INVERTER

Plug-in option

A8NDPV1

INSTRUCTION MANUAL

PROFIBUS communication interface

Doc.Id. HMSI-216-127
Rev. 1.00

HMS *Connecting Devices™*

HALMSTAD • CHICAGO • KARLSRUHE • TOKYO • BEIJING • MILANO • MULHOUSE • COVENTRY • PUNE • COPENHAGEN

Important User Information

This document is intended to provide a good understanding of the functionality offered by the A8NDPV1 PROFIBUS Option Board. The document only describes the features that are specific to the option board. For general information regarding the FR-A800 inverter, consult the FR-A800 inverter design guides.

The reader of this document is expected to be familiar with high level software design, and communication systems in general. The use of advanced PROFIBUS-specific functionality may require in-depth knowledge in PROFIBUS networking internals and/or information from the official PROFIBUS specifications. In such cases, the people responsible for the implementation of this product should either obtain the PROFIBUS specification to gain sufficient knowledge or limit their implementation in such a way that this is not necessary.

Liability

Every care has been taken in the preparation of this manual. Please inform HMS Industrial Networks AB of any inaccuracies or omissions. The data and illustrations found in this document are not binding. We, HMS Industrial Networks AB, reserve the right to modify our products in line with our policy of continuous product development. The information in this document is subject to change without notice and should not be considered as a commitment by HMS Industrial Networks AB. HMS Industrial Networks AB assumes no responsibility for any errors that may appear in this document.

There are many applications of this product. Those responsible for the use of this device must ensure that all the necessary steps have been taken to verify that the applications meet all performance and safety requirements including any applicable laws, regulations, codes, and standards.

HMS Industrial Networks AB will under no circumstances assume liability or responsibility for any problems that may arise as a result from the use of undocumented features, timing, or functional side effects found outside the documented scope of this product. The effects caused by any direct or indirect use of such aspects of the product are undefined, and may include e.g. compatibility issues and stability issues.

The examples and illustrations in this document are included solely for illustrative purposes. Because of the many variables and requirements associated with any particular implementation, HMS Industrial Networks AB cannot assume responsibility for actual use based on these examples and illustrations.

Intellectual Property Rights

HMS Industrial Networks AB has intellectual property rights relating to technology embodied in the product described in this document. These intellectual property rights may include patents and pending patent applications in the US and other countries.

Trademark Acknowledgements

Anybus® is a registered trademark of HMS Industrial Networks AB. All other trademarks are the property of their respective holders.

⚠ CAUTION

- This is a class A product. In a domestic environment this product may cause radio interference in which case the user may be required to take adequate measures.
- **ESD Note**
  This product contains ESD (Electrostatic Discharge) sensitive parts that may be damaged if ESD control procedures are not followed. Static control precautions are required when handling the product.
  Failure to observe this may cause damage to the product.

A8NDPV1 PROFIBUS Option Board User Manual

Rev 1.00

Copyright© HMS Industrial Networks AB

March 2014 Doc Id HMSI-216-127

Thank you for choosing this Mitsubishi Inverter plug-in option for the Mitsubishi FR-A800 Series Inverter. This Instruction Manual gives handling information and precautions for use of this equipment. Incorrect handling may cause an unexpected failure or damage. In order to ensure optimal performance, please read this manual carefully prior to use of the equipment.

Please forward this manual to the end user of the equipment.

This section is specifically about safety matters

Do not attempt to install, operate, maintain or inspect this product until you have read through this Instruction Manual and any related documents carefully, and can use the equipment correctly. Do not use this product until you have a full working knowledge of the equipment, safety information and instructions.

In this Instruction Manual, the safety instruction levels are classified into "WARNING" and "CAUTION" levels.

⚠ WARNING	Assumes that incorrect handling may cause hazardous conditions, resulting in death or severe injury.
⚠ CAUTION	Assumes that incorrect handling may cause hazardous conditions, resulting in moderate or slight injury, or may cause physical damage only.

Please note that even the **⚠ CAUTION** level may lead to a serious consequence depending on conditions. Please be sure to follow the instructions of both levels as they are critical to personnel safety.

SAFETY INSTRUCTIONS

Electric Shock Prevention

⚠ WARNING

- Do not open any cover on the inverter while power is on or while the inverter is running, as an electrical shock may result.
- Do not operate the inverter with any cover or wiring cover removed, as accidental contact with exposed high-voltage terminals and internal components may occur, resulting in an electrical shock.
- If power is off do not remove any cover except when necessary for wiring or periodic inspection. While any cover is removed, accidental contact with exposed high-voltage terminals and internal components may occur, resulting in an electrical shock.
- Prior to starting wiring or inspection, confirm that input power to the inverter has been switched off via observation of the inverter's display panel. Additionally, wait for at least 10 minutes after removal of input power, and then confirm that all residual voltage has been dissipated by using a voltage meter. Internal DC bus capacitors may contain high voltages for several minutes after removal of input power, resulting in a dangerous situation should anything come into contact with them.
- All personnel involved in the installation or inspection of this equipment should be fully competent to perform the required work.
- Always install plug-in options prior to wiring main power.
- Do not touch the plug-in option with wet hands.
- Do not subject the cables to scratches, excessive stress, heavy loads or pinching.

Injury Prevention

⚠ CAUTION

- To prevent explosions or similar damage, apply only the voltages specified in the instruction manual to each terminal.
- To prevent explosions or similar damage, ensure that all cables are properly connected to the correct terminals.
- To prevent explosions or similar damage, observe all wiring polarity indicators.
- To prevent burns from hot components, do not touch the inverter while power is on, or for some time after power is removed.

Additional Instructions

Please note the following points to prevent equipment damage, injury or electrical shock.

Transportation and mounting

⚠ CAUTION

- Do not install or operate the plug-in option if it is damaged or has parts missing.
- Do not stand or rest heavy objects on the equipment.
- Check that the mounting orientation is correct.
- Prevent conductive items such as screws and metal fragments, or flammable substances such as oil from entering the inverter.

Trial run

⚠ CAUTION

- To prevent unexpected equipment movement, confirm and adjust all required parameters prior to starting operation.

Usage

⚠ WARNING

- Do not modify the equipment.
- Do not remove any inverter or option parts unless specifically instructed to do so in this manual.

⚠ CAUTION

- Performing a "parameter clear" or "all parameter clear" will reset all inverter parameters to their factory default settings. After performing one of these operations, remember to reenter any custom parameter values prior to starting operation.
- To prevent damage from electric discharge, always touch a grounded piece of metal prior to touching any equipment.

Maintenance, inspection and parts replacement

⚠ CAUTION

- Do not perform hi-pot tests on the equipment.

Disposal

⚠ CAUTION

- Contact the local or state environmental agency in your area for details on the disposal of electrical components and packaging.

General instruction

For clarity purposes, illustrations in this manual man be drawn with covers or safety guards removed. Ensure all covers and safety guards are properly installed prior to starting operation.

Table of Contents

About This Document 1
Related Documents 1
Download 1
Document History 2
Conventions & Terminology 3
Support 3

1. Pre-Operation Instructions 5
1.1 General 5
1.2 Product Overview 5
1.3 Features 5
1.4 Unpacking and Product Confirmation 6
1.4.1 Shipment Confirmation 6
1.4.2 Component Overview 6
1.5 Environmental Specifications 7

2. Installation 9
2.1 Pre-installation Instructions 9
2.2 Installation Procedure 9
2.3 Network Connector (DSUB, female) 12
2.4 LED Indicators 13

3. Get Started 15
3.1 Physical Installation 15
3.2 Download GSD file 15
3.3 Inverter setup 15
3.4 GX Works (Q-CPU) Telegram 1 example 16
3.5 GX Works (Q-CPU) Telegram 102 example 29
3.6 GX Works (Q-CPU) Acyclic communication example 34
3.6.1 Reading a parameter (Sequence 1) 34
3.6.2 Reading an array of parameters (Sequence 3) 37
3.6.3 Changing parameters (Sequence 2) 38
3.7 GX Works (Q-CPU) Simple Ladder Telegram 1 example 42
3.8 GX Works (Q-CPU) Simple Ladder Telegram 102 example 50
3.9 GX Works (Q-CPU) Simple Ladder Acyclic communication example 54
3.9.1 Reading a parameter (Sequence 1) 54
3.9.2 Changing parameters (Sequence 2) 59
3.10 GX Works (FX-CPU) Telegram 1 example 65
3.11 GX Works (FX-CPU) Telegram 102 example 77

3.12 GX Works (FX-CPU) Acyclic communication example....82
3.12.1 Reading a parameter (Sequence 1)....82
3.12.2 Changing parameters (Sequence 2)....85
3.13 TIA Portal Telegram 1 example....89
3.14 TIA Portal Telegram 102 example....98
3.15 TIA Portal Acyclic communication example....101
3.15.1 Reading a parameter (Sequence 1)....101
3.15.2 Changing parameters (Sequence 2)....107
3.16 SIMATIC STEP7 example....111
3.16.1 Creating a Configuration....111
3.16.2 Download Configuration....116
3.16.3 Run....116
3.17 Parameter Settings....116

4. Inverter Settings....117
4.1 Inverter Parameters....117
4.2 Option Board Parameters....117
4.3 Operation Mode Setting....119

5. Identifying Option Board....121
5.1 Set Slave Address....121

6. PROFIBUS DP-V1 Implementation....123
6.1 General....123
6.2 Electronic Data Sheet (GSD)....123
6.3 DAP....123
6.4 I&M....123

7. Data Exchange 125
7.1 General Information 125
7.2 Inverter parameters (Acyclic Data Exchange) 126
7.3 Monitor Data (Acyclic and Cyclic Data Exchange) 126
7.4 Drive Profile Parameters (Acyclic Data Exchange) 129
7.4.1 PROFIdrive Parameters 129
7.4.2 Setpoint- and Actual Value (P915/P916) 130
7.4.3 Signal List (P923) 130
7.4.4 Drive Reset (P972) 130
7.5 General State Diagram 131
7.6 Process Data (Cyclic Data Exchange) 132
7.6.1 General 132
7.6.2 Signals 132
7.6.3 Telegram Types 134
7.6.4 Vendor Specific Access to Parameters 135
7.7 Acyclic Data Exchange 136
7.7.1 Explanation of fields used in requests 137
7.7.2 Data format type table 137
7.7.3 Error table 138
7.7.4 Sequence 1: Request parameter value, single 138
7.7.5 Sequence 1: Parameter response positive 138
7.7.6 Sequence 1: Parameter response negative 139
7.7.7 Sequence 2: Change parameter value 139
7.7.8 Sequence 2: Parameter response positive 139
7.7.9 Sequence 2: Parameter response negative 139
7.7.10 Sequence 3: Request parameter value, several array elements 140
7.7.11 Sequence 3: Parameter response positive 140
7.7.12 Sequence 3: Parameter response negative 140

8. Diagnostics 141

9. Troubleshooting 143

A. Translation of Signal Numbers 145

Index 147

About This Document

For more information, documentation etc., please visit the ME website, 'https://eu3a.mitsubishielectric.com'.

Related Documents

Document	Author
Installation guideline of Mitsubishi inverter drive	ME
Instruction manual of Mitsubishi inverter drive	ME
Q Series Profibus Master QJ71PB92V manual: https://eu3a.mitsubishielectric.com/fa/en/mymitsubishi/download_manager?id=1839	ME
FX Series Profibus Master FX3U-64DP-M manual: https://eu3a.mitsubishielectric.com/fa/en/mymitsubishi/download_manager?id=1862	ME
GX Configurator-DP 7.10L Software Manual: https://eu3a.mitsubishielectric.com/fa/en/mymitsubishi/download_manager?id=1404	ME

Download

The following websites are available for downloads:

Website	Region
https://eu3a.mitsubishielectric.com	EU
http://www.meau.com/eprise/main/sites/public/Products/Variable_Frequency_Drives/A800/default	Americas
GSD File: https://eu3a.mitsubishielectric.com/fa/en/mymitsubishi/download_manager?id=10168 (MyMitsubishi Login required)	EU
GX Configurator DP 7.10L: https://eu3a.mitsubishielectric.com/fa/en/mymitsubishi/download_manager?id=10160 (MyMitsubishi Login required)	EU
Profibus A800 function blocks: https://eu3a.mitsubishielectric.com/fa/en/mymitsubishi/download_manager?id=10172 (MyMitsubishi Login required)	EU
Profibus A800 DPV1 function blocks: https://eu3a.mitsubishielectric.com/fa/en/mymitsubishi/download_manager?id=9189 (MyMitsubishi Login required)	EU
Sample programs: https://eu3a.mitsubishielectric.com/fa/en/mymitsubishi/download_manager?id=10170 (MyMitsubishi Login required)	EU

Document History

Summary of Recent Changes (... 1.00)

Change	Page(s)

Revision List

Revision	Date	Author(s)	Chapter(s)	Description
1.00	03/2014			

Conventions & Terminology

The following conventions are used throughout this manual:

- Numbered lists provide sequential steps
- Bulleted lists provide information, not procedural steps
- The term 'module' refers to the communication module.
- Hexadecimal values are written in the format NNNNh, where NNNN is the hexadecimal value.

Support

MITSUBISHI ELECTRIC EUROPE
EUROPE B.V.
German Branch
Gothaer Straße 8
D-40880 Ratingen
Phone: +49 (0) 21 02 / 486-0
Hotline: +49 2102 1805 000-765 /-766
Fax: +49 (0) 21 02 / 4 86-1 12 0
e-mail: megfa-mail@meg.mee.com
https://eu3a.mitsubishielectric.com

MITSUBISHI ELECTRIC USA
AUTOMATION
500 Corporate Woods Parkway
Vernon Hills, Illinois 60061
Phone: +1 847-478-2100
Fax: +1 847-478- 0327
www.MEAU.com

MITSUBISHI ELECTRIC JAPAN
CORPORATION
Tokyo Bldg.
2-7-3 Marunouchi Chiyoda-Ku
Tokyo 100-8310
Phone: +81 (0) 3 / 32 18 31 76
Fax: +81 (0) 3 / 32 18 24 22

Please refer to the drive manual for other region contact addresses.

1. Pre-Operation Instructions

1.1 General

The FR-A800 series from Mitsubishi Electric (ME), is a family of frequency inverters. The communication modules, option boards, enabling communication on different industrial networks, are developed and produced by HMS Industrial Networks.

Examples of applications for the frequency inverters are:

- Lifting equipment
- Warehouse systems
- Extruders
- Centrifuges

1.2 Product Overview

The A8NDPV1 PROFIBUS Option Board allows information to be transferred seamlessly between an FR-A800 series inverter and a PROFIBUS network with minimal configuration requirements. The interface installs directly onto the inverter's control board, and presents a standard DSUB port for connection to the PROFIBUS network.

The option board is connected directly to the control board of the inverter and communicates with the inverter via a built-in communication port. Note that when the inverter's network communication port is used by the A8NDPV1 PROFIBUS Option Board, it is unavailable for use by any other network.

Before using the option board, please familiarize yourself with the product and be sure to thoroughly read the instructions and precautions contained in this manual. In addition, please make sure that this instruction manual is delivered to the end user of the product, and keep this instruction manual in a safe place for future reference or unit inspection.

1.3 Features

- PROFIBUS DPV1 communication according to IEC 61158 Type 3
- Drive operation according to PROFIdrive V4.1 [PDTS]
  - Supports Application class 1 functionality
- Diagnostic support
- Automatic baud rate detection
- Baud rates up to 12 Mbit supported
- Support for Set Slave Address (SSA) functionality, i.e. node address can be set from the network by a configuration tool or by a PROFIBUS master
- Customized GSD-file provided
- Up to 64 bytes of I/O data in each direction

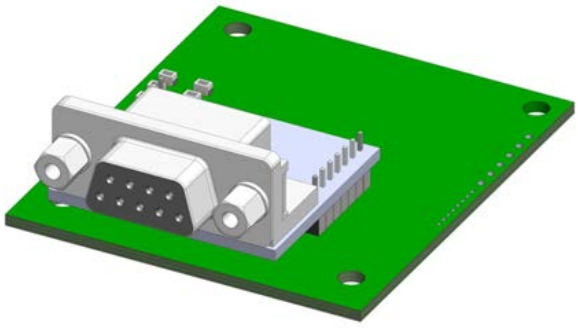
1.4 Unpacking and Product Confirmation

1.4.1 Shipment Confirmation

Check the enclosed items. Confirm that the correct quantity of each item was received, and that no damage occurred during shipment.

1.4.2 Component Overview

Included in the package are the following items.

Item	No. of pcs	
PCB board	1	
M# x 6 mm screw	3	
Board spacer	2	
LED cover Note: this picture only shows the label, not the cover!	1	
PE plate	1	

1.5 Environmental Specifications

Item	Specification
Operating Temperature	-10º to +50º Celsius (ambient of the drive, non-freezing)
Storage Temperature	-40º to +65º Celsius
Relative Humidity	93% non condensing
Vibration	Max acceleration amplitude: 10 m/s^2 at 9 - 200 Hz Max displacement amplitude: 3 mm at 2 - 9 Hz
Grounding	Connected to inverter frame ground through the PE plate / isolated from inverter control power common
Power Supply	Supplied from inverter
Cooling Method	Self cooled
Communication Speed	Up to 12 Mbit

The A8NPDPV1 interface is lead-free / RoHS-compliant.

2. Installation

2.1 Pre-installation Instructions

	⚠ WARNING
⚠	To avoid damage to the inverter or plug-in option board, never install or remove a plug-in option board while the inverter's input power is on. Make sure that the inverter's power is OFF.

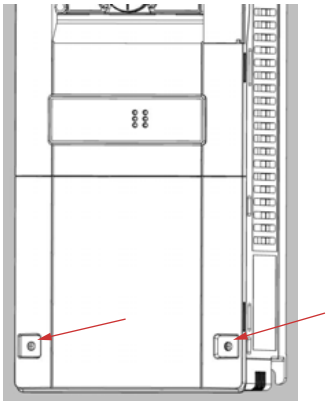
Physical installation of the option board is a two-step process. First, the card will be mounted onto an available option connector on the inverter's control board. Second, the card will be connected to the PROFIBUS network using a PROFIBUS cable.

2.2 Installation Procedure

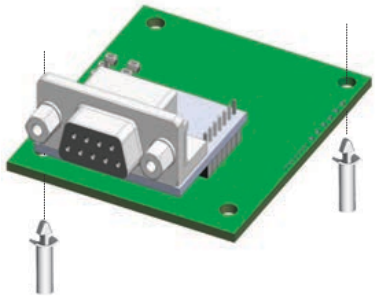
① Make sure that power is off.

② Remove both lids of the FR-A800.

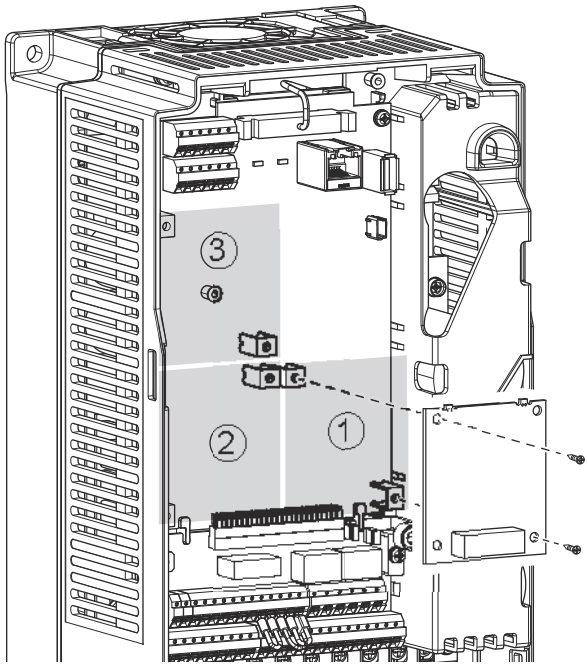
- Unscrew the two screws in the bottom corners of the inverter.
- Remove the lid covering the lower front of the inverter.
- Unscrew the screw in the bottom right corner of the lid covering the upper front of the inverter.
- Remove the lid.



③ Put the included studs in the holes at the right top and left bottom corners of the PCB.



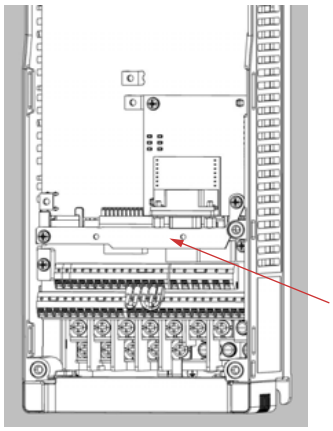
④ Position the A8NDPV1 PROFIBUS Option Board at the option slot 1 as shown in the image. This is the only position that will allow network connectivity.



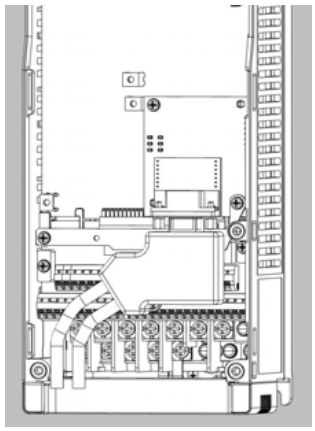
⑤ Fasten the option board by tightening the included screws at the left top and right bottom corners. The PE plate is attached along with the screw in the right bottom corner.

Note:	Over-tightening the screws will damage the board.

⑥ Fasten the other end of the PE plate with another screw as shown in the picture.

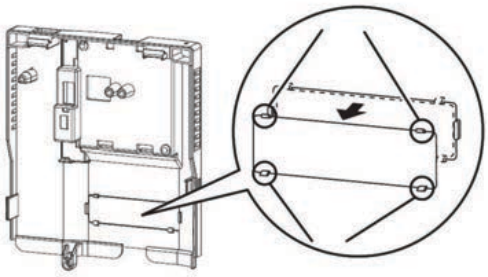


⑦ Attach the network cable to the network connector on the option board.

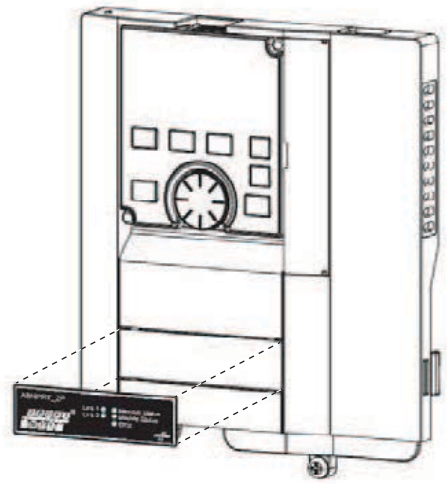


⑧ To fit the LED cover on the front cover of the drive, do as follows:

– Cut the bridges, using nippers, on the upper front cover.



– Snap the LED cover into the front cover of the drive.



⑨ Fasten both front covers, top front cover first.

The option board is now mounted and power can be applied.

Removal

① Remove both lids of the FR-A800 inverter.

② Remove the network cable.

③ Remove the screws.

④ Carefully remove the option board by lifting it straight up.

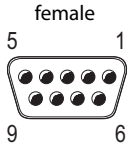
⑤ Remove the board spacers.

⑥ Replace the lids.

2.3 Network Connector (DSUB, female)

The option board provides connection to PROFIBUS through a female DSUB connector.

Pin	Name	Description
1	NC	
2	NC	
3	RxD/TxD-P	Receive/Transmit data P; B-line
4	CNTR-P	Repeater control signal (Direction control); RTS signal
5	DGND	Data ground (Reference voltage to VP)
6	VP	Power supply; (PSV)
7	NC	
8	RxD/TxD-N	Receive/Transmit data N; A-line
9	NC	
Shield	PE	



2.4 LED Indicators

LED	State	Status
Network status	Off	Offline or no power
	Green	Online, data exchange
	Green, 1 flash	Online, clear
	Red, 1 flash	Module initialized, parametrization error.
	Red, 2 flashes	Module initialized, configuration error.
	Red	Fatal error. (Network status LED and Module status LED will both be red.)
Module status	Off	Not initialized: no power or setup not finished
	Green	Operational
	Green, 1 flash	Module initialized, diagnostic event(s) present
	Red	Exception error: the option board is in the exception state. If the Network status LED also indicates solid red, a fatal error has occurred.
Error	Off	Communication with main unit is working without problems.
	Red	Error in communication with main unit
	Red, flashing (2 blinks)	Invalid process data parameter mapped
	Red, flashing (3 blinks)	Too many process data parameters mapped

3. Get Started

All example programs described in this chapter are available for download from the MyMitsubishi website. See section “Download” on page 1 for a direct link.

3.1 Physical Installation

How to physically install the option board is described in “Installation Procedure” on page 9.

3.2 Download GSD file

Download the appropriate GSD file from the Mitsubishi Electric website, see section “Download” on page 1 for a direct link.

3.3 Inverter setup

This chapter describes the first part of setting up a system using an A800 inverter controlled by Profibus. It assumes that the A8NDPV1 option card has been installed correctly, and all inverter parameters are set to default values. This setup will be used for all get started example programs.

The first step is to set the FDL address of the inverter Profibus slave. One way to do this is to set parameter 1305 to the desired address, and parameter 1307 to 1. After restarting the inverter, the option card will be available under the input address. The valid range for the FDL address is 0-125.

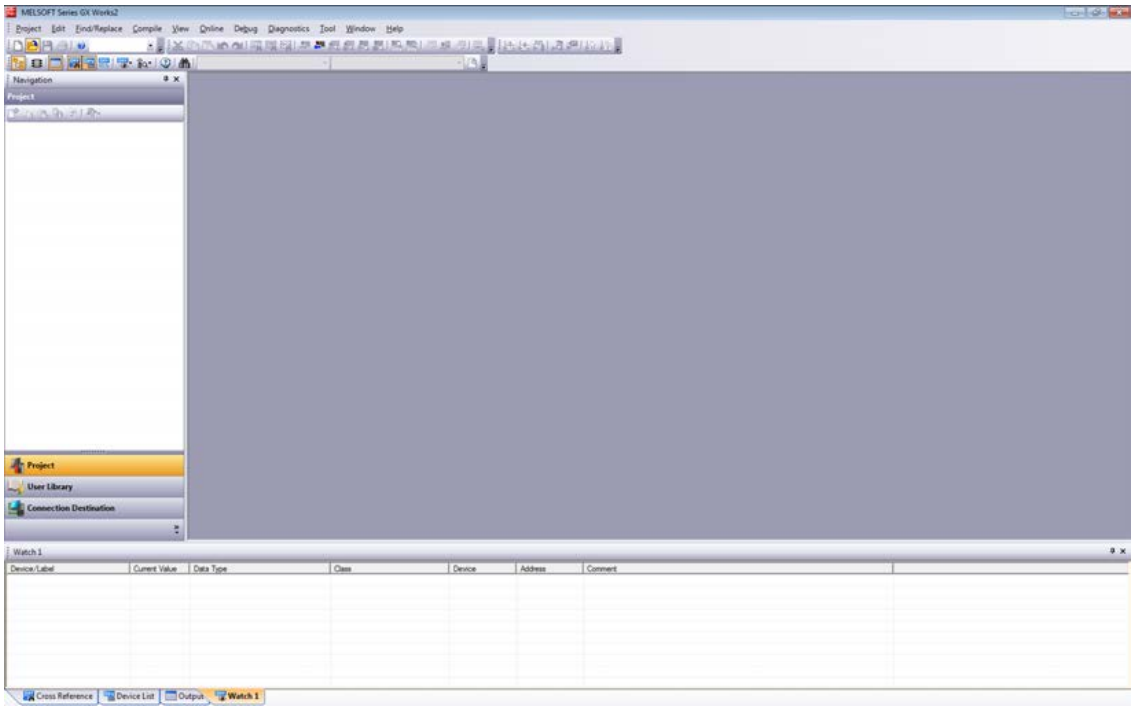
Parameter No.	Value
1305	(1) FDL Address
1307	(1)

The next step is to change the operation mode of the inverter to network mode. Do this, by leaving the default setting of parameter 79 (0), and setting parameter 340 to 1. After restarting the inverter, it should operate in network mode. For detailed settings, please refer to A800 manual.

3.4 GX Works (Q-CPU) Telegram 1 example

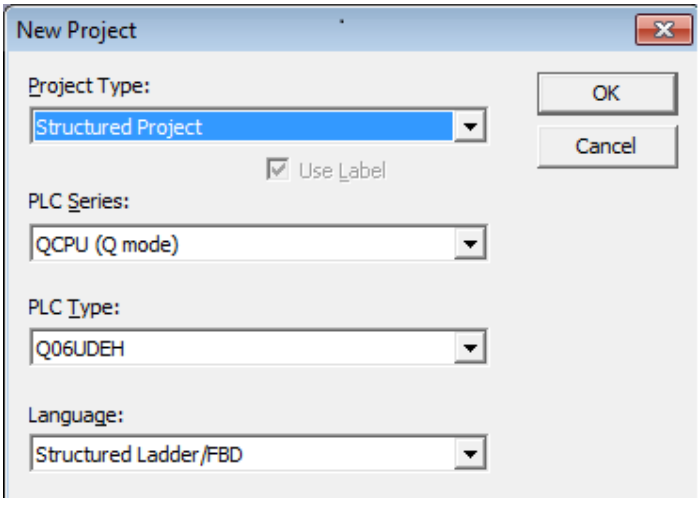
This chapter describes running the inverter through the A8NDPV1 Profibus option card using GX Works 2 with a QCPU and QJ71PB92V Profibus Master. GX Configurator DP 7.10L or newer is required to perform the setup.

① Start with a new project.

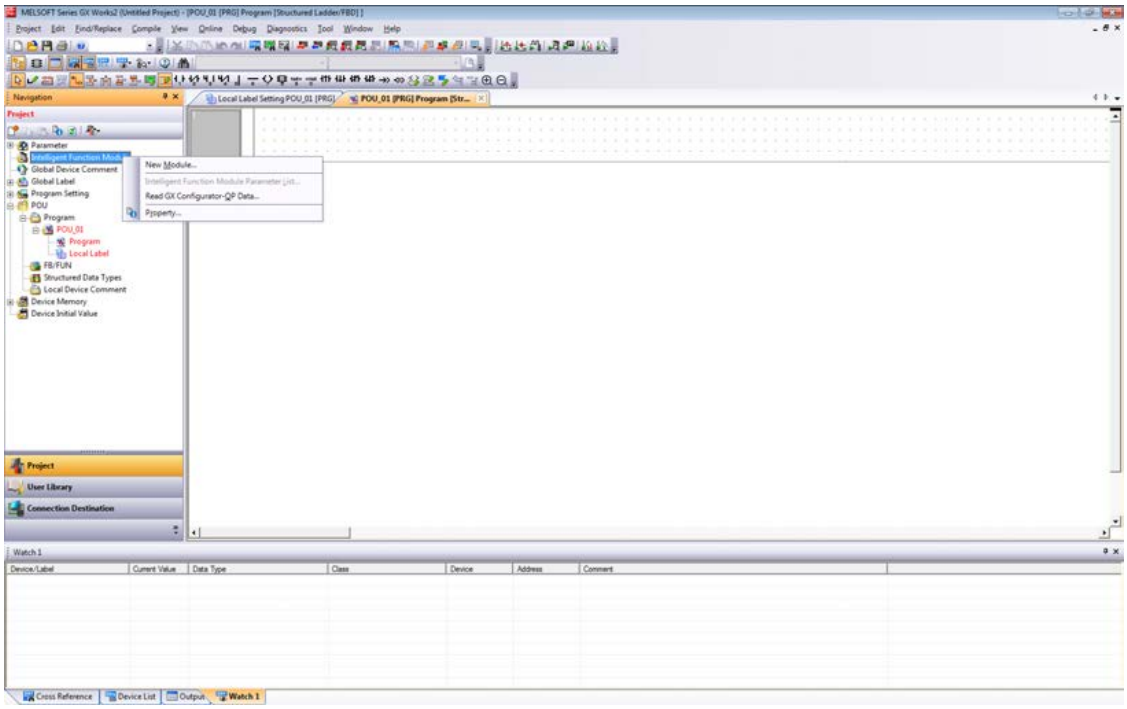


② Select:

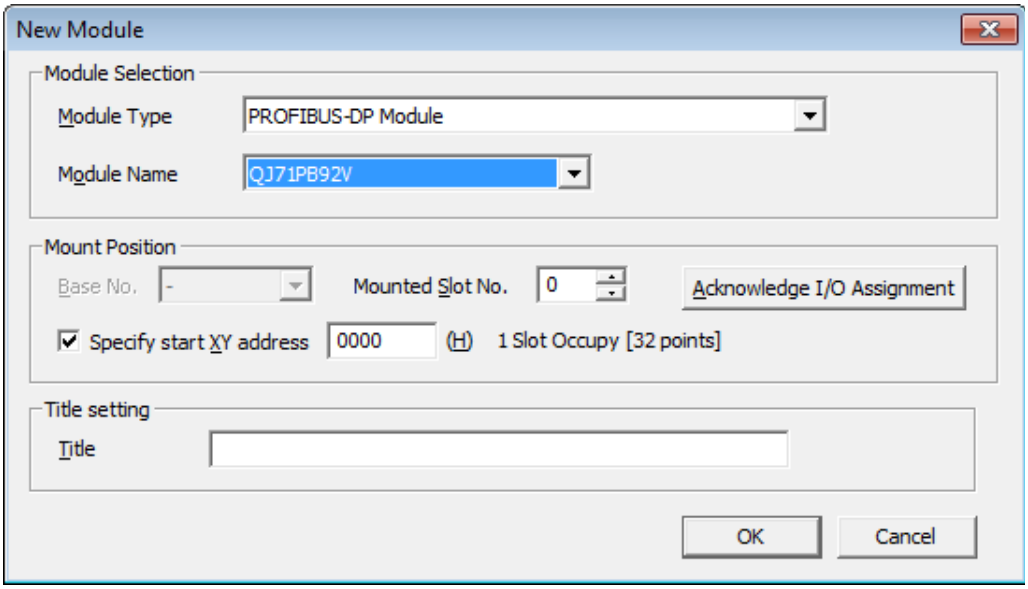
- Project type: Structured project;
- PLC Series: QCPU (Q mode);
- PLC type: select the cpu;
- Language: Structured Ladder/FBD.



