPN-AB6218 |
| FP-FNS Block (PROFINET IO) | PROFINET IO | AFPN-AB6221 |

1.3 Expansion Restrictions and Current Limitations

1.3.1 Expansion Restrictions for the FP2-FNS Unit

The number of FP2-FNS units is restricted by the size of the FP2 backplane.

1.3.2 Expansion Restrictions for the FP Σ FNS Unit

The FP Σ -FNS units are connected to the left side of the control unit via the FP Σ expansion connector. Up to 4 expansion units can be connected to the left side of the control unit.

1.3.3 Limitations on Current Consumption

The 5V DC power used to drive the internal circuit of each unit is supplied from the power supply unit of the FP2 through the internal bus of the backplane or from the FP Σ control unit through the expansion connector.

Pay attention to the combination of units so that the rated capacity of the power supply is not exceeded.

Chapter 2

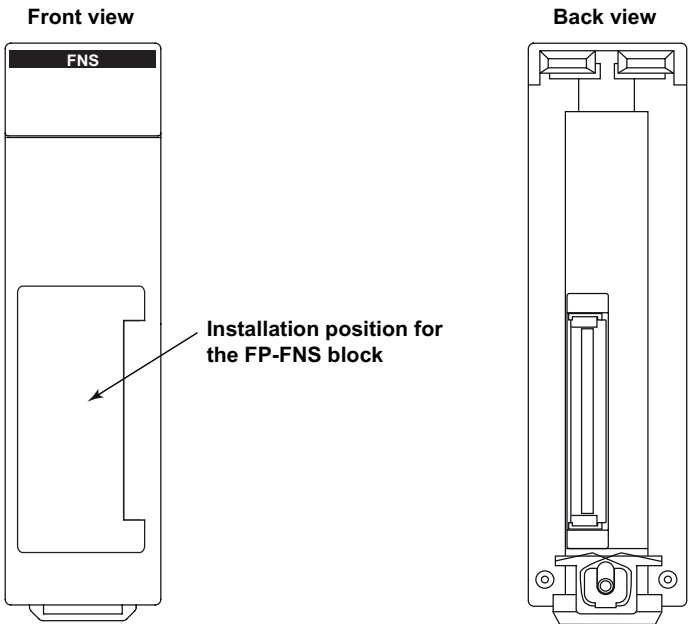
Parts and Functions

2.1 Fieldbus Slave Units

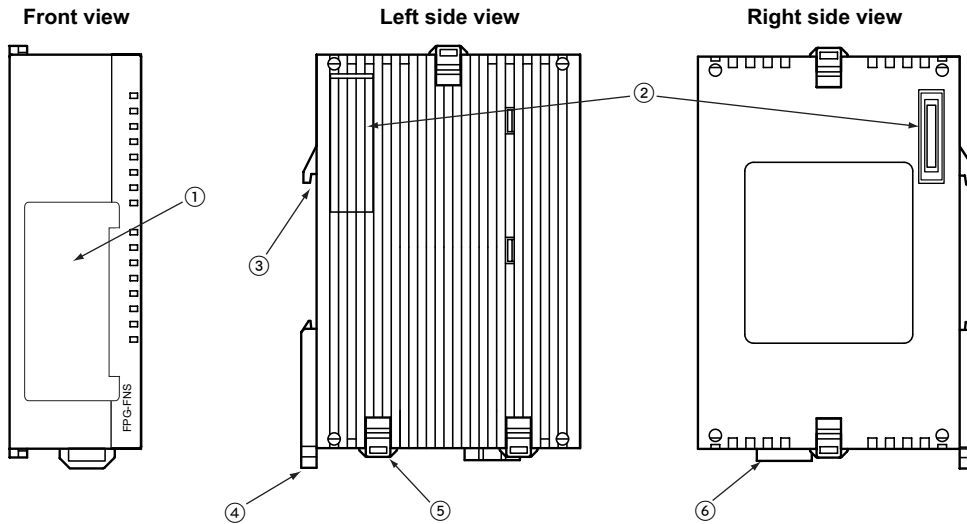
FP2 and FPΣ Fieldbus Slave Units (see page 2) are preassembled to include:

- an FP2 FNS unit (see page 7)
- or an FPΣ FNS unit (see page 8)
- and the corresponding FP-FNS block (see page 9).

2.2 FP2 FNS Unit



2.3 FPΣ FNS Unit



① **Installation position for FP-FNS block**

② **FPΣ expansion connector**

Used to connect the unit to the **control unit or other expansion units**.

③ **DIN standard rail attachment**

④ **DIN rail attachment lever**

Used for easy attachment to a DIN rail. The lever is also used for installation on the FP0 slim 30 type mounting plate (part no. AFP0811).

⑤ **Expansion hook**

Used to secure an expansion unit. The hook is also used for installation on the FP0 flat type mounting plate (part no. AFP0804).

⑥ **Function earth connector**

At least one of the pins must be connected to function earth to achieve proper EMC behavior.

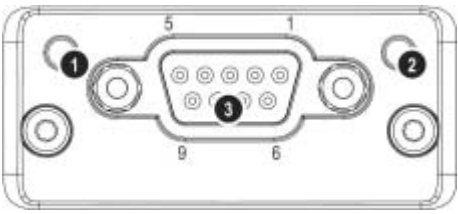
The FPΣ-FNS unit is connected to the left side of the control unit via the FPΣ expansion connector.

2.4 FP-FNS Blocks

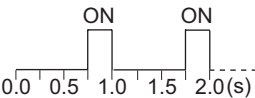
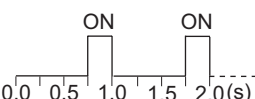
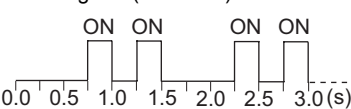
Various FP-FNS blocks are available to meet your networking needs.

2.4.1 FP-FNS Block PROFIBUS DP

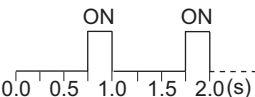
This FP-FNS block connects the unit to a PROFIBUS network.

| Front view | No. | Item |
|---|-----|---------------------------|
|  | 1 | Operation mode |
| | 2 | Status |
| | 3 | PROFIBUS connector (DB9F) |

Operation Mode

| State | Indication | Comments |
|---|------------------------------|---|
| Off | Not online/No power | - |
| Green | Online, data exchange | - |
| Flashing green  | Online, clear | - |
| Flashing red (1 flash)  | Parametrization error | - |
| Flashing red (2 flashes)  | PROFIBUS configuration error | Slave configuration does not match master configuration |

Status

| State | Indication | Comments |
|---|--|-------------------------------------|
| Off | No power or not initialized | FP-FNS state = 'SETUP' or 'NW_INIT' |
| Green | Initialized | FP-FNS has left the 'NW_INIT' state |
| Flashing green  | Initialized, diagnostic event(s) present | Extended diagnostic bit is set |
| Red | Exception error | FP-FNS state = 'EXCEPTION' |

PROFIBUS connector, DB9F, 9-pin Sub-D female

| Pin | Signal | Description |
|---------|---------------------------|--|
| 1 | - | - |
| 2 | - | - |
| 3 | B Line | Positive RxD/TxD, RS485 level |
| 4 | RTS | Request to send |
| 5 | GND | Bus ground (isolated) |
| 6 | +5V bus output (see note) | +5V termination power (isolated) |
| 7 | - | - |
| 8 | A Line | Negative RxD/TxD, RS485 level |
| 9 | - | - |
| Housing | Cable shield | <ul style="list-style-type: none"> FPΣ: Internally connected to the function earth connector of the FNS unit. FP2: Internally connected to the earth terminal of the power unit. |

**◆ NOTE**

Any current drawn from pin 6, the +5V bus output pin, will affect the total power consumption.

2.4.2 FP-FNS Block DeviceNet

This FP-FNS block connects the unit to a DeviceNet network.

| Front view | No. | Item |
|------------|-----|---------------------|
| | 1 | Network status LED |
| | 2 | Module status LED |
| | 3 | DeviceNet connector |

Network Status**◆ NOTE**

During start-up, an LED test is performed according to the DeviceNet standard.

| State | Indication |
|----------------------|---|
| Off | Not online/No power |
| Green | Online, one or more connections are established |
| Flashing green (1Hz) | Online, no connections established |
| Red | Critical link failure |

| State | Indication |
|--------------------|-----------------------------------|
| Flashing red (1Hz) | One or more connections timed out |

Module Status



◆ NOTE

During start-up, an LED test is performed according to the DeviceNet standard.

| State | Indication |
|----------------------|--|
| Off | No power or not initialized |
| Green | Operating in normal condition |
| Flashing green (1Hz) | Missing or incomplete configuration, device needs to be configured |
| Red | Unrecoverable fault(s) |
| Flashing red (1Hz) | Recoverable fault(s) |

DeviceNet Connector

| Pin | Signal | Description |
|-----|--------|--|
| 1 | V- | Negative bus supply voltage (see note) |
| 2 | CAN_L | CAN low bus line |
| 3 | SHIELD | Cable shield |
| 4 | CAN_H | CAN high bus line |
| 5 | V+ | Positive bus supply voltage (see note) |



◆ NOTE

Mandatory 24V bus power.

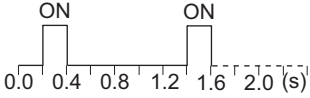
2.4.3 FP-FNS Block CANopen

This FP-FNS block connects the unit to a CANopen network.

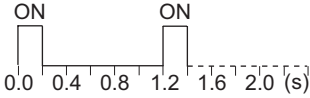
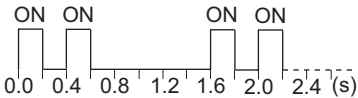
AFPN-AB6218

| Front view | No. | Item |
|------------|-----|-------------------|
| | 1 | RUN LED |
| | 2 | ERROR LED |
| | 3 | CANopen interface |

RUN

| State | Indication |
|---|--|
| Off | No power or device is in "Exception" state |
| Flickering green (10Hz) | Automatic baud rate detection |
| Single flash green  | Device stopped |
| Blinking green (2.5Hz) | Device is in "pre-operational" state |
| Green | "Operational" state |
| Red | Fatal event encountered. Bus interface is in physically passive state. |

ERROR

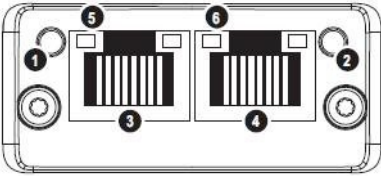
| State | Indication |
|--|---|
| Off | No power or device is in working condition |
| Single flash red  | A bus error counter has reached warning limit |
| Flickering red (10Hz) | LSS (Layer Setting Service) in progress |
| Double flash red  | Error control event has occurred |
| Red | Bus off or fatal event |

CANopen Interface for AFPN-AB6218

| Pin | Signal | Description |
|-----|---------|-----------------------------------|
| 1 | — | — |
| 2 | CAN_L | CAN low bus line (dominant low) |
| 3 | CAN_GND | Negative bus power supply input |
| 4 | — | — |
| 5 | — | — |
| 6 | — | — |
| 7 | CAN_H | CAN high bus line (dominant high) |
| 8 | — | — |
| 9 | — | — |

2.4.4 FP-FNS Block PROFINET IO

This FP-FNS block connects the unit to a PROFINET IO network.

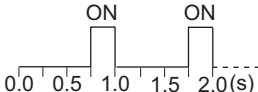
| Front view | No. | Item |
|---|-----|----------------------------|
|  | 1 | Network status LED |
| | 2 | Module status LED |
| | 3 | Ethernet port 1 |
| | 4 | Ethernet port 2 |
| | 5 | Link/Activity LED (port 1) |
| | 6 | Link/Activity LED (port 2) |

Network Status



◆ NOTE

During start-up, a test sequence is performed on this LED.

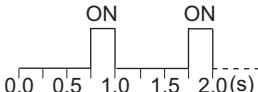
| State | Indication | Comments |
|--|---------------|---|
| Off | Offline | No power, or no connection with the IO controller |
| Green | Online (RUN) | <ul style="list-style-type: none">Connection with IO controller establishedIO controller in RUN state |
| Green, flashing  | Online (STOP) | <ul style="list-style-type: none">Connection with IO controller establishedIO controller in STOP state |

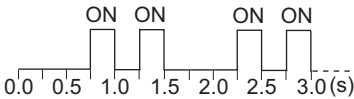
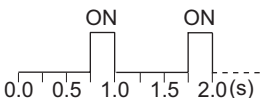
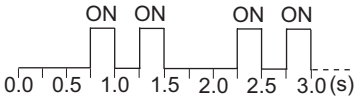
Module Status




◆ NOTE

During start-up, a test sequence is performed on this LED.

| State | Indication | Comments |
|---|-----------------------------|-------------------------------------|
| Off | No power or not initialized | FP-FNS state = 'SETUP' or 'NW_INIT' |
| Green | Normal operation | FP-FNS has left the 'NW_INIT' state |
| Green, 1 flash  | Diagnostic event(s) | Diagnostic event(s) present |

| State | Indication | Comments |
|---|---------------------|---|
| Green, 2 flashes  | Blink | Used by engineering tools to identify the node on the network. |
| Red | Exception error | FP-FNS state = 'EXCEPTION' |
| Red, 1 flash  | Configuration Error | Expected configuration by controller differs from real configuration. |
| Red, 2 flashes  | IP Address Error | IP address not set |
| Red, 3 flashes* | Station Name Error | Station Name not set |
| Red, 4 flashes* | Internal Error | FP-FNS has encountered a major internal error. |

*3 flashes: 

*4 flashes: 

LINK/Activity LED

| LED State | Indication | Comments |
|--------------------------|------------|---|
| Off | No Link | No link, no communication present |
| Green | Link | Ethernet link established, no communication present |
| Green, flickering (10Hz) | Activity | Ethernet link established, communication present |

Ethernet interface, RJ45

The Ethernet interface operates at 100Mbit, full duplex, as required by PROFINET.