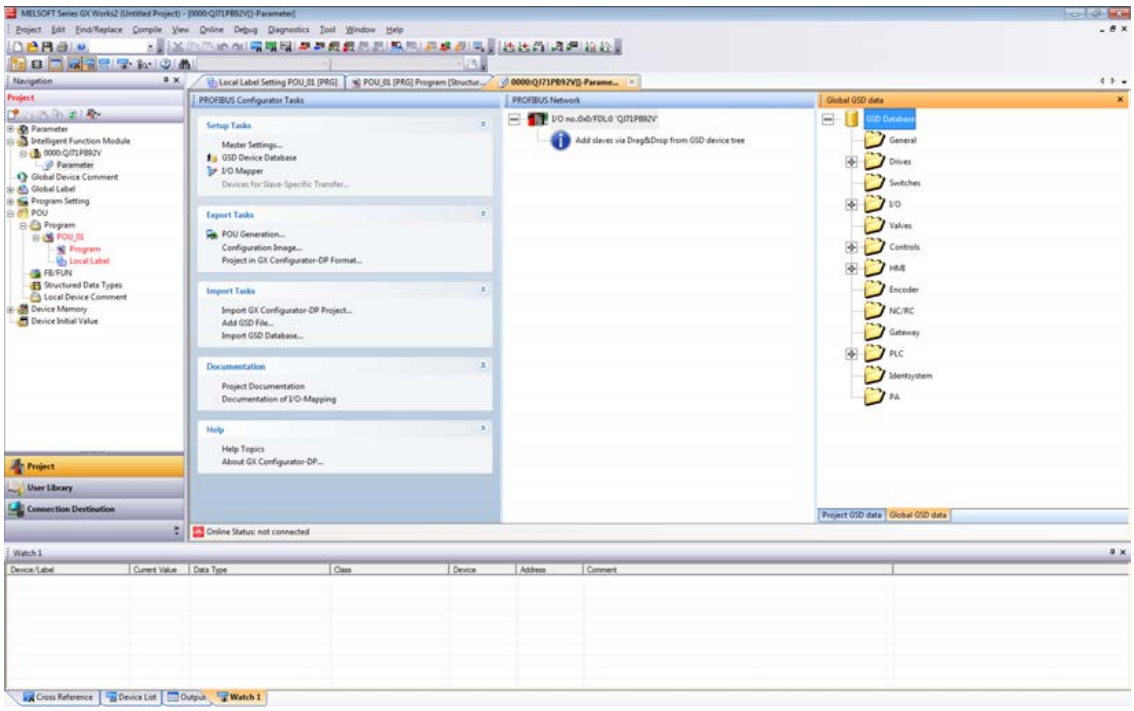
③ Add a new Intelligent Function Module from the menu on the left (right click and select new):



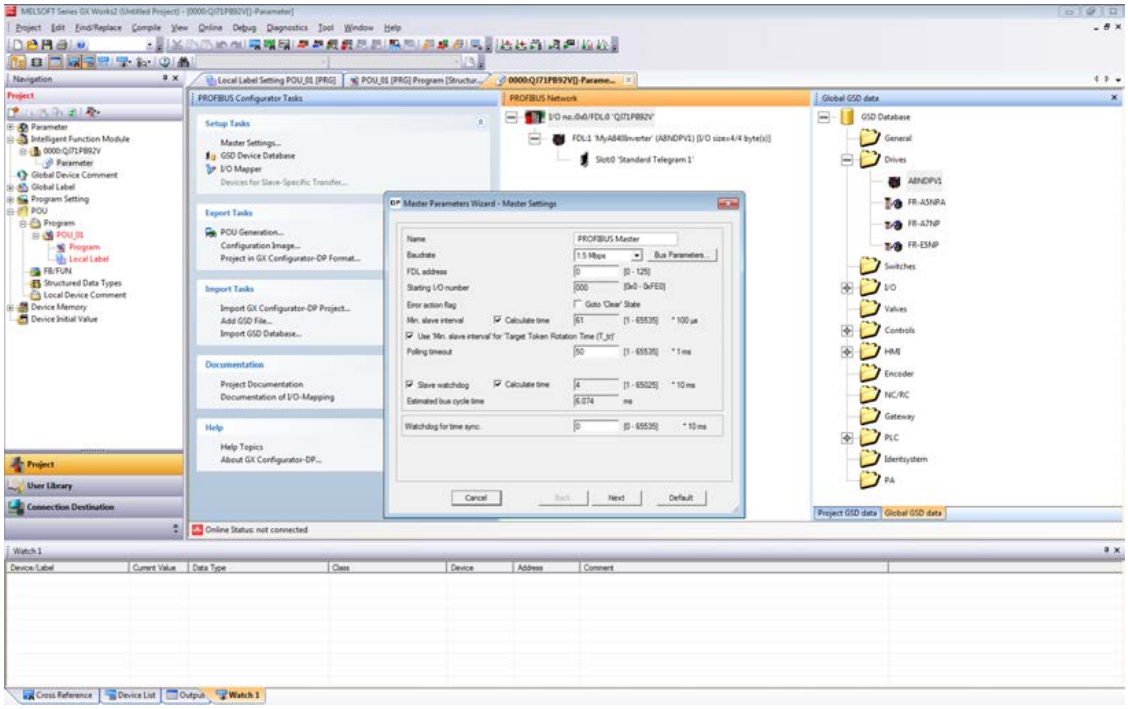
④ In the new window, click Module Type and select “PROFIBUS-DP Module”. Module Name should be set to QJ71PB92V. Save this setting by clicking OK.



⑤ The Profibus-DP module should be inserted into the Intelligent Function Module tree. Expand it, and double click on Parameter. This should open the Profibus Configurator.

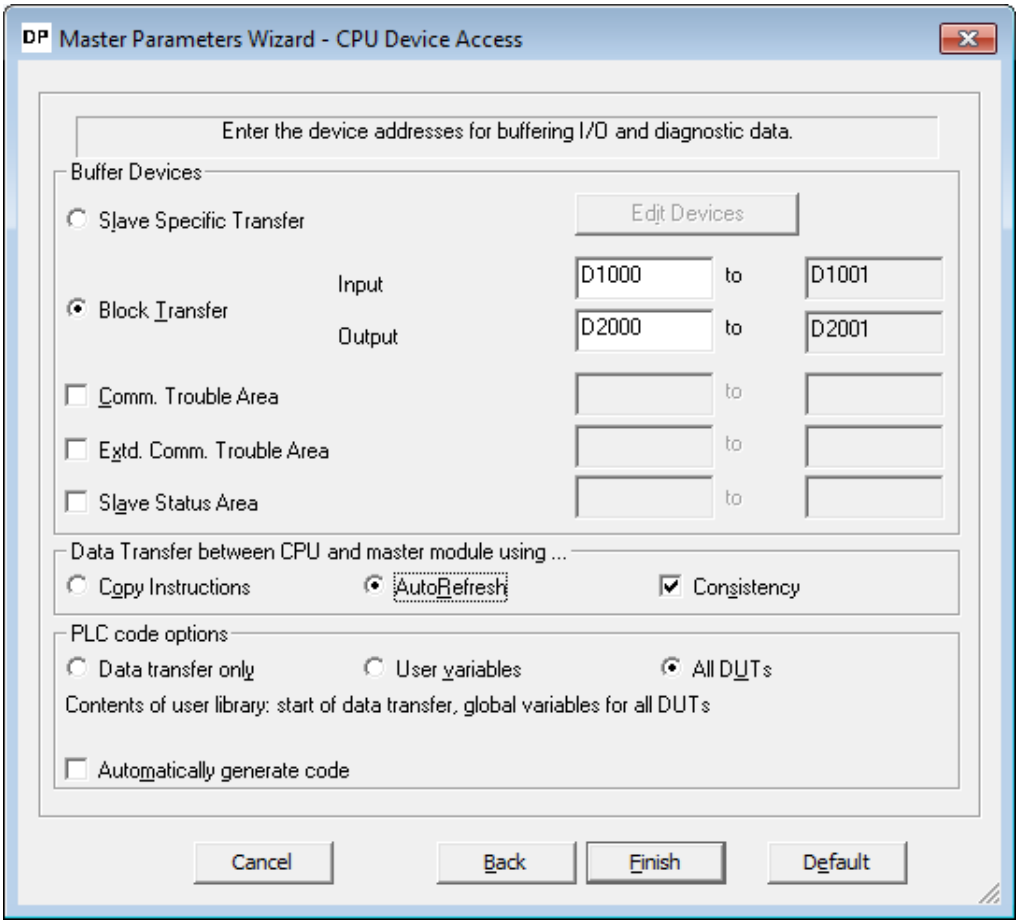


⑥ First setup the Profibus Master. Double click on the master module (in the screenshot visible as “I/O no.:0x0 FDL:0 ‘QJ71PB92V’. This will bring up the “Master Parameters Wizard”.

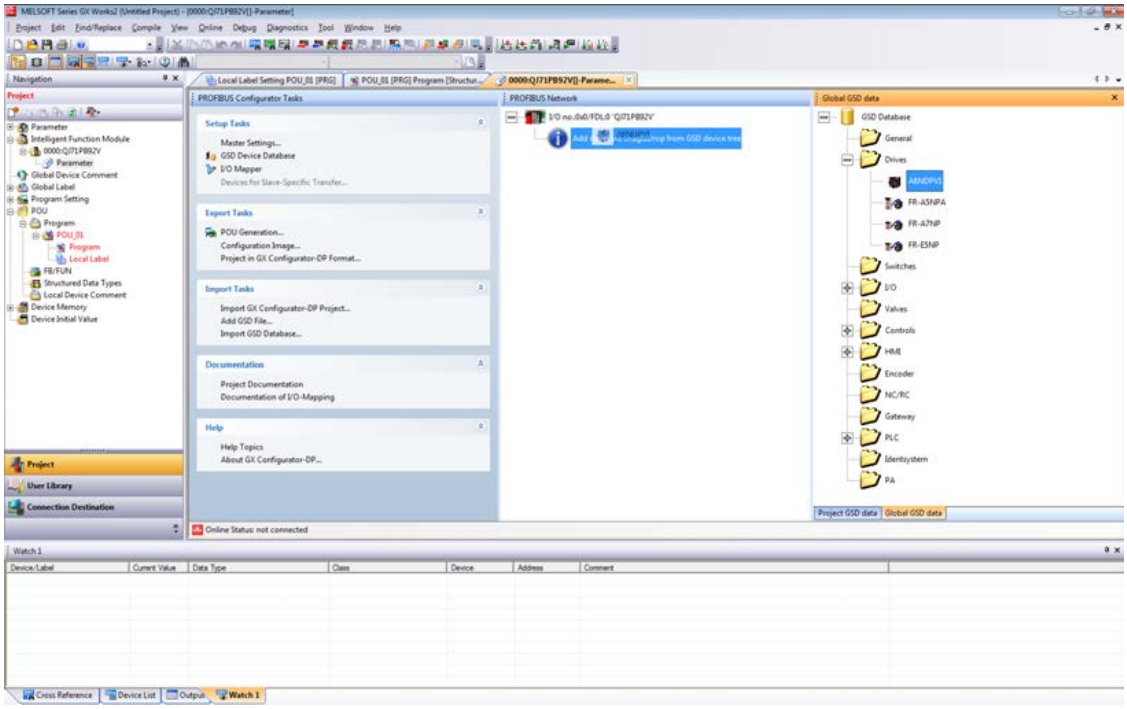


⑦ It is possible to change the Baudrate, FDL address and other settings as needed. For this tutorial these settings will not be changed. Click Next to proceed.

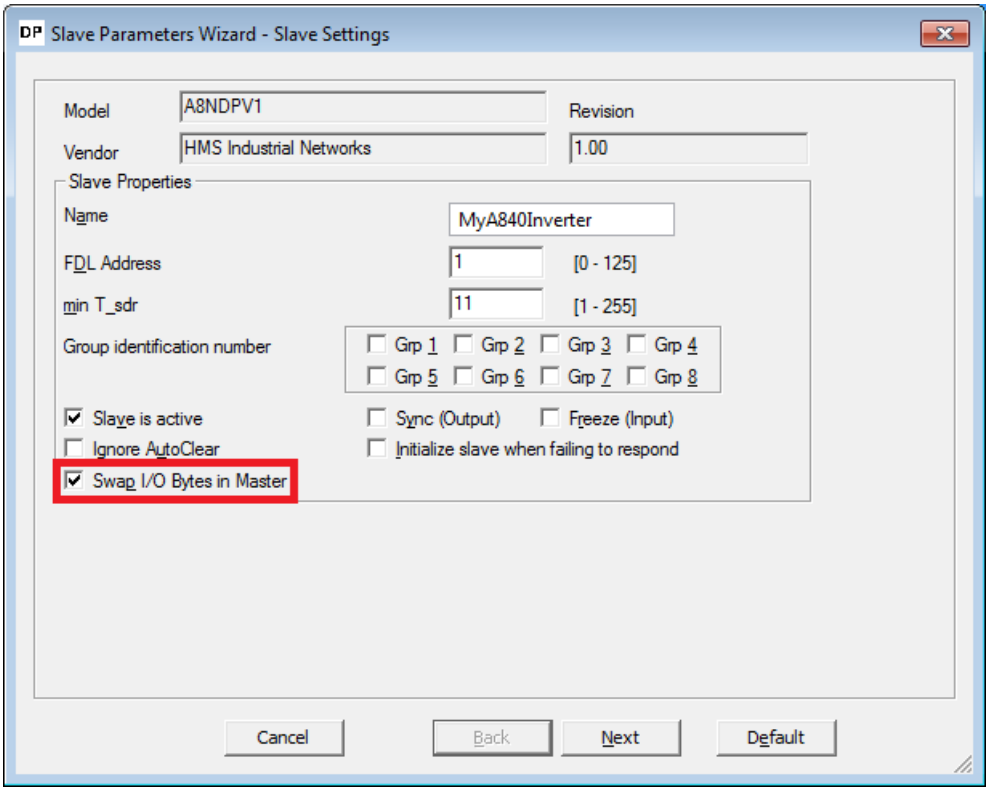
⑧ On this screen the buffer devices, and data transfer between Profibus Master and CPU can be changed. For this tutorial use AutoRefresh with Consistency. Click Finish to save settings and close this wizard.



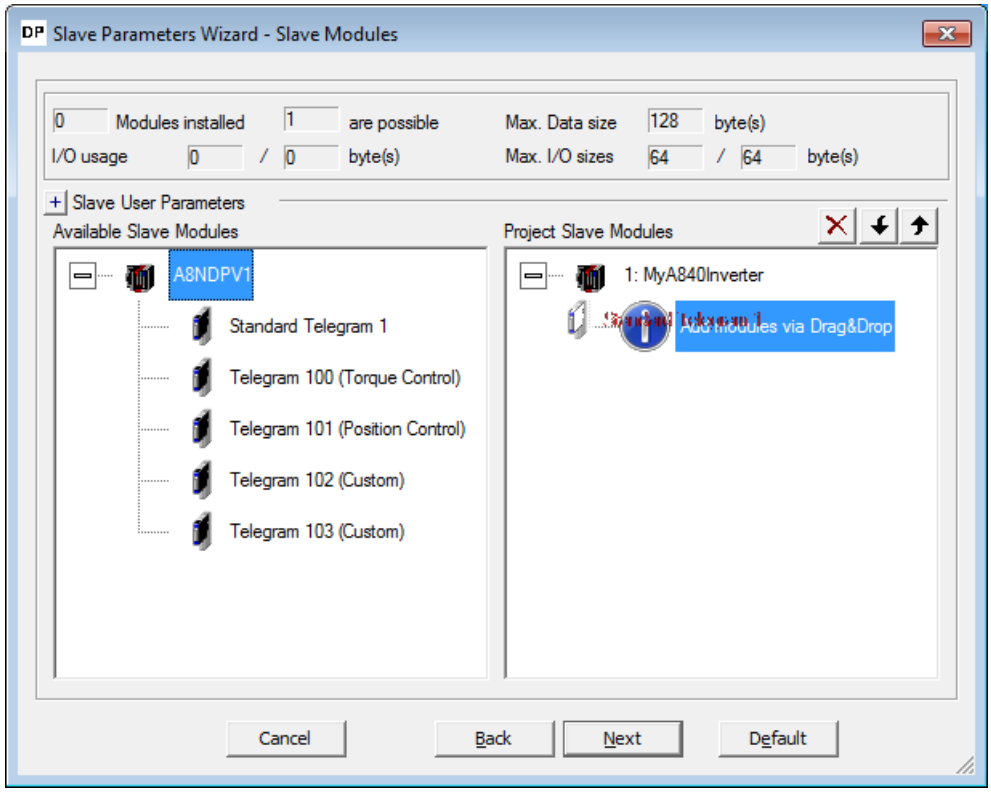
⑨ In the Global GSD data section, expand drives and search for A8NDPV1. If it is not available, right click and select Add GSD File. Search for the GSD File on the drive (included with this manual), afterwards the A8NDPV1 should appear under drives. Drag this option to the Profibus Master, like on the screenshot.



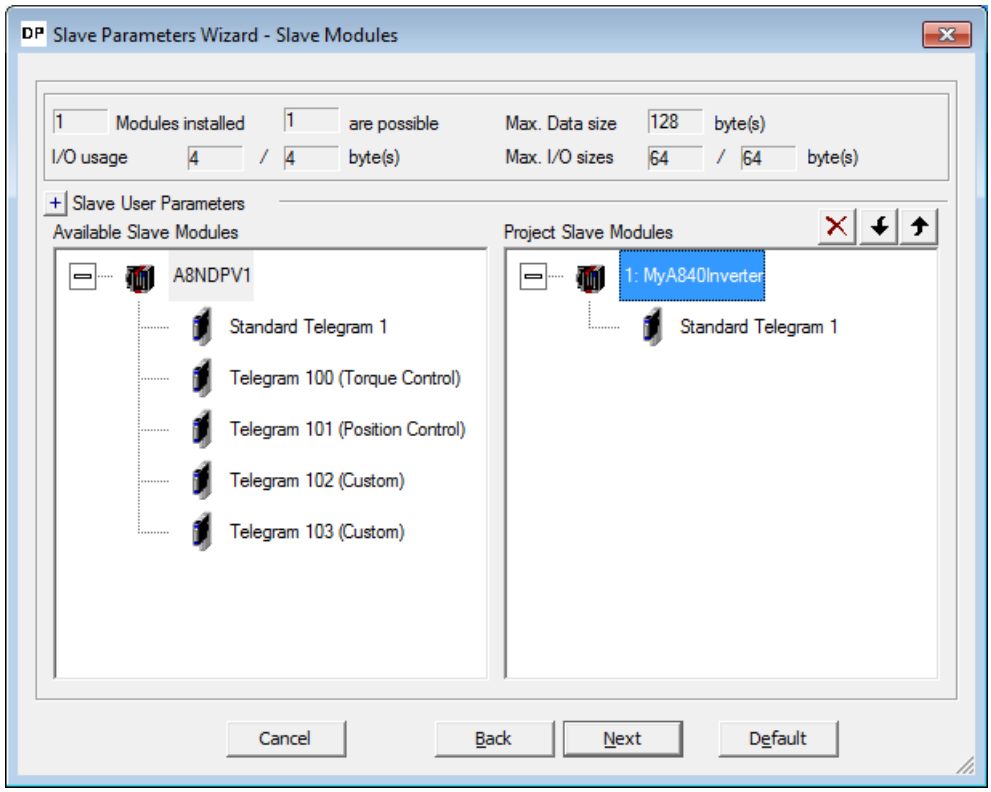
The “Slave Parameters Wizard” should be shown. Enter a name, select 1 for FDL Address, and make sure to select “Swap I/O Bytes in Master”. All other settings can remain as default. Click Next to proceed.



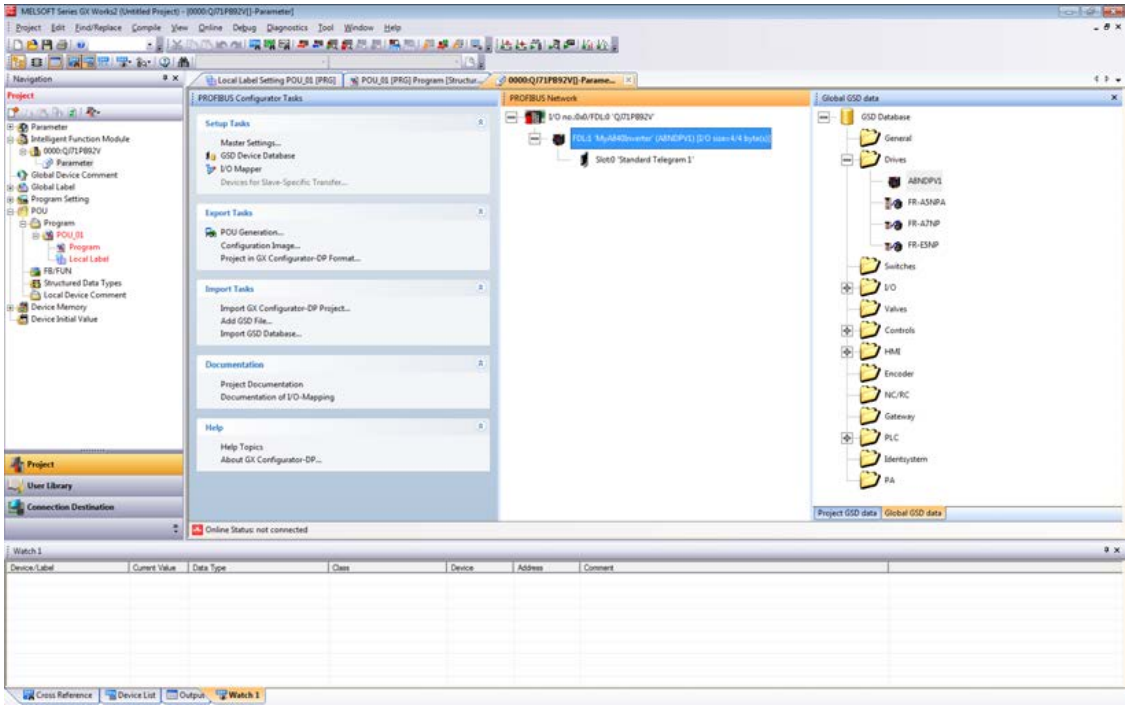
⑩ Select the communication protocol used to communicate with the option card. For this tutorial use “Standard Telegram 1”. Drag this option from the “Available Slave Modules” to “Project Slave Modules”. The A8NDPV1 supports only one module.



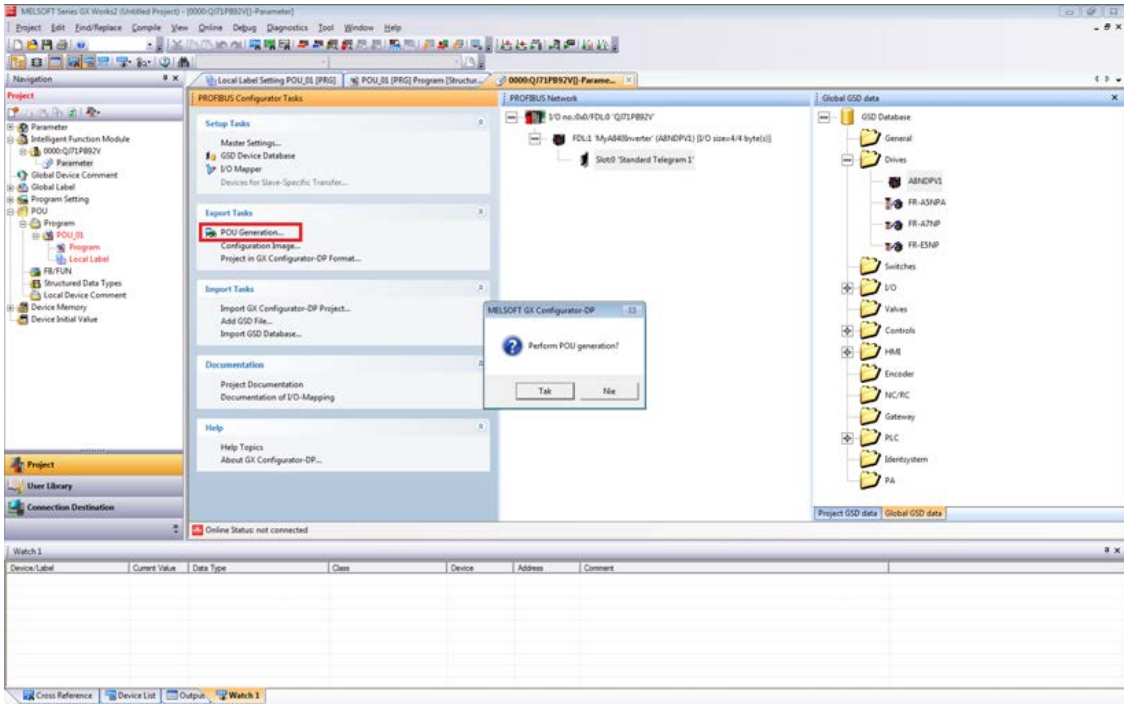
The result should be:



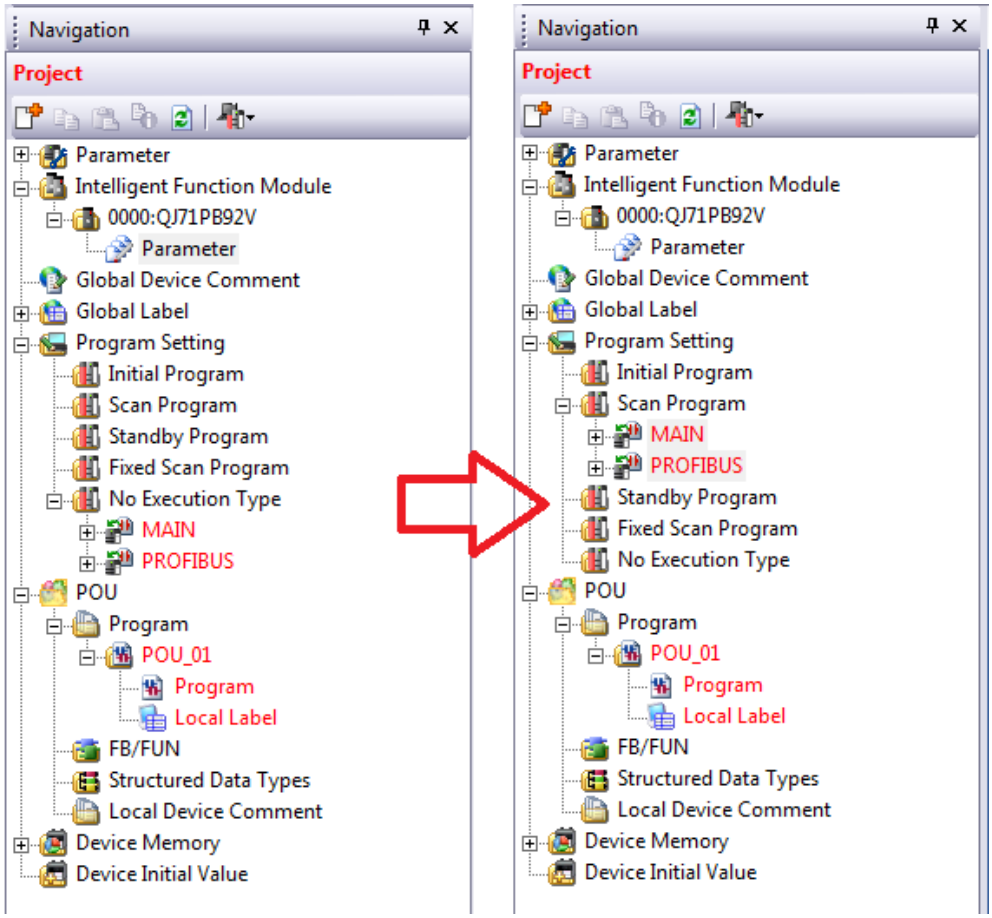
⑪ Click Next to proceed. No changes are required on this screen, after clicking Next proceed to the final screen and click Finish. The configured Profibus Slave should appear in the Profibus Network window, like below.



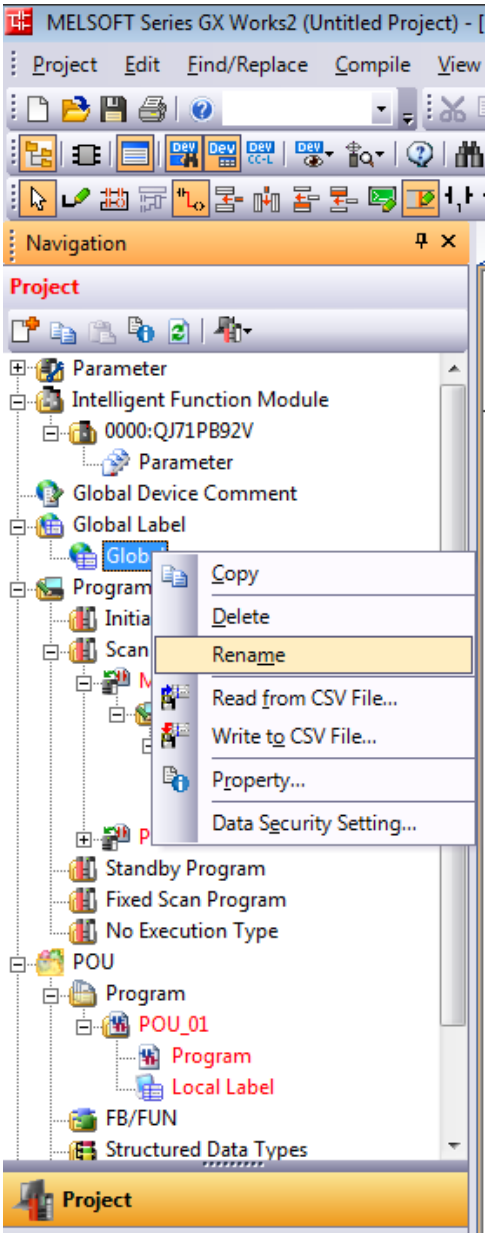
⑫ From the configurator's left menu select POU Generation. This will create the user libraries that can be used to control the inverter.



⑬ From the GX Works 2 Navigation window, expand Program settings. There should be two tasks under “No Execution Type” – MAIN and PROFIBUS. Drag both tasks to “Scan Program” to allow them to execute.



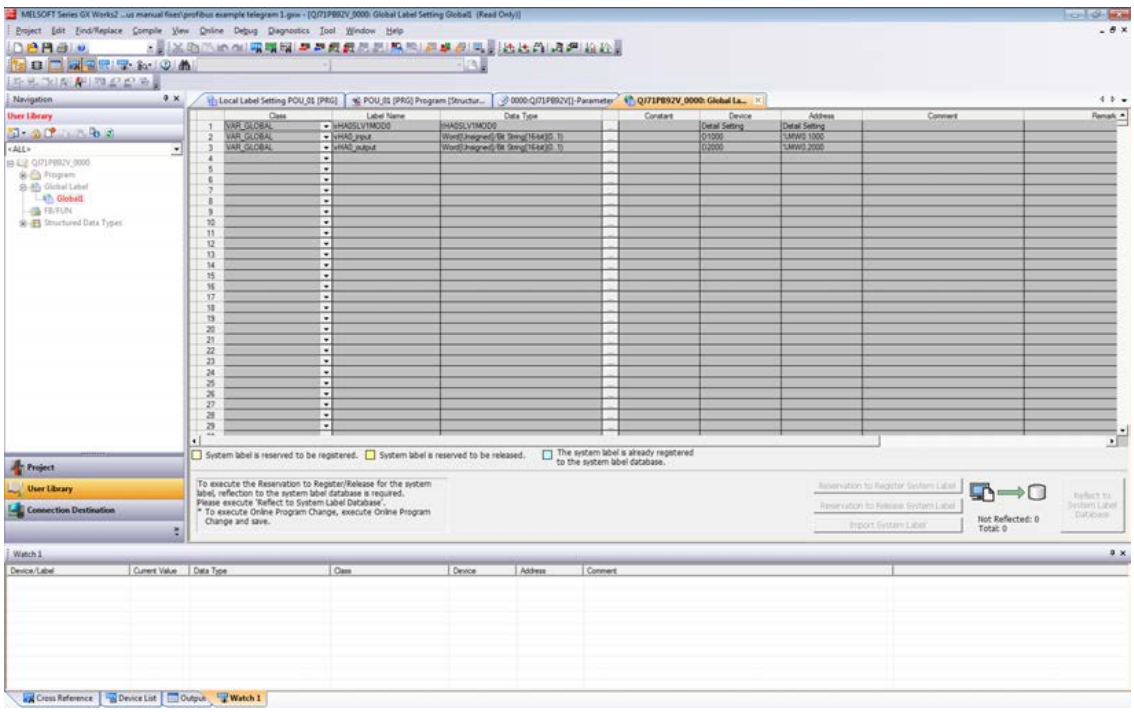
⑭ The POU Generation creates a library containing global variable definitions. The global label name conflicts with the project global label name setup by default in GX Works 2. In order to avoid conflict, change the name of the project global label from “Global1” to “Global”. If this is not done, an error will occur upon compilation.



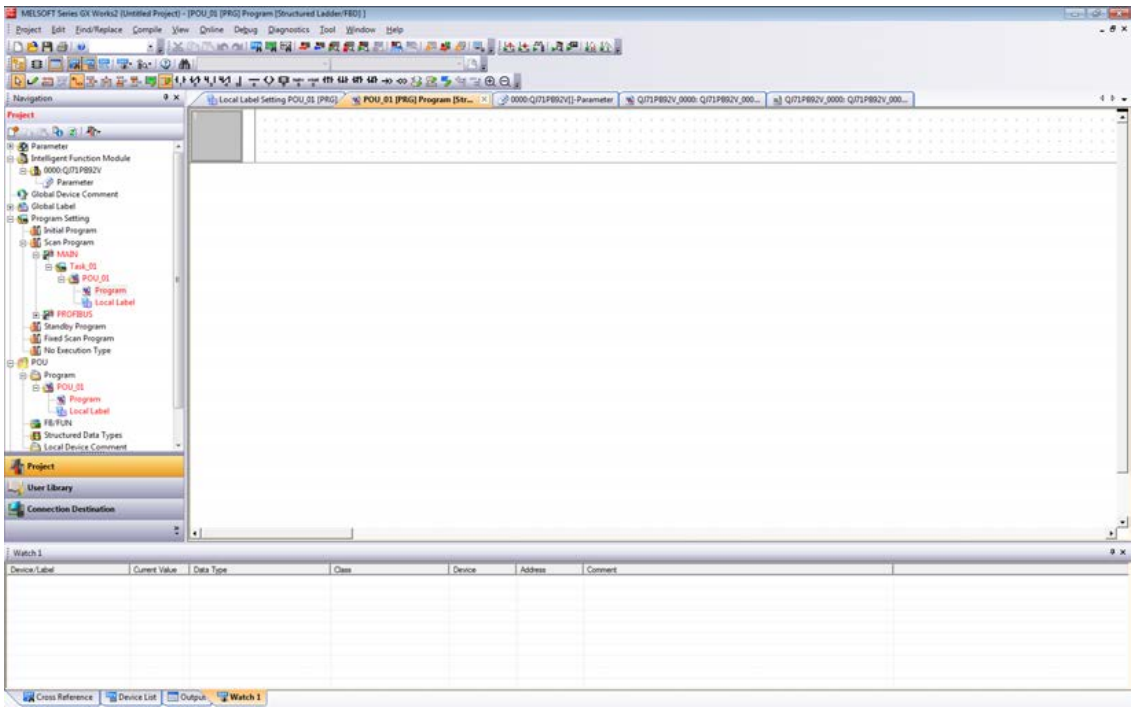
⑮ The Profibus controller is setup and ready to run after compilation and write to the PLC. Proceed to write the program to get the inverter running under a set frequency.

⑯ Go to the User Library section, select the library generated by Profibus configurator and expand Global Label section. Double click on the global variables “Global1”. Take note of the name of the global variables representing the input and output parameters to the inverter Profibus controller. In this case it should be “vHA0_input” and “vHA0_output”. Both are an array of two words containing:

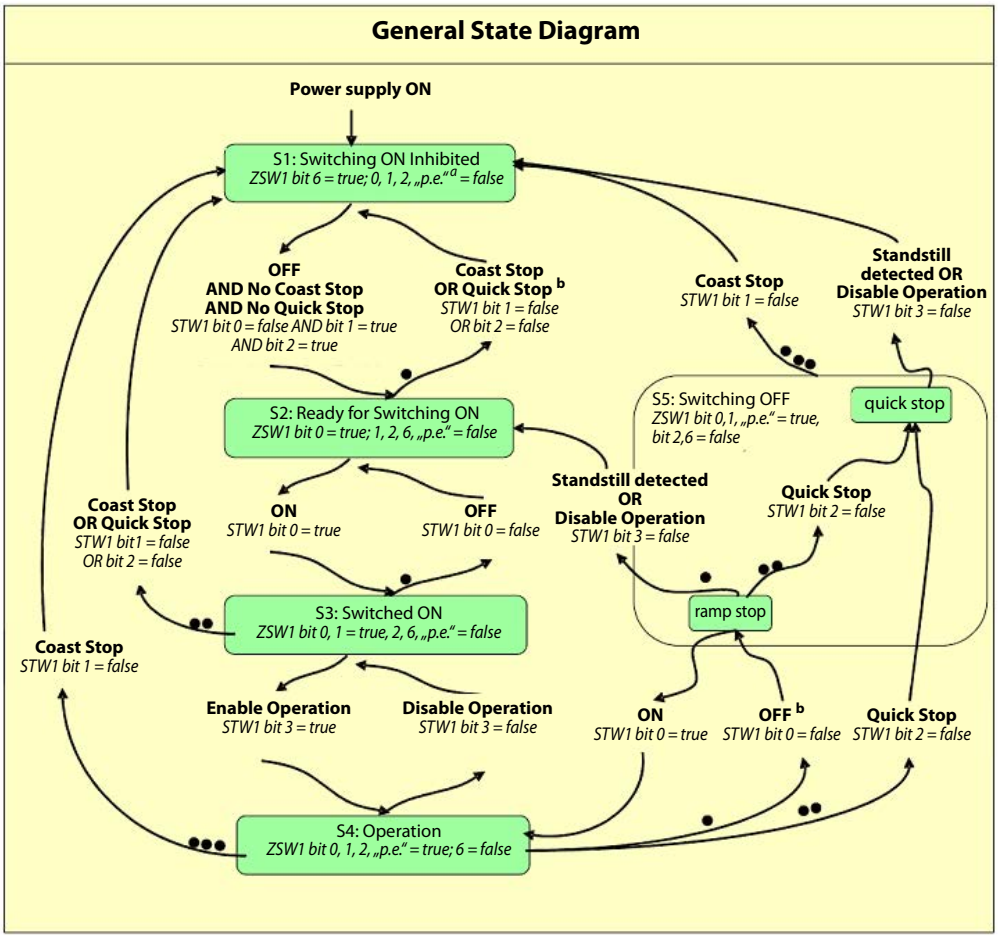
– For vHA0_input: [0]=ZSW1, [1]=NIST_A.
– For vHA0_output: [0]=STW1, [1]=NSOLL_A.



⑰ Return to the project section. From the MAIN Task, select the Program of POU_01. This Program should be blank, this is where the example user program will be written.



⑱ After powering up the inverter and Profibus master, connection will be established. Expect to see bit 6 (Switching on inhibated) set in ZSW1. Below is a simplified state diagram, dependent on control word 1 (STW1).

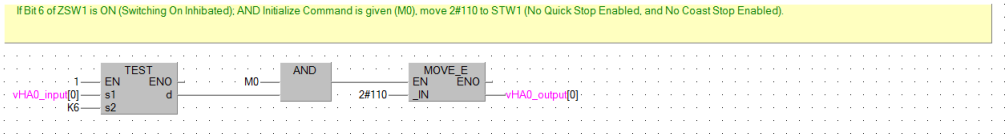


Control Word 1 (STW1) description

State / Bit combination	Bit 10	Bit 7	Bit 3	Bit 2	Bit 1	Bit 0
Description	**Control By PLC**	**Fault acknowl-edge**	**Enable operation**	**No quick stop**	**No coast stop**	**On**
Switching on Inhibited	-	-	-	-	0	-
	-	-	-	0	-	-
Ready To Switch On	-	-	-	1	1	-
Switched On	-	-	-	1	1	1
Operation	-	-	1	1	1	1
Rotation	1	-	1	1	1	1
Fault reset	-	0 =>1	-	-	-	-

Set both No Coast Stop and No Quick Stop to reset the Switching On Inhibited bit in ZSW1. To do this set bits 1 and 2 in STW1, that is global label vHA0_output[0].

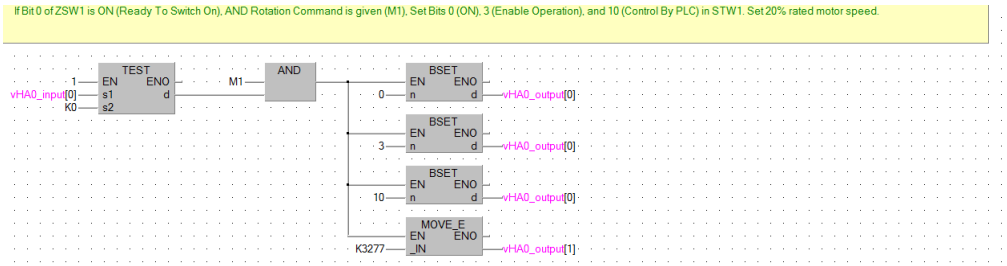
Input the following ladder block to allow the inverter to enter “Ready For Switching On” status, after connection is established, and initialization command is given (M0).



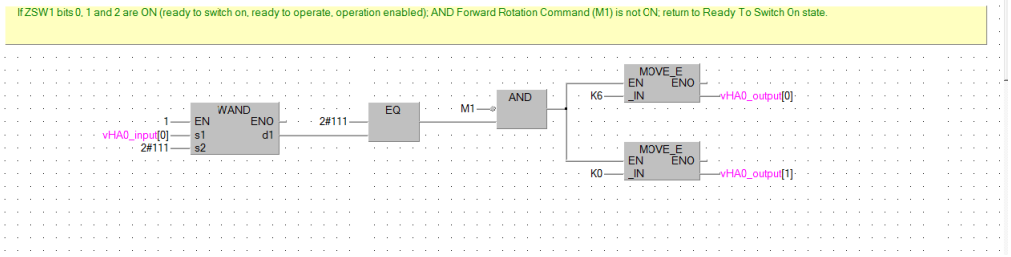
⑲ This initialization will result in setting bit 0 in ZSW1 (Ready To Switch On). It is now possible to switch on the inverter, and start operation. To do this, enable bits 0 (ON/Off), 3 (enable operation), and 10 (Control By PLC) of STW1; while leaving bits 1 and 2 enabled.

At this moment it is also possible to set the desired rotation frequency. Assuming the motor is rated for 50 Hz (inverter parameter 3) and a frequency of 10 Hz is requested, set 20% of the rated motor speed to achieve this frequency. The value in Profidrive to allow full power is 16384, so setting 20% of this value (3276,8 round to 3277) will give the expected result. This value shall be set in NSOLL_A.

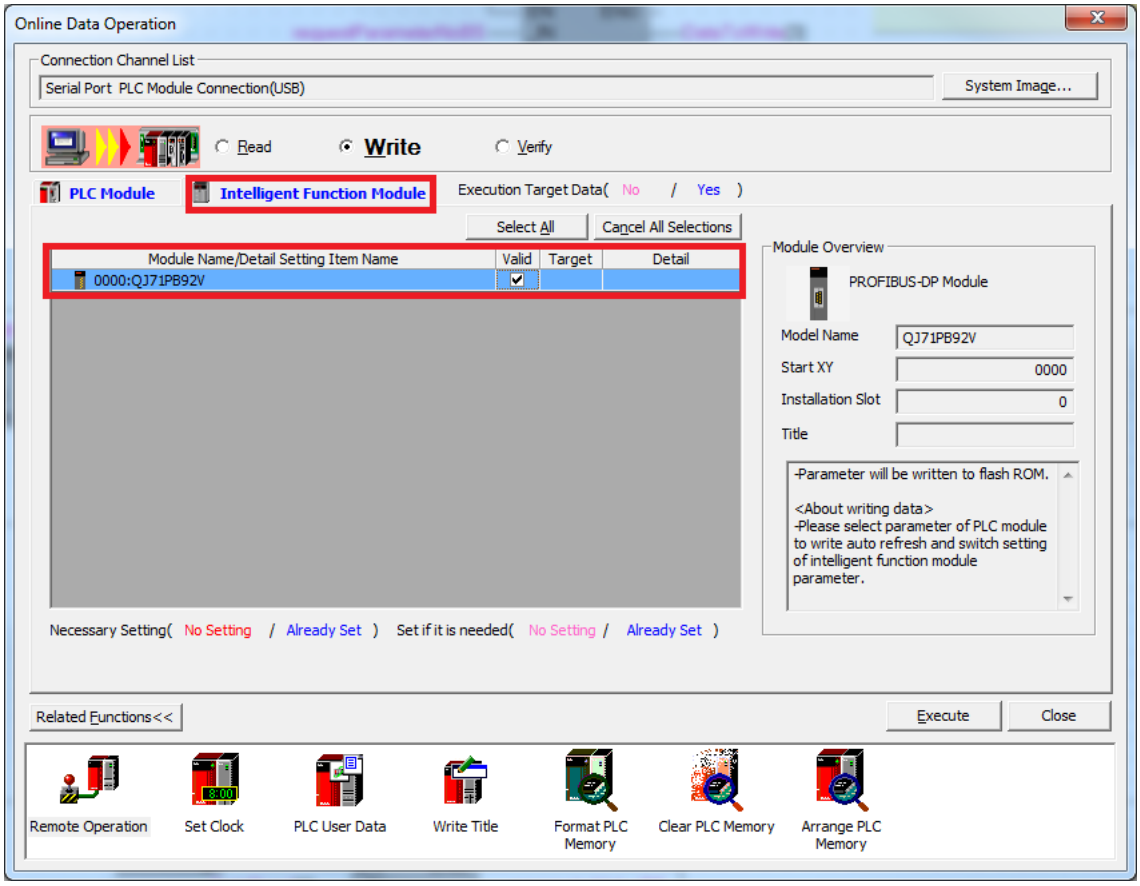
Add the following ladder block to enable rotation command after initialization, and giving rotation command (M1).



⑳ Finally add the code to stop the drive, when M1 is reset. In a new ladder block, check whether the first 3 bits of ZSW1 are ON. This condition means that the drive is in operation mode. If this condition is met, and M1 is not ON, reset the set point speed to zero, and set STW1 as 6.



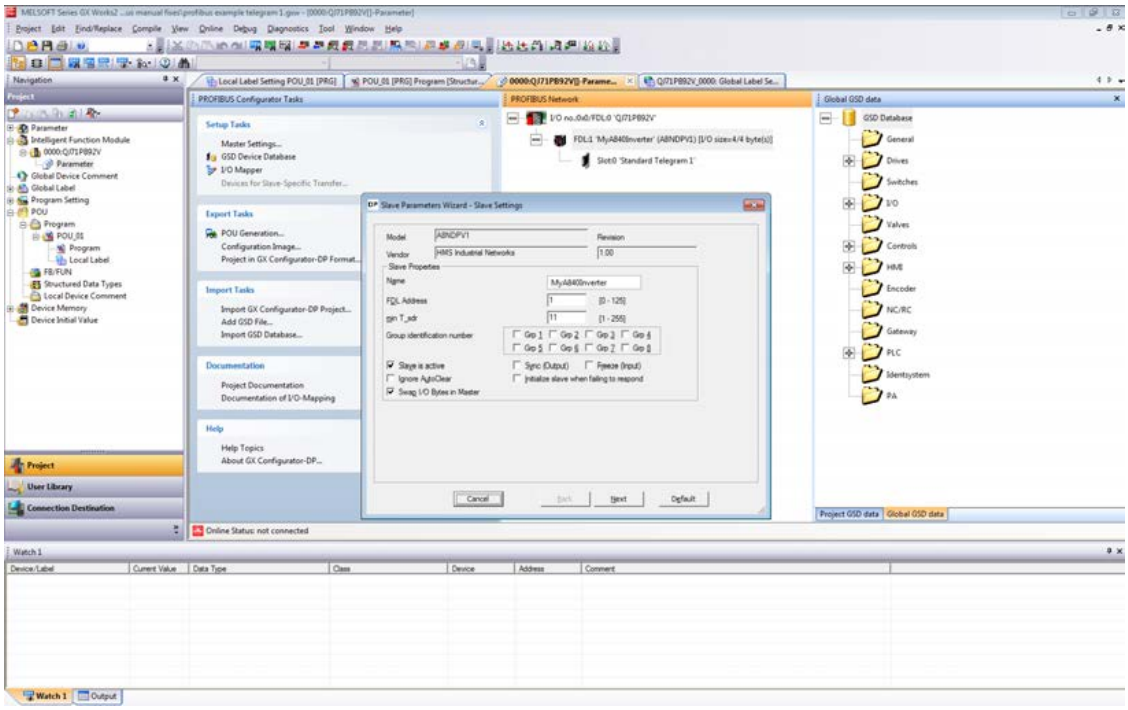
㉑ Compile, and write the program and parameters to the PLC. After resetting the PLC, and powering up the inverter, set bits M0 and M1 to get the inverter running. Be sure to select the intelligent function module, like on the screenshot below.



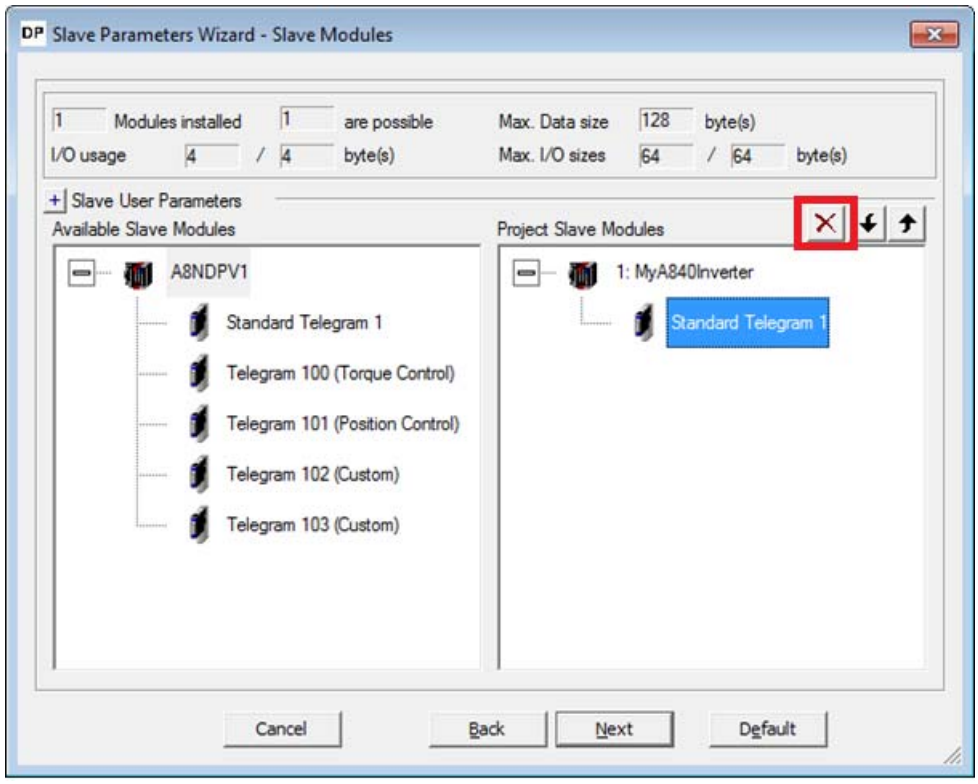
3.5 GX Works (Q-CPU) Telegram 102 example

This chapter describes running the inverter through the A8NDPV1 Profibus option card using GX Works 2 with a QCPU and QJ71PB92V Profibus Master. First perform the setup using telegram 1.

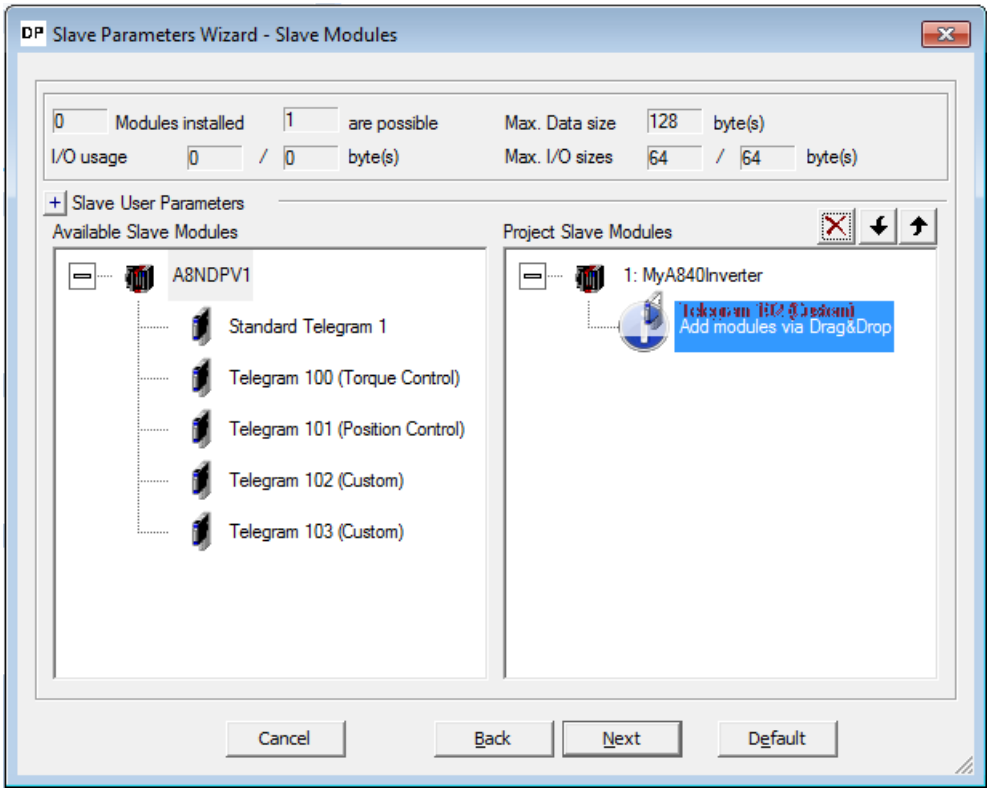
① Enter the Profibus Configurator and double click on the inverter to bring up the Slave Parameters Wizard. Click Next to proceed.



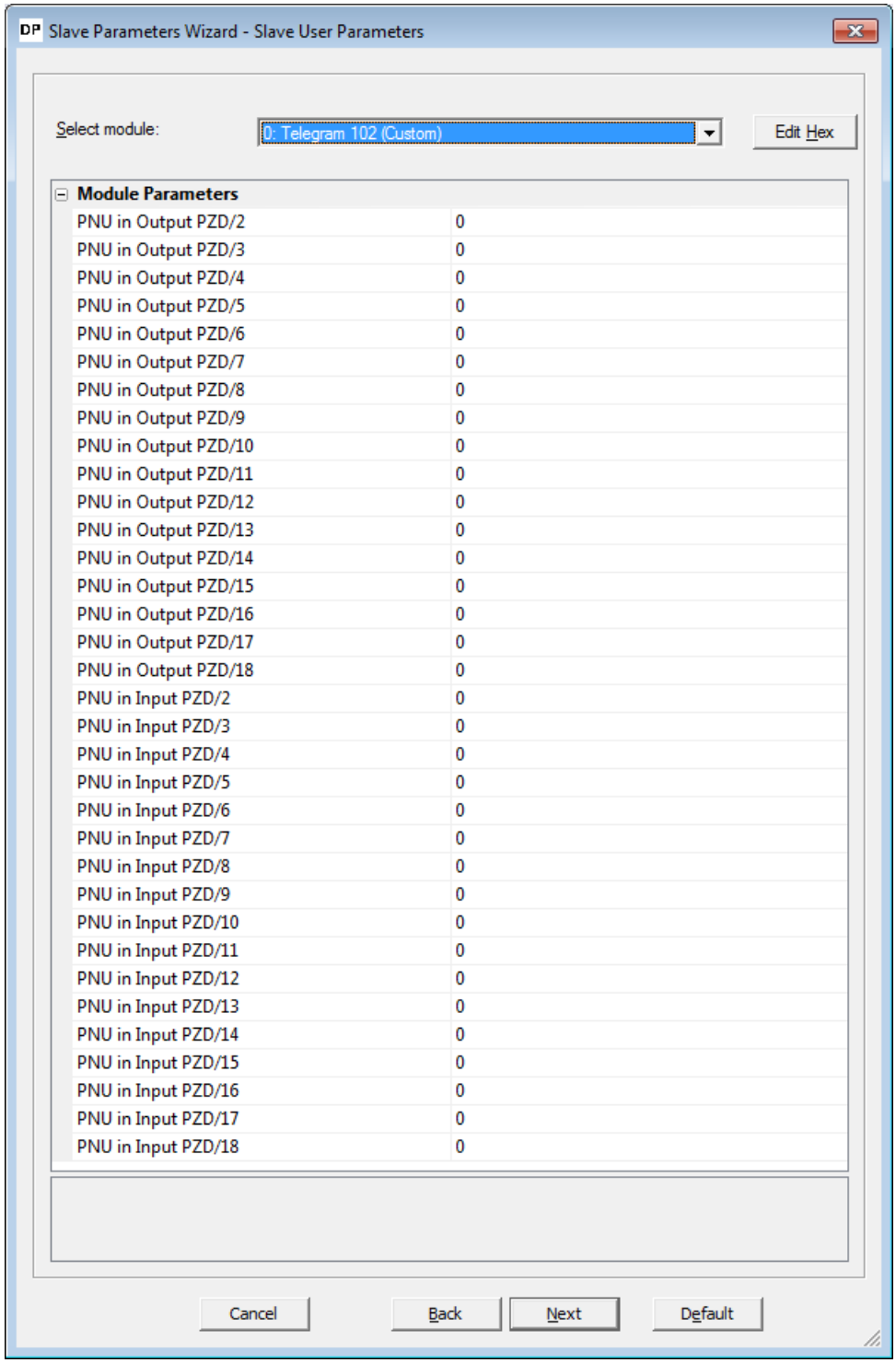
② Select “Standard Telegram 1” in the Project Slave Modules section. Click the red ‘x’ above this section to clear this slot.



③ From the Available Slave Modules, drag “Telegram 102 (Custom)” to the Project Slave Modules section. Click Next to proceed.

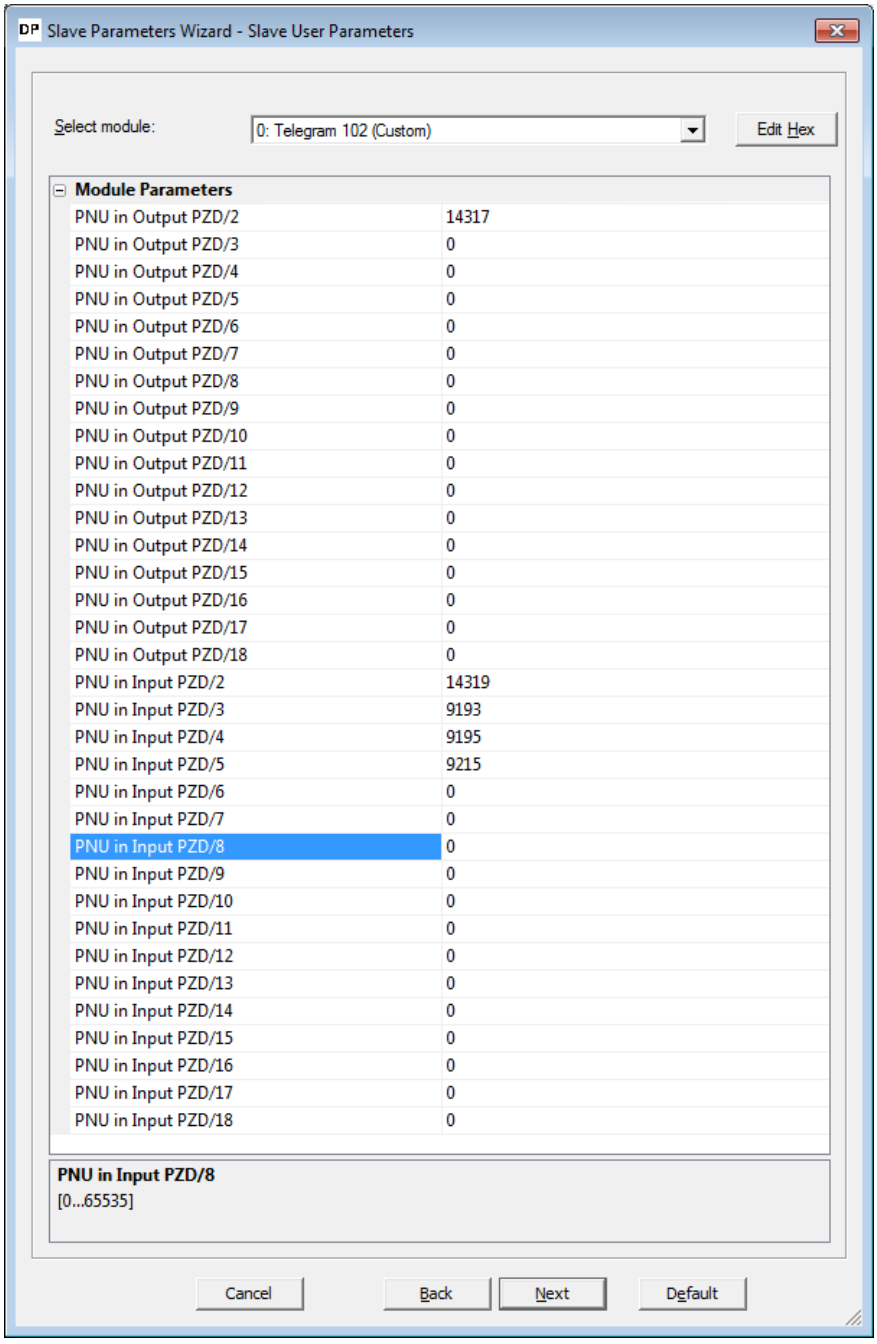


④ From the “select module” dropdown, select slot 0: Telegram 102 (Custom). It is now possible to setup the input and outputs of the telegram. There are 18 inputs and outputs used in this telegram, one of each is reserved for control/status word (leaving 17 inputs/outputs free for custom setup).

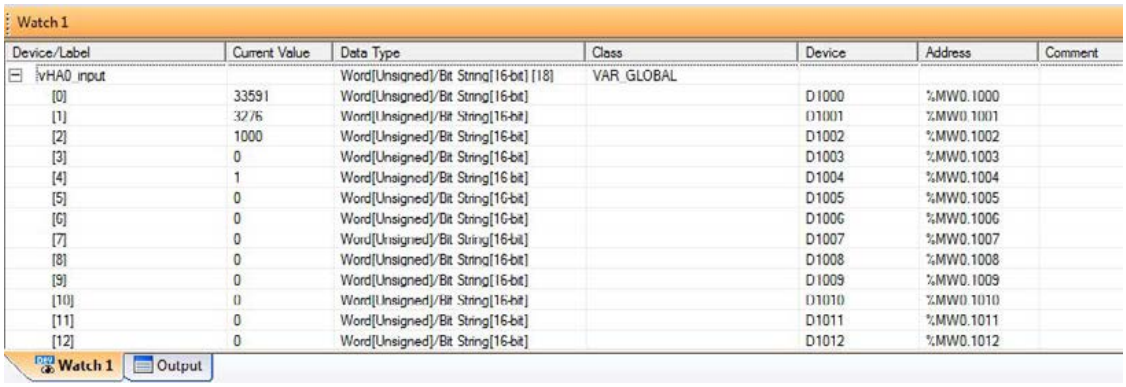


⑤ The setup of the telegram is done by entering signal/parameter numbers into the right column. All monitor data can be viewed as input, and some of the profile parameters as input/output. Please keep in mind, that not all parameters can be accessed via cyclic communication. The PROFIdrive parameter numbers (PNUs) available for use are listed in the Data Exchange subchapters 7.1to 7.4. An example setup is provided:

Slot	Description	PNU
Output PZD/2	Speed set point	14317
Input PZD/2	Actual speed set point	14319
Input PZD/3	Output frequency	9193
Input PZD/4	Output voltage	9195
Input PZD/5	Actual operation time	9215



⑥ Click Next, and Finish to proceed. Complete the POU Generation, rebuild the project, and write it to the PLC. The additional monitor data can be viewed in the appropriate position of the vHA0_input array.