Chapter 3

Specifications

3.1 FNS Unit General Specifications

| Item | Description |
|-----------------------|---|
| Operating temperature | 0 to +55°C/32 to +131°F |
| Storage temperature | -20 to +70°C/-4 to +158°F |
| Operating humidity | 30 to 85% RH (non-condensing) |
| Storage humidity | 30 to 85% RH (non-condensing) |
| Vibration resistance | 10 to 55Hz, 1 cycle/min: double amplitude of 0.75mm/0.030in., 10 min. on 3 axes |
| Shock resistance | Shock of 98m/s ² or more, 4 times on 3 axes |
| Operation condition | Free from corrosive gases and excessive dust |
| Current consumption | 55mA or less at 5V |
| Weight (main unit) | FP2-FNS: 88g FPΣ-FNS: 61g |

3.2 FP-FNS Block General Specifications

3.2.1 FP-FNS Block PROFIBUS DP General Specifications

| Item | Description |
|-----------------------|---|
| Operating temperature | 0 to +55°C/32 to +131°F |
| Storage temperature | -20 to +70°C/-4 to +158°F |
| Operating humidity | 30 to 85% RH (non-condensing) |
| Storage humidity | 30 to 85% RH (non-condensing) |
| Vibration resistance | 10 to 55Hz, 1 cycle/min: double amplitude of 0.75mm/0.030in., 10 min. on 3 axes |
| Shock resistance | Shock of 98m/s ² or more, 4 times on 3 axes |
| Immunity | EN61000-4-2, EN61000-4-3, EN61000-4-4, EN61000-4-5, EN61000-4-6 |
| Operation condition | Free from corrosive gases and excessive dust |
| Insulation resistance | Min. 100MΩ (measured with a 500V DC megger) |
| Breakdown voltage | 500V AC, 1 min. between DC external terminal and ground terminal |
| Current consumption | 230mA or less at 5V |
| Weight | 31g |

3.2.2 FP-FNS Block DeviceNet General Specifications

| Item | Description |
|-----------------------|--|
| Operating temperature | 0 to +55°C/32 to +131°F |
| Storage temperature | -20 to +70°C/-4 to +158°F |
| Operating humidity | 30 to 85% RH (non-condensing) |
| Storage humidity | 30 to 85% RH (non-condensing) |
| Vibration resistance | 10 to 55 Hz, 1 cycle/min: double amplitude of 0.75mm/0.030in., 10 min. on 3 axes |
| Shock resistance | Shock of 98m/s ² or more, 4 times on 3 axes |
| Immunity | EN61000-4-2, EN61000-4-3, EN61000-4-4, EN61000-4-5, EN61000-4-6 |
| Operation condition | Free from corrosive gases and excessive dust |
| Insulation resistance | Min. 100MΩ (measured with a 500 V DC megger) |
| Breakdown voltage | 500V AC, 1 min. between DC external terminal and ground terminal |
| Current consumption | 65mA or less at 5V; additional 140mA for bus power at 24V |
| Weight | 32g |

3.2.3 FP-FNS Block CANopen General Specifications

| Item | Description |
|-----------------------|---|
| Operating temperature | 0 to +55°C/32 to +131°F |
| Storage temperature | -20 to +70°C/-4 to +158°F |
| Operating humidity | 30 to 85% RH (non-condensing) |
| Storage humidity | 30 to 85% RH (non-condensing) |
| Vibration resistance | 10 to 55Hz, 1 cycle/min: double amplitude of 0.75mm/0.030in., 10 min. on 3 axes |
| Shock resistance | Shock of 98m/s ² or more, 4 times on 3 axes |
| Immunity | EN61000-4-2, EN61000-4-3, EN61000-4-4, EN61000-4-5, EN61000-4-6 |
| Operation condition | Free from corrosive gases and excessive dust |
| Insulation resistance | Min. 100MΩ (measured with a 500V DC megger) |
| Breakdown voltage | 500V AC, 1 min. between DC external terminal and ground terminal |
| Current consumption | 65mA or less at 5V; additional 140mA for bus power at 24V |
| Weight | 32g |

3.2.4 FP-FNS Block PROFINET IO General Specifications

| Item | Description |
|-----------------------|---|
| Operating temperature | 0 to +55°C/32 to +131°F |
| Storage temperature | -20 to +70°C/-4 to +158°F |
| Operating humidity | 30 to 85% RH (non-condensing) |
| Storage humidity | 30 to 85% RH (non-condensing) |
| Vibration resistance | 10 to 55Hz, 1 cycle/min: double amplitude of 0.75mm/0.030in., 10 min. on 3 axes |
| Shock resistance | Shock of 98m/s ² or more, 4 times on 3 axes |
| Immunity | EN61000-4-2, EN61000-4-3, EN61000-4-4, EN61000-4-5, EN61000-4-6 |
| Operation condition | Free from corrosive gases and excessive dust |
| Insulation resistance | Min. 100MΩ (measured with a 500V DC megger) |
| Breakdown voltage | 500V AC, 1 min. between DC external terminal and ground terminal |
| Current consumption | 375mA or less at 5V |
| Weight | 31g |

3.3 FP-FNS Block Communication Specifications

| Item | PROFIBUS | DeviceNet | CANopen | PROFINET IO |
|-------------------------------|--|---|--|---|
| Baud rate | <ul style="list-style-type: none"> Automatic baud rate detection 9.6kbaud to 12Mbaud | <ul style="list-style-type: none"> Automatic baud rate detection 125kbps to 500kbps | <ul style="list-style-type: none"> Automatic baud rate detection 10kbps to 1Mbps | <ul style="list-style-type: none"> 100Mbit/s full duplex |
| Isolation | Galvanically isolated bus electronics | Galvanically isolated bus electronics | Galvanically isolated bus electronics | Galvanically isolated bus electronics |
| Connection types | DP-V0: process data is accessed from the PROFIBUS network as cyclical I/O data | <ul style="list-style-type: none"> Cyclic connections COS (Change of State) Bit strobe connections Polled connections Explicit connections | PDO (Process Data Object) Exchange via: <ul style="list-style-type: none"> Cyclic Synchronous Acyclic Synchronous COS Timer-driven connections | <ul style="list-style-type: none"> PROFINET IO conformance class 2 Cyclic data exchange via PROFINET IO Real Time (RT) communication, 2ms cycle time |
| Maximum inputs/outputs | 76 words altogether for inputs and outputs (in units of 1, 2 or 4 words) | E.g. for cyclic connections: 128 words in each direction | 128 words (for TPDOs and RPDOs) | 128 words of Real Time I/O data in each direction |
| Additional features | Diagnostic support | <ul style="list-style-type: none"> UCMM capable CIP Parameter Object Diagnostic support | Diagnostic support | Diagnostic support |
| Interface | DB9F (9-pin Sub-D female) | 5-pin terminal block | <ul style="list-style-type: none"> 5-pin terminal block (AFPN-AB6202) 9-pin Sub-D male (AFPN-AB6218) | Integrated 2-port switch: 2 x RJ45 socket |

Chapter 4

Installation and Wiring

4.1 Fastening the FP-FNS Block



◆ CAUTION

Pins may bend!

To ensure that the pins in the FP-FNS do not bend or break, which will ruin the FP-FNS unit, read the following installation instructions carefully and follow them precisely.



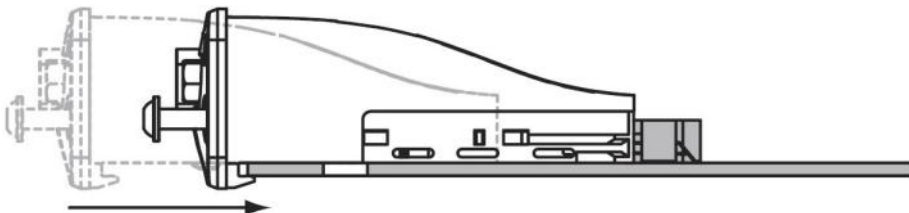
◆ NOTE

Make sure you are not electrostatically charged before you touch the FP-FNS block: the discharge of static electricity can damage parts and equipment.



◆ Procedure

1. **CAREFULLY** insert the FP-FNS block into the FNS unit's installation port. Do not force the block into the unit! Do not bend the pins!



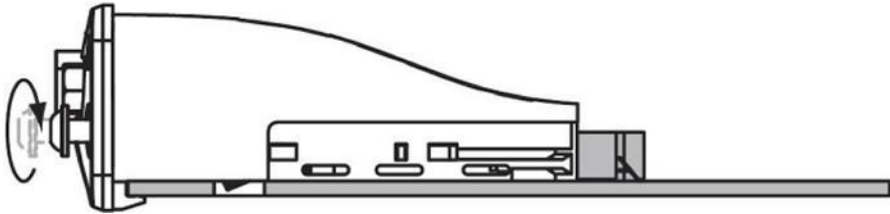
Make sure that the FP-FNS block is properly placed in the installation port of the FNS unit and properly guided in the slot so that there is no space between the FP-FNS block and the PCB.

2. **Push the FP-FNS block into the main unit until it stops. Do not force it!**

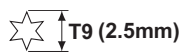
If the block stops with 5mm of space remaining until it is flush with the surface of the FP-FNS unit, the pins are not aligned properly! Pull the block out and reinsert it carefully, making sure it is properly guided.

3. While flush with the unit's surface, tighten the mounting screws.

Make sure the mounting mechanics fit into the fastening support holes of the PCB.



When tightening the FP-FNS block, use a TORX driver with a blade size of T9. The recommended tightening torque is 0.25Nm.



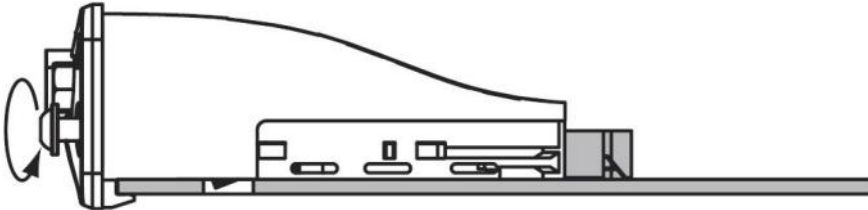
TORX® are registered trademarks of Acument™ Global Technologies.

4.2 Removing the FP-FNS Block

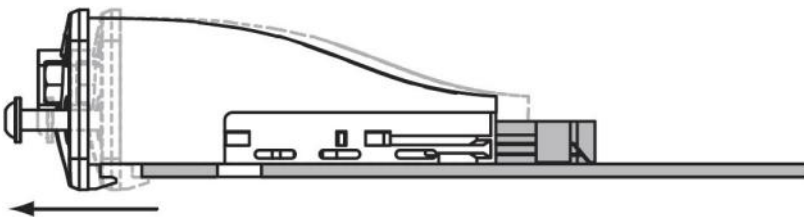


◆ Procedure

1. Loosen the mounting screws.



2. Pull the FP-FNS block out of the installation port of the FNS unit.



4.3 Installation of the FP2/FPΣ Unit



Warning!

Read the following notes carefully before installing the unit!

Failure to follow these instructions could lead to fire or damage the equipment.

Installation environment

- Be sure to install the unit in locations designed for electrical equipment, e.g. in a closed metal cabinet such as a switch cabinet.

Avoid installing the unit in the following locations:

- Ambient temperatures outside the range of 0°C to 55°C.
- Ambient humidity outside the range of 30% to 85% RH (at 25°C, non-condensing)
- Sudden temperature changes causing condensation
- Inflammable or corrosive gases
- Excessive airborne dust, metal particles or salts
- Benzine, paint thinner, alcohol or other organic solvents or strong alkaline solutions such as ammonia or caustic soda
- Excessive vibration or shock
- Direct sunlight
- Water or oil in any form including spray or mist

Static electricity

- Before touching the unit or equipment, always touch some grounded metal to discharge any static electricity you may have generated (especially in dry locations). The discharge of static electricity can damage parts and equipment.

Avoid noise interference from the following sources:

- Influence from power transmission lines, high voltage equipment, power cables, power equipment, radio transmitters, or any other equipment that would generate high switching surges.
- If noise occurs in the power supply line even after the above countermeasures are taken, it is recommended to supply power through an insulation transformer, noise filter, or the like.

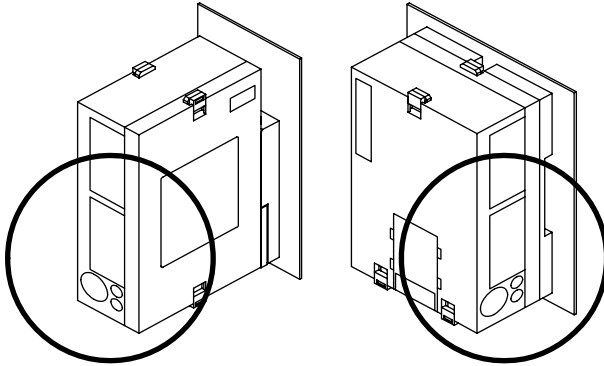
Cleaning

- Do not use thinner based cleaners because they deform the unit case and fade the

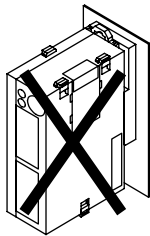
colors.

Measures regarding heat discharge

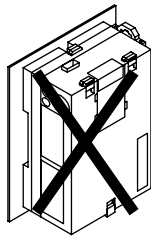
- Always install the CPU orientated with the TOOL port facing outward on the bottom in order to prevent the generation of heat.



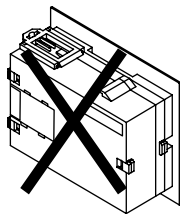
- Do **NOT** install the CPU as shown below.



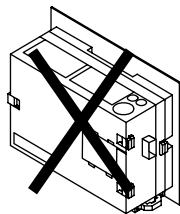
Upside-down



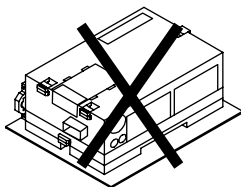
Air duct blocked



Input and output
connectors face down



Input and output
connectors on top



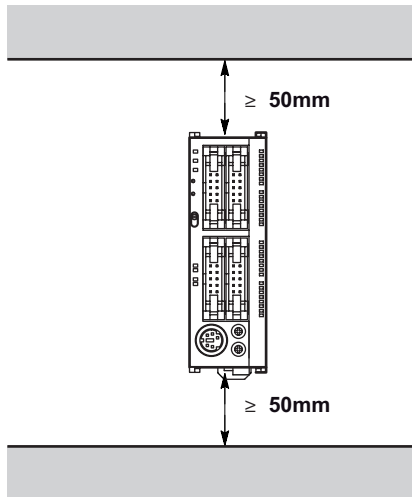
Horizontal
installation of the unit

- Do not install the unit above devices which generate heat such as heaters,

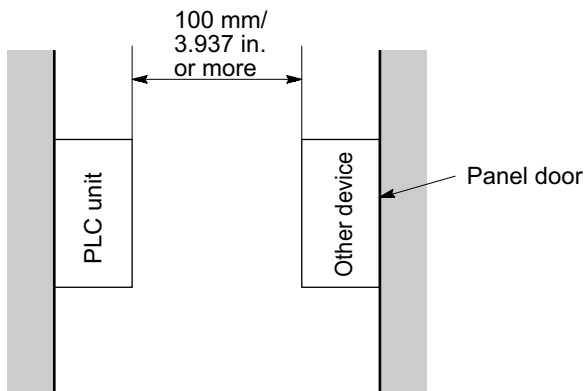
transformers or large scale resistors.