3.6 GX Works (Q-CPU) Acyclic communication example

This chapter describes using acyclic communication through the A8NDPV1 Profibus option card using GX Works 2 with a QCPU and QJ71PB92V Profibus Master. It contains examples of reading and writing individual parameters, as well as arrays. First perform the setup using telegram 1.

The process of acyclic communication parameter read consists of a write request, and a response read. For details please refer to the Acyclic Data Exchange subchapter 7.7. These operations are done using libraries supplied with the QJ71PB92V module (lib_dpv1). To perform acyclic communication without this library, please refer to the Simple Ladder example (chapter “GX Works (Q-CPU) Simple Ladder Acyclic communication example” on page 54).

3.6.1 Reading a parameter (Sequence 1)

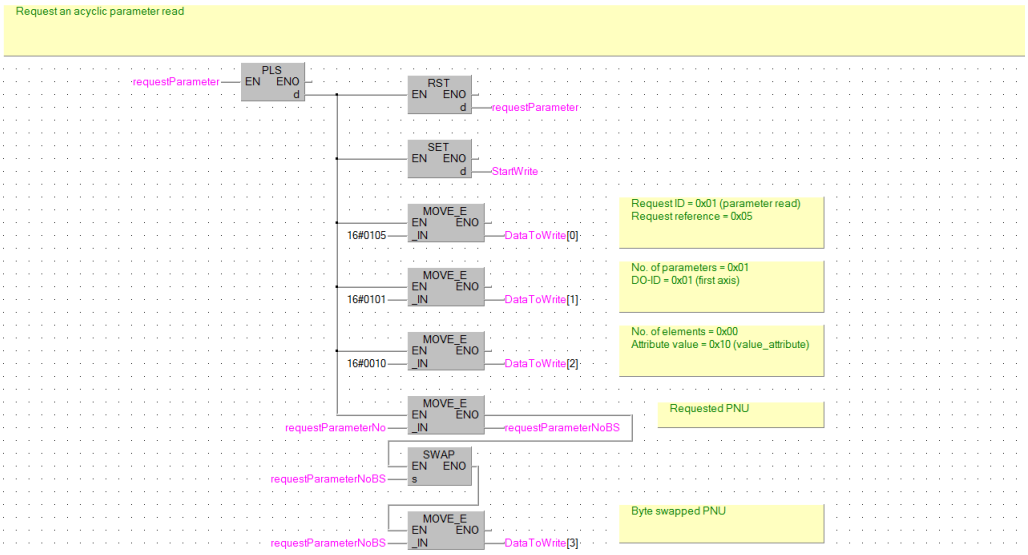
Creating the request

Start by preparing the content for the parameter read request. The data request has the following format:

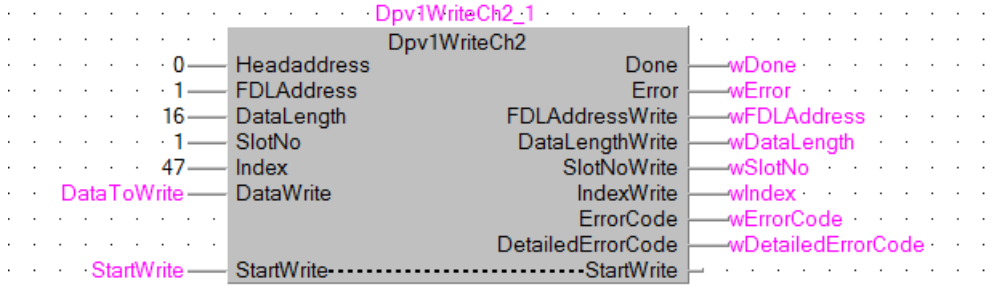
Byte no.	Description	Value
0	Request ID	0x01
1	Request reference	0x01-0xFF
2	No. of parameters	0x01
3	DO-ID	0x01
4	No. of elements	0x00
5	Attribute value	0x10 (value attribute)
6-7	Parameter number	Byte swapped PNU

Request reference is any valid number, this value is mirrored back in the response and can be used to distinguish multiple requests. The parameter number needs to be byte swapped(the SWAP instruction can be used to swap the lower and higher byte of a word).

In the code below *requestParameterNo* contains the unswapped parameter number, while *requestParameterNoBS* contains the byte swapped version. The content of the parameter request(*DataToWrite*), and the request to perform it (*StartWrite*), are shown in the screenshot below:



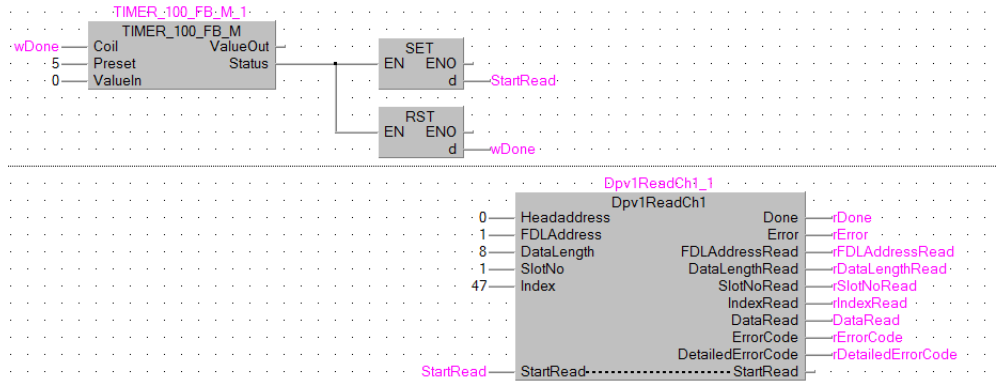
Proceed to issue the actual request using a function block from the *lib_dpv1* library. Pick any *Dpv1WriteCh** function block, using different channels allows for simultaneous communication to multiple slaves.



Receiving the response

If the request is completed successfully (bit *wDone* is ON), then perform a response read after a delay of 500ms. Due to the nature of Profidrive acyclic communication, the response may not be immediately available after execution of the request. Waiting a small amount of time significantly increases the chance of receiving a response on the first try. Another solution is to keep executing read requests, until a proper read request is received.

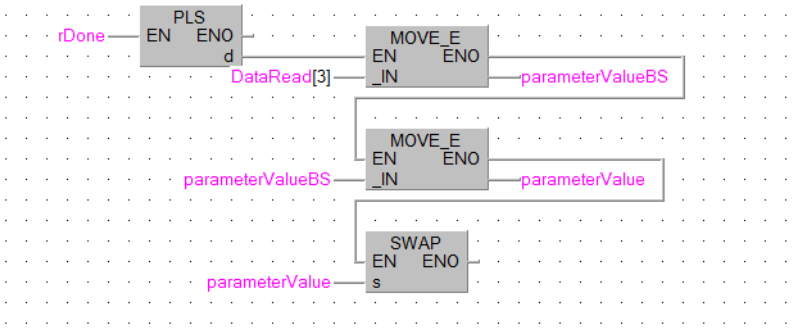
It should also be noted, that 240 bytes can always be read using the *Dpv1ReadCh** function blocks. This is the maximum allowed length of a response. In case of a shorter response, only the available data will be read.



If the response is received successfully (bit *rDone* is ON), then the received data can be accessed from *DataRead*. The format of a positive response is as follows:

Byte no.	Description	Value
0	Request ID	0x01
1	Request reference	0x01-0xFF (same as in request)
2	No. of parameters	0x01
3	DO-ID	0x01
4	No. of values	0x01
5	Format	See data format type table
6-7	Parameter value	Byte swapped parameter value

Extract the parameter value from *DataRead*, and byte swap it to receive the actual value:

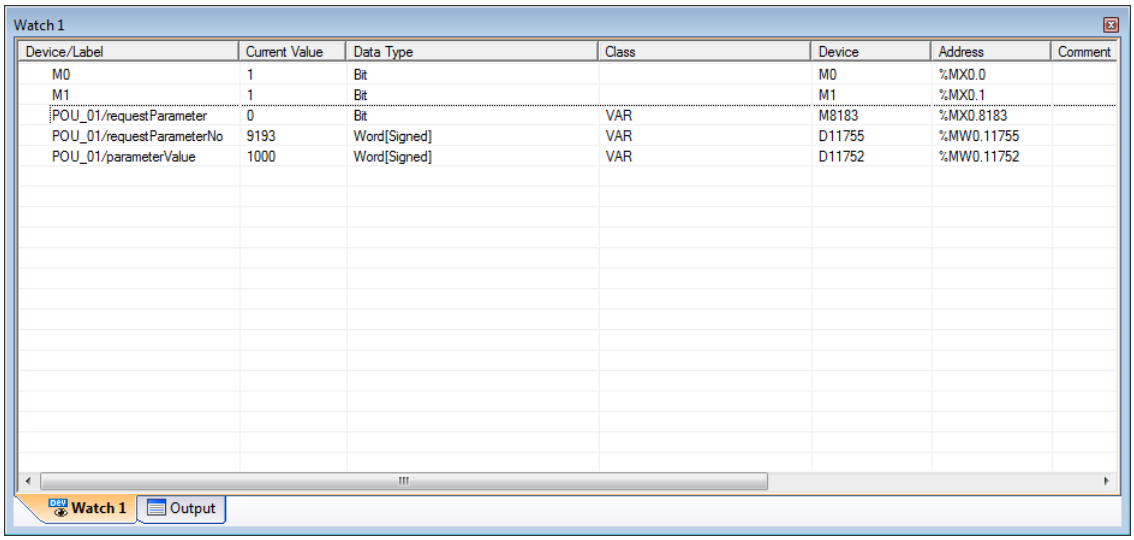


In case of a negative response, the format is as follows:

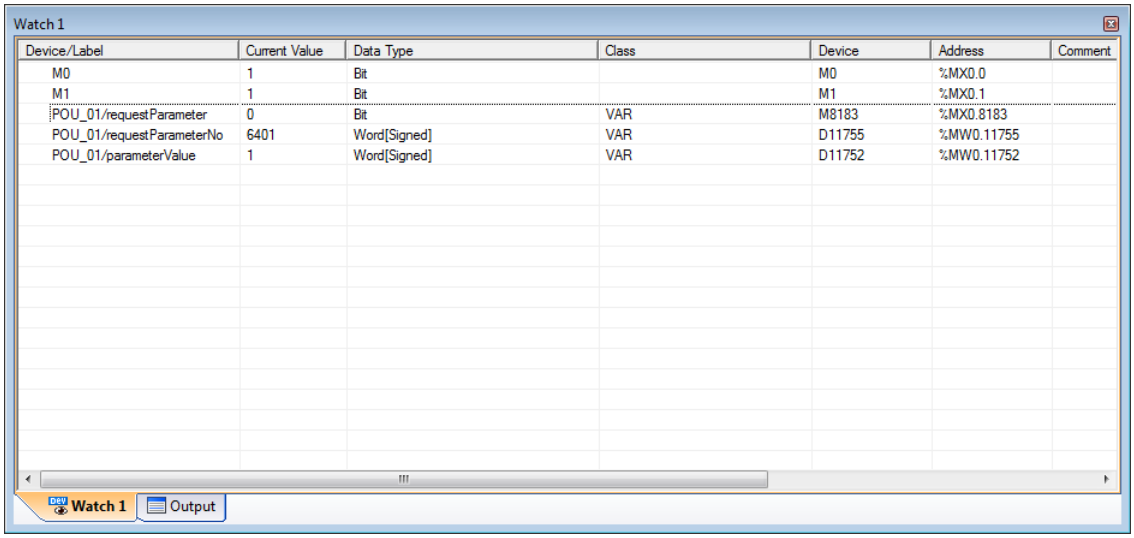
Byte no.	Description	Value
0	Request ID	0x81
1	Request reference	0x01-0xFF (same as in request)
2	No. of parameters	0x01
3	DO-ID	0x01
4	No. of values	0x01
5	Format	0x44 (error)
6-7	Error value	Byte swapped error value (see error table)

Executing example requests

After compiling the program, and writing it to the PLC, it is possible to execute parameter read requests. First, get the inverter running by setting bits M0 and M1. Now read the actual frequency by writing PNU 9193 to *requestParameterNo* and setting bit *requestParameter*. The result should be a value of 1000 in *parameterValue* which corresponds to 10Hz. The PROFIdrive parameter numbers (PNUs) available for use are listed in the Data Exchange subchapters 7.1to 7.4.



Try reading an inverter parameter, for example the FDL address of the inverter (parameter 1305 + 5096 offset = PNU 6401). The expected result is 1. It is possible to read any inverter parameter in this way.

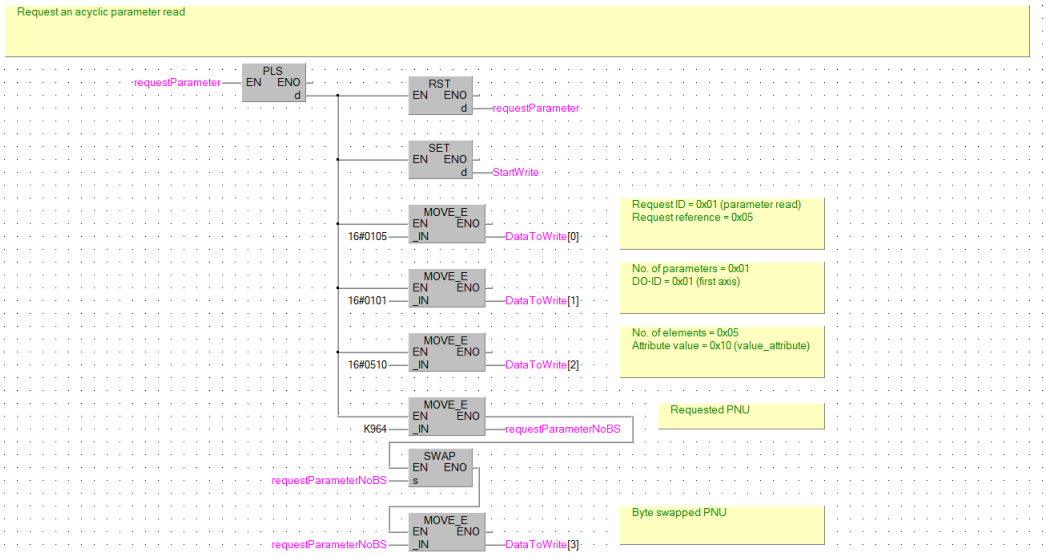


3.6.2 Reading an array of parameters (Sequence 3)

Certain parameters are available in the form of an array. It is possible to access individual elements of the array be changing the subindex. It is also possible to retrieve multiple elements of the array by executing sequence 3. This requires only changing byte no. 4 of the request (No. of values) to the amount of array values wanted. The response will contain additional parameter values added to the end of the response data.

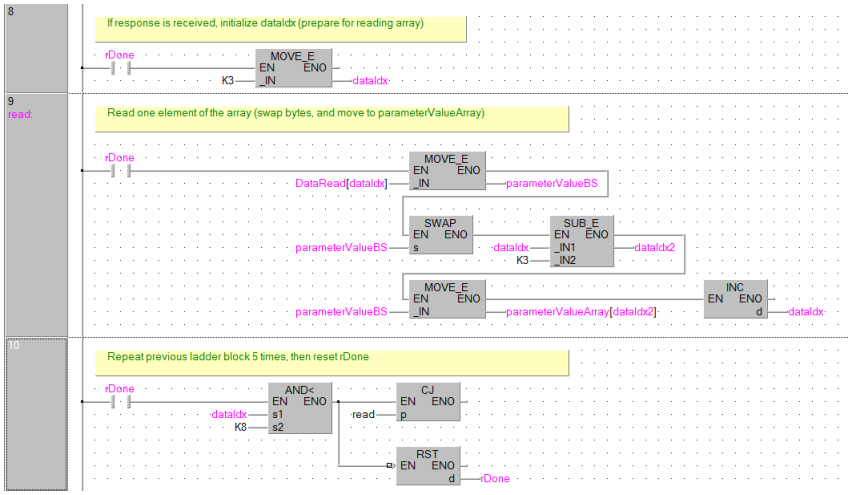
Creating the request

For this example all 5 array elements of PNU 964 (Device identification) will be requested. Modify the parameter request as described above:



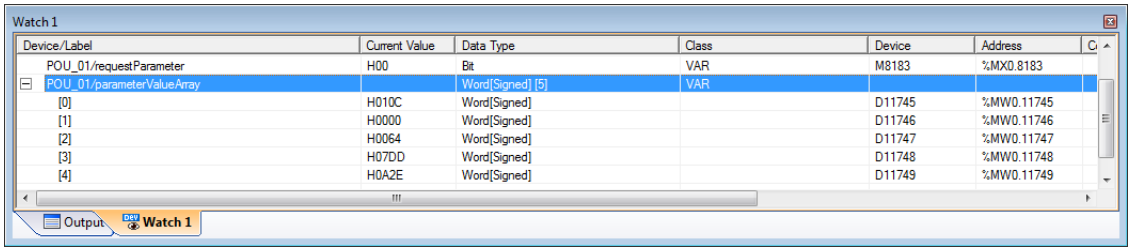
Reading the response

Modify the code reading the response data, to read all 5 elements, and byte swap each one. An example procedure how to do this is shown below. The byte-swapped elements will be available in the *parameterValueArray*.



Executing example request

After executing a request, the response data is available in the *parameterValueArray* variable.



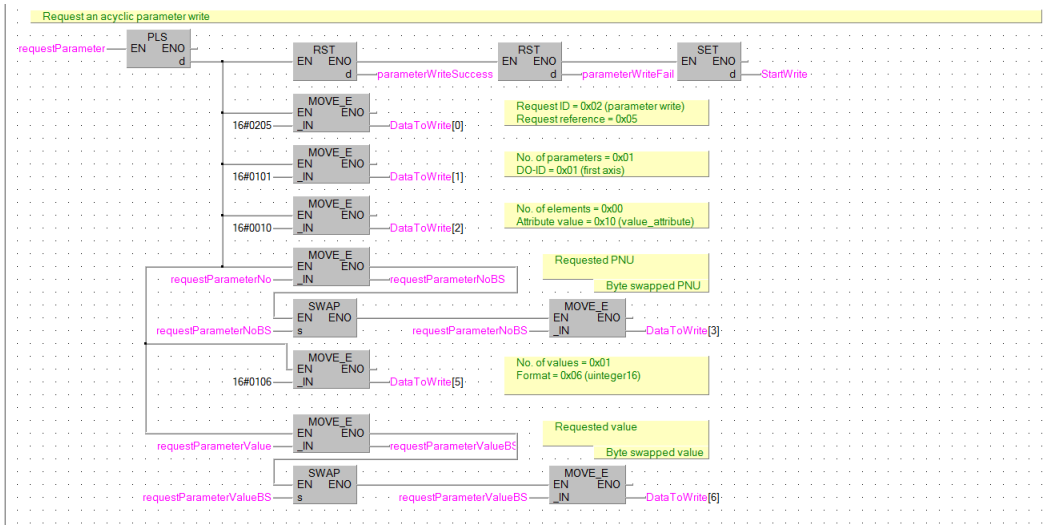
3.6.3 Changing parameters (Sequence 2)

Creating the request

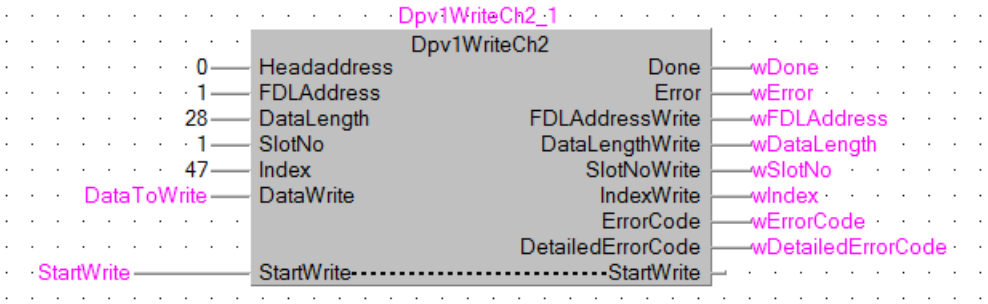
The format of the write request is as follows:

Byte no.	Description	Value
0	Request ID	0x02
1	Request reference	0x01-0xFF
2	No. of parameters	0x01
3	DO-ID	0x01
4	No. of elements	0x00
5	Attribute value	0x10 (value attribute)
6-7	Parameter number	Byte swapped PNU
8-9	Subindex (irrelevant)	0x00
10	No. of values	0x01
11	Format	0x06 (UINTEGER16 for all inverter parameters) See data format type table
12-13	Set value	Byte swapped set value

Modify the ladder block preparing the content of the request.

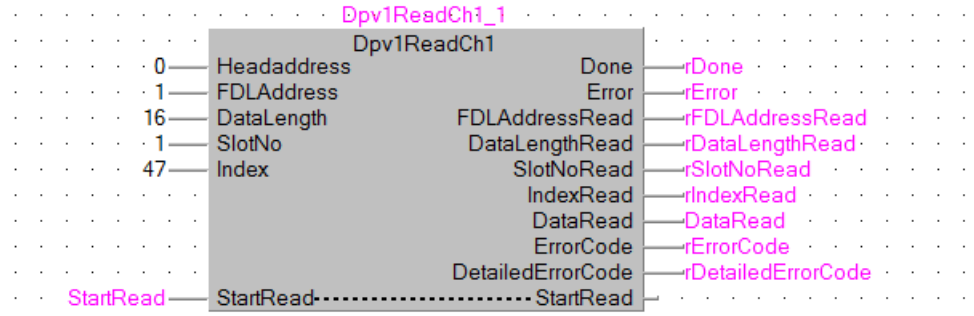


Change the actual length of the write request. Change the length to 28 hex characters in the ladder block that executes the request.



Reading the response

The length of the expected read response is up to 8 bytes.



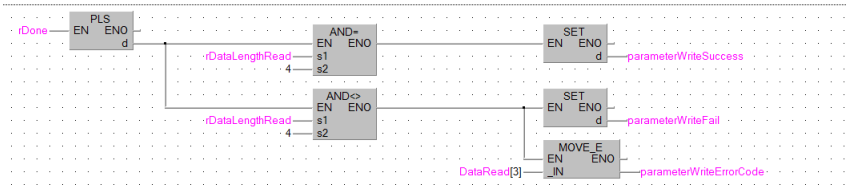
Finally modify the ladder block processing the response data. A positive response consists of just 4 bytes of data in the following format:

Byte no.	Description	Value
0	Request ID	0x02
1	Request reference	0x01-0xFF (same as in request)
2	No. of parameters	0x01
3	DO-ID	0x01

In case of a negative response, the format is as follows:

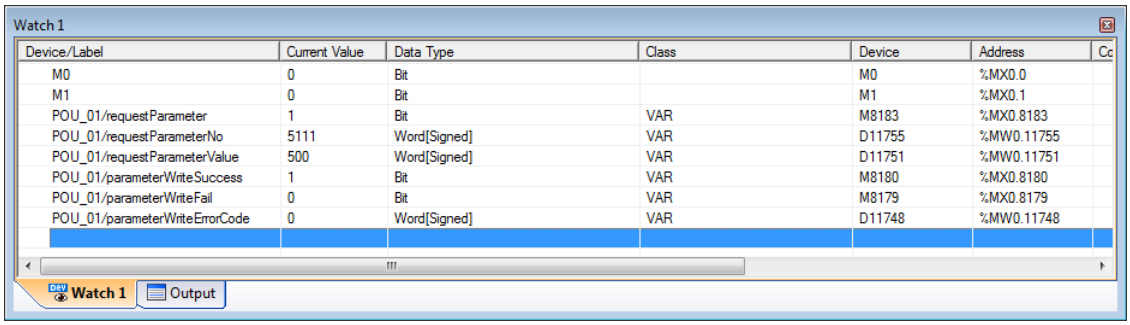
Byte no.	Description	Value
0	Request ID	0x82
1	Request reference	0x01-0xFF (same as in request)
2	No. of parameters	0x01
3	DO-ID	0x01
4	No. of values	0x01
5	Format	0x44 (error)
6-7	Error value	Byte swapped error value (see error table)

After a successful response is received (bit *rDone* is ON), check the length of the received response to determine, whether the parameter write request was processed successfully. If the length (*rDataLengthRead*) is equal to 4, set a bit indicating a successful parameter write. If this is not true, a bit indicating parameter write failure shall be set, and the error code shall be extracted to variable *parameterWriteErrorCode*.

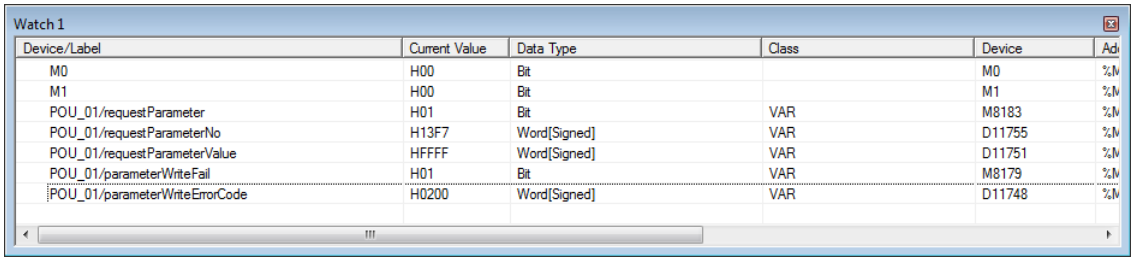


Executing example requests

To perform a parameter write, set the PNU of the parameter to change in *requestParameterNo*. As an example, to change parameter 15 (Jog frequency), add the offset 5096 and write the result to this variable. Set variable *requestParameterValue* to the requested value. The value that will be set, will be the output of multiplication of the set value, and the minimum setting increments, as according to the parameter list in the inverter manual. For parameter 15, the minimum setting increment is 0.01Hz. If a jog frequency of 5Hz is requested, set *requestParameterValue* to 500. Finally execute the request by setting bit *requestParameter* to ON. A successful write will result in setting of the *parameterWriteSuccess* bit.



Test receiving a negative response, by trying to write an out of range value. After trying to write 0xFFFF as the value of parameter 15, a negative response with error code 0x02 (LOW_OR_HIGH_LIMIT_EXCEEDED) will be received, which is the expected behavior.

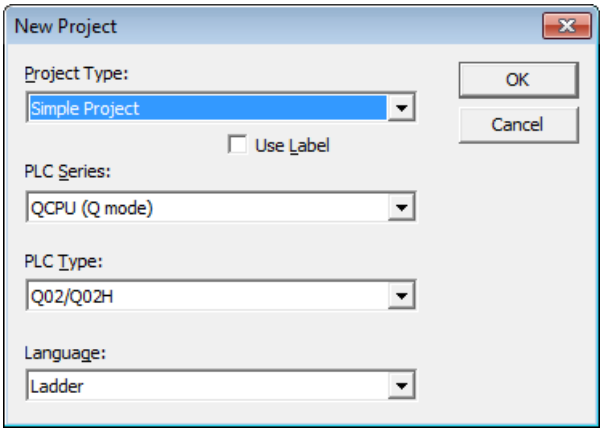


3.7 GX Works (Q-CPU) Simple Ladder Telegram 1 example

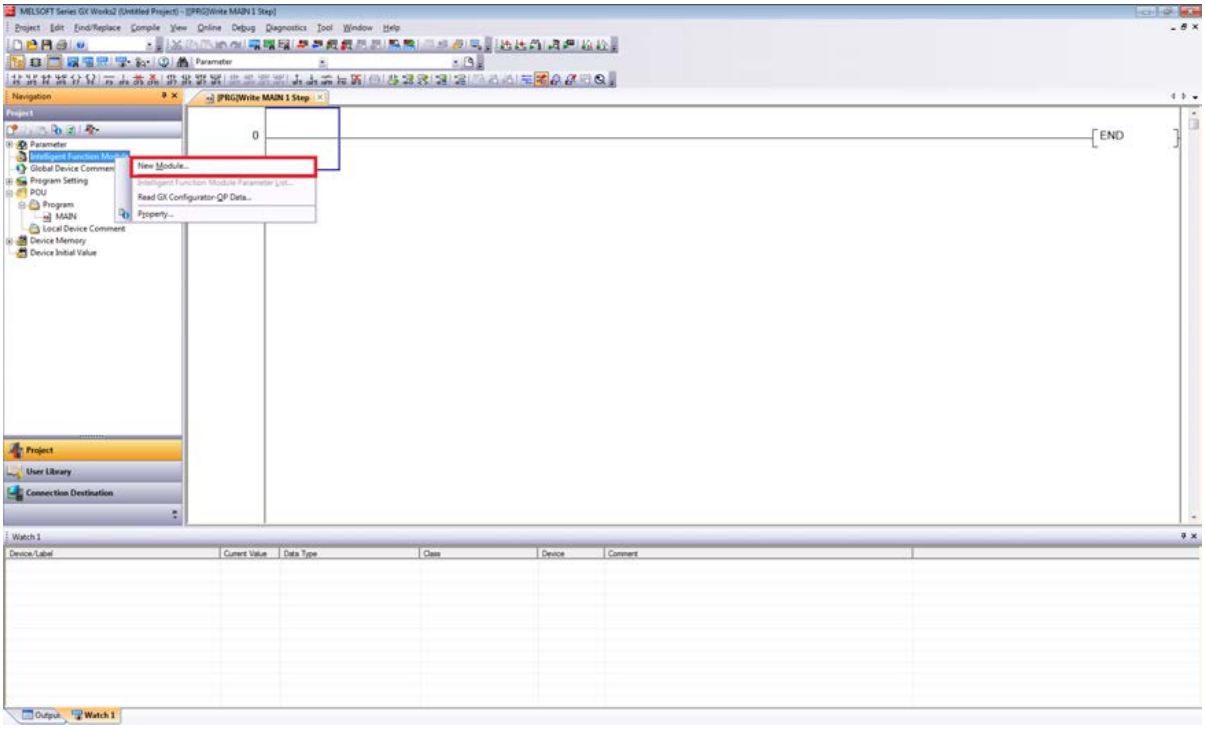
This chapter describes running the inverter through the A8NDPV1 Profibus option card using GX Works 2 in Simple Ladder mode with a QCPU and QJ71PB92V Profibus Master. GX Configurator DP 7.10L or newer is required to perform the setup.

① Start with a new project in simple ladder mode. Select:

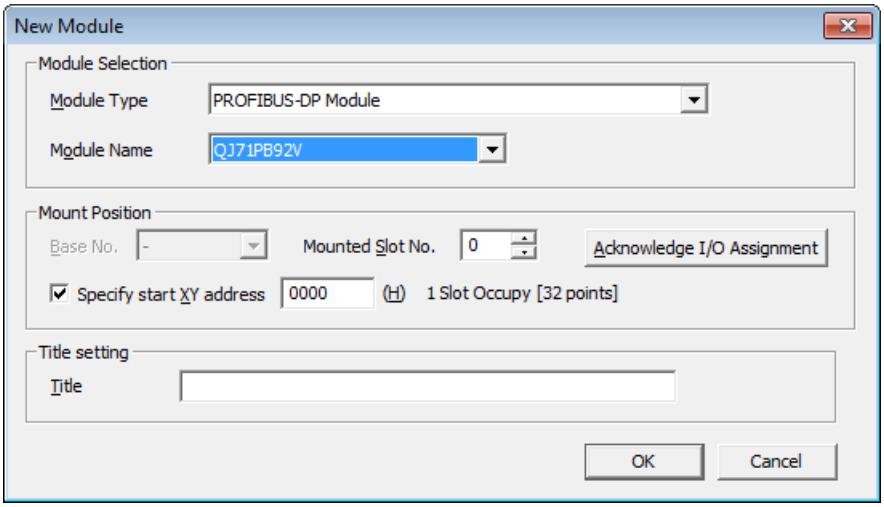
- Project type: Simple project;
- PLC Series: QCPU (Q mode);
- PLC type: select the cpu;
- Language: Ladder.



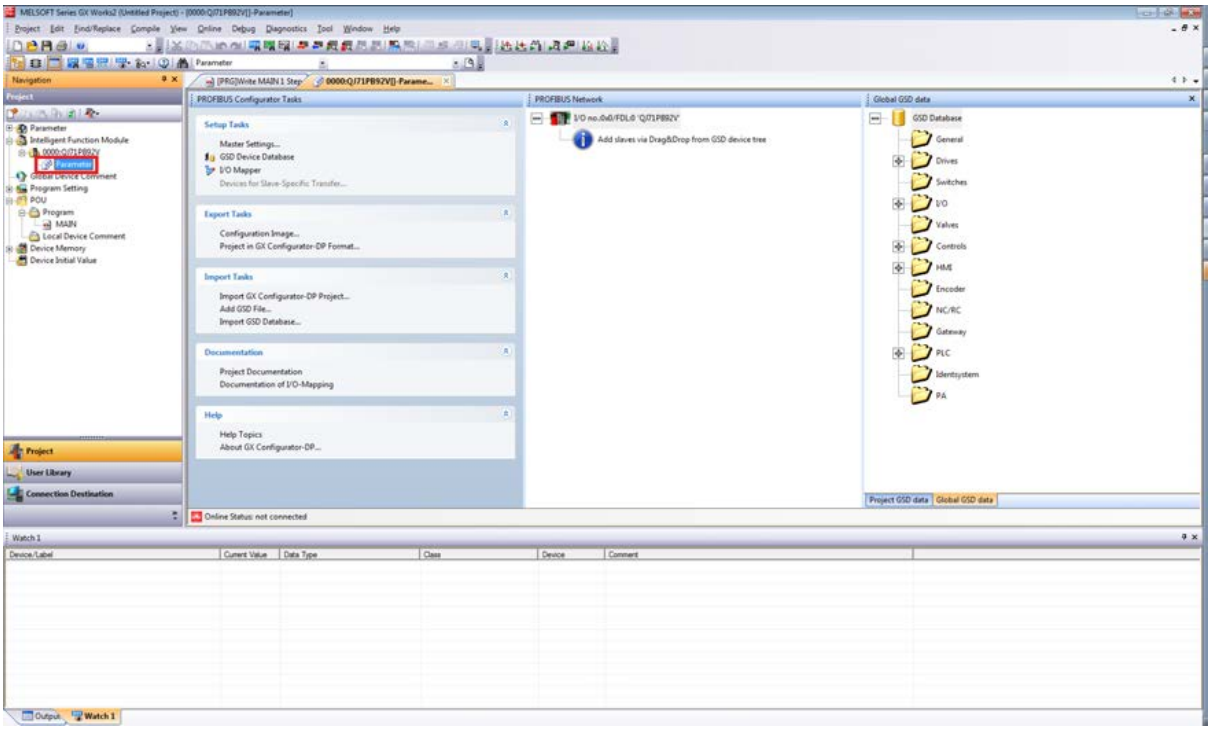
② Add a new Intelligent Function Module from the menu on the left (right click and select new):



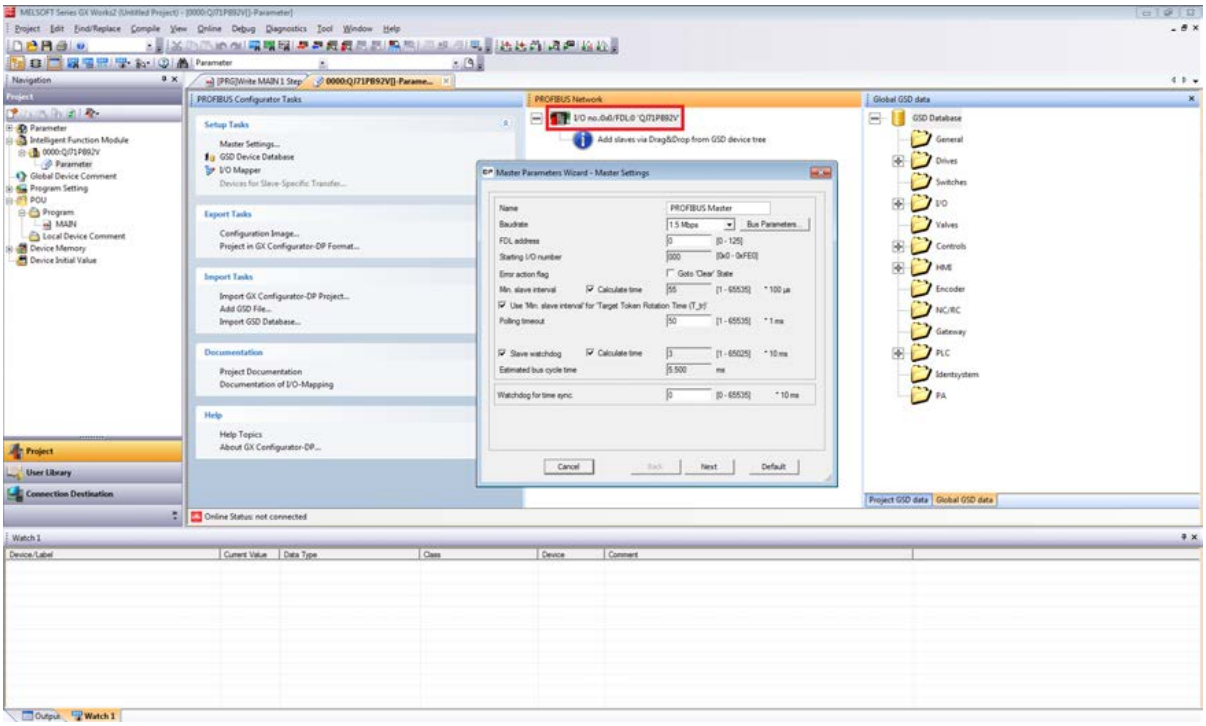
③ In the new window, click Module Type and select “PROFIBUS-DP Module”. Module Name should be set to QJ71PB92V. Save this setting by clicking OK.



④ The Profibus-DP module should be inserted into the Intelligent Function Module tree. Expand it, and double click on Parameter. This should open the Profibus Configurator.

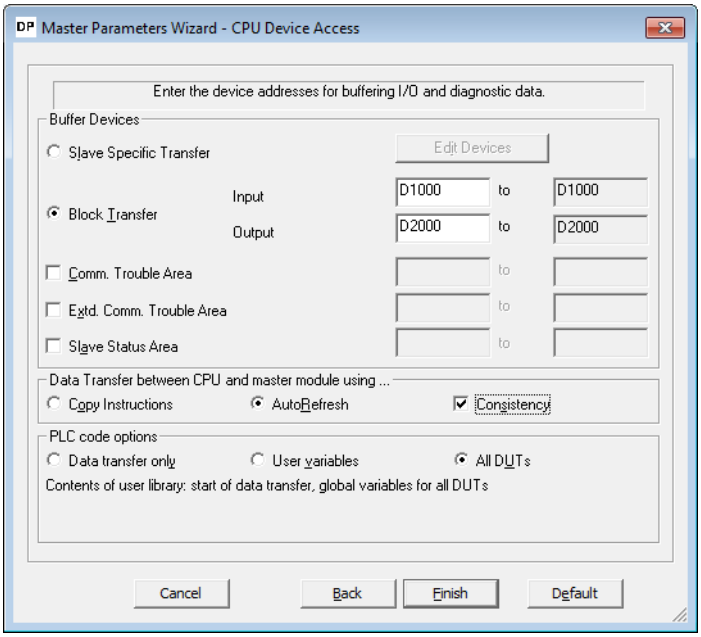


⑤ First setup the Profibus Master. Double click on the master module (in the screenshot visible as “I/O no.:0x0 FDL:0 ‘QJ71PB92V’. This will bring up the “Master Parameters Wizard”.

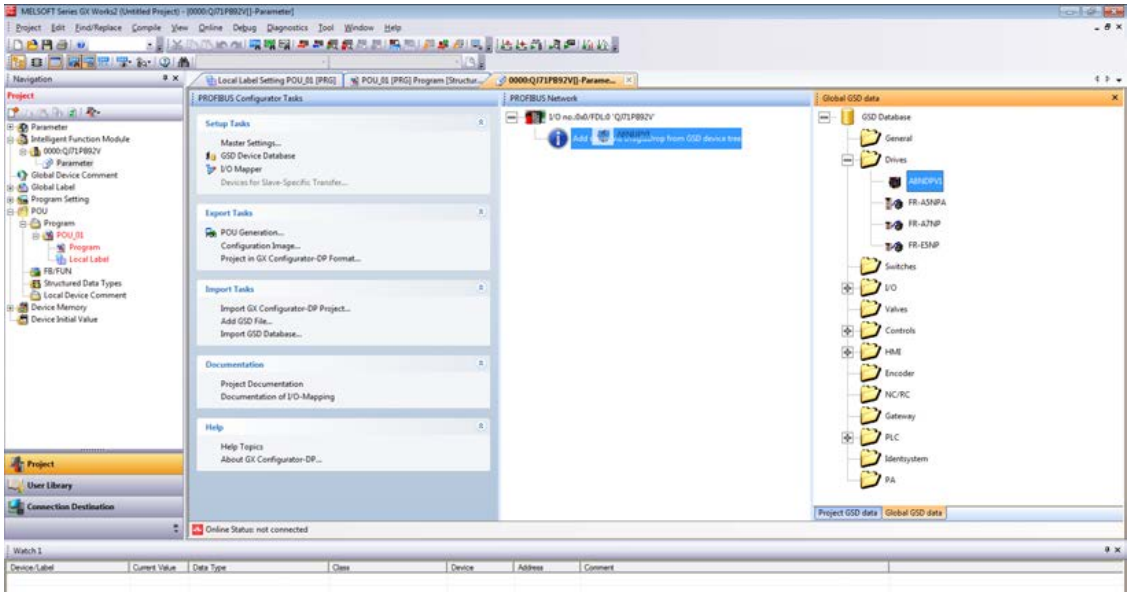


⑥ The Baudrate, FDL address and other settings can be changed as needed. For this tutorial these settings will not be changed. Click Next to proceed.

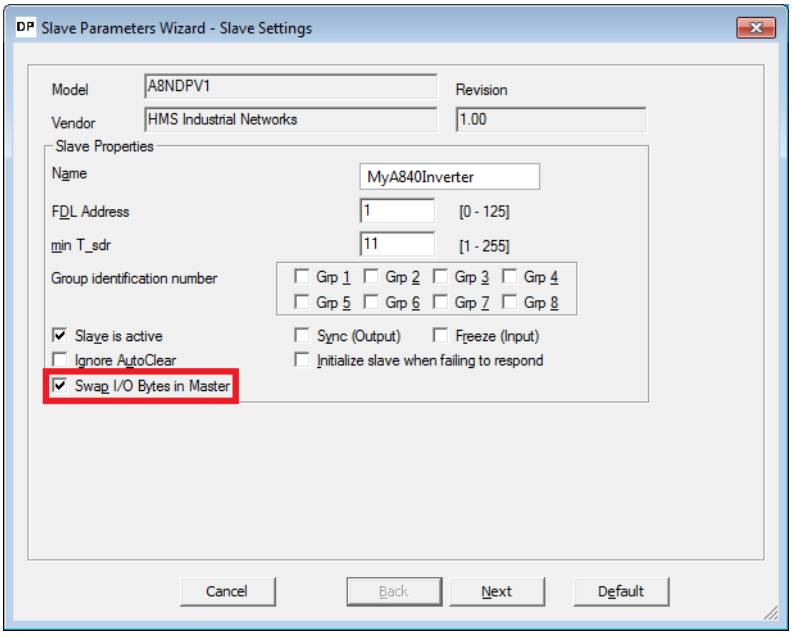
⑦ On this screen it is possible to change the buffer devices, and data transfer between Profibus Master and CPU. For this tutorial use AutoRefresh with Consistency. Click Finish to save settings and close this wizard.



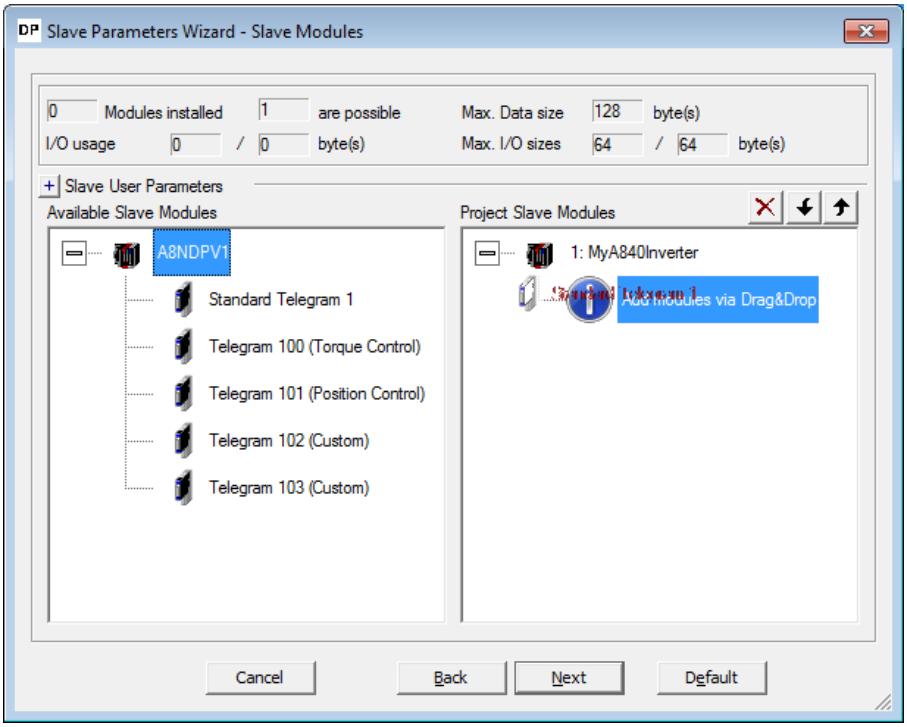
⑧ In the Global GSD data section, expand drives and search for A8NDPV1. If it is not available, right click and select Add GSD File. Search for the GSD File on the drive (included with this manual), afterwards the A8NDPV1 should appear under drives. Drag this option to the Profibus Master, like on the screenshot.



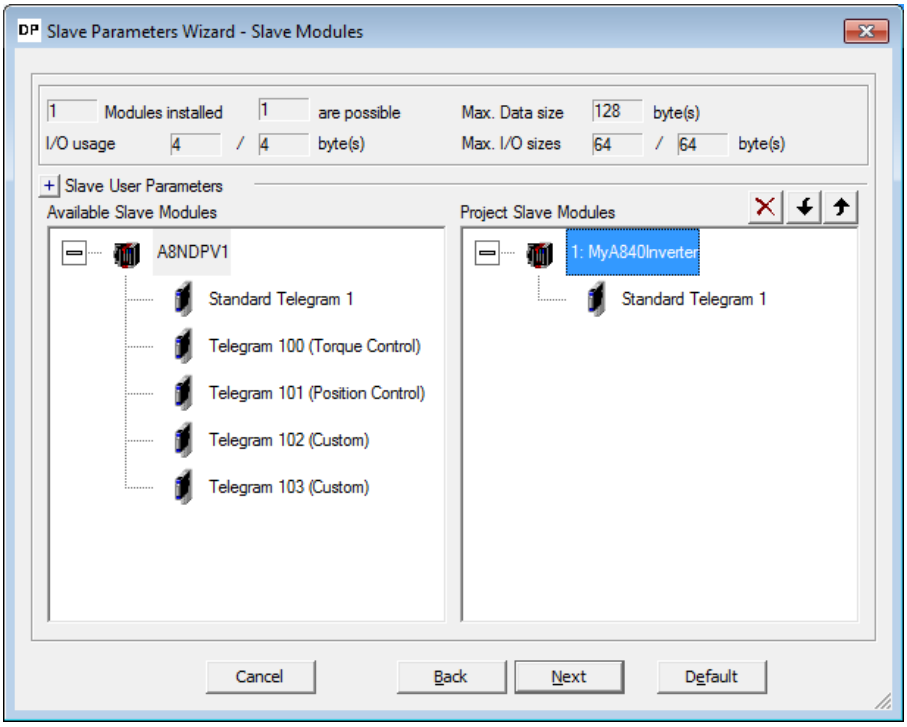
The “Slave Parameters Wizard” should now be shown. Enter a name, select 1 for FDL Address, and make sure to select “Swap I/O Bytes in Master”. All other settings can remain as default. Click Next to proceed.



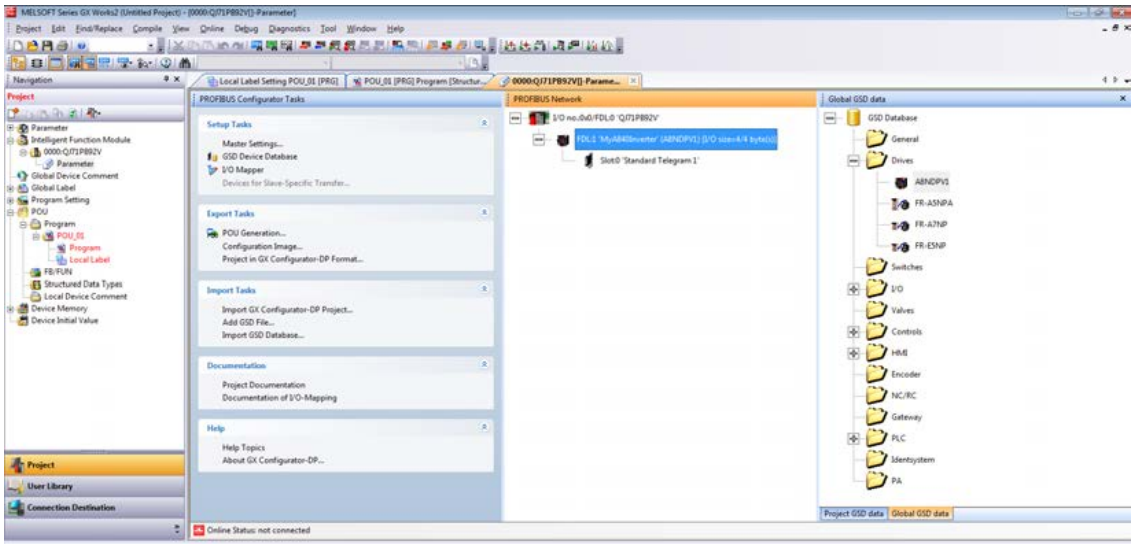
⑨ Select the communication protocol used to communicate with the option card. For this tutorial use “Standard Telegram 1”. Drag this option from the “Available Slave Modules” to “Project Slave Modules”. The A8NDPV1 supports only one module.



The result should be:



⑩ Click Next to proceed. No changes are required on this screen. After clicking Next proceed to the final screen and click Finish. The configured Profibus Slave should appear in the Profibus Network window, like below.



⑪ As of now, the Profibus controller is setup and ready to run after compilation and write to the PLC. Proceed to write the program to get the inverter running under a set frequency. Below is a table of PLC devices and their corresponding meaning in PROFIdrive

Device	Description
D1000 (input)	ZSW1
D1001 (input)	NIST_A
D2000 (output)	STW1
D2001 (output)	NSOLL_A

⑫ Start the ladder program by initializing Profibus communication.

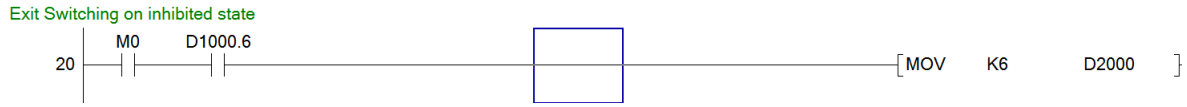
⑬ After powering up the inverter and Profibus master, connection will be established. Expect to see bit 6 (Switching on inhibated) set in ZSW1. Refer to the simplified state diagram (chapter GX Works (Q-CPU) Telegram 1 example), dependent on control word 1 (STW1).

Control Word 1 (STW1) description

State / Bit combination	Bit 10	Bit 7	Bit 3	Bit 2	Bit 1	Bit 0
Description	**Control By PLC**	**Fault acknowl-edge**	**Enable operation**	**No quick stop**	**No coast stop**	**On**
Switching on Inhibited	-	-	-	-	0	-
	-	-	-	0	-	-
Ready To Switch On	-	-	-	1	1	-
Switched On	-	-	-	1	1	1
Operation	-	-	1	1	1	1
Rotation	1	-	1	1	1	1
Fault reset	-	0=>1	-	-	-	-

Both No Coast Stop and No Quick Stop need to be set to reset the Switching On Inhibited bit in ZSW1. To do this set bits 1 and 2 in STW1, that is D2000.

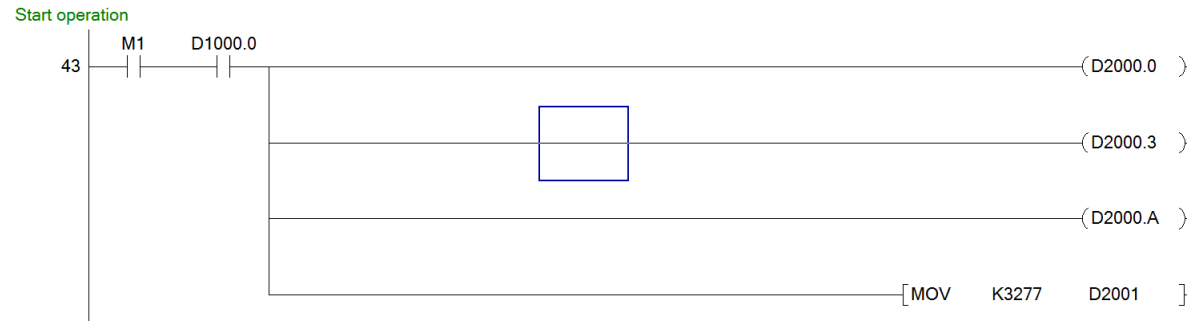
Input the following ladder to allow the inverter to enter “Ready For Switching On” status, after connection is established, and initialization command is given (M0).



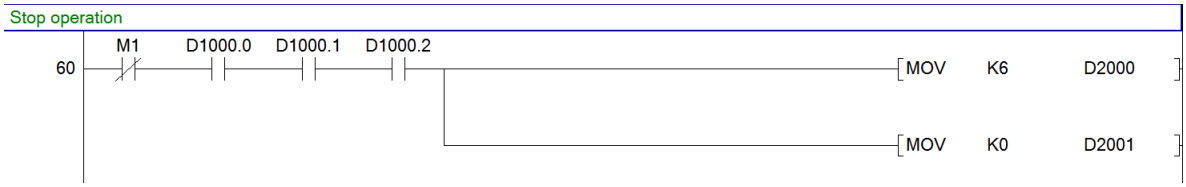
⑭ This initialization will result in setting bit 0 in ZSW1 (Ready To Switch On). Switch on the inverter, and start operation. To do this enable bits 0 (ON/Off), 3 (enable operation), and 10 (Control By PLC) of STW1; while leaving bits 1 and 2 enabled.

⑮ At this moment it is also possible to set the desired rotation frequency. Assuming the motor is rated for 50 Hz (inverter parameter 3) and a frequency of 10 Hz is requested, set 20% of the rated motor speed to achieve this frequency. The value in Profidrive to allow full power is 16384, so setting 20% of this value (3276,8 round to 3277) will give the expected result. This value shall be set in NSOLL_A (D2001).

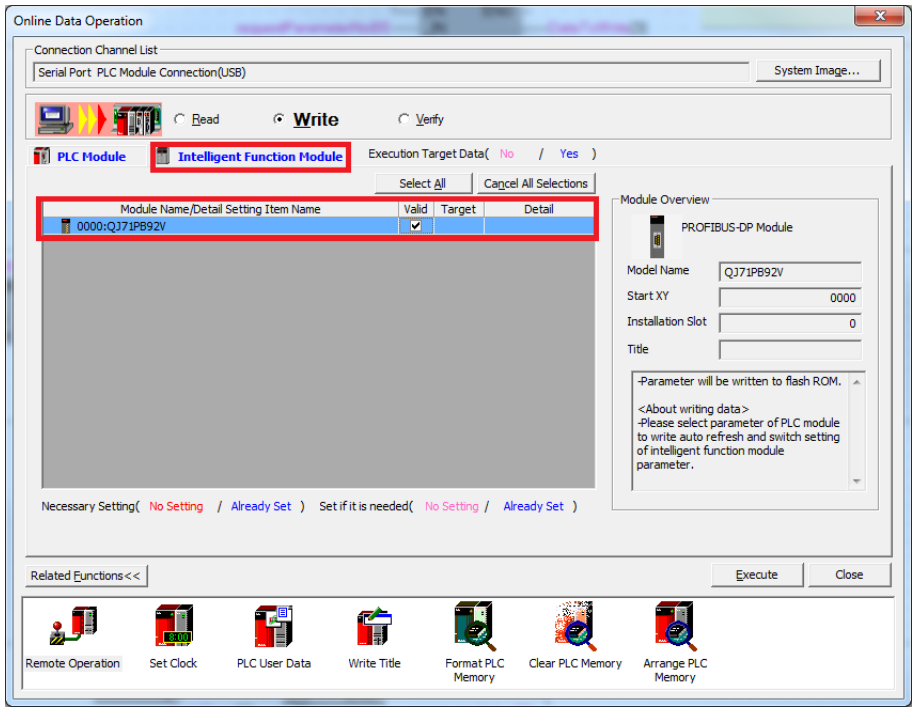
Add the following ladder block to enable rotation command after initialization, and giving rotation command (M1).



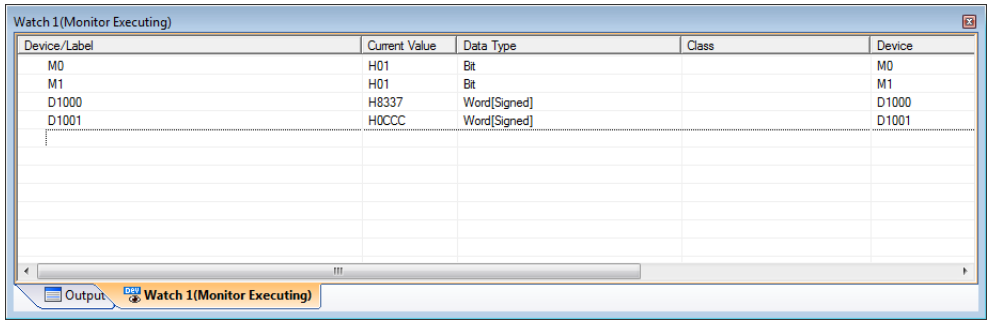
⑯ Finally add the code to stop the drive, when M1 is reset. Check whether the first 3 bits of ZSW1 are ON. This condition means that the drive is in operation mode. If this condition is met, and M1 is not ON, reset the set point speed to zero, and set STW1 as 6.



⑰ Compile, and write the program and parameters to the PLC. Be sure to select the intelligent function module, like on the screenshot below.



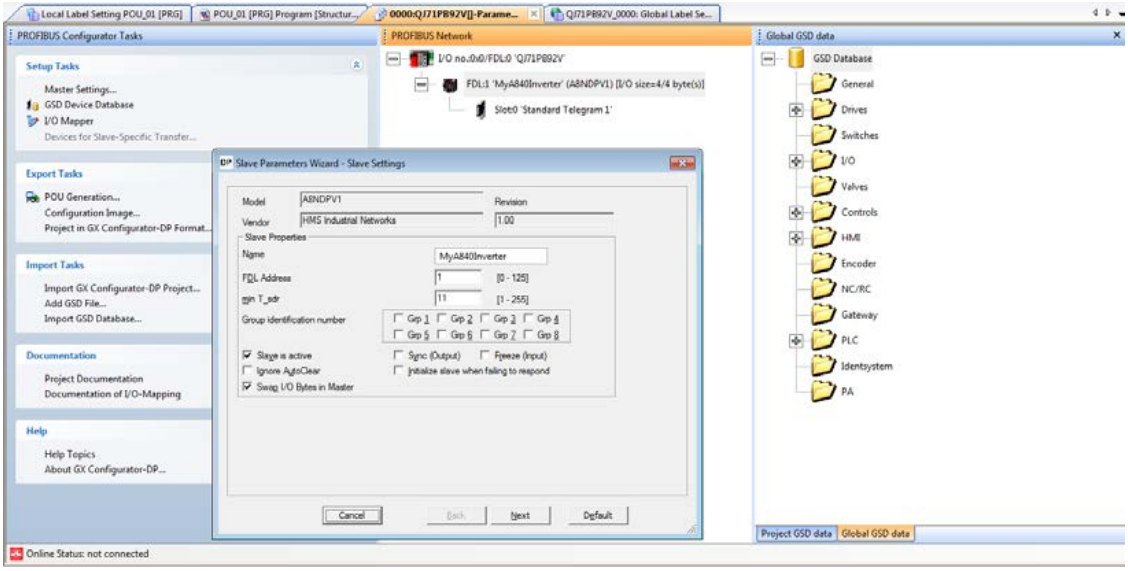
After resetting the PLC, and powering up the inverter set bits M0 and M1 to get the inverter up and running.



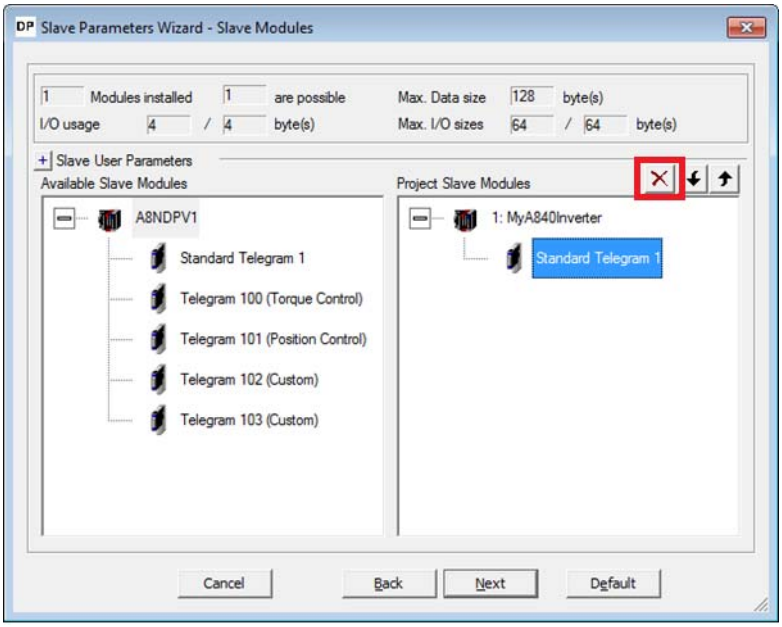
3.8 GX Works (Q-CPU) Simple Ladder Telegram 102 example

This chapter describes running the inverter through the A8NDPV1 Profibus option card using GX Works 2 in Simple Ladder mode with a QCPU and QJ71PB92V Profibus Master. First perform the setup using telegram 1.

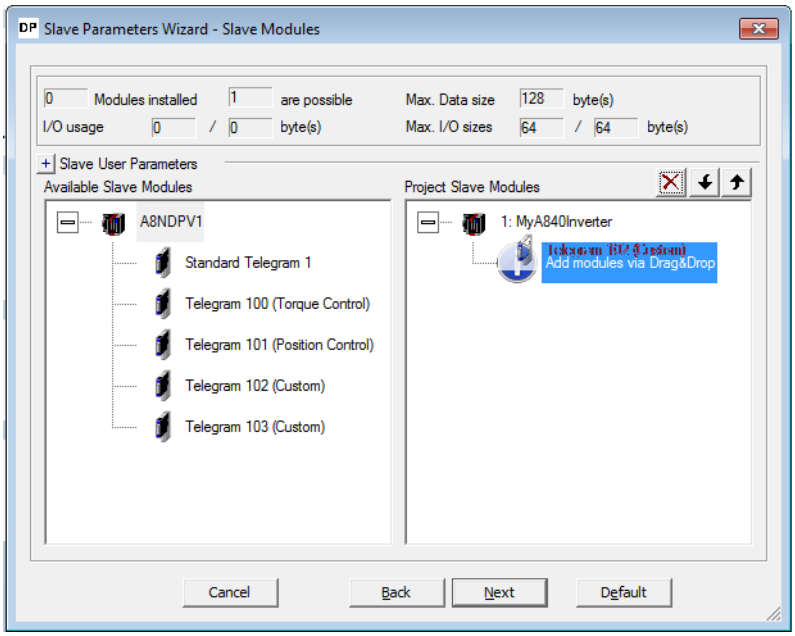
① Enter the Profibus Configurator and double click on the inverter to bring up the Slave Parameters Wizard. Click Next to proceed.