Installation space

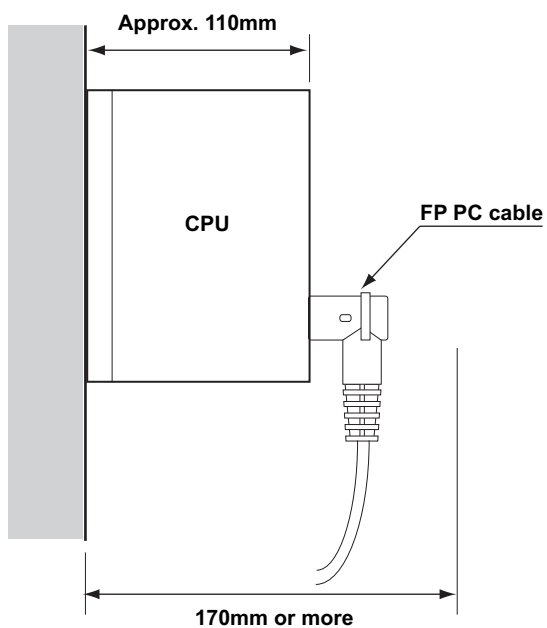
- Leave at least 50mm/1.97in. of space between the wiring ducts of the unit and other devices to allow heat radiation and unit replacement.



- Maintain a minimum of 100mm/3.937in. between devices to avoid adverse affects from noise and heat when installing a device or panel door to the front of the unit.



- For the FP2/FP2SH, keep the first 170mm from the PLC front surface clear of objects to allow the connecting of programming tools. For the FPΣ, the distance should be at least 130mm.



4.4 Mounting Methods

FPΣ-FNS Unit

You can attach up to 4 expansion units, including the FPΣ-FNS unit, to the left side of the FPΣ CPU. You can mount all units on a DIN rail.



◆ REFERENCE

For more information, please refer to the **FPΣ User's Manual**.

FP2-FNS Unit

Install the FP2-FNS unit on the FP2 backplane. You can mount the backplane on a DIN rail.



◆ REFERENCE

For more information, please refer to the **FP2 Hardware Manual**.

4.5 Cable Selection

Select a cable suitable for the network used.

PROFIBUS

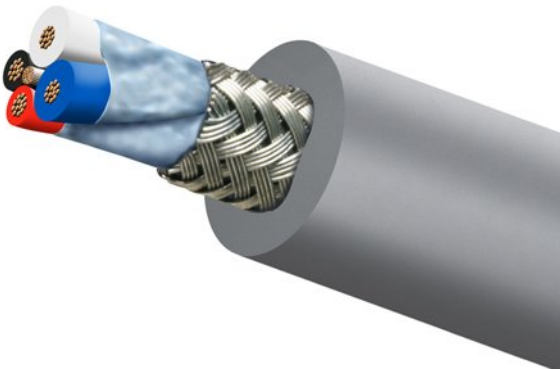
Use a standard PROFIBUS cable and a standard 9-pin Sub-D PROFIBUS connector.

CANopen

Use a standard CANopen cable and a standard 9-pin Sub-D CANopen connector.

DeviceNet

Use a standard DeviceNet cable.



The round cable contains five wires: one twisted pair (red and black) for 24V DC power, one twisted pair (blue and white) for signal, and a drain wire (bare).

You can find proposals for standard cables on the Open DeviceNet Vendor Association's Web site (ODVA): <http://www.odva.org>. (<http://www.odva.org/default.aspx?tabid=84>)

PROFINET

Use a standard PROFINET Ethernet cable and a standard RJ45 connector.

4.6 Wiring of the FP-FNS Blocks

4.6.1 FP-FNS Block PROFIBUS DP Wiring

Use a standard PROFIBUS cable and standard 9-pin Sub-D male PROFIBUS connectors.

We recommend using a straight (0°) bus interface connector (e.g. PR 103-658). When a horizontal (90°) bus interface connector is used, the cables will be directed toward the top of the unit, which may cause difficulties when installing other devices in a control cabinet.

4.6.2 DeviceNet Wiring

Open style connector/suitable wire

DeviceNet has a standard open style connector.

If additional connectors are needed, use the standard CAN 5-poles open style connectors manufactured by Phoenix Contact.

| No. of contacts | Phoenix Contact product ID | |
|-----------------|-----------------------------|-------------|
| 5 | Model no. | Product no. |
| | MSTB 2,5/ 5-ST-5,08 ABGY AU | 1849037 |



Terminal block for DeviceNet

For a suitable wire, please refer to cable selection (see page 30).

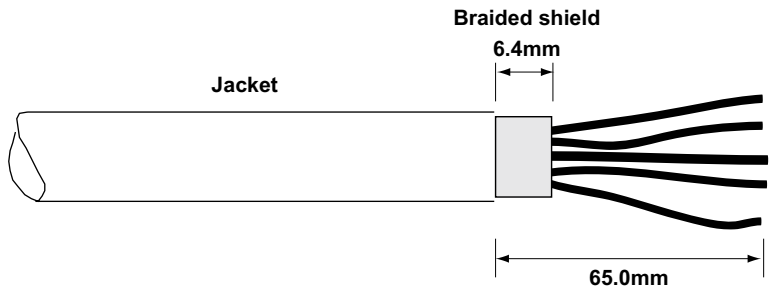
Wiring method

Attach a plug-in, open style connector to a cable.

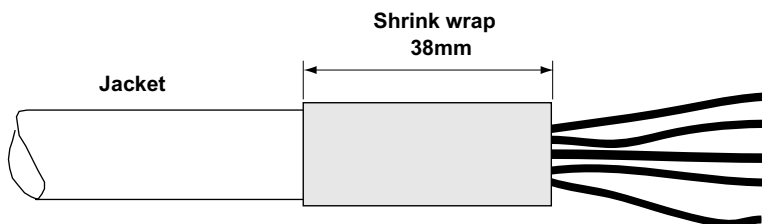


◆ Procedure

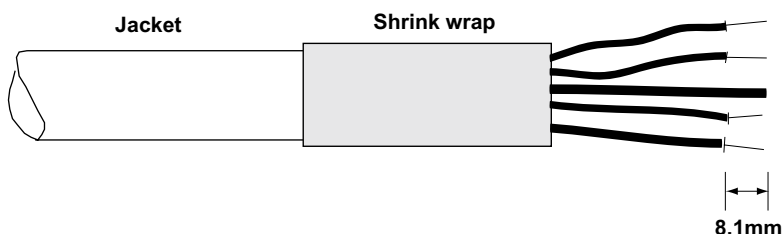
1. Strip 65mm (2.6in.) to 75mm (3in.) of the outer jacket from the end of the cable, leaving no more than 6.4mm (0.25in.) of the braided shield exposed.



2. Wrap the end of the cable with 38mm (1.5in.) of shrink wrap, covering part of the exposed conductors and part of the trunk line insulation.



3. Strip 8.1mm (0.32in.) of the insulation from the end of each of the insulated conductors.



4. Insert each conductor into the appropriate clamping cavity of the open style connector or the screw terminal on the device, according to the color of the cable insulation:

| Wire color | Wire identity | Usage |
|------------|---------------|--------|
| White | CAN_H | Signal |
| Blue | CAN_L | Signal |
| Bare | Drain | Shield |
| Black | V- | Power |
| Red | V+ | Power |

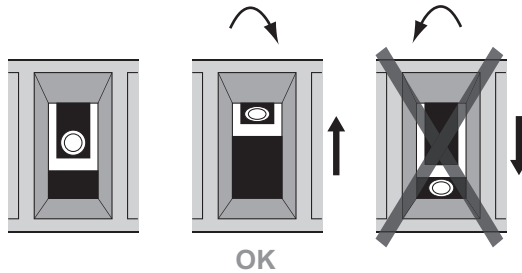
5. Tighten the clamping screws to secure each conductor. The male contacts of the device connector must match the female contacts of the connector.



◆ NOTE

- When removing the wire's insulation, be careful not to scratch the core wire.
- Do not twist the wires to connect them.

- **Do not solder the wires to connect them. The solder may break due to vibration.**



4.6.3 FP-FNS Block CANopen Wiring

Use a standard CANopen cable and standard 9-pin Sub-D female CANopen connectors.

We recommend using a straight (0°) bus interface connector. When a horizontal (90°) bus interface connector is used, the cables will be directed toward the top of the unit, which may cause difficulties when installing other devices in a control cabinet.

4.6.4 FP-FNS Block PROFINET IO Wiring

PROFINET uses a transmission rate of 100Mbit/s in full-duplex mode for data communication. Therefore, the cables used must fulfill these requirements. Use a standard, shielded, twisted-pair Ethernet cable (100 BASE TX) with a minimum category 5 rating and at least four wires. For example, STP5 is a shielded, twisted pair cable of category 5.

Please use standard RJ45 connectors. RJ45 connectors are available with different IP degrees of protection.

The maximum distance between two devices should not exceed 100m.

4.7 Wiring of the FPΣ-FNS Unit

The FPΣ-FNS unit has a spring-cage connection type (2-pin) or screw (3-pin) terminal block on its lower side to connect to function earth. As the pins are internally bridged, one of the pins should be connected to function earth for proper EMC behaviour. Use the following items for wiring.

Accessory terminal block

If additional connectors are needed, use the connector manufactured by Phoenix Contact.

| No. of contacts | Phoenix Contact Model no. | Phoenix Contact Product no. |
|-----------------|---------------------------|-----------------------------|
| 2 | FK-MC 0.5/2-ST-2.5 | 18 81 32 5 |
| 3 | MC 1,5/ 3-ST-3,5 | 18 40 37 9 |

Suitable wire for spring-cage connection type terminal (2-pin)

| No. of wires | Size | Cross-sectional area |
|--------------|-----------|-------------------------|
| 1 | AWG 26-20 | 0.14-0.5mm ² |

Suitable wire for screw terminal (3-pin)

| No. of wires | Size | Cross-sectional area |
|--------------|-----------|-------------------------|
| 1 | AWG 28-16 | 0.14-1.5mm ² |



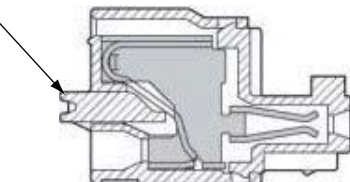
◆ NOTE

Either fixed or flexible wires can be used to connect the function earth.

Wiring method for the spring-cage connection type

Fixed wires with a diameter > 0.2mm² and flexible wires with a wire end ferrule can be plugged in the clamp. When using smaller diameters or flexible wires without a ferrule, you must push the orange opening lever to plug in the wire.

Opening lever



◆ CAUTION

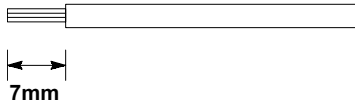
- When removing the wire's insulation, be careful not to scratch the core wire.
- Do not twist the wires to connect them.

- Do not solder the wires to connect them. The solder may break from vibrations.
- After wiring, make sure stress is not applied to the wire.

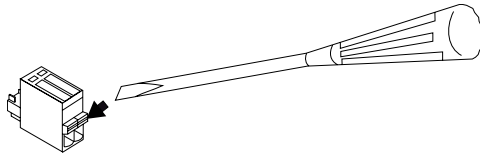


◆ Procedure

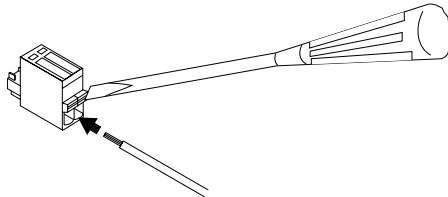
1. Remove a portion of the wire's insulation.



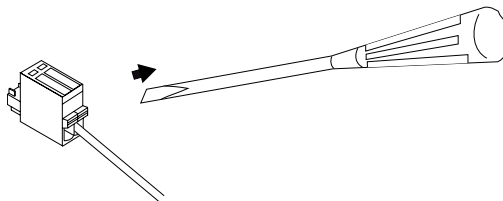
2. Press the orange opening lever of the connector using a tool such as a flat-blade screwdriver.



3. Insert the wire into the connector until it stops while pressing the opening lever.



4. Release the opening lever.



Chapter 5

Programming Examples for FPWIN Pro

5.1 General Information

In these programming examples for Control FPWIN Pro, several different functions and function blocks are used, which are explained in the following sections.

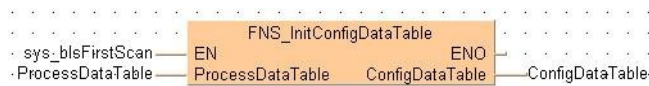
Make sure you use at least version 5.2.3 of FPWIN Pro, into which the functions necessary for programming the FP-FNS blocks are integrated.

These example programs are used to configure the various FNS units and to start communication with the specific network.

The functions and function blocks used in these programming examples depend on the FP-FNS block used. They can be used for either the FP2-FNS or FP Σ -FNS unit.

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

5.2 FNS_InitConfigDataTable Function



The **FNS_InitConfigDataTable** function creates a **ConfigDataTable** from the variable **ProcessDataTable**. This **ConfigDataTable** is necessary to configure the FP-FNS block.

Make sure that the size of the variable **ConfigDataTable** corresponds to the structure of the **ProcessDataTable**, e.g. if the **ProcessDataTable** consists of three entries, then the **ConfigDataTable** variable should be an "Array[0..2] of WORD", whose size matches the number of entries. If the **ProcessDataTable** variable has only one entry (e.g. WORD), then the **ConfigDataTable** variable should be an "Array[0..0] of WORD" (with size 1).

Allowed data types for the input of the **FNS_InitConfigDataTable** are all 16-bit (INT, WORD), 32-bit (DINT, DWORD, TIME (32 bits), REAL) and 64-bit variables or arrays of them. 64-bit variables are defined as 2-dimensional arrays, e.g. "Array[0..0,0..3] of INT" is a 64-bit variable, while "Array[0..3] of INT" represents an array with four elements of 16-bit variables.

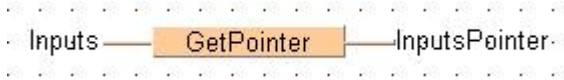
The data types BOOL, STRING and arrays of these types are NOT allowed at the input of the function **FNS_InitConfigDataTable**.

The output **ConfigDataTable** of the function must be an array of WORD.

In the programming example, both variables **ConfigIn** and **ConfigOut** must have a size of three to accommodate the three elements of the DUT's inputs and outputs.

If no inputs or no outputs are used, just omit the corresponding network when creating the configuration data.

5.3 GetPointer Function



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 FPCWIN Pro online help.

5.4 Programming Example, FP-FNS Block ProfibusDP

After you install the FNS Library, you can start programming.



◆ Procedure

1. Create the Data Unit Types (DUTs) for inputs and outputs.
2. Create input and output variables of the type of DUT generated in the previous step in the global variable list.
3. Generate the configuration data table for inputs and outputs by using the function `FNS_InitConfigDataTable` (see page 39). Make sure that the size of the `FNS_InitConfigDataTable` output variable corresponds to the DUT.
4. Create pointers of the input, output and `ConfigDataTable` variables and provide them to the `FNS_ProfibusDP` function block together with the corresponding variables.

Data Unit Types (DUTs)

In the following picture you can see all possible data types and how the different variables (16-bit, 32-bit and 64-bit) can be defined.

| Input [DUT] | | | | |
|-------------|-------------------------------|---------------------------|-----------|-------------------------|
| | Identifier | Type | Initial | Comment |
| 0 | INT_16bits | INT | 0 | |
| 1 | WORD_16bits | WORD | 0 | |
| 2 | DINT_32bits | DINT | 0 | |
| 3 | DWORD_32bits | DWORD | 0 | |
| 4 | REAL_32bits | REAL | 0.0 | |
| 5 | TIME_32bits | TIME | T#0s | |
| 6 | SIXTYFOUR_bits_INT | ARRAY [0..0,0..3] OF INT | [4(0)] | second dimension size 4 |
| 7 | SIXTYFOUR_bits_WORD_v1 | ARRAY [0..0,0..3] OF WORD | [4(0)] | second dimension size 4 |
| 8 | SIXTYFOUR_bits_WORD_v2 | ARRAY [0..0,1..4] OF WORD | [4(0)] | second dimension size 4 |
| 9 | TWO_Elements_of_INT_16bits | ARRAY [0..1] OF INT | [2(0)] | |
| 10 | THREE_Elements_of_WORD_16bits | ARRAY [0..2] OF WORD | [3(0)] | |
| 11 | FOUR_Elements_of_DINT_32bits | ARRAY [0..3] OF DINT | [4(0)] | |
| 12 | FIVE_Elements_of_DWORD_32bits | ARRAY [0..4] OF DWORD | [5(0)] | |
| 13 | SIX_Elements_of_REAL_32bits | ARRAY [0..5] OF REAL | [6(0.0)] | |
| 14 | SEVEN_Elements_of_TIME_32bits | ARRAY [0..6] OF TIME | [7(T#0s)] | |
| 15 | EIGHT_Elements_of_64bits | ARRAY [0..7,0..3] OF INT | [32(0)] | second dimension size 4 |
| 16 | NINE_Elements_of_64bits | ARRAY [0..8,0..3] OF WORD | [36(0)] | second dimension size 4 |
| 17 | | | | |

64-bit variables are declared by creating a two-dimensional array, whereas the second dimension must have a size of four. The first dimension specifies the number of elements of this type.

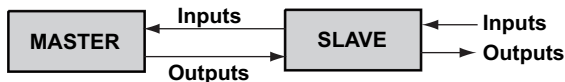
In this programming example both variables, the input and output process data, consist of three elements: a 16-bit, a 32-bit and a 64-bit variable:

| Input [DUT] | | | | |
|-------------|------------|------------------------|---------|---------|
| | Identifier | Type | Initial | Comment |
| 0 | I_1W | INT | 0 | |
| 1 | I_2W | DINT | 0 | |
| 2 | I_4W | ARRAY [0..0,0..3] O... | [4(0)] | |

| Output [DUT] | | | | |
|--------------|------------|------------------------|---------|---------|
| | Identifier | Type | Initial | Comment |
| 0 | O_1W | INT | 0 | |
| 1 | O_2W | DINT | 0 | |
| 2 | O_4W | ARRAY [0..0,0..3] O... | [4(0)] | |

Input process data represents data that will be sent to the master. Thus, from the slave's point-of-view, it has to be regarded as output data.

Output process data represents data received from the master. Thus, from the slave's point-of-view, it has to be regarded as input data.



◆ NOTE

The order in which inputs and outputs are mapped to the process data is significant and must be replicated in the master configuration. Inputs are mapped to the process data previous to the outputs.

Global Variable List

To use the DUTs for further programming and to pass on the process data to an application program declare the following global variable with the type of DUT that was created in the previous step. The global variables are afterwards accessed by the variable class VAR_EXTERNAL in the example program's header.

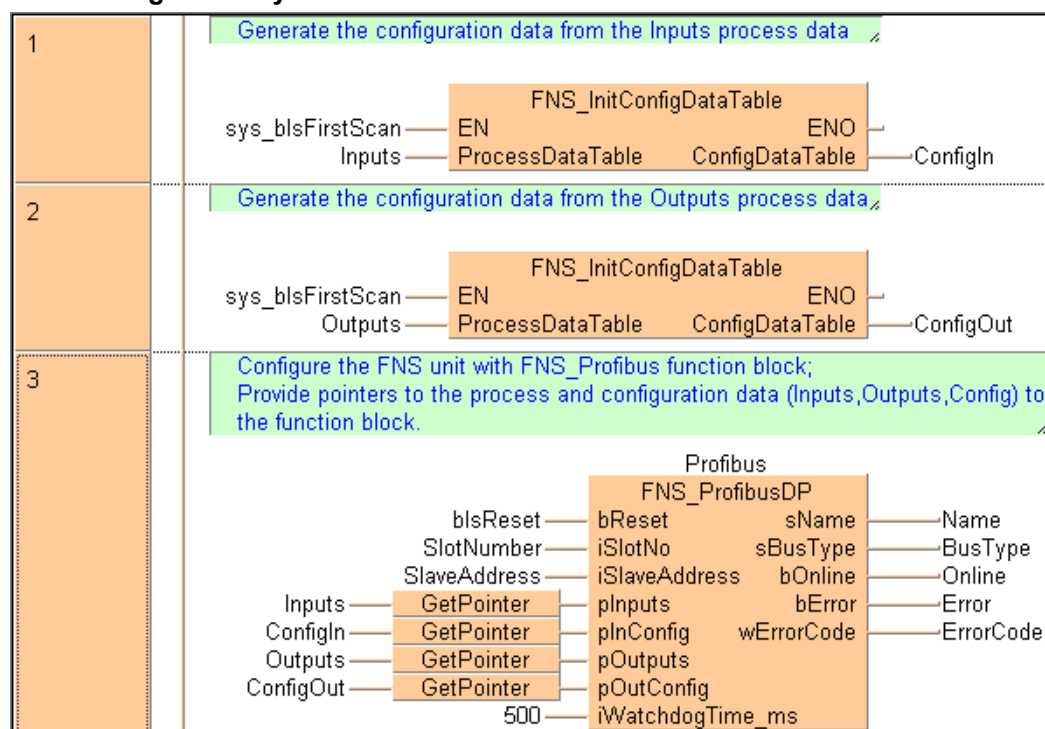
| Global Variables | | | | | | | |
|------------------|------------|------------|--------|-------------|--------|---------|--------------------------|
| | Class | Identifier | FP ... | IEC Address | Type | Initial | A... |
| 0 | VAR_GLOBAL | Inputs | | | Input | | <input type="checkbox"/> |
| 1 | VAR_GLOBAL | Outputs | | | Output | | <input type="checkbox"/> |

POU Header

| Profibus | | | | | |
|----------|--------------|--------------|----------------------|---------|---------|
| | Class | Identifier | Type | Initial | Comment |
| 0 | VAR | SlotNumber | INT | 0 | |
| 1 | VAR | SlaveAddress | INT | 2 | |
| 2 | VAR | ConfigIn | ARRAY [0..2] OF WORD | [3(0)] | |
| 3 | VAR | ConfigOut | ARRAY [0..2] OF WORD | [3(0)] | |
| 4 | VAR | Name | STRING[16] | " | |
| 5 | VAR | BusType | STRING[16] | " | |
| 6 | VAR | Online | BOOL | FALSE | |
| 7 | VAR | Error | BOOL | FALSE | |
| 8 | VAR | ErrorCode | WORD | 0 | |
| 9 | VAR | bIsReset | BOOL | FALSE | |
| 10 | VAR | Profibus | FNS_ProfibusDP | | |
| 11 | VAR_EXTERNAL | Inputs | Input | | |
| 12 | VAR_EXTERNAL | Outputs | Output | | |

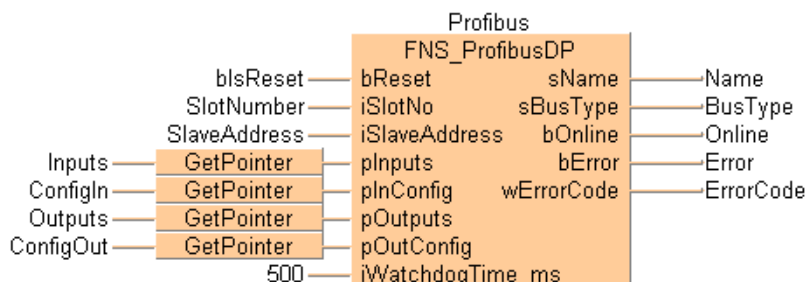
In the POU header, all variables that are required for the program are declared. The size of the variables **ConfigIn** and **ConfigOut** must correspond to the number of entries in the DUTs input and output.

Ladder Diagram Body



In the ladder diagram body you can see an instance of the **FNS_ProfibusDP** function block called **ProfibusDP**, and how the inputs, outputs and configuration data have to be supplied to the function block.

5.4.1 FNS_ProfibusDP Function Block



The FNS_ProfibusDP function block configures the FP-FNS block ProfibusDP. It has to be supplied with information about the configuration, the input and output size, and network-specific data.

If no inputs or no outputs are used, just leave the corresponding pins unconnected.

PLC types: available for FP2/FP2SH and FPΣ.

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

Inputs

| Input | Data Type | Function |
|------------------|-----------|---|
| bReset | BOOL | Reset pin; network block will be reset while bReset is set |
| iSlotNo | INT | Installation position of the FNS unit |
| iSlaveAddress | INT | PROFIBUS slave address. Values from 0 to 125. |
| pInputs | POINTER | Pointer to the input's process data table |
| pInConfig | POINTER | Pointer to the input's configuration data table |
| pOutputs | POINTER | Pointer to the output's process data table |
| pOutConfig | POINTER | Pointer to the output's configuration data table |
| iWatchdogTime_ms | POINTER | Watchdog timeout value for unit in ms. Valid values from 1 to 32767. 0: default of 700ms. |

Outputs

| Output | Data Type | Function |
|------------|------------|--|
| sName | STRING[16] | Name of installed FP-FNS block |
| sBusType | STRING[20] | Network type of installed FP-FNS block |
| bOnline | BOOL | Flag for online status |
| bError | BOOL | Error flag |
| wErrorCode | WORD | Error code if error flag is set |

List of error codes for the FP-FNS block ProfibusDP

| Errorcode | Indication |
|-----------|--|
| 16#0000 | No error |
| 16#0001 | PROFIBUS configuration error: master and slave configuration do not correspond |
| 16#0002 | Process data area is too large (max.76 words) |
| 16#0005 | FP-FNS block is not installed correctly |
| 16#0007 | FP-FNS block has incorrect provider ID |
| 16#0008 | Wrong FP-FNS block installed |
| 16#0009 | Invalid slave address |
| 16#000A | Exception state entered; application watchdog timeout |

5.5 Programming Example, FP-FNS Block DeviceNet

After you install the FNS Library, you can start programming.



◆ Procedure

1. Create the Data Unit Types (DUTs) for inputs and outputs.
2. Create input and output variables of the type of DUT generated in the previous step in the global variable list.
3. Generate the configuration data table for inputs and outputs by using the function `FNS_InitConfigDataTable` (see page 39). Make sure that the size of the `FNS_InitConfigDataTable` output variable corresponds to the DUT.
4. Create pointers of the input, output and `ConfigDataTable` variables and provide them to the `FNS_DeviceNet` function block together with the corresponding variables.

Data Unit Types (DUTs)

In the following picture you can see all possible data types and how the different variables (16-bit, 32-bit and 64-bit) can be defined.

| | Identifier | Type | Initial | Comment |
|----|-------------------------------|---------------------------|-----------|-------------------------|
| 0 | INT_16bits | INT | 0 | |
| 1 | WORD_16bits | WORD | 0 | |
| 2 | DINT_32bits | DINT | 0 | |
| 3 | DWORD_32bits | DWORD | 0 | |
| 4 | REAL_32bits | REAL | 0.0 | |
| 5 | TIME_32bits | TIME | T#0s | |
| 6 | SIXTYFOUR_bits_INT | ARRAY [0..0,0..3] OF INT | [4(0)] | second dimension size 4 |
| 7 | SIXTYFOUR_bits_WORD_v1 | ARRAY [0..0,0..3] OF WORD | [4(0)] | second dimension size 4 |
| 8 | SIXTYFOUR_bits_WORD_v2 | ARRAY [0..0,1..4] OF WORD | [4(0)] | second dimension size 4 |
| 9 | TWO_Elements_of_INT_16bits | ARRAY [0..1] OF INT | [2(0)] | |
| 10 | THREE_Elements_of_WORD_16bits | ARRAY [0..2] OF WORD | [3(0)] | |
| 11 | FOUR_Elements_of_DINT_32bits | ARRAY [0..3] OF DINT | [4(0)] | |
| 12 | FIVE_Elements_of_DWORD_32bits | ARRAY [0..4] OF DWORD | [5(0)] | |
| 13 | SIX_Elements_of_REAL_32bits | ARRAY [0..5] OF REAL | [6(0.0)] | |
| 14 | SEVEN_Elements_of_TIME_32bits | ARRAY [0..6] OF TIME | [7(T#0s)] | |
| 15 | EIGHT_Elements_of_64bits | ARRAY [0..7,0..3] OF INT | [32(0)] | second dimension size 4 |
| 16 | NINE_Elements_of_64bits | ARRAY [0..8,0..3] OF WORD | [36(0)] | second dimension size 4 |

64-bit variables are declared by creating a two-dimensional array, whereas the second dimension must have a size of four. The first dimension specifies the number of elements of this type.