② Select “Standard Telegram 1” in the Project Slave Modules section. Click the red ‘x’ above this section to clear this slot.



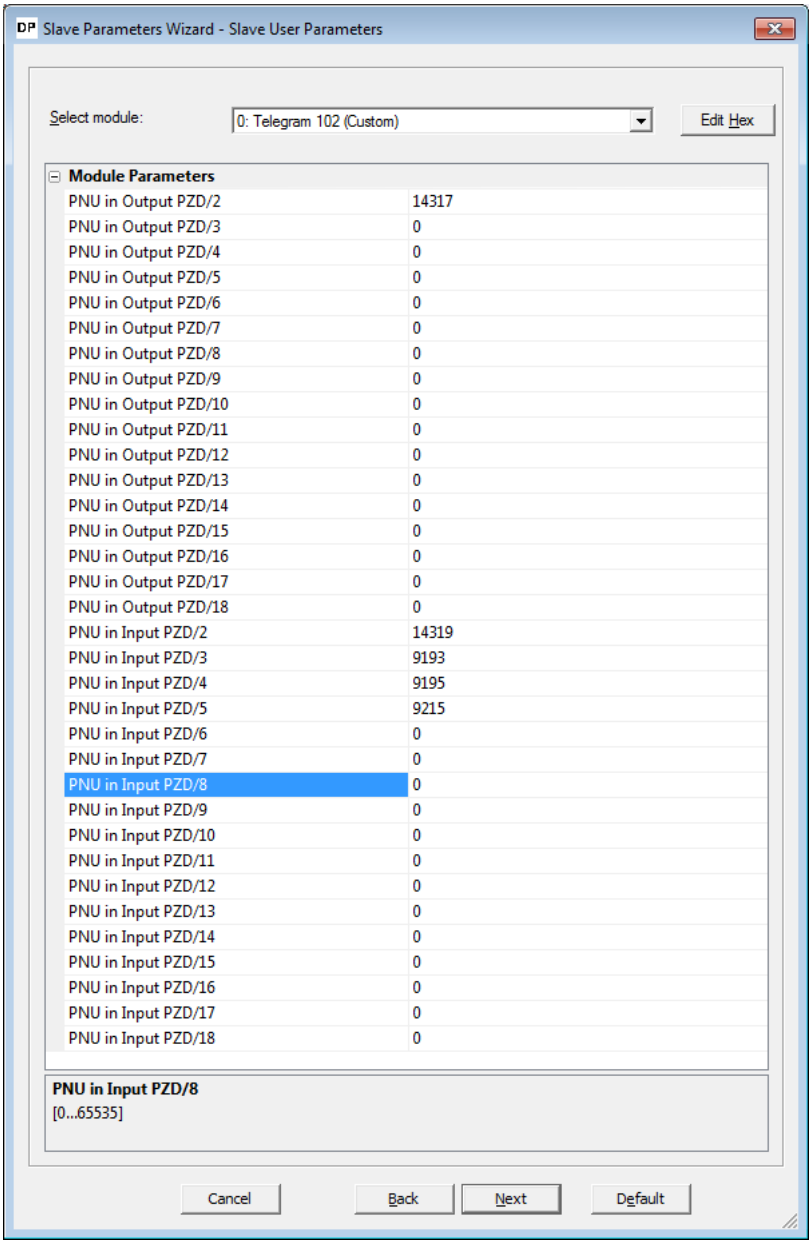
③ From the Available Slave Modules, drag “Telegram 102 (Custom)” to the Project Slave Modules section. Click Next to proceed.



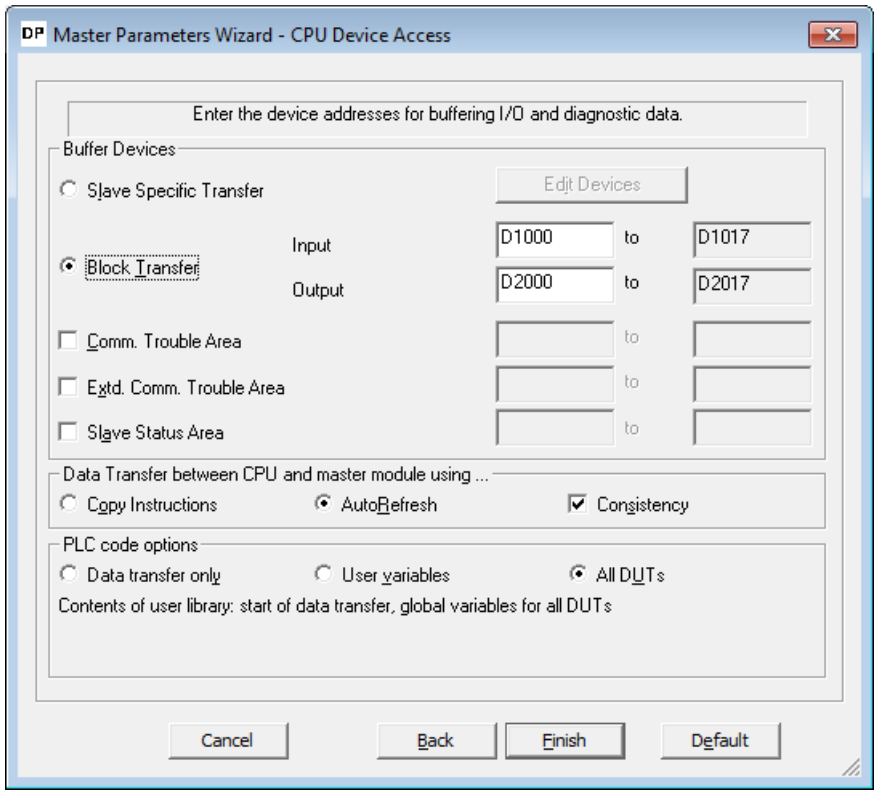
④ From the “select module” dropdown, select slot 0: Telegram 102 (Custom). It is now possible to setup the input and outputs of the telegram. There are 18 inputs and outputs used in this telegram, one of each is reserved for control/status word (leaving 17 inputs/outputs free for custom setup).

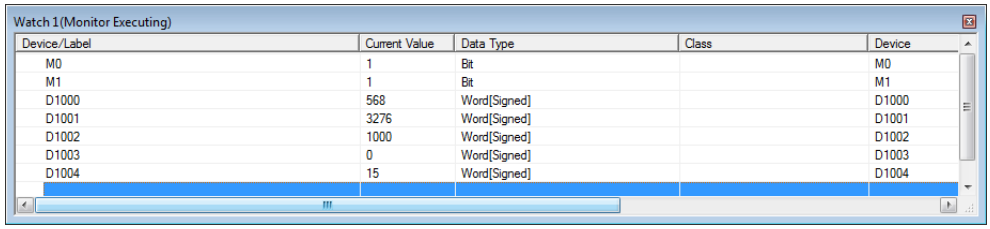
⑤ The setup of the telegram is done by entering signal/parameter numbers into the right column. All monitor data can be viewed as input, and some of the profile parameters as input/output. Please keep in mind, that not all parameters can be accessed via cyclic communication. The PROFIdrive parameter numbers (PNUs) available for use are listed in the Data Exchange subchapters 7.1to 7.4. Below is an example setup:

Slot	Description	PNU
Output PZD/2	Speed set point	14317
Input PZD/2	Actual speed set point	14319
Input PZD/3	Output frequency	9193
Input PZD/4	Output voltage	9195
Input PZD/5	Actual operation time	9215


Slave Parameters Wizard - Slave User Parameters
Select module:
0: Telegram 102 (Custom)
Edit Hex
Module Parameters
PNU in Output PZD/2 14317
PNU in Output PZD/3 0
PNU in Output PZD/4 0
PNU in Output PZD/5 0
PNU in Output PZD/6 0
PNU in Output PZD/7 0
PNU in Output PZD/8 0
PNU in Output PZD/9 0
PNU in Output PZD/10 0
PNU in Output PZD/11 0
PNU in Output PZD/12 0
PNU in Output PZD/13 0
PNU in Output PZD/14 0
PNU in Output PZD/15 0
PNU in Output PZD/16 0
PNU in Output PZD/17 0
PNU in Output PZD/18 0
PNU in Input PZD/2 14319
PNU in Input PZD/3 9193
PNU in Input PZD/4 9195
PNU in Input PZD/5 9215
PNU in Input PZD/6 0
PNU in Input PZD/7 0
PNU in Input PZD/8 0
PNU in Input PZD/9 0
PNU in Input PZD/10 0
PNU in Input PZD/11 0
PNU in Input PZD/12 0
PNU in Input PZD/13 0
PNU in Input PZD/14 0
PNU in Input PZD/15 0
PNU in Input PZD/16 0
PNU in Input PZD/17 0
PNU in Input PZD/18 0
PNU in Input PZD/8
[0...65535]
Cancel
Back
Next
Default

⑥ Click *Next*, and *Finish* to proceed. Save the project, and write it to the PLC. The additional monitor data can be viewed in the appropriate buffer device. To view the devices that are occupied by Profibus, double click on the Profibus master to bring up the *Master Parameters Wizard* and click *Next*. The buffer devices section displays which device addresses are occupied.





3.9 GX Works (Q-CPU) Simple Ladder Acyclic communication example

This chapter describes using acyclic communication through the A8NDPV1 Profibus option card using GX Works 2 in Simple Ladder mode with a QCPU and QJ71PB92V Profibus Master. It contains examples of reading and writing individual parameters. First perform the setup using telegram 1. For additional information refer to the QJ71PB92V manual.

The process of acyclic communication parameter read consists of a write request, and a response read. For details please refer to the Acyclic Data Exchange subchapter 7.7.

3.9.1 Reading a parameter (Sequence 1)

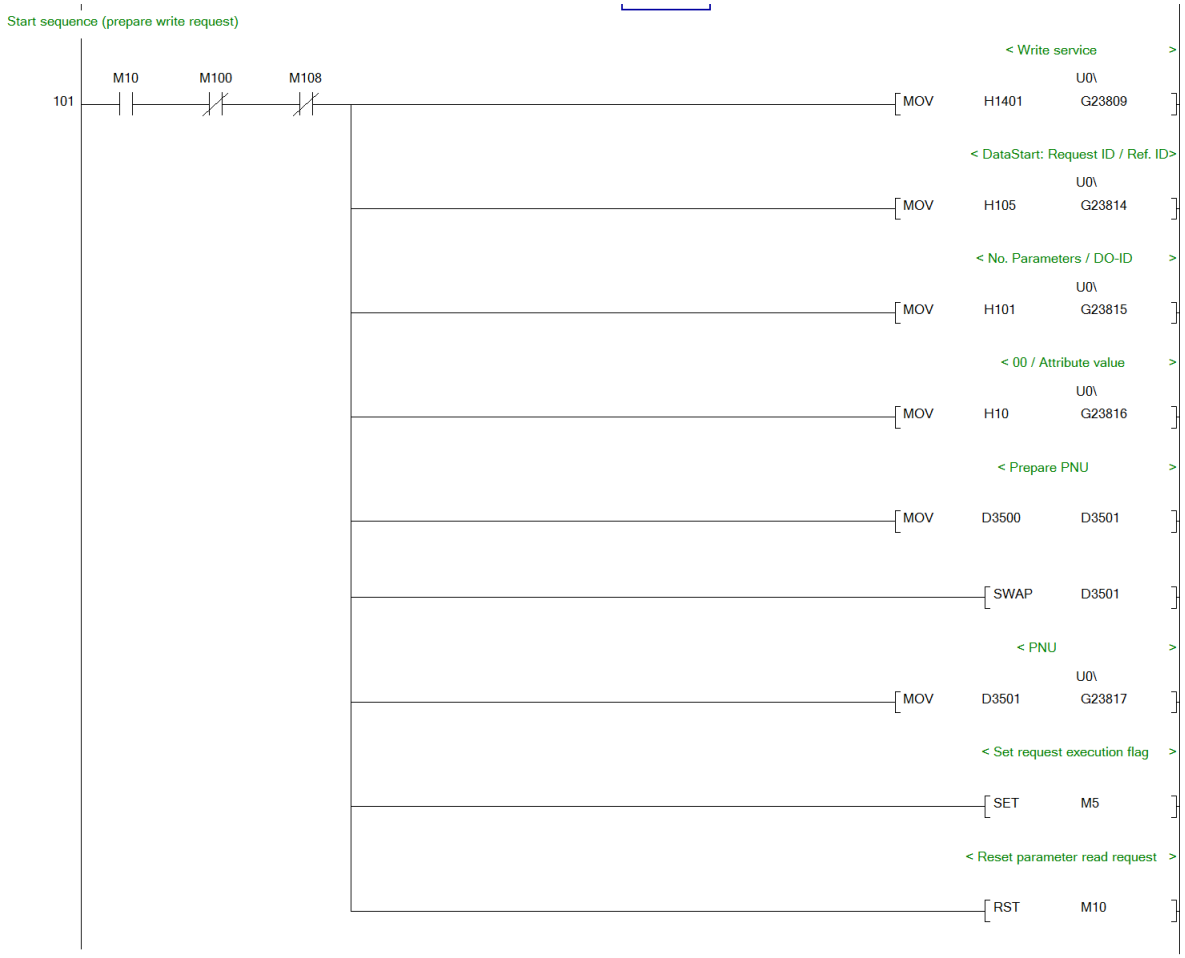
Creating the request

The first part of the program should read the acceptance/completion status of the acyclic request. This can be done with the following code:

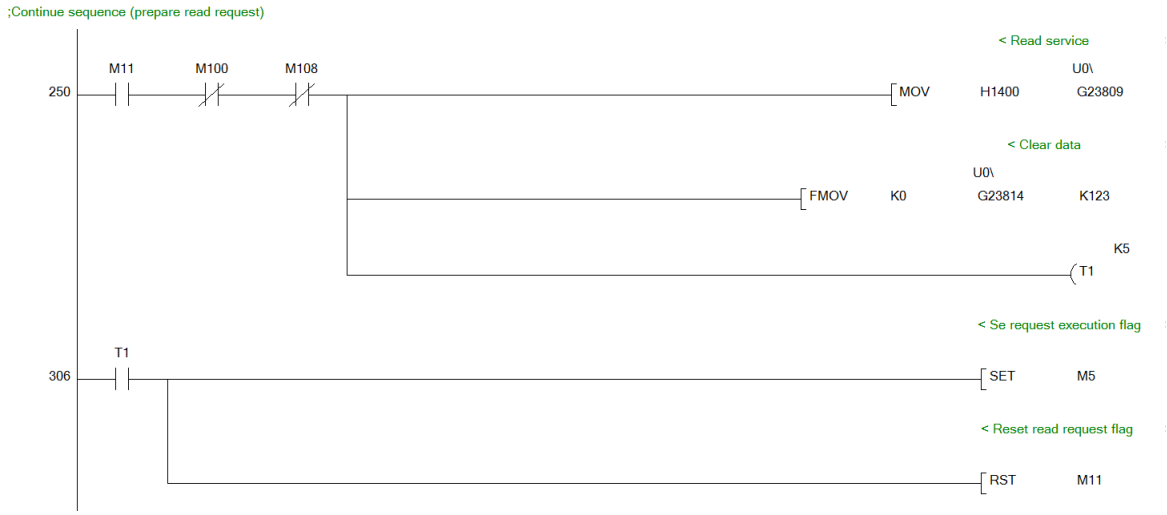
Prepare the content for the parameter read request. The data request has the following format:

Byte no.	Description	Value
0	Request ID	0x01
1	Request reference	0x01-0xFF
2	No. of parameters	0x01
3	DO-ID	0x01
4	No. of elements	0x00
5	Attribute value	0x10 (value attribute)
6-7	Parameter number	Byte swapped PNU

Request reference is any valid number, this value is mirrored back in the response and can be used to distinguish multiple requests. The parameter number needs to be byte swapped (the SWAP instruction can be used to swap the lower and higher byte of a word). Below is an example on how to prepare the request data:

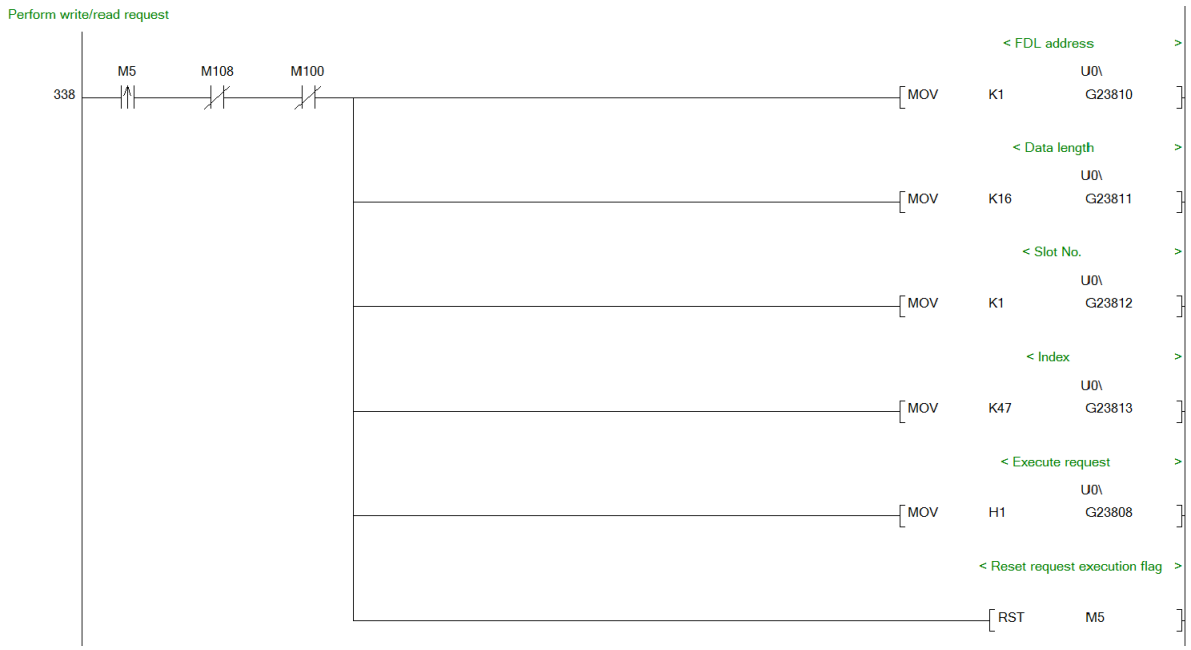


The code above (after executing the write to the Profibus slave) will return information regarding the success of making the request, not the actually requested data. Before making the actual request, prepare the data for the read request:



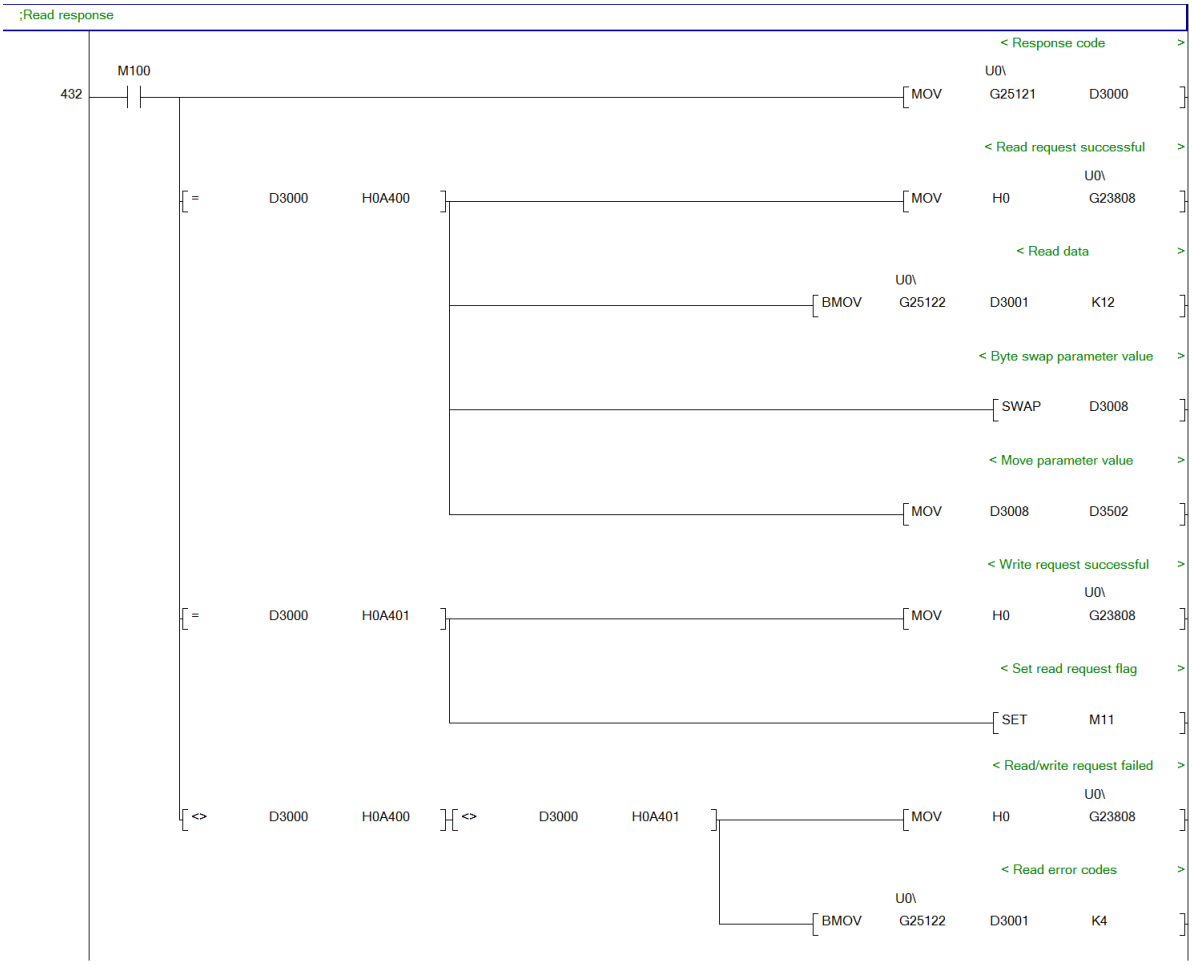
In the program there is a delay of 500ms between receiving confirmation of the write request, and executing the read request. Due to the nature of Profidrive acyclic communication, the response may not be immediately available after execution of the request. Waiting a small amount of time significantly increases the chance of receiving a response on the first try. Another solution is to keep executing read requests, until a proper read request is received.

Finally include the code that executes the prepared request:



Receiving the response

The requests are completed after buffer memory bit G25120.0 turns ON. Successful completion is achieved if buffer memory G25121 is equal to 0xA400 for read request and 0xA401 for write request.



After the write request is executed successfully, expect G25121 (D3000) to equal 0xA401. Set M11 to continue the sequence and issue the read request. This time a value of 0xA400 means a successful execution. The received data starts from buffer memory address G25127 (D3006). In buffer memory G25121-25126 (D3000-D3005) data regarding the acyclic request is held, as described in the QJ71PB92V manual. The format of a positive response is as follows:

Byte no.	Description	Value
0	Request ID	0x01
1	Request reference	0x01-0xFF (same as in request)
2	No. of parameters	0x01
3	DO-ID	0x01
4	No. of values	0x01
5	Format	See data format type table
6-7	Parameter value	Byte swapped parameter value

In case of a negative response, the format is as follows:

Byte no.	Description	Value
0	Request ID	0x81
1	Request reference	0x01-0xFF (same as in request)
2	No. of parameters	0x01
3	DO-ID	0x01
4	No. of values	0x01
5	Format	0x44 (error)
6-7	Error value	Byte swapped error value (see error table)

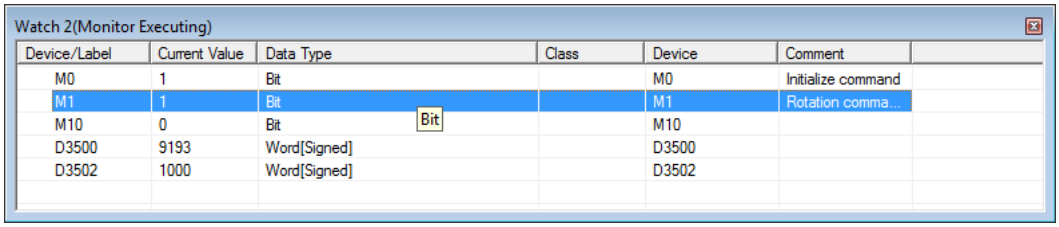
The parameter value (or error value) can be therefore read from D3008. This value is byte swapped and saved to address D3502.

Executing example requests

In the code above the following devices are used:

Device	Description
M5	Perform acyclic request
M10	Trigger sequence / Prepare write request
M11	Continue sequence / Prepare read request
M100 – M108	Acceptance/completion status
D3000	Acyclic response code
D3001 – D3013	PROFIdrive response data (byte swapped)
D3500	PNU to read
D3501	Byte-swapped PNU no.
D3502	PNU value

After compiling the program, and writing it to the PLC, it is possible to execute parameter read requests. First, get the inverter running by setting bits M0 and M1. Now read the actual frequency by writing PNU 9193 to *D3500* and setting bit *M10*. The result should be a value of 1000 in *3502* which corresponds to 10Hz. The PROFIdrive parameter numbers (PNUs) available for use are listed in the Data Exchange subchapters 7.1 to 7.4.



3.9.2 Changing parameters (Sequence 2)

Creating the request

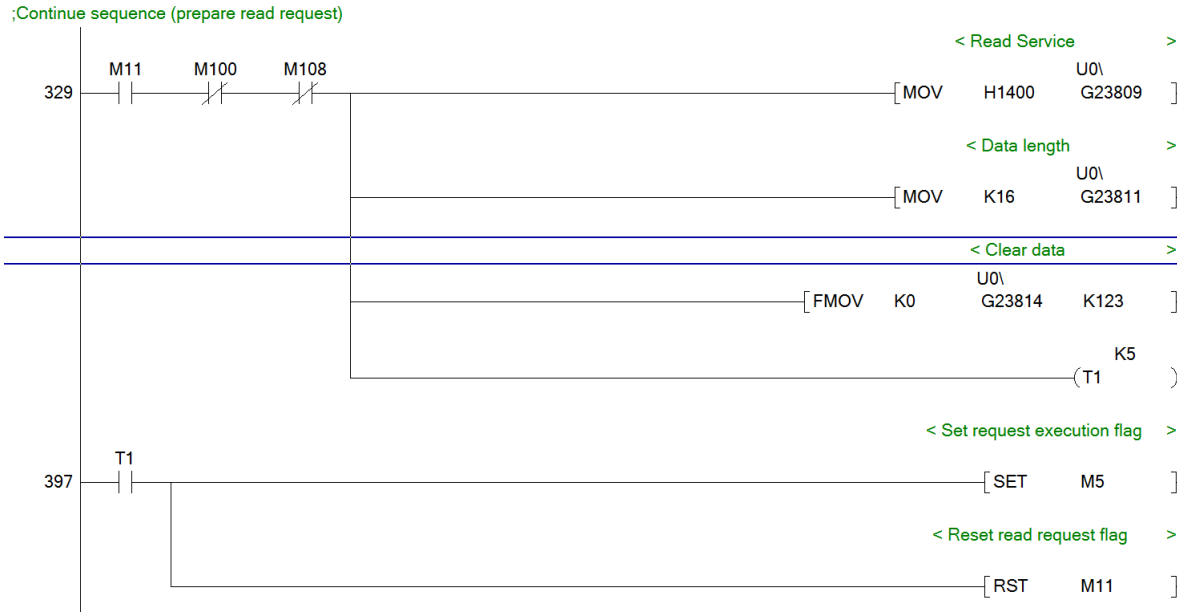
The idea of executing a change (write) of a PNU is the same as reading one. The only difference is the change of the data request / response.

The format of the write request is as follows:

Byte no.	Description	Value
0	Request ID	0x02
1	Request reference	0x01-0xFF
2	No. of parameters	0x01
3	DO-ID	0x01
4	No. of elements	0x00
5	Attribute value	0x10 (value attribute)
6-7	Parameter number	Byte swapped PNU
8-9	Subindex (irrelevant)	0x00
10	No. of values	0x01
11	Format	0x06 (UINTEGER16 for all inverter parameters) See data format type table
12-13	Set value	Byte swapped set value

Modify the content of the write request.

Modify the read request:



Modify the execution of the request:

Reading the response

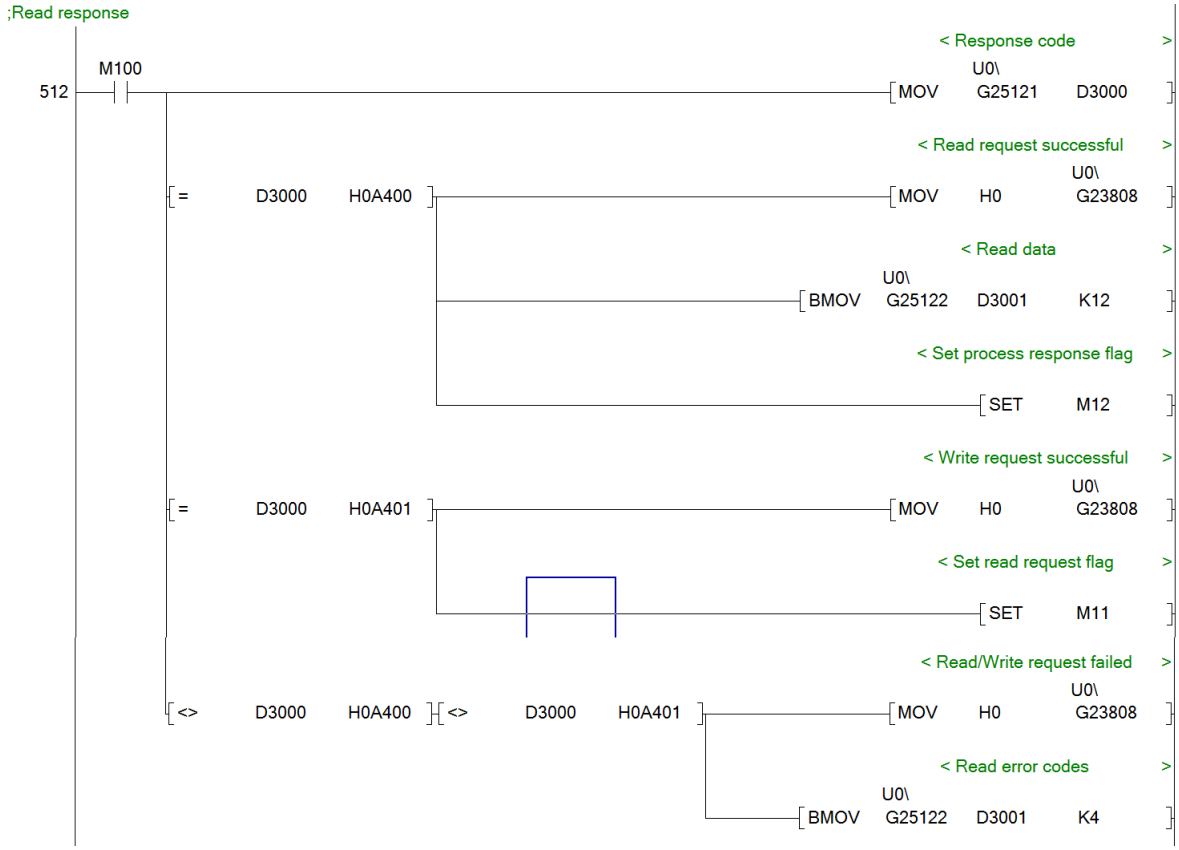
A positive response consists of just 4 bytes of data in the following format:

Byte no.	Description	Value
0	Request ID	0x02
1	Request reference	0x01-0xFF (same as in request)
2	No. of parameters	0x01
3	DO-ID	0x01

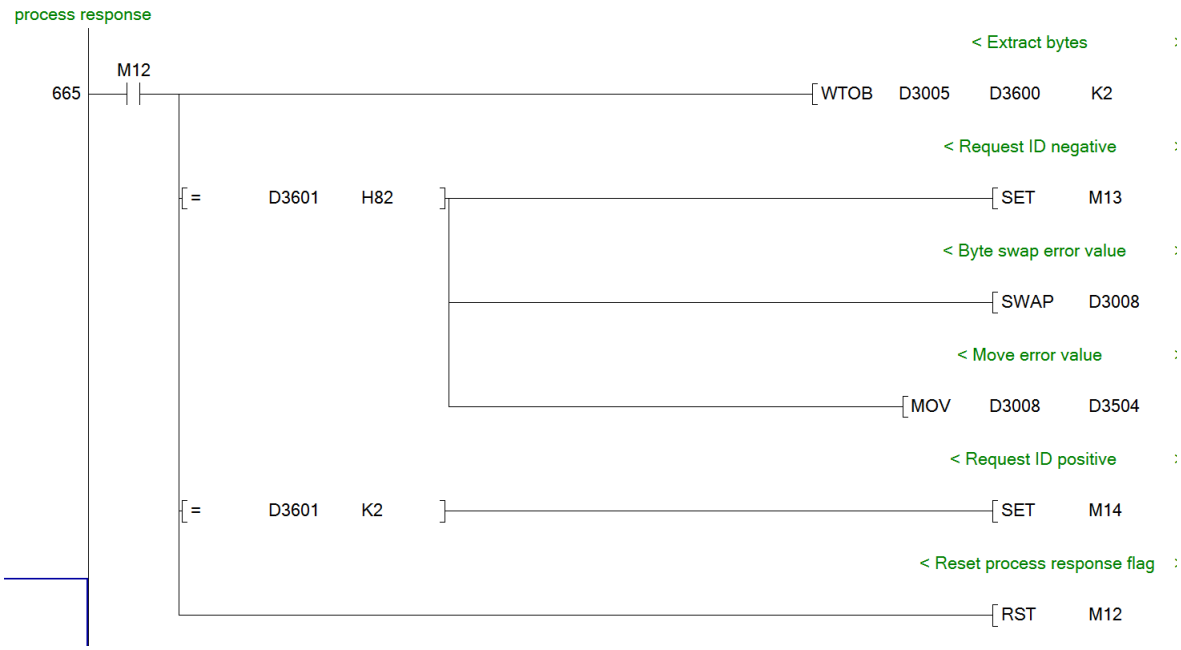
In case of a negative response, the format is as follows:

Byte no.	Description	Value
0	Request ID	0x82
1	Request reference	0x01-0xFF (same as in request)
2	No. of parameters	0x01
3	DO-ID	0x01
4	No. of values	0x01
5	Format	0x44 (error)
6-7	Error value	Byte swapped error value (see error table)

The requests are completed after buffer memory bit G25120.0 turns ON. Successful completion is achieved if buffer memory G25121 is equal to 0xA400 for read request and 0xA401 for write request. If the write request is executed successfully, the program will continue with the read request, and if this is also successful bit M12 will be set ON to process the response.