In this programming example both variables, the input and output process data, consist of three elements: a 16-bit, a 32-bit and a 64-bit variable:

| Input [DUT] | | | | |
|-------------|------------|------------------------|---------|---------|
| | Identifier | Type | Initial | Comment |
| 0 | I_1W | INT | 0 | |
| 1 | I_2W | DINT | 0 | |
| 2 | I_4W | ARRAY [0..0,0..3] O... | [4(0)] | |

| Output [DUT] | | | | |
|--------------|------------|------------------------|---------|---------|
| | Identifier | Type | Initial | Comment |
| 0 | O_1W | INT | 0 | |
| 1 | O_2W | DINT | 0 | |
| 2 | O_4W | ARRAY [0..0,0..3] O... | [4(0)] | |

Produced data represents data that will be sent to the master. Thus, from the slave's point-of-view, it has to be regarded as output data.

Consumed data represents data received from the master. Thus, from the slave's point-of-view, it has to be regarded as input data.



Global Variable List

To use the DUTs for further programming and to pass on the process data to an application program declare the following global variable with the type of DUT that was created in the previous step. The global variables are afterwards accessed by the variable class VAR_EXTERNAL in the example program's header.

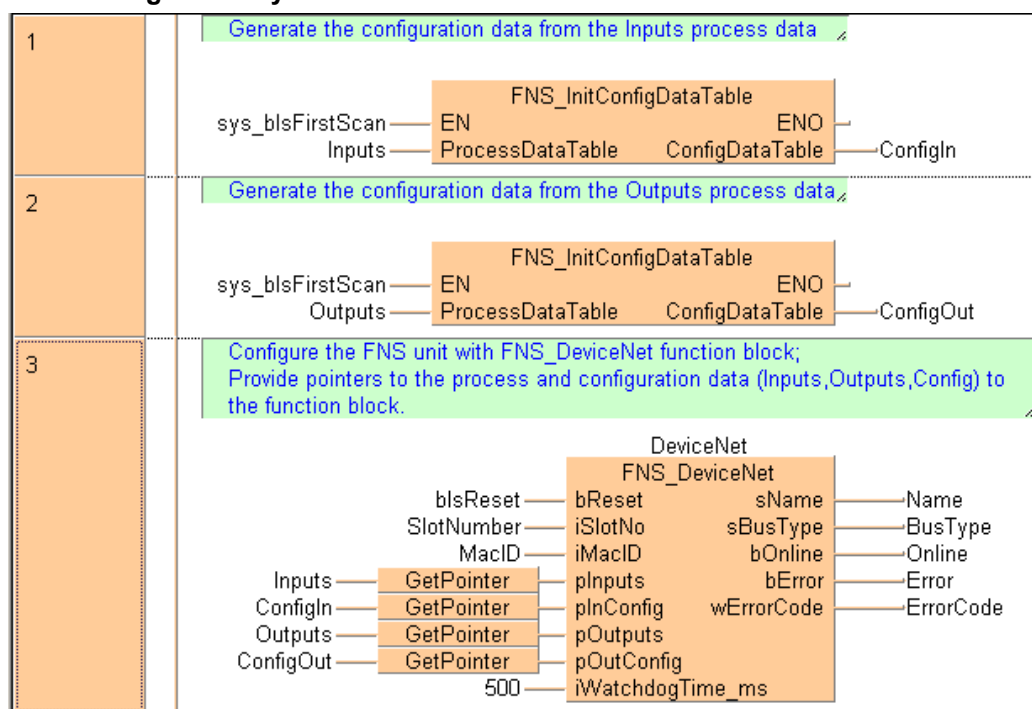
| Global Variables | | | | | | | | |
|------------------|------------|------------|--------|-------------|--------|---------|--------------------------|---------|
| | Class | Identifier | FP ... | IEC Address | Type | Initial | A... | Comment |
| 0 | VAR_GLOBAL | Inputs | | | Input | | <input type="checkbox"/> | |
| 1 | VAR_GLOBAL | Outputs | | | Output | | <input type="checkbox"/> | |

POU Header

| DeviceNet | | | | | | |
|-----------|--------------|------------|----------------------|---------|---------|--|
| | Class | Identifier | Type | Initial | Comment | |
| 0 | VAR | SlotNumber | INT | 0 | | |
| 1 | VAR | MacID | INT | 2 | | |
| 2 | VAR | ConfigIn | ARRAY [0..2] OF WORD | [3(0)] | | |
| 3 | VAR_EXTERNAL | Outputs | Output | | | |
| 4 | VAR | ConfigOut | ARRAY [0..2] OF WORD | [3(0)] | | |
| 5 | VAR | Name | STRING[16] | | | |
| 6 | VAR | BusType | STRING[16] | | | |
| 7 | VAR | Online | BOOL | FALSE | | |
| 8 | VAR | Error | BOOL | FALSE | | |
| 9 | VAR | ErrorCode | WORD | 0 | | |
| 10 | VAR_EXTERNAL | Inputs | Input | | | |
| 11 | VAR | bIsReset | BOOL | FALSE | | |
| 12 | VAR | DeviceNet | FNS_DeviceNet | | | |

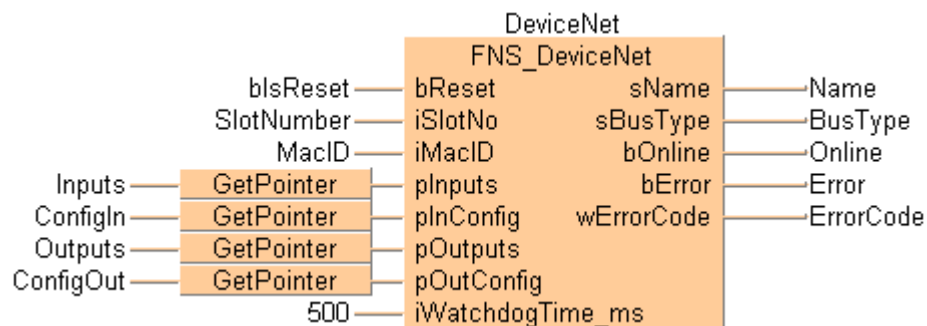
In the POU header, all variables that are required for the program are declared. The size of the variables **ConfigIn** and **ConfigOut** must correspond to the number of entries in the DUTs input and output.

Ladder Diagram Body



In the ladder diagram body you can see an instance of the **FNS_DeviceNet** function block called **DeviceNet**, and how the inputs, outputs and configuration data have to be supplied to the function block.

5.5.1 FNS_DeviceNet Function Block



The **FNS_DeviceNet** function block configures the FP-FNS block **DeviceNet**. It has to be supplied with information about the configuration, the input and output size and network-specific data.

If no inputs or no outputs are used, just leave the corresponding pins unconnected.

PLC types: available for FP2/FP2SH and FPΣ.

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

Inputs

| Input | Data Type | Function |
|------------------|-----------|---|
| bReset | BOOL | Reset pin; network block will be reset while bReset is set |
| iSlotNo | INT | Installation position of the FNS unit |
| iMacID | INT | DeviceNet address; Values from 0 to 63. |
| pInInputs | POINTER | Pointer to the input's process data table |
| pInConfig | POINTER | Pointer to the input's configuration data table |
| pOutOutputs | POINTER | Pointer to the output's process data table |
| pOutConfig | POINTER | Pointer to the output's configuration data table |
| iWatchdogTime_ms | INT | Watchdog timeout value for unit in ms. Valid values from 1 to 32767. 0: default of 700ms. |

Outputs

| Output | Data Type | Function |
|------------|------------|--|
| sName | STRING[16] | Name of installed FP-FNS block |
| sBusType | STRING[20] | Network type of installed FP-FNS block |
| bOnline | BOOL | Flag for online status |
| bError | BOOL | Error flag |
| wErrorCode | WORD | Error code if error flag is set |

List of error codes for FP-FNS block DeviceNet

| Errorcode | Indication |
|-----------|---|
| 16#0000 | No error |
| 16#0002 | Process data area is too large (max.128 Words in each direction) |
| 16#0003 | Reset Request Error |
| 16#0004 | Bus off or cable disconnected, or no connection established between master and slave (wrong Mac ID or process data configuration) |
| 16#0005 | FP-FNS block is not installed correctly |
| 16#0007 | FP-FNS block has incorrect provider ID |
| 16#0008 | Wrong FP-FNS block installed |
| 16#0009 | Invalid Mac ID |
| 16#000A | Exception state entered; application watchdog timeout; unit needs resetting |

5.6 Programming Example, FP-FNS Block CANopen

After you install the FNS Library, you can start programming.



◆ Procedure

1. Create the Data Unit Types (DUTs) for inputs and outputs.
2. Create input and output variables of the type of DUT generated in the previous step in the global variable list.
3. Generate the configuration data table for inputs and outputs by using the function `FNS_InitConfigDataTable` (see page 39). Make sure that the size of the `FNS_InitConfigDataTable` output variable corresponds to the DUT.
4. Create pointers of the input, output and `ConfigDataTable` variables and provide them to the `FNS_CANopen` function block together with the corresponding variables.

Data Unit Types (DUTs)

In the following picture you can see all possible data types and how the different variables (16-bit, 32-bit and 64-bit) can be defined.

| CANStructure [DUT] | | | | |
|--------------------|--------------------|---------------------------|---------|---------|
| | Identifier | Type | Initial | Comment |
| 0 | INT_16bit | INT | 0 | |
| 1 | INT_16bit_v2 | ARRAY [0..0] OF INT | [1(0)] | |
| 2 | ARRAY_INT_4x16bit | ARRAY [0..3] OF INT | [4(0)] | |
| 3 | WORD_16bit | WORD | 0 | |
| 4 | WORD_16bit_v2 | ARRAY [0..0] OF WORD | [1(0)] | |
| 5 | ARRAY_WORD_3x16bit | ARRAY [0..2] OF WORD | [3(0)] | |
| 6 | DINT_32bit | DINT | 0 | |
| 7 | DWORD_32bit | DWORD | 0 | |
| 8 | REAL_32bit | REAL | 0.0 | |
| 9 | TIME_32bit | TIME | T#0s | |
| 10 | INTEGER64 | ARRAY [0..0,0..3] OF INT | [4(0)] | |
| 11 | UNSIGNED64 | ARRAY [0..0,0..3] OF WORD | [4(0)] | |
| 12 | UNSIGNED64_v2 | ARRAY [0..0,1..4] OF WORD | [4(0)] | |

64-bit variables are declared by creating a two-dimensional array, whereas the second dimension must have a size of four. The first dimension specifies the number of elements of this type.

In the CANopen network, each entry of the DUT is represented as one PDO (Process Data Object). Each PDO can carry up to 4 words (64 bits) of data. The `FNS_CANopen` function block supports up to 32 TPDOs and 32 RPDOs. The exact representation of the process data depends on the structure of the .eds file. Only the data types that are supported in each .eds file can be used.

The .eds files from Panasonic Electric Works Europe AG only support one data type for all 32 RPDOs and 32 TPDOs, so please choose the .eds file that best suits your needs. The following .eds files are available at the moment:

- FNS_32PDO_UNSIGNED.EDS, supports the data type unsigned16 (WORD) only
- FNS_32PDO_INTEGER.EDS, supports the data type integer16 (INT) only
- FNS_32PDO_64UNSIGNED.EDS, supports the data type unsigned64 only
- FNSCO4_64IO.EDS, only four RPDOs and four TPDOs are supported

If a mixture of data types is used, you can either handle the data in your application program or use the file "FNSCO4_64IO.EDS". This file only supports up to four TPDOs and four RPDOs, but several different data types can be mixed. Please note: only one entry per PDO is allowed, so each PDO can consist of one data type only.

Independent of the .eds file used, due to the mapping scheme of the process data, a PDO can only be composed of variables of the same data type.

Each entry of the DUT is represented as an individual manufacturer-specific object in the CANopen object dictionary, whereas each element of a DUT is assigned to one subindex of the object, according to the table below. DUTs with one element can be regarded as a one-dimensional array with one element; DUTs with more than one element (arrays) are represented as a one-dimensional array with several elements.

In this programming example both variables, the input and output process data, consist of three elements:

- a 16-bit integer variable (PDO1)
- an array of a 16-bit integer variable with 2 elements (PDO2)
- an array of a 16-bit integer variable with 4 elements (PDO3)

| InputCANStructure [DUT] | | | | |
|-------------------------|------------|---------------------|---------|---------|
| | Identifier | Type | Initial | Comment |
| 0 | PDO1 | INT | 0 | |
| 1 | PDO2 | ARRAY [0..1] OF INT | [2(0)] | |
| 2 | PDO3 | ARRAY [0..3] OF INT | [4(0)] | |

| OutputCANStructure [DUT] | | | | |
|--------------------------|------------|---------------------|---------|---------|
| | Identifier | Type | Initial | Comment |
| 0 | PDO1 | INT | 0 | |
| 1 | PDO2 | ARRAY [0..1] OF INT | [2(0)] | |
| 2 | PDO3 | ARRAY [0..3] OF INT | [4(0)] | |

Thus in this programming example, the input structure **InputCANStructure** can be found at the following indexes:

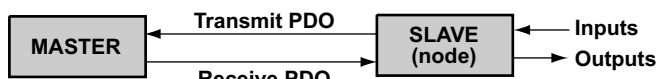
- InputsCAN PDO1: index **2001h**, subindex **01h**
- InputsCAN PDO2: index **2002h**, subindex **01h** and **02h**
- InputsCAN PDO3: index **2003h**, subindex **01h** to **04h**

According to the list above, the output structure **OutputCANStructure** can be found at the following indexes:

- OutputsCAN PDO1: index **2021h**, subindex **01h**
- OutputsCAN PDO2: index **2022h**, subindex **01h** and **02h**
- OutputsCAN PDO3: index **2023h**, subindex **01h** to **04h**

Transmit PDO represents data that will be sent to the master. Thus, from the slave's point-of-view, it has to be regarded as output data.

Receive PDO represents data received from the master. Thus, from the slave's point-of-view, it has to be regarded as input data.



Global Variable List

To use the DUTs for further programming and to pass on the process data to an application program declare the following global variable with the type of DUT that was created in the previous step. The global variables are afterwards accessed by the variable class VAR_EXTERNAL in the example program's header.

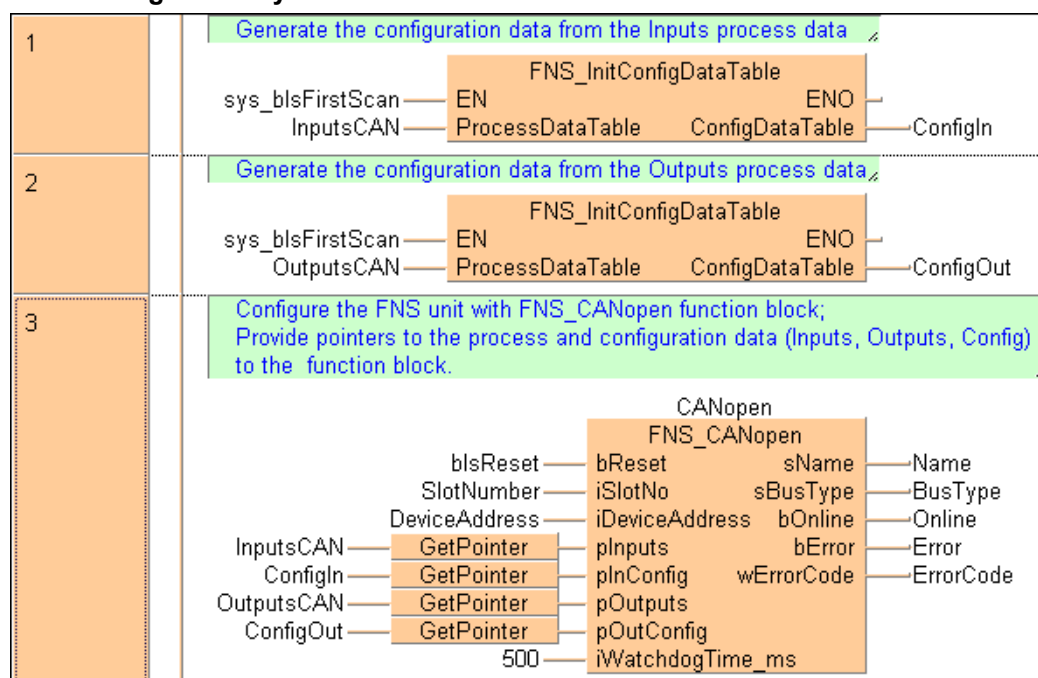
| Global Variables | | | | | | | | |
|------------------|------------|------------|------------|-------------|-------------------|---------|--------|-------|
| | Class | Identifier | FP Addr... | IEC Addr... | Type | Initial | Aut... | Comme |
| 2 | VAR_GLOBAL | InputsCAN | | | InputCANStructure | | | |
| 3 | VAR_GLOBAL | OutputsCAN | | | OuputCANStructure | | | |

POU Header

| CANopen | | | | | |
|---------|--------------|---------------|----------------------|---------|---------|
| | Class | Identifier | Type | Initial | Comment |
| 0 | VAR | SlotNumber | INT | 0 | |
| 1 | VAR | DeviceAddress | INT | 2 | |
| 2 | VAR | ConfigIn | ARRAY [0..2] OF WORD | [3(0)] | |
| 3 | VAR | ConfigOut | ARRAY [0..2] OF WORD | [3(0)] | |
| 4 | VAR | Name | STRING[16] | " | |
| 5 | VAR | BusType | STRING[16] | " | |
| 6 | VAR | Online | BOOL | FALSE | |
| 7 | VAR | Error | BOOL | FALSE | |
| 8 | VAR | ErrorCode | WORD | 0 | |
| 9 | VAR | bIsReset | BOOL | FALSE | |
| 10 | VAR | CANopen | FNS_CANopen | | |
| 11 | VAR_EXTERNAL | InputsCAN | InputCANStructure | | |
| 12 | VAR_EXTERNAL | OutputsCAN | OuputCANStructure | | |

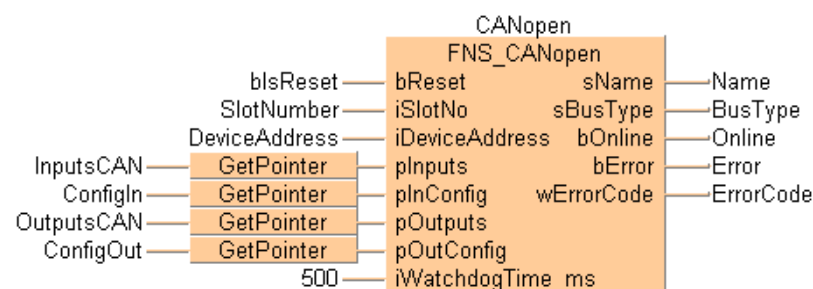
In the POU header, all variables that are required for the program are declared. The size of the variables **ConfigIn** and **ConfigOut** must correspond to the number of entries in the DUTs input and output.

Ladder Diagram Body



In the ladder diagram body you can see an instance of the **FNS_CANopen** function block called **CANopen**, and how the inputs, outputs and configuration data have to be supplied to the function block.

5.6.1 FNS_CANopen Function Block



The **FNS_CANopen** function block configures the FP-FNS block **CANopen**. It has to be supplied with information about the configuration, the input and output size and network-specific data.

If no inputs or no outputs are used, just leave the corresponding pins unconnected.

PLC types: available for FP2/FP2SH and FPΣ.

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

Inputs

| Input | Data Type | Function |
|------------------|-----------|---|
| bReset | BOOL | Reset pin; network block will be reset while bReset is set. |
| iSlotNo | INT | Installation position of the FNS unit |
| iDeviceAddress | INT | CANopen address; values from 1 to 127. |
| pInputs | POINTER | Pointer to the input's process data table |
| pInConfig | POINTER | Pointer to the input's configuration data table |
| pOutputs | POINTER | Pointer to the output's process data table |
| pOutConfig | POINTER | Pointer to the output's configuration data table |
| iWatchdogTime_ms | INT | Watchdog timeout value for unit in ms. Valid values from 1 to 32767. 0: default of 700ms. |

Outputs

| Output | Data Type | Function |
|------------|------------|--|
| sName | STRING[16] | Name of installed FP-FNS block |
| sBusType | STRING[20] | Network type of installed FP-FNS block |
| bOnline | BOOL | Flag for online status |
| bError | BOOL | Error flag |
| wErrorCode | WORD | Error code if error flag is set |

List of error codes for FP-FNS block CANopen

| Errorcode | Indication |
|-----------|---|
| 16#0000 | No error |
| 16#0002 | Process data area is too large (max. 32 PDOs, i.e. max. 128 words in each direction) |
| 16#0003 | Reset request error |
| 16#0004 | Bus off or cable disconnected, or no connection established between master and slave (wrong device address or process data configuration) |
| 16#0005 | FP-FNS block is not installed correctly |
| 16#0007 | FP-FNS block has incorrect provider ID |
| 16#0008 | Wrong FP-FNS block installed |
| 16#0009 | Invalid device address |
| 16#000A | Exception state entered; application watchdog timeout; unit needs resetting |

5.7 Programming Example, FP-FNS Block Profinet IO

After you install the FNS Library, you can start programming.



◆ Procedure

1. Create the Data Unit Types (DUTs) for inputs and outputs.
2. Create input and output variables of the type of DUT generated in the previous step in the global variable list.
3. Generate the configuration data table for inputs and outputs by using the function `FNS_InitConfigDataTable` (see page 39). Make sure that the size of the `FNS_InitConfigDataTable` output variable corresponds to the DUT.
4. Create pointers of the input, output and `ConfigDataTable` variables and provide them to the `FNS_ProfinetIO` function block together with the corresponding variables.

Data Unit Types (DUTs)

In the following picture you can see all possible data types and how the different variables (16-bit, 32-bit and 64-bit) can be defined.

| Input [DUT] | | | | |
|----------------------------------|---------------------------|-----------|-------------------------|--|
| Identifier | Type | Initial | Comment | |
| 0 INT_16bits | INT | 0 | | |
| 1 WORD_16bits | WORD | 0 | | |
| 2 DINT_32bits | DINT | 0 | | |
| 3 DWORD_32bits | DWORD | 0 | | |
| 4 REAL_32bits | REAL | 0.0 | | |
| 5 TIME_32bits | TIME | T#0s | | |
| 6 SIXTYFOUR_bits_INT | ARRAY [0..0,0..3] OF INT | [4(0)] | second dimension size 4 | |
| 7 SIXTYFOUR_bits_WORD_v1 | ARRAY [0..0,0..3] OF WORD | [4(0)] | second dimension size 4 | |
| 8 SIXTYFOUR_bits_WORD_v2 | ARRAY [0..0,1..4] OF WORD | [4(0)] | second dimension size 4 | |
| 9 TWO_Elements_of_INT_16bits | ARRAY [0..1] OF INT | [2(0)] | | |
| 10 THREE_Elements_of_WORD_16bits | ARRAY [0..2] OF WORD | [3(0)] | | |
| 11 FOUR_Elements_of_DINT_32bits | ARRAY [0..3] OF DINT | [4(0)] | | |
| 12 FIVE_Elements_of_DWORD_32bits | ARRAY [0..4] OF DWORD | [5(0)] | | |
| 13 SIX_Elements_of_REAL_32bits | ARRAY [0..5] OF REAL | [6(0.0)] | | |
| 14 SEVEN_Elements_of_TIME_32bits | ARRAY [0..6] OF TIME | [7(T#0s)] | | |
| 15 EIGHT_Elements_of_64bits | ARRAY [0..7,0..3] OF INT | [32(0)] | second dimension size 4 | |
| 16 NINE_Elements_of_64bits | ARRAY [0..8,0..3] OF WORD | [36(0)] | second dimension size 4 | |
| 17 | | | | |

64-bit variables are declared by creating a two-dimensional array, whereas the second dimension must have a size of four. The first dimension specifies the number of elements of this type.

The FNS PROFINET IO Device handles the plugging of modules and submodules automatically according to the following scheme:

- A DAP (Device Access Point) is plugged into Slot 0
- Modules are added beginning with the DUT Inputs followed by the DUT Outputs
- Each module occupies a single slot
- Each entry of a DUT results in one module being added
- One-dimensional array entries in a DUT result in an equal number of modules being

added

- Two-dimensional array entries in a DUT (used for 64-bit variables) result in the same number of modules being added as the size of the first dimension of the array.
- One sub-module per module

Each slot can carry up to 4 words (64 bits) of data. The FNS_ProfinetIO function block supports up to 64 slots for input and/or output process data. Only the data types that are supported in the GSDML-file (.xml) can be used.

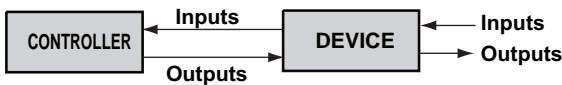
In this programming example both variables, the input and output process data, consist of three elements: a 16-bit, a 32-bit and a 64-bit variable:

| Input [DUT] | | | | |
|-------------|------------|------------------------|---------|---------|
| | Identifier | Type | Initial | Comment |
| 0 | I_1W | INT | 0 | |
| 1 | I_2W | DINT | 0 | |
| 2 | I_4W | ARRAY [0..0,0..3] O... | [4(0)] | |

| Output [DUT] | | | | |
|--------------|------------|------------------------|---------|---------|
| | Identifier | Type | Initial | Comment |
| 0 | O_1W | INT | 0 | |
| 1 | O_2W | DINT | 0 | |
| 2 | O_4W | ARRAY [0..0,0..3] O... | [4(0)] | |

Input process data represents data that will be sent to the controller. Thus, from the device's point-of-view, it has to be regarded as output data.

Output process data represents data received from the controller. Thus, from the device's point-of-view, it has to be regarded as input data.



◆ NOTE

The order in which inputs and outputs are mapped to the process data is significant and must be replicated in the master configuration. Inputs are mapped to the process data previous to the outputs.

Global Variable List

To use the DUTs for further programming and to pass on the process data to an application program declare the following global variable with the type of DUT that was created in the previous step. The global variables are afterwards accessed by the variable class VAR_EXTERNAL in the example program's header.