If a successful parameter write request was performed (request ID = 0x02), bit M14 shall turn on. In case of a failure (request ID = 0x82) bit M13 will be ON, and the error code will be stored in D3504.

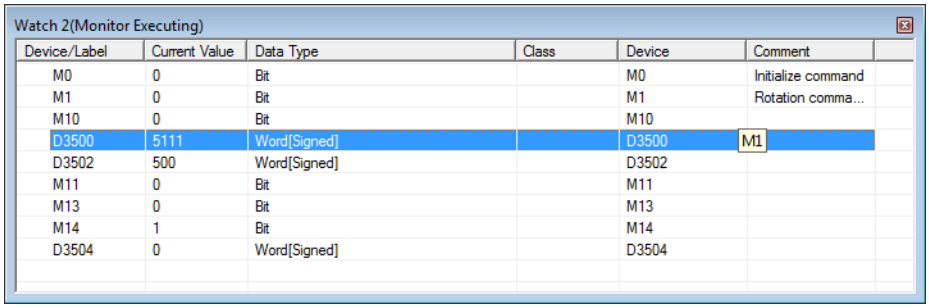


Executing example requests

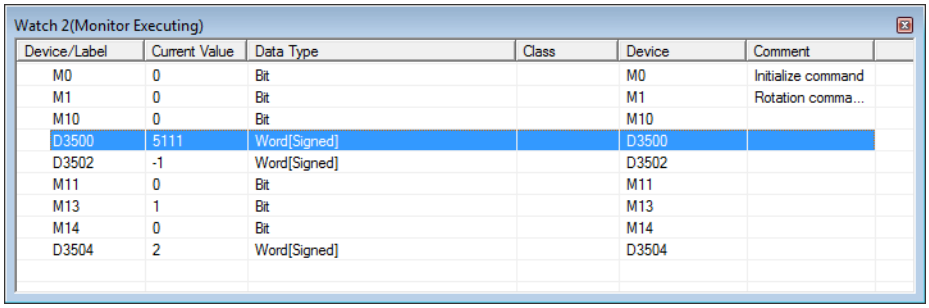
In this program the following devices are used:

Device	Description
M5	Perform acyclic request
M10	Trigger sequence / Prepare write request
M11	Continue sequence / Prepare read request
M13	Parameter write failed
M14	Parameter write success
M100 – M108	Acceptance/completion status
D3000	Acyclic response code
D3001 – D3013	PROFIdrive response data
D3500	PNU to change
D3501	Byte-swapped PNU no.
D3502	PNU value to set
D3503	Byte-swapped PNU value to set
D3504	PROFIdrive error value

To perform a parameter write, set the PNU of the parameter to change in *D3500*. In order to change parameter 15 (Jog frequency), add the offset 5096 and write the result to *D3500*. Set *D3502* to the requested value. The value that will be set, will be the output of multiplication of the set value, and the minimum setting increments, as according to the parameter list in the inverter manual. For parameter 15, the minimum setting increment is 0.01Hz. If a jog frequency of 5Hz is requested, set *D3502* to 500. Finally execute the request by setting bit *M10* to ON. A successful write will result in setting of the *M14* bit.



Test receiving a negative response, by trying to write an out of range value. After trying to write 0xFFFF as the value of parameter 15, a negative response with error code 0x02 (LOW_OR_HIGH_LIMIT_EXCEEDED) will be received, which is the expected behavior.

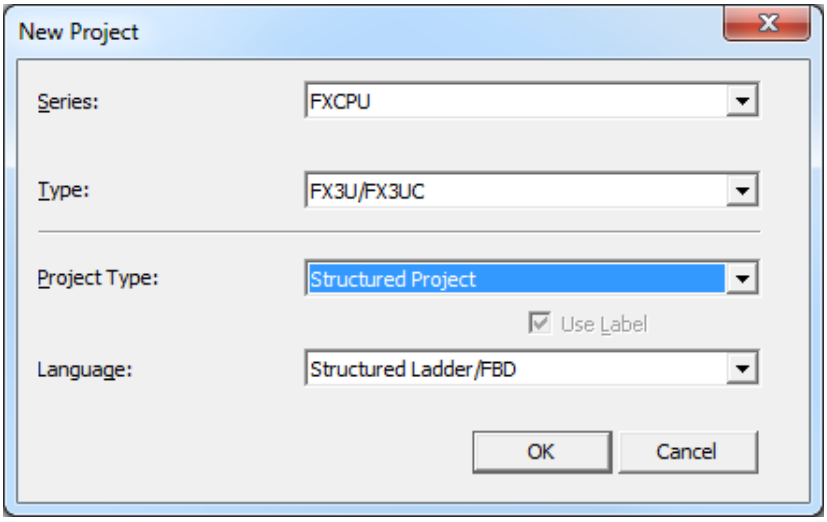


3.10 GX Works (FX-CPU) Telegram 1 example

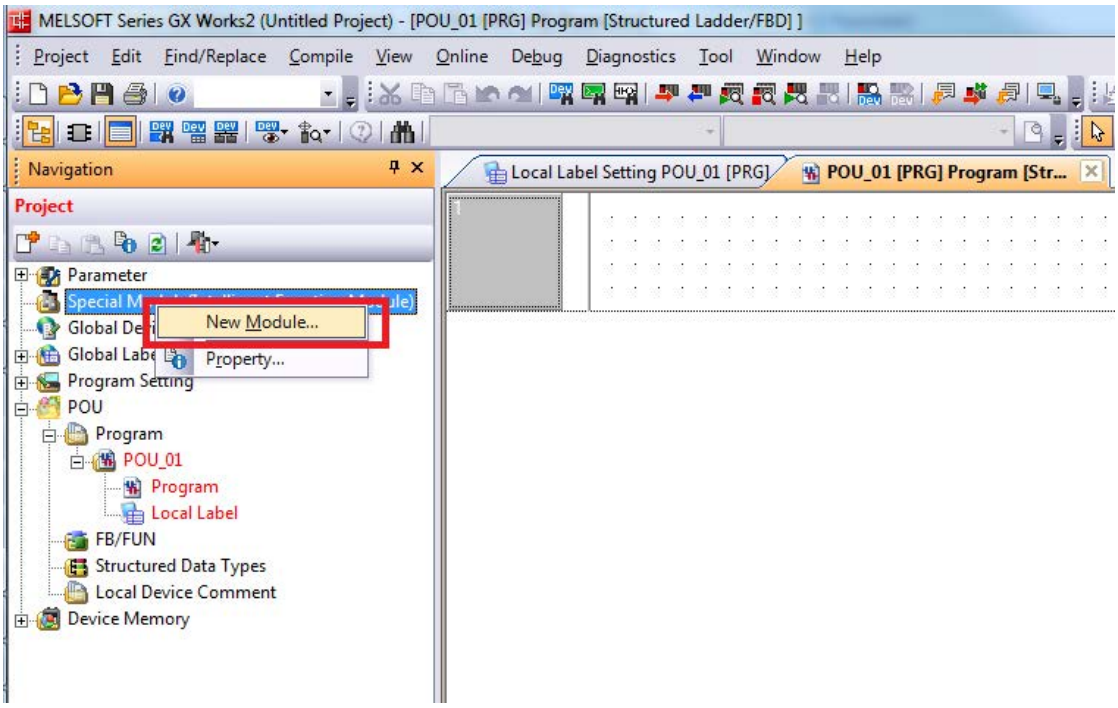
This chapter describes running the inverter through the A8NDPV1 Profibus option card using GX Works 2 (version 1.501X or newer) with a FXCPU and FX3U-64DP-M Profibus Master. GX Configurator DP 7.10L or newer is required to perform the setup.

① Start with a new project.
Select:

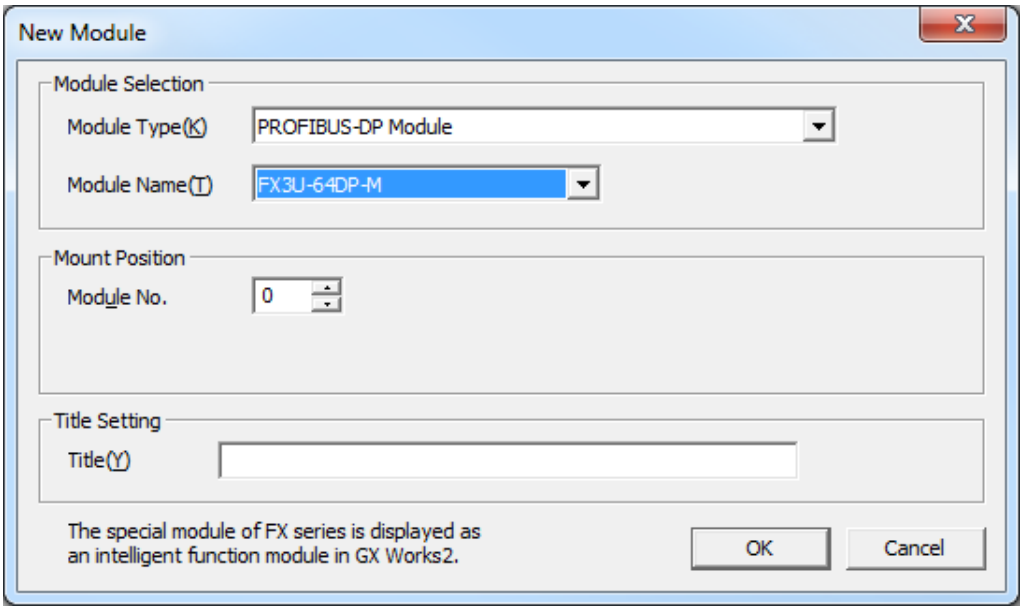
- Series: FXCPU;
- PLC type: Select the cpu;
- Project type: Structured project;
- Language: Structured Ladder/FBD.



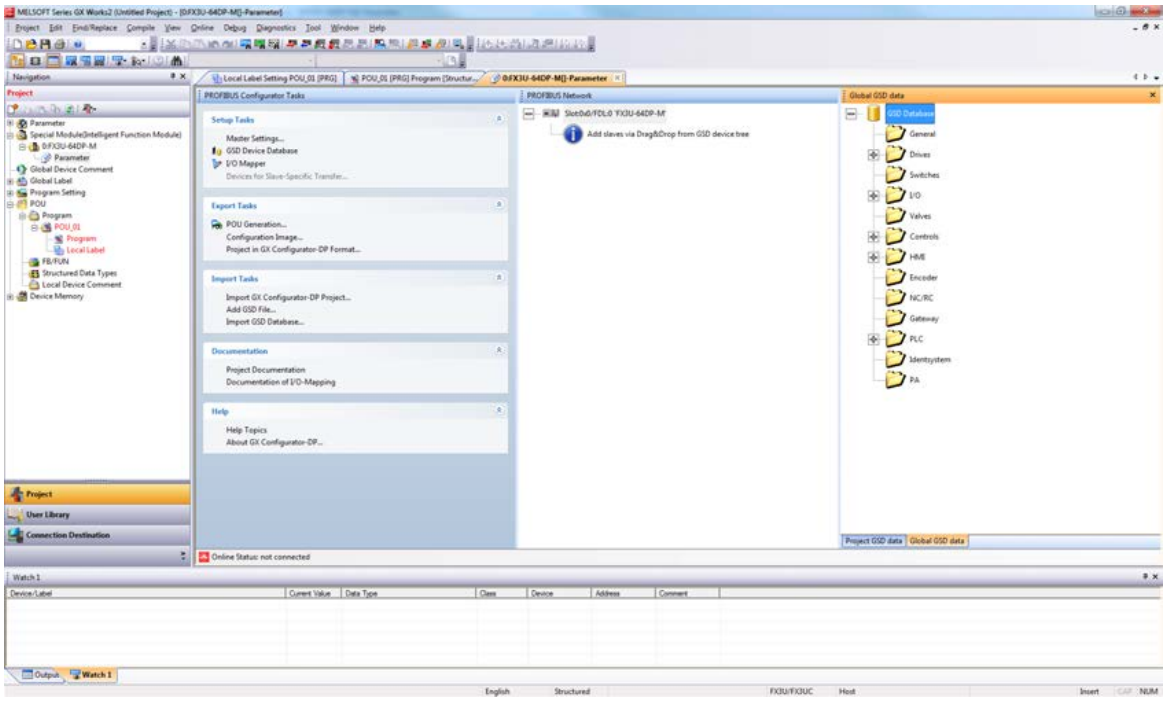
② Add a new Special Module (Intelligent Function Module) from the menu on the left (right click and select new):



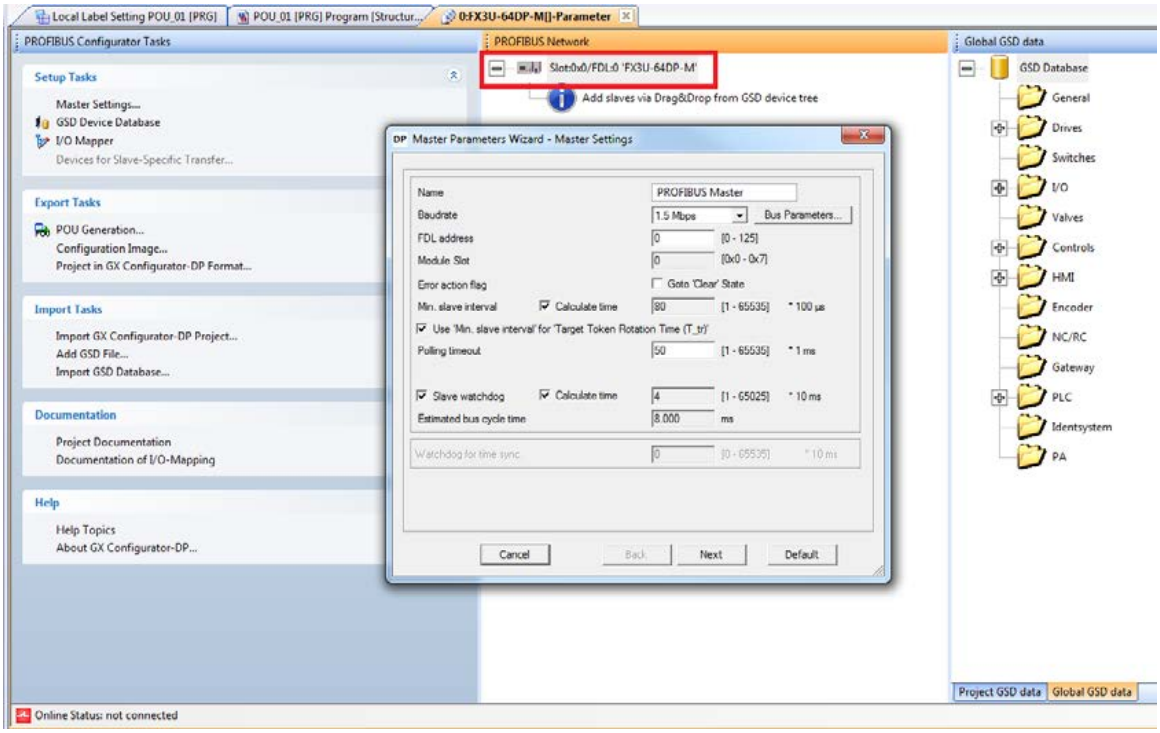
③ In the new window, click Module Type and select “PROFIBUS-DP Module”. Module Name should be set to FX3U-64DP-M. Save this setting by clicking OK.



④ The Profibus-DP module should be inserted into the Intelligent Function Module tree. Expand it, and double click on Parameter. This should open the Profibus Configurator.

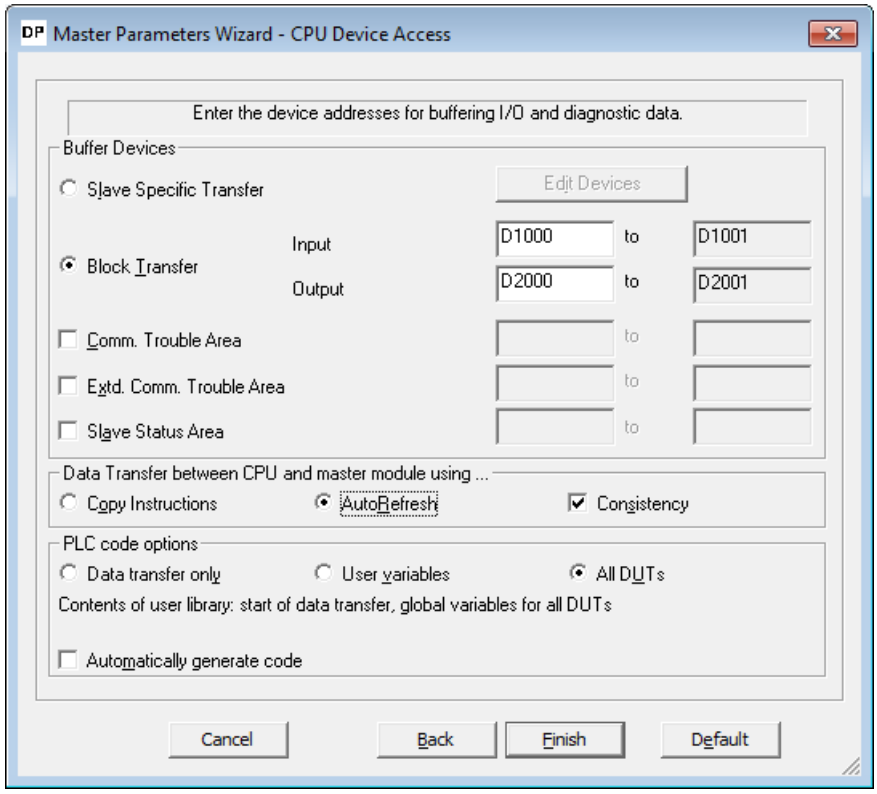


⑤ First setup the Profibus Master. Double click on the master module (in the screenshot visible as “Slot:0x0/FDL:0 ‘FX3U-64DP-M”. This will bring up the “Master Parameters Wizard”.

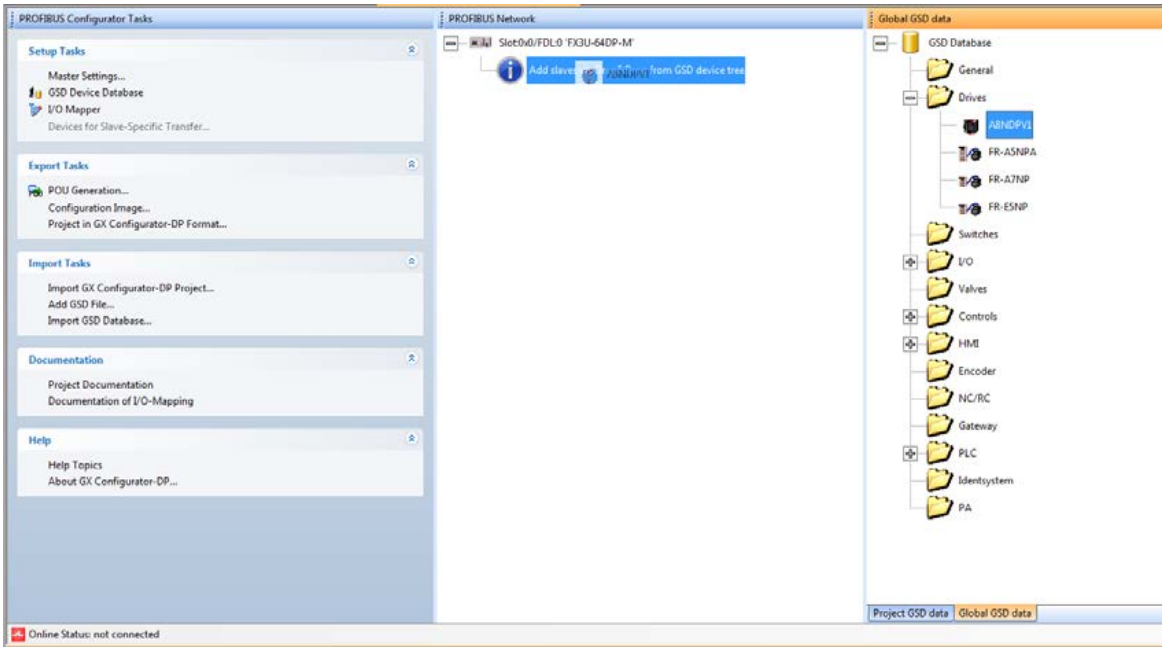


⑥ It is possible to change the Baudrate, FDL address and other settings as needed. For this tutorial these settings will not be changed. Click Next to proceed.

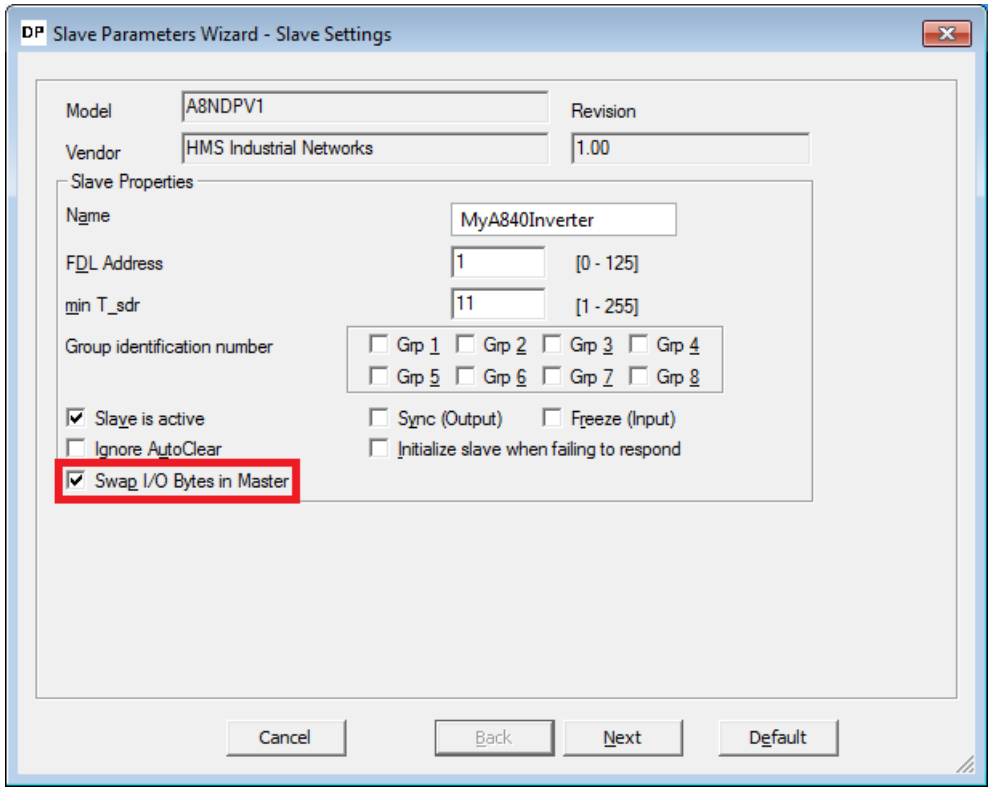
⑦ On this screen the buffer devices, and data transfer between Profibus Master and CPU can be changed. For this tutorial use AutoRefresh with Consistency. Click Finish to save settings and close this wizard.



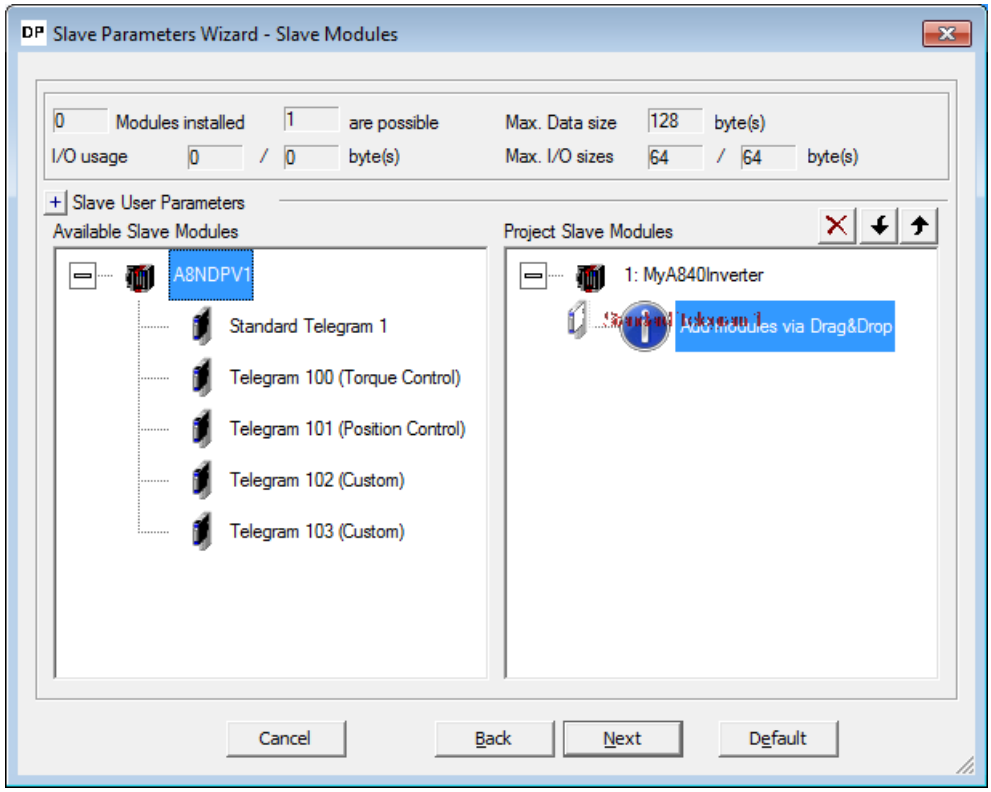
⑧ In the Global GSD data section, expand drives and search for A8NDPV1. If it is not available, right click and select Add GSD File. Search for the GSD File on the drive (included with this manual), afterwards the A8NDPV1 should appear under drives. Drag this option to the Profibus Master, like on the screenshot.



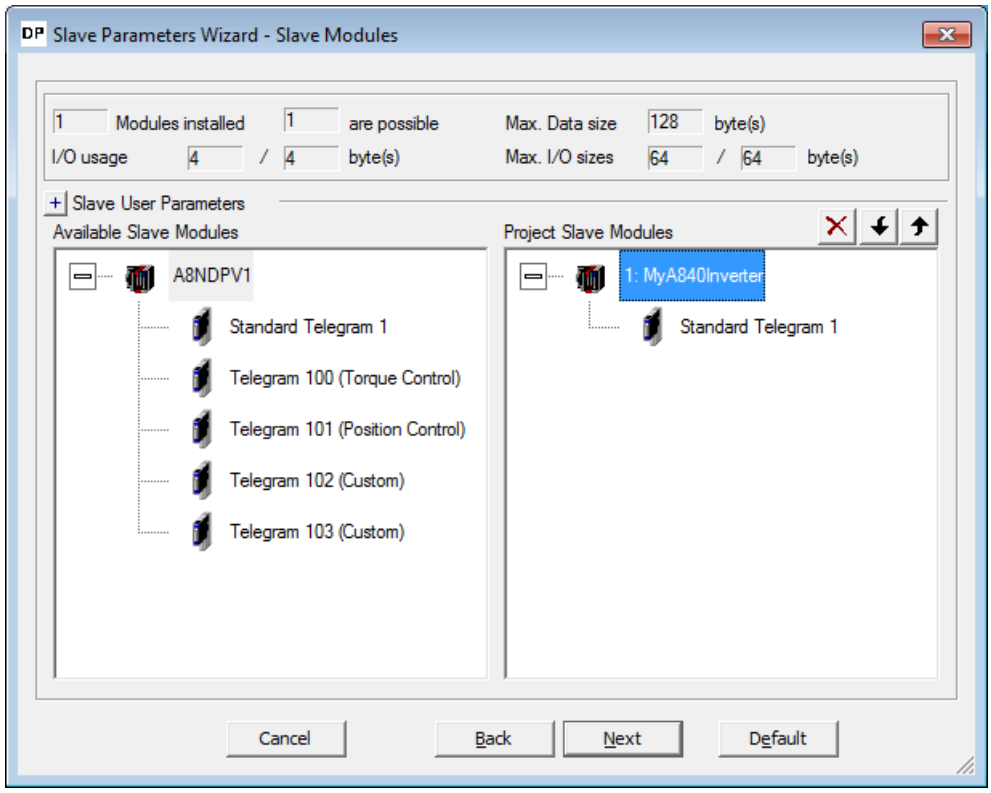
The “Slave Parameters Wizard” should be shown. Enter a name, select 1 for FDL Address, and make sure to select “Swap I/O Bytes in Master”. All other settings can remain as default. Click Next to proceed.



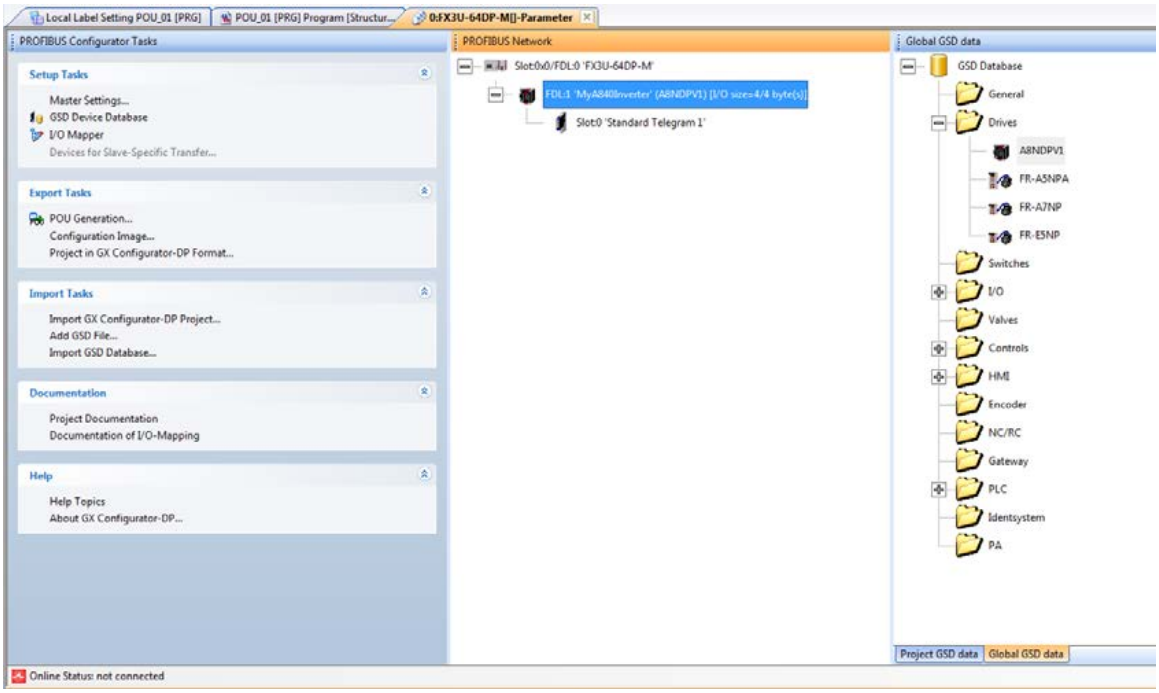
⑨ Select the communication protocol used to communicate with the option card. For this tutorial use “Standard Telegram 1”. Drag this option from the “Available Slave Modules” to “Project Slave Modules”. The A8NDPV1 supports only one module.