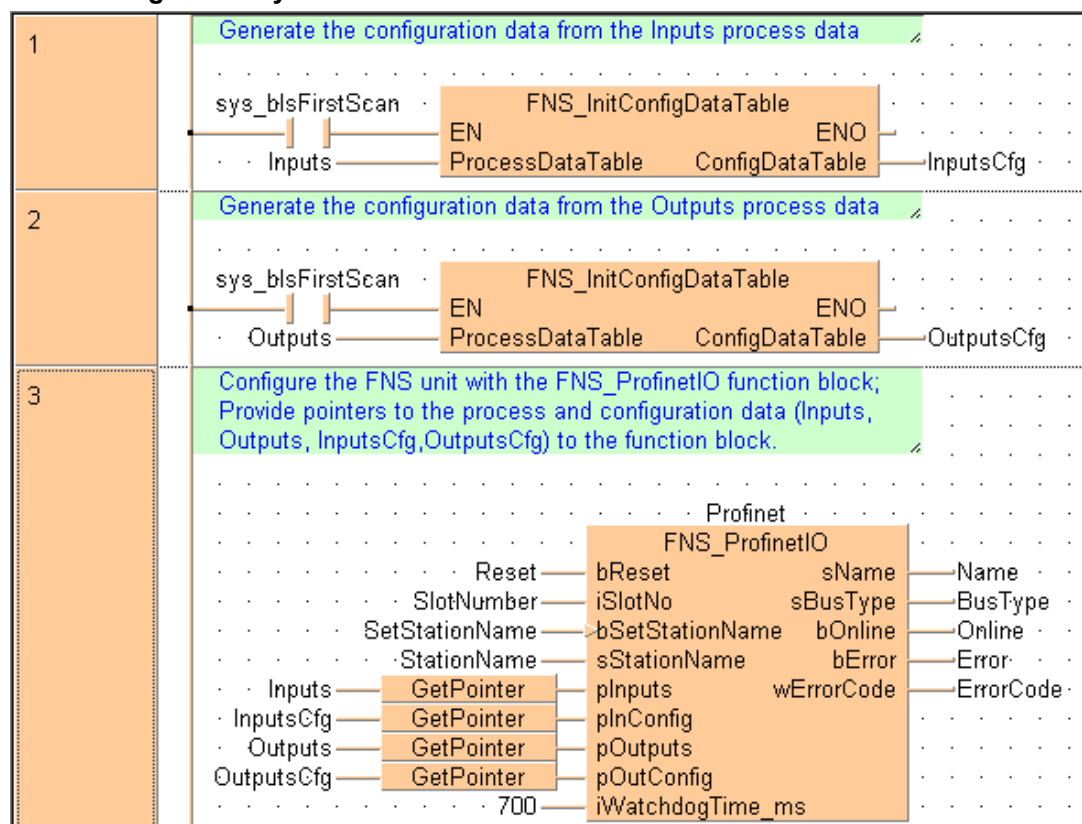
| Global Variables | | | | | | | |
|------------------|------------|------------|--------|-------------|--------|---------|--------------------------|
| | Class | Identifier | FP ... | IEC Address | Type | Initial | A... |
| 0 | VAR_GLOBAL | Inputs | | | Input | | <input type="checkbox"/> |
| 1 | VAR_GLOBAL | Outputs | | | Output | | <input type="checkbox"/> |

POU Header

| PROFINET_IO | | | | | |
|-------------|--------------|----------------|----------------------|---------------|---------|
| | Class | Identifier | Type | Initial | Comment |
| 0 | VAR_EXTERNAL | Inputs | Input | | |
| 1 | VAR_EXTERNAL | Outputs | Output | | |
| 2 | VAR | InputsCfg | ARRAY [0..2] OF W... | [3(0)] | |
| 3 | VAR | OutputsCfg | ARRAY [0..2] OF W... | [3(0)] | |
| 4 | VAR | Reset | BOOL | FALSE | |
| 5 | VAR | SlotNumber | INT | 0 | |
| 6 | VAR | SetStationName | BOOL | FALSE | |
| 7 | VAR | StationName | STRING[254] | 'abcc-prt-... | |
| 8 | VAR | Name | STRING[16] | " | |
| 9 | VAR | BusType | STRING[20] | " | |
| 10 | VAR | Online | BOOL | FALSE | |
| 11 | VAR | Error | BOOL | FALSE | |
| 12 | VAR | ErrorCode | WORD | 0 | |
| 13 | VAR | Profinet | FNS_ProfinetIO | | |

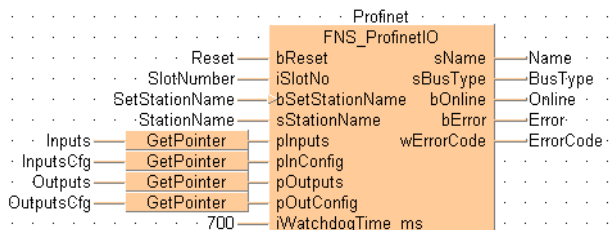
In the POU header, all variables that are required for the program are declared. The size of the variables **InputsCfg** and **OutputsCfg** must correspond to the number of entries in the DUTs Input and Output.

Ladder Diagram Body



In the ladder diagram body you can see an instance of the FNS_ProfinetIO function block called Profinet, and how the inputs, outputs and configuration data have to be supplied to the function block.

5.7.1 FNS_ProfinetIO Function Block



The FNS_ProfinetIO function block configures the FP-FNS block ProfinetIO. It has to be supplied with information about the configuration, the input and output size and network-specific data.

If no inputs or no outputs are used, just leave the corresponding pins unconnected.

PLC types: available for FP2/FP2SH and FPΣ.

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

Inputs

| Input | Data Type | Function |
|------------------|-----------|--|
| bReset | BOOL | Reset pin; network block will be reset while bReset is set. |
| iSlotNo | INT | Installation position of the FNS unit |
| bSetStationName | BOOL | A rising edge of this input sets the string stored in sStationName as the station's name and performs a power-up reset of the unit. |
| sStationName | STRING | The Station Name identifies the PROFINET IO unit in the PROFINET network. If this value is set with bSetStationName while the connection with the IO controller is established, the unit will reset so changes can take effect. Changes made through DCP will take immediate effect without reset. |
| pInputs | POINTER | Pointer to the input's process data table |
| pInConfig | POINTER | Pointer to the input's configuration data table |
| pOutputs | POINTER | Pointer to the output's process data table |
| pOutConfig | POINTER | Pointer to the output's configuration data table |
| iWatchdogTime_ms | INT | Watchdog timeout value for unit in ms. Valid values from 1 to 32767. 0: default of 700ms. |

Outputs

| Output | Data Type | Function |
|------------|------------|--|
| sName | STRING[16] | Name of installed FP-FNS block |
| sBusType | STRING[20] | Network type of installed FP-FNS block |
| bOnline | BOOL | Flag for online status |
| bError | BOOL | Error flag |
| wErrorCode | WORD | Error code if error flag is set |

List of error codes for FP-FNS block CANopen

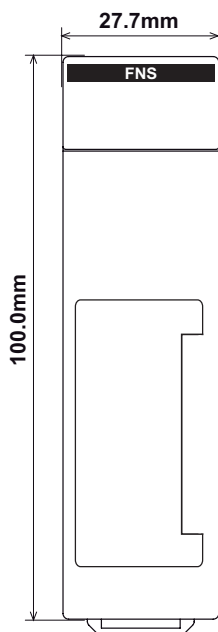
| Errorcode | Indication |
|-----------|--|
| 16#0000 | No error |
| 16#0001 | Controller and Device process data configuration do not match |
| 16#0002 | Process data area is too large (max. 64 slots, max. 128 words in each direction) |
| 16#0004 | Bus off or cable disconnected, or no link established between controller and device. |
| 16#0005 | FP-FNS block is not installed correctly |
| 16#0007 | FP-FNS block has incorrect provider ID |
| 16#0008 | Wrong FP-FNS block installed |
| 16#000A | Exception state entered; application watchdog timeout; unit needs resetting |

Chapter 6

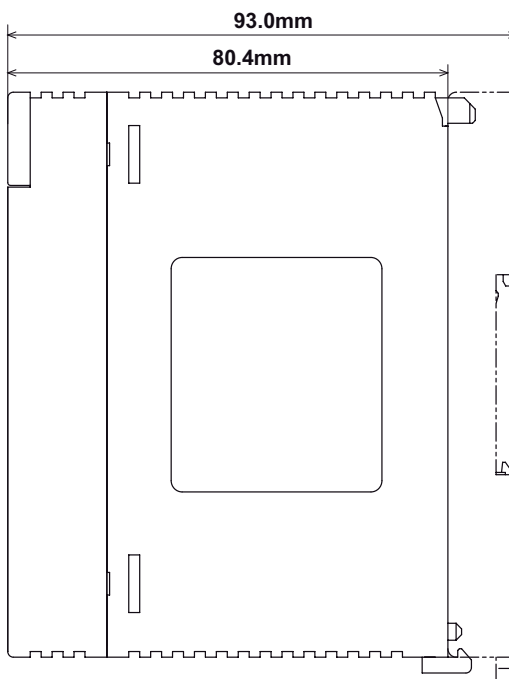
Outline Dimensions

6.1 Outline Dimensions of FP2-FNS Unit

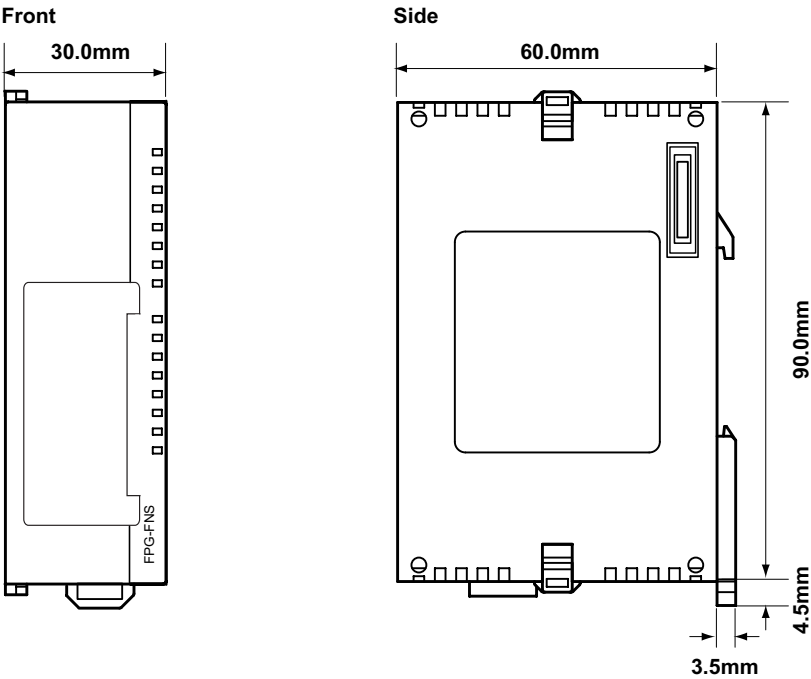
Front



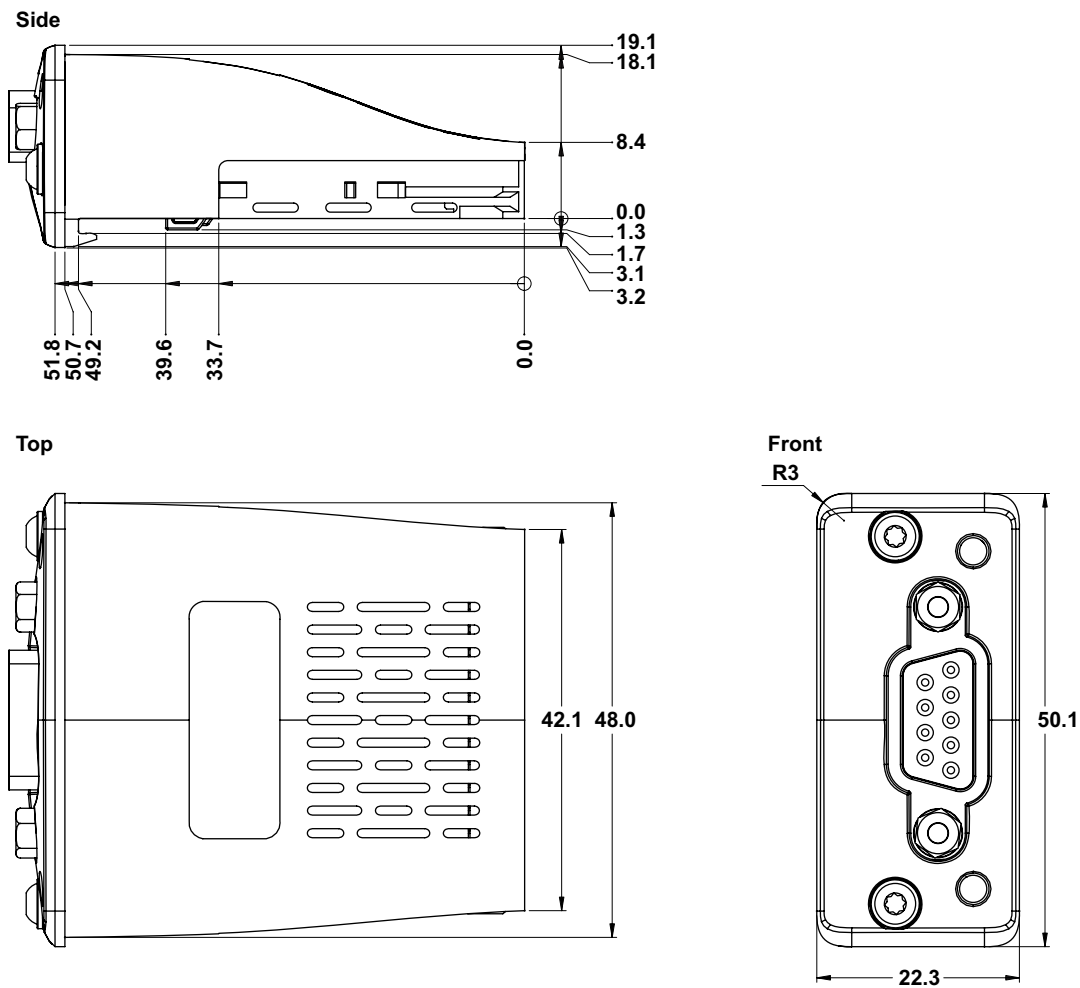
Side



6.2 Outline Dimensions of FPΣ FNS Unit

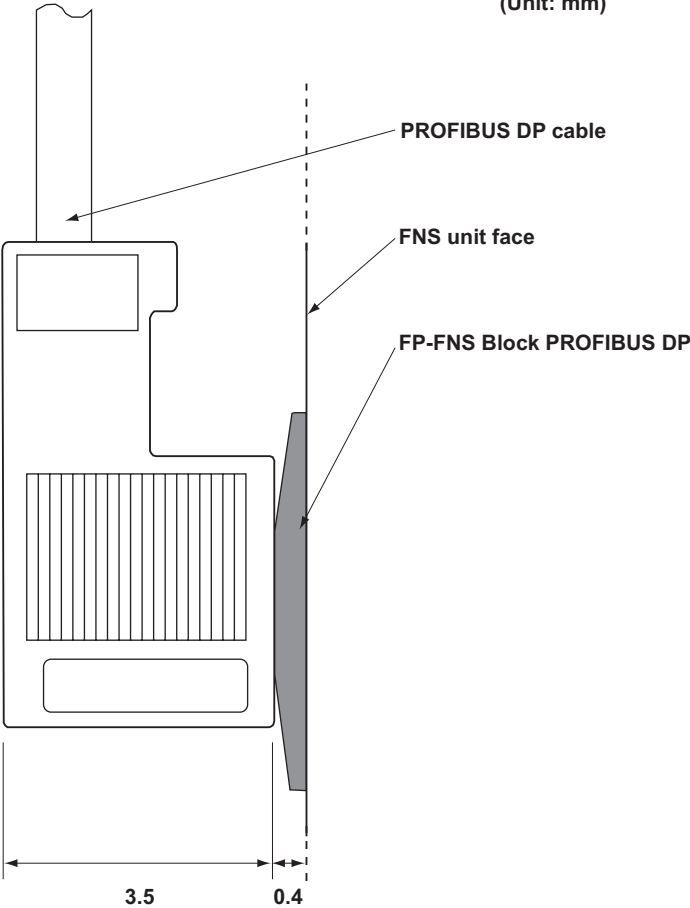


6.3 Dimensions of the FP-FNS Blocks



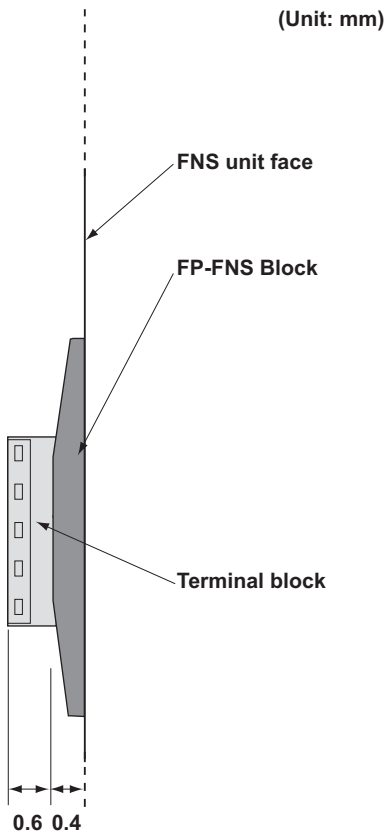
6.4 Dimensions with FNS Blocks and Cables

FP-FNS Block PROFIBUS DP or CANopen, example
(Unit: mm)



FP-FNS Block DeviceNet or PROFINET IO

For DeviceNet and PROFINET IO, how far the cable protrudes from the FNS unit face depends on the cable and connector you choose and how you connect it.



Index

A

| | |
|---------------------------|----|
| Accessory connector..... | 34 |
| Ambient humidity | 16 |
| Ambient temperature | 16 |

B

| | |
|------------------------|----|
| Backplane | 29 |
| Baud rate..... | 19 |
| BEFORE BEGINNING | i |

C

| | |
|--------------------------------|--------|
| Cable Selection | 30 |
| CANopen cable | 30 |
| CANopen Interface | 11 |
| Conductor..... | 31 |
| Configuration Data Table | 41, 46 |
| Connection types | 19 |
| Connectors | |
| Additional connectors | 31 |
| Consumption current..... | 16 |
| Current Consumption | |
| limitations | 4 |

D

| | |
|---|--------|
| Data Unit Types | 41, 46 |
| DeviceNet cable | 30 |
| DeviceNet Connector..... | 10 |
| DeviceNet Wiring | 31 |
| Diagnostic support | 19 |
| Dimensions of the FP-FNS Blocks | 64 |
| Dimensions with FNS Blocks and Cables | 65 |
| DIN rail attachment lever | 8 |

| | |
|------------------------------------|----|
| DIN rails | |
| attachment | 29 |
| DIN standard rail attachment | 8 |

E

| | |
|--|--------|
| Error codes | 44, 48 |
| FNS blockProfibusDP | 44 |
| FNS_DeviceNet | 48 |
| Expansion hook | 8 |
| Expansion Restrictions and Current | |
| Limitations | 4 |
| Expansion Restrictions for the FP2-FNS | |
| Unit | 4 |
| Expansion Restrictions for the FPΣ FNS | |
| Unit | 4 |

F

| | |
|--|--------|
| Fastening the FP-FNS Block | 22 |
| Features and Restrictions | 1 |
| Female contacts..... | 31 |
| Fieldbus Slave Units | 2, 6 |
| Flexible Network Slave (FNS) Units and | |
| Blocks | 3 |
| Flexible Network Slave Units | |
| Product no. | 3 |
| FNS..... | 3 |
| FNS blocks | |
| wiring..... | 31 |
| FNS libraries | 38 |
| FNS Library | 41, 46 |
| FNS Unit General Specifications | 16 |
| FNS_CANopen Function Block | 53 |
| FNS_DeviceNet Function Block | 48 |
| FNS_InitConfigDataTable Function | 39 |
| FNS_ProfibusDP Function Block..... | 44 |
| FNS_ProfinetIO Function Block | 58 |
| FP2 Flexible Network Slave Unit | 3 |
| FP2 FNS Unit..... | 7 |

| | |
|--|----|
| FP-FNS Block CANopen..... | 11 |
| FP-FNS Block CANopen General Specifications | 18 |
| FP-FNS Block CANopen Wiring | 33 |
| FP-FNS Block Communication Specifications | 19 |
| FP-FNS Block DeviceNet..... | 10 |
| FP-FNS Block DeviceNet General Specifications | 17 |
| FP-FNS Block General Specifications | 17 |
| FP-FNS Block Profibus Wiring | 31 |
| FP-FNS Block PROFIBUS DP | 9 |
| FP-FNS Block PROFIBUS DP General Specifications | 17 |
| FP-FNS Block PROFIBUS DP Wiring..... | 31 |
| FP-FNS Block PROFINET IO | 13 |
| FP-FNS Block PROFINET IO General Specifications | 18 |
| FP-FNS Block PROFINET IO Wiring | 33 |
| FP-FNS Blocks..... | 9 |
| Product no..... | 3 |
| FPS Flexible Network Slave Unit..... | 3 |
| FPS-FNS Unit Wiring | 34 |
| FP-Sigma expansion connector..... | 8 |
| FPΣ FNS Unit..... | 8 |

G

| | |
|---------------------------|------------|
| General Information | 38 |
| GetPointer Function | 40 |
| Global Variable List..... | 41, 46, 50 |

I

| | |
|------------------------------|--------|
| Important Symbols | ii |
| Input process data | 41, 46 |
| Installation and Wiring..... | 21 |

| | |
|---------------------------------------|----|
| Installation of the FP2/FPΣ Unit..... | 25 |
| Interface | 19 |
| Isolation..... | 19 |

L

| | |
|--|----|
| Ladder Diagram Body | 41 |
| LED test | 10 |
| Limitations on Current Consumption | 4 |

M

| | |
|------------------------------|----|
| Male contacts | 31 |
| Maximum Inputs/Outputs | 19 |
| Mounting Methods | 29 |

O

| | |
|--|----|
| Open style connector | 31 |
| Operation condition..... | 16 |
| Outline Dimensions..... | 61 |
| Outline Dimensions of FP2-FNS Unit | 62 |
| Outline Dimensions of FPΣ FNS Unit | 63 |

P

| | |
|--|----|
| Parts and Functions | 5 |
| PDO | 50 |
| POU Header | 41 |
| Power supply unit..... | 4 |
| Profibus connector | 30 |
| Profibus cable | 30 |
| Programming Example, FP-FNS Block CANopen | 50 |
| Programming Example, FP-FNS Block DeviceNet..... | 46 |
| Programming Example, FP-FNS Block ProfibusDP | 41 |
| Programming Example, FP-FNS Block Profinet IO | 55 |

| | |
|--|----|
| Programming Examples for FPWIN Pro | 37 |
| Protective earth 2-pin connector housing | 8 |

R

| | |
|---------------------------|----|
| Record of Changes | 70 |
| Removing the FP-FNS Block | 24 |

S

| | |
|---------------------|----|
| Shock resistance | 16 |
| Specifications | 15 |
| Storage humidity | 16 |
| Storage temperature | 16 |

V

| | |
|----------------------|----|
| Vibration resistance | 16 |
|----------------------|----|

W

| | |
|---------------------------------|----|
| Weight | 16 |
| Wire color | 31 |
| Wire identity | 31 |
| Wiring | |
| FNS blocks | 31 |
| Wiring method | 31 |
| Wiring of the FP-FNS Blocks | 31 |
| Wiring of the FP-Sigma-FNS Unit | 34 |

Record of Changes

| Manual No. | Date | Description of changes |
|----------------|---------------|--|
| ACGM0160V10END | March 2007 | First edition |
| ACGM0160V11END | May 2007 | <ul style="list-style-type: none">• Product nos. for FNS units and blocks removed (part nos. and product nos. have been harmonized)• Addition of CANopen function block for Control FPWIN Pro |
| ACGM0160V20END | May 2008 | <ul style="list-style-type: none">• New CANopen block (9-pin Sub-D male interface)• Improvements in function blocks for FPWIN Pro for all networks |
| ACGM0160V21EN | November 2008 | Note added to Profibus programming example that inputs are mapped to the process data before outputs. |
| ACGM0160V30EN | March 2009 | FP2 and FPΣ Fieldbus Slave Units added. These products are preassembled and include the FNS Unit and corresponding FP-FNS Block. Manual renamed to reflect this change. |
| ACGM0160V4EN | October 2009 | PROFINET IO Fieldbus Slave Unit added. CANopen block, standard 5-pole open type connector (AFPN-AB6202), discontinued and removed. |

North America

Europe

Asia Pacific

China

Japan

Panasonic Electric Works Global Sales Companies



Europe

| | | |
|--------------------|---|---|
| ► Headquarters | Panasonic Electric Works Europe AG | Rudolf-Diesel-Ring 2, 83607 Holzkirchen, Tel. +49 (0) 8024 648-0, Fax +49 (0) 8024 648-111, www.panasonic-electric-works.com |
| ► Austria | Panasonic Electric Works Austria GmbH PEW Electronic Materials Europe GmbH | Rep. of PEWDE, Josef Madersperger Str. 2, 2362 Biedermannsdorf, Tel. +43 (0) 2236-26846, Fax +43 (0) 2236-46133, www.panasonic-electric-works.at Ennshafenstraße 30, 4470 Enns, Tel. +43 (0) 7223 883, Fax +43 (0) 7223 88333, www.panasonic-electronic-materials.com |
| ► Benelux | Panasonic Electric Works Sales Western Europe B.V. | De Rijn 4, (Postbus 211), 5684 PJ Best, (5680 AE Best), Netherlands, Tel. +31 (0) 499 372727, Fax +31 (0) 499 372185, www.panasonic-electric-works.nl |
| ► Czech Republic | Panasonic Electric Works Czech s.r.o. | Prumtyslová 1, 34815 Planá, Tel. (+420)-374799990, Fax (+420)-374799999, www.panasonic-electric-works.cz |
| ► France | Panasonic Electric Works Sales Western Europe B.V. | Successale française, 10, rue des petits ruisseaux, 91371 Verrières le Buisson, Tél. +33 (0) 1 6013 5757, Fax +33 (0) 1 6013 5758, www.panasonic-electric-works.fr |
| ► Germany | Panasonic Electric Works Deutschland GmbH | Rudolf-Diesel-Ring 2, 83607 Holzkirchen, Tel. +49 (0) 8024 648-0, Fax +49 (0) 8024 648-555, www.panasonic-electric-works.de |
| ► Hungary | Panasonic Electric Works Europe AG | Magyarországi Közvetlen Kereskedelmi Képviselet, 1117 Budapest, Neumann János u. 1., Tel. +36(0)1482 9258, Fax +36 (0) 1482 9259, www.panasonic-electric-works.hu |
| ► Ireland | Panasonic Electric Works UK Ltd. | Dublin, Tel. +353 (0) 14600969, Fax +353 (0) 14601131, www.panasonic-electric-works.co.uk |
| ► Italy | Panasonic Electric Works Italia s.r.l. | Via del Commercio 3-5 (Z.I. Ferlina), 37012 Bussolengo (VR), Tel. +39 (0) 456752711, Fax +39 (0) 456700444, www.panasonic-electric-works.it |
| ► Nordic Countries | Panasonic Electric Works Nordic AB PEW Fire & Security Technology Europe AB | Sjögångsvägen 10, 19272 Sollentuna, Sweden, Tel. +46 859476680, Fax +46 859476690, www.panasonic-electric-works.se |
| ► Poland | Panasonic Electric Works Polska sp. z o.o. | Jungmansgatan 12, 21119 Malmö, Tel. +46 40697-7000, Fax +46 40697-7099, www.panasonic-fire-security.com |
| ► Portugal | Panasonic Electric Works España S.A. | Al. Krakowska 4/6, 02-284 Warszawa, Tel. +48 (0) 22 338-11-33, Fax +48 (0) 22 338-12-00, www.panasonic-electric-works.pl |
| ► Spain | Panasonic Electric Works España S.A. | Portuguese Branch Office, Avda Adelino Amaro da Costa 728 R/C J, 2750-277 Cascais, Tel. +351 214812520, Fax +351 214812529 |
| ► Switzerland | Panasonic Electric Works Schweiz AG | Barajas Park, San Severo 20, 28042 Madrid, Tel. +34 913293875, Fax +34 913292976, www.panasonic-electric-works.es |
| ► United Kingdom | Panasonic Electric Works UK Ltd. | Grundstrasse 8, 6343 Rotkreuz, Tel. +41 (0) 417997050, Fax +41 (0) 417997055, www.panasonic-electric-works.ch Sunrise Parkway, Linford Wood, Milton Keynes, MK14 6LF, Tel. +44(0) 1908 231555, +44(0) 1908 231599, www.panasonic-electric-works.co.uk |

North & South America

| | | |
|-------|----------------------------|---|
| ► USA | PEW Corporation of America | 629 Central Avenue, New Providence, N.J. 07974, Tel. +1-908-464-3550, Fax +1-908-464-8513, www.pewa.panasonic.com |
|-------|----------------------------|---|

Asia Pacific/China/Japan

| | | |
|-------------|---|--|
| ► China | Panasonic Electric Works (China) Co., Ltd. | Level 2, Tower W3, The Tower Oriental Plaza, No. 2, East Chang An Ave., Dong Cheng District, Beijing 100738, Tel. (010) 5925-5988, Fax (010) 5925-5973, www.pewc.panasonic.cn |
| ► Hong Kong | Panasonic Electric Works (Hong Kong) Co., Ltd. | RM1205-9, 12/F, Tower 2, The Gateway, 25 Canton Road, Tsimshatsui, Kowloon, Hong Kong, Tel. (8520) 2956-3118, Fax (852) 2956-0398 |
| ► Japan | Panasonic Electric Works Co., Ltd. | 1048 Kadoma, Kadoma-shi, Osaka 571-8686, Japan, Tel. (06)-6908-1050, Fax (06)-6908-5781 http://panasonic-electric-works.net |
| ► Singapore | Panasonic Electric Works Asia Pacific Pte. Ltd. | 101 Thomson Road, #25-03/05, United Square, Singapore 307591, Tel. (06255)-5473, Fax (06253)-5689 |