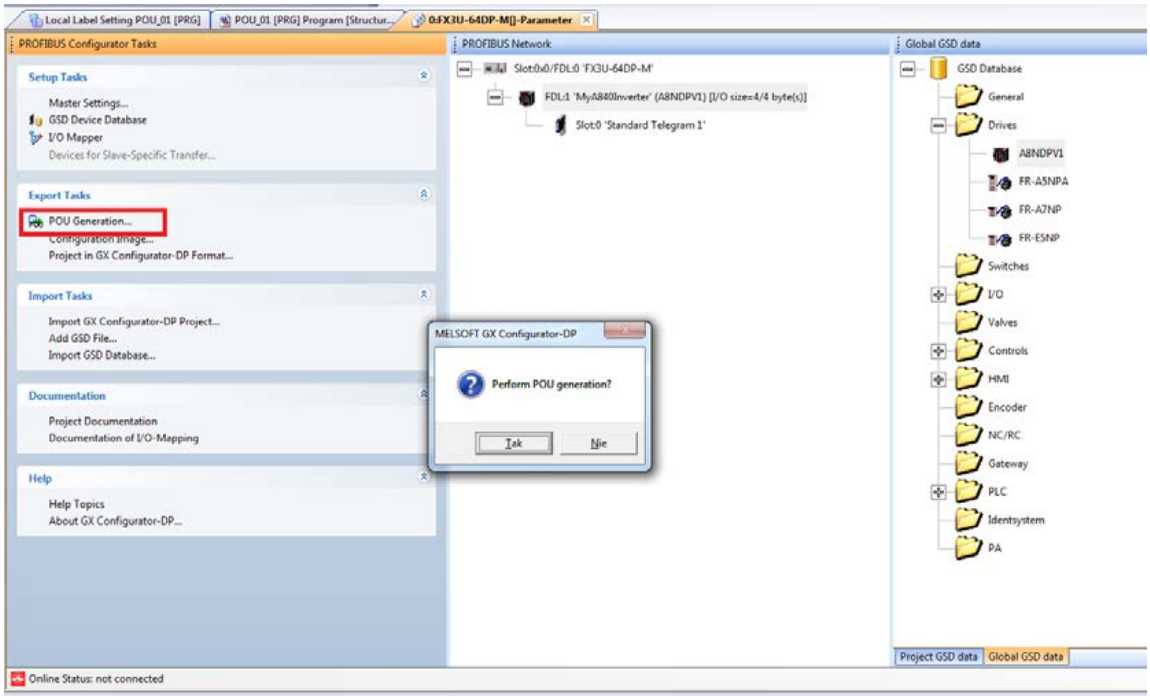
The result should be:



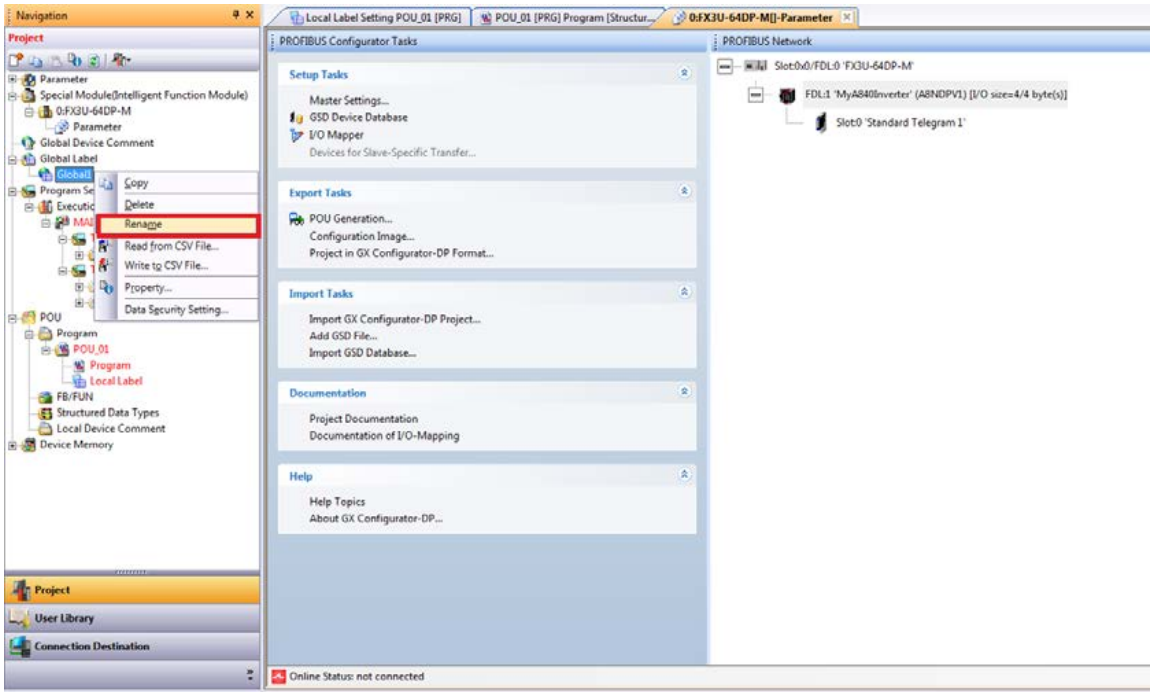
⑩ Click Next to proceed. No changes are required on this screen, after clicking Next proceed to the final screen and click Finish. The configured Profibus Slave should appear in the Profibus Network window, like below.



⑪ From the configurator's left menu select POU Generation. This will create the user libraries that can be used to control the inverter.

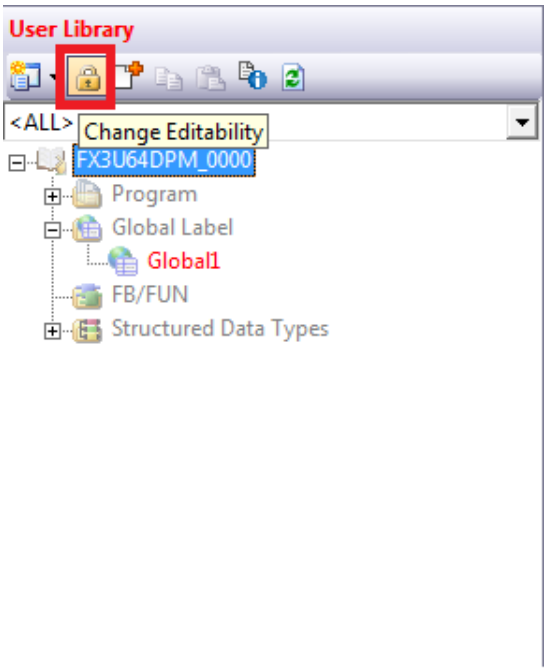


⑫ The POU Generation creates a library containing global variable definitions. The global label name conflicts with the project global label name setup by default in GX Works 2. In order to avoid conflict, change the name of the project global label from “Global1” to “Global”. If this is not done, an error will occur upon compilation.

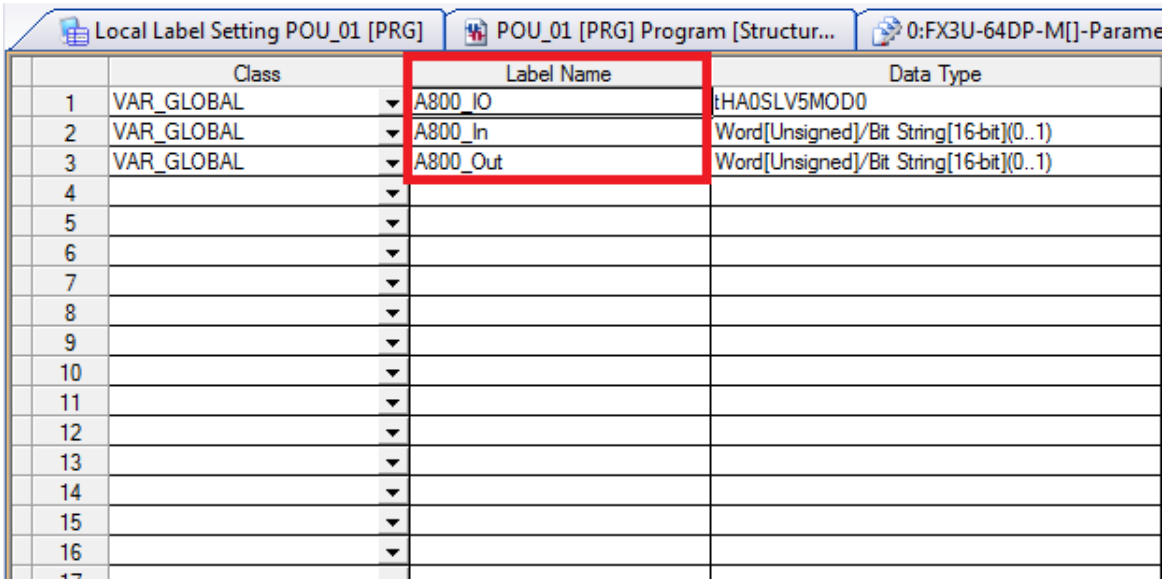


⑬ The Profibus controller is setup and ready to run after compilation and write to the PLC. Proceed to write the program to get the inverter running under a set frequency.

⑭ Go to the User Library section, select the library generated by Profibus configurator and click on the lock pad icon to unlock it.

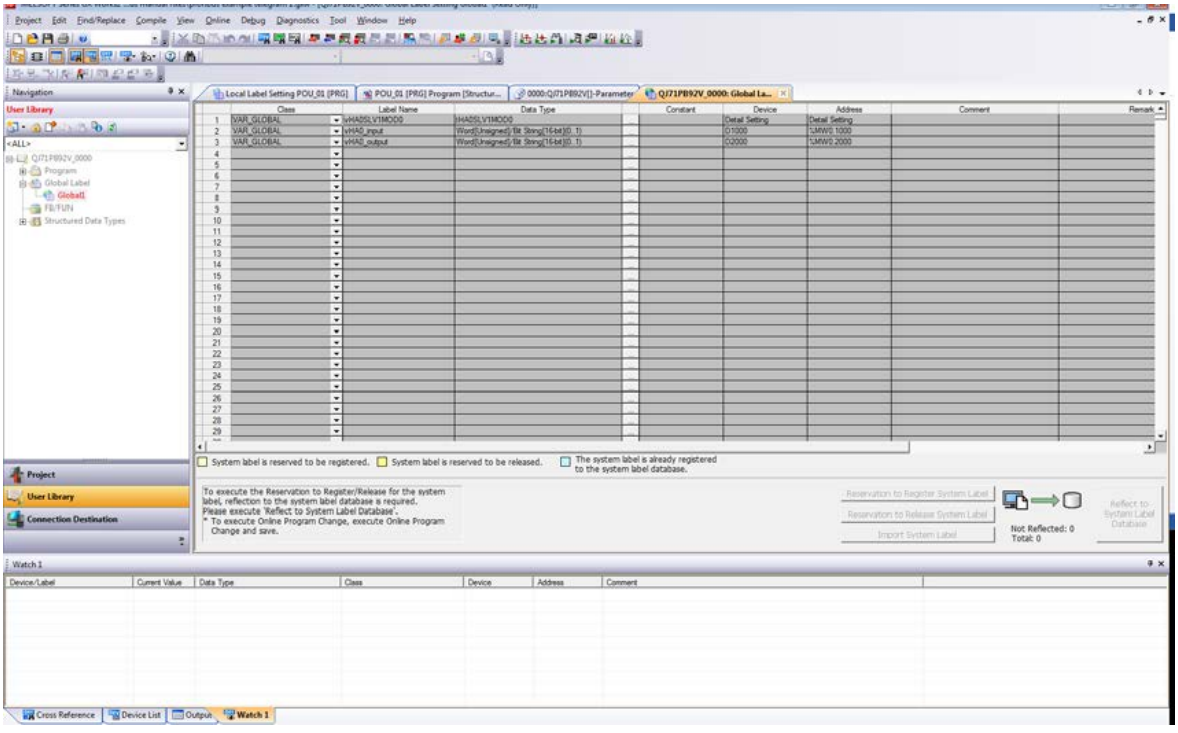


⑮ Enter the global label definition file and modify the label names to your liking. To allow easy following of this tutorial, please use the names like on the screenshot below:

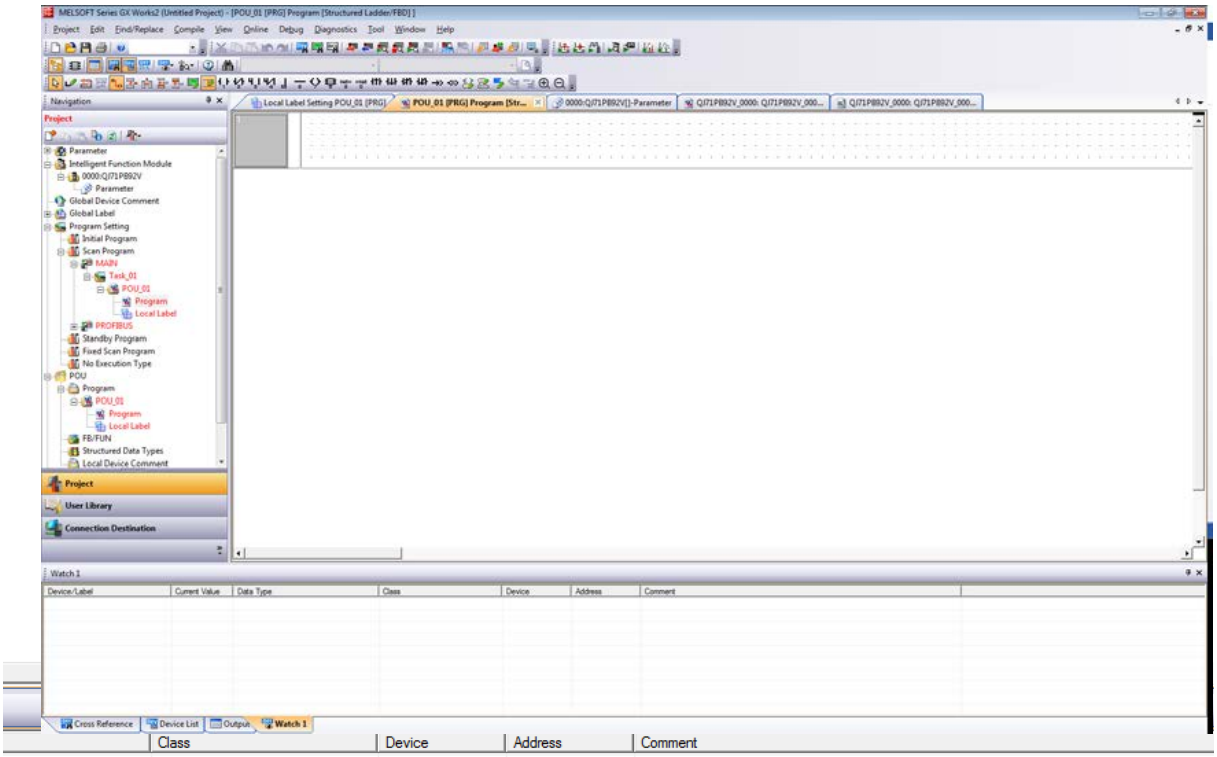


⑯ The data is defined as follows :

- For A800_In: [0] = ZSW1, [1] = NIST_A.
- For A800_Out: [0] = STW1, [1] = NSOLL_A.



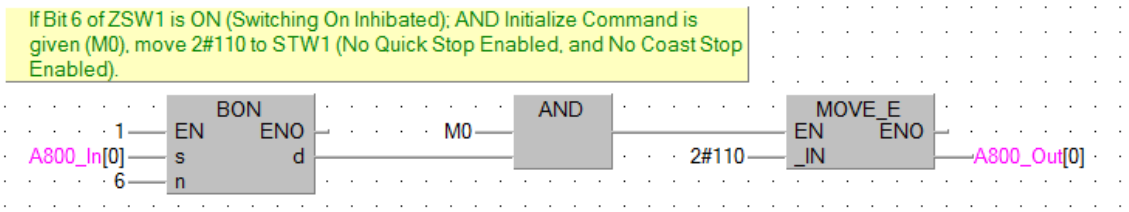
⑰ Return to the project section. From the MAIN Task, select the Program of POU_01. This Program should be blank, this is where the example user program will be written.



⑱ After powering up the inverter and Profibus master, connection will be established. Expect to see bit 6 (Switching on inhibated) set in ZSW1. For details refer to chapter 7.5 "General State Diagram"

Set both No Coast Stop and No Quick Stop to reset the Switching On Inhibited bit in ZSW1. To do this set bits 1 and 2 in STW1, that is global label vHA0_output[0].

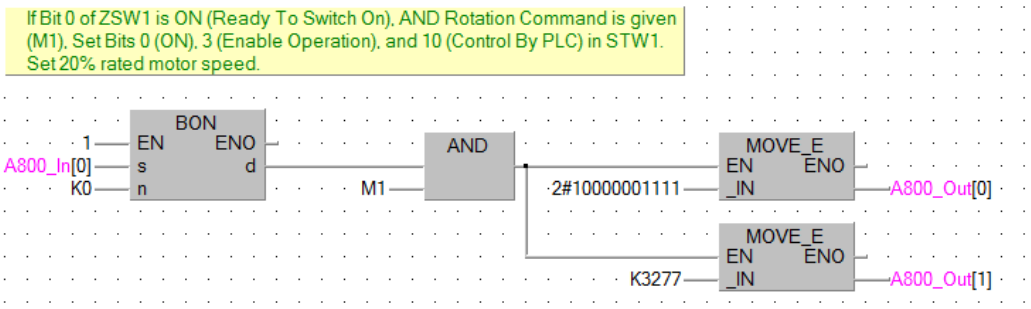
Input the following ladder block to allow the inverter to enter “Ready For Switching On” status, after connection is established, and initialization command is given (M0).



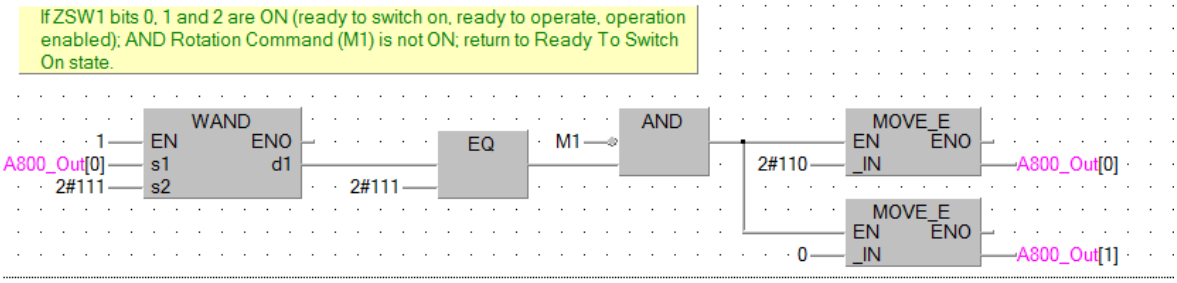
⑲ This initialization will result in setting bit 0 in ZSW1 (Ready To Switch On). It is now possible to switch on the inverter, and start operation. To do this, enable bits 0 (ON/Off), 3 (enable operation), and 10 (Control By PLC) of STW1; while leaving bits 1 and 2 enabled.

At this moment it is also possible to set the desired rotation frequency. Assuming the motor is rated for 50 Hz (inverter parameter 3) and a frequency of 10 Hz is requested, set 20% of the rated motor speed to achieve this frequency. The value in Profidrive to allow full power is 16384, so setting 20% of this value (3276,8 round to 3277) will give the expected result. This value shall be set in NSOLL_A.

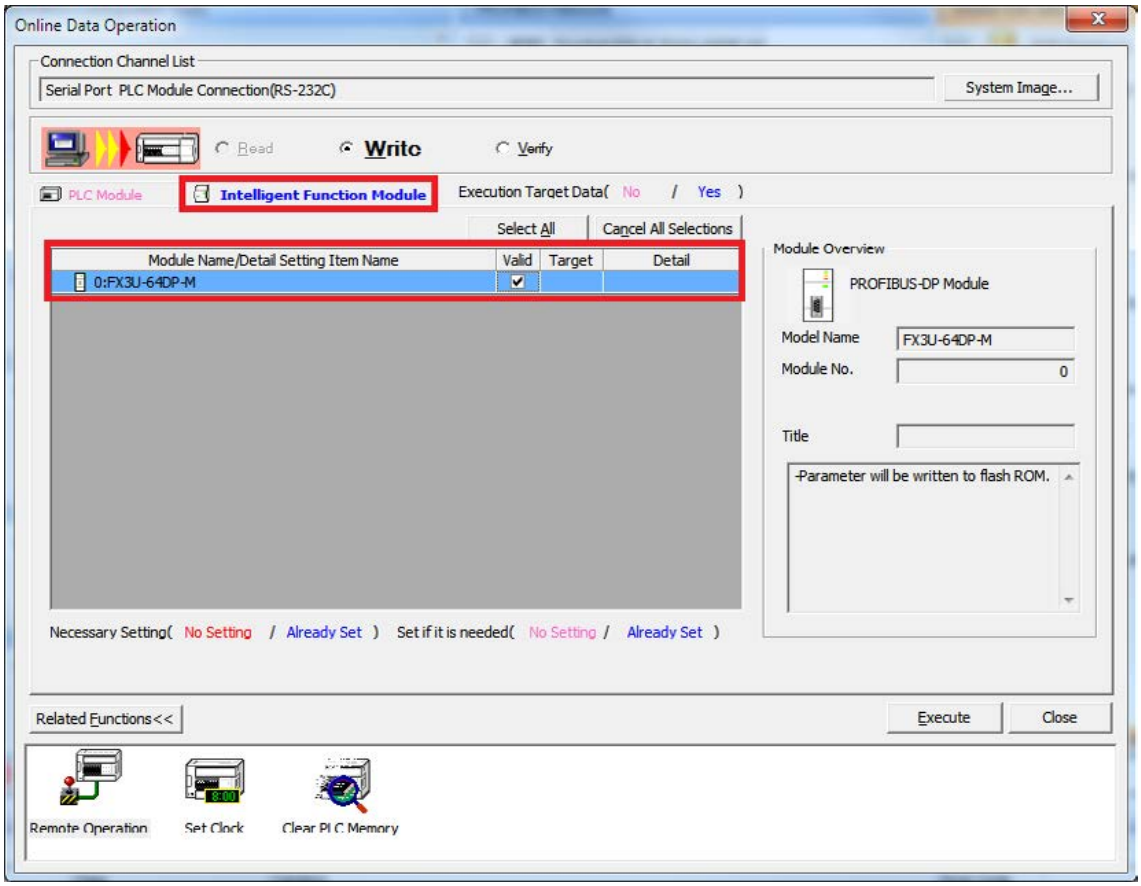
Add the following ladder block to enable rotation command after initialization, and giving rotation command (M1).



⑳ Finally add the code to stop the drive, when M1 is reset. In a new ladder block, check whether the first 3 bits of ZSW1 are ON. This condition means that the drive is in operation mode. If this condition is met, and M1 is not ON, reset the set point speed to zero, and set STW1 as 6.



㉑ Compile, and write the program and parameters to the PLC. Be sure to select the intelligent function module, like on the screenshot below.



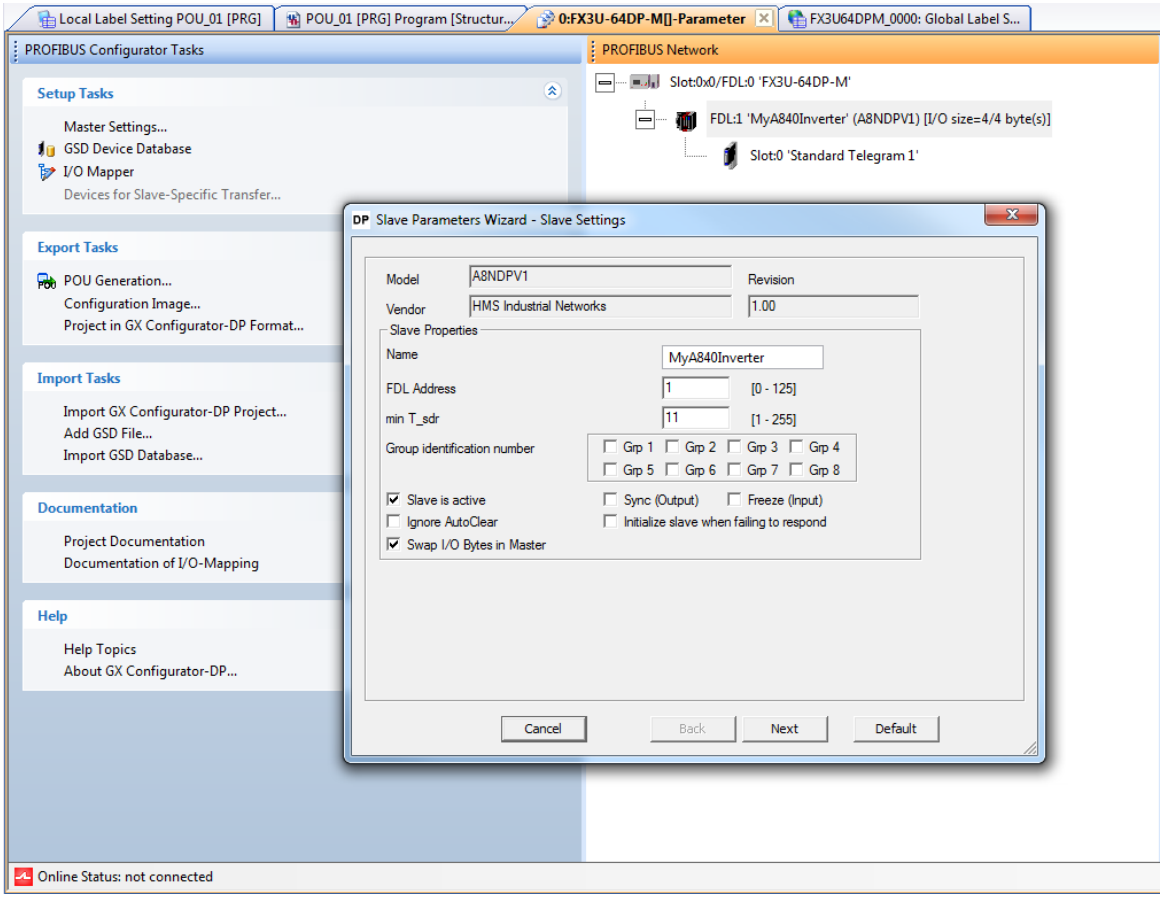
If this option is not visible, then the firmware on your FX3U-64DP-M module does not support the new intelligent module function. Please upgrade the firmware, or download the module setup using the standalone GX Configurator DP.

After resetting the PLC, and powering up the inverter, set bits M0 and M1 to get the inverter running.

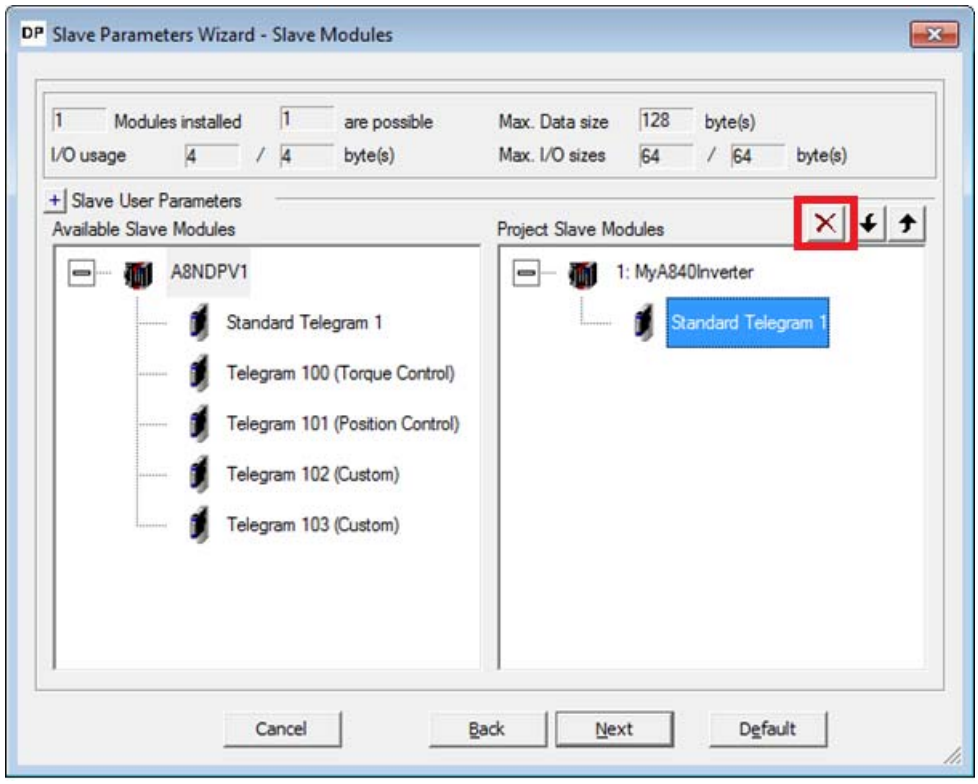
3.11 GX Works (FX-CPU) Telegram 102 example

This chapter describes running the inverter through the A8NDPV1 Profibus option card using GX Works 2 (version 1.501X or newer) with a FXCPU and FX3U-64DP-M Profibus Master.

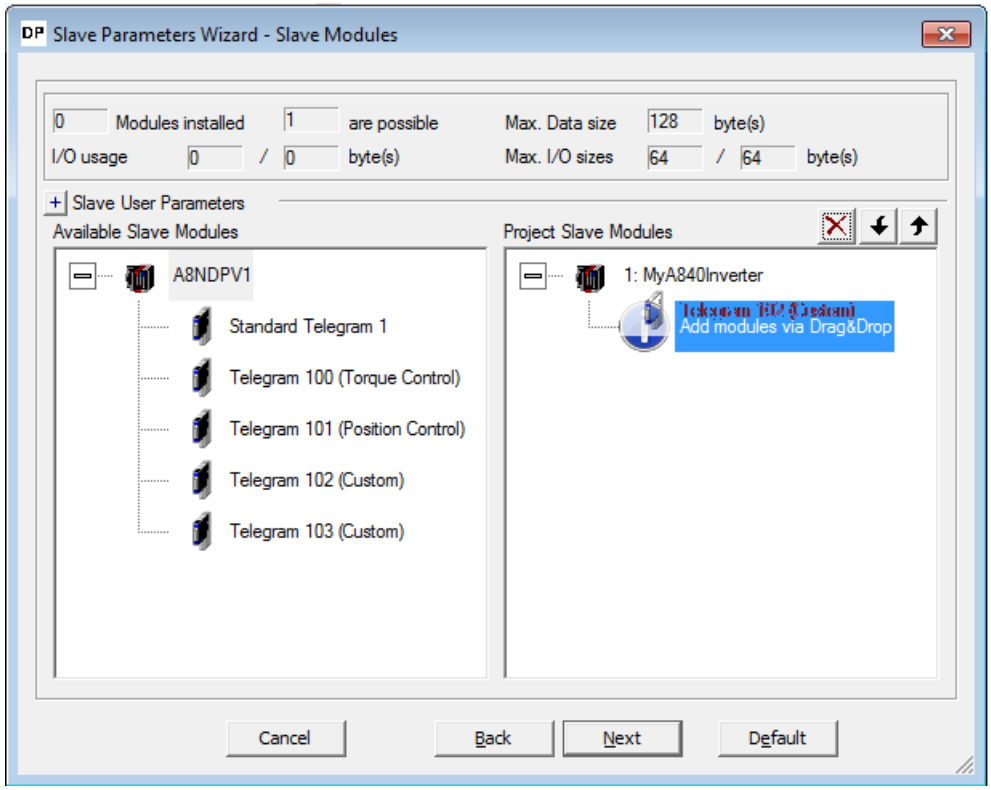
① Enter the Profibus Configurator and double click on the inverter to bring up the Slave Parameters Wizard. Click Next to proceed.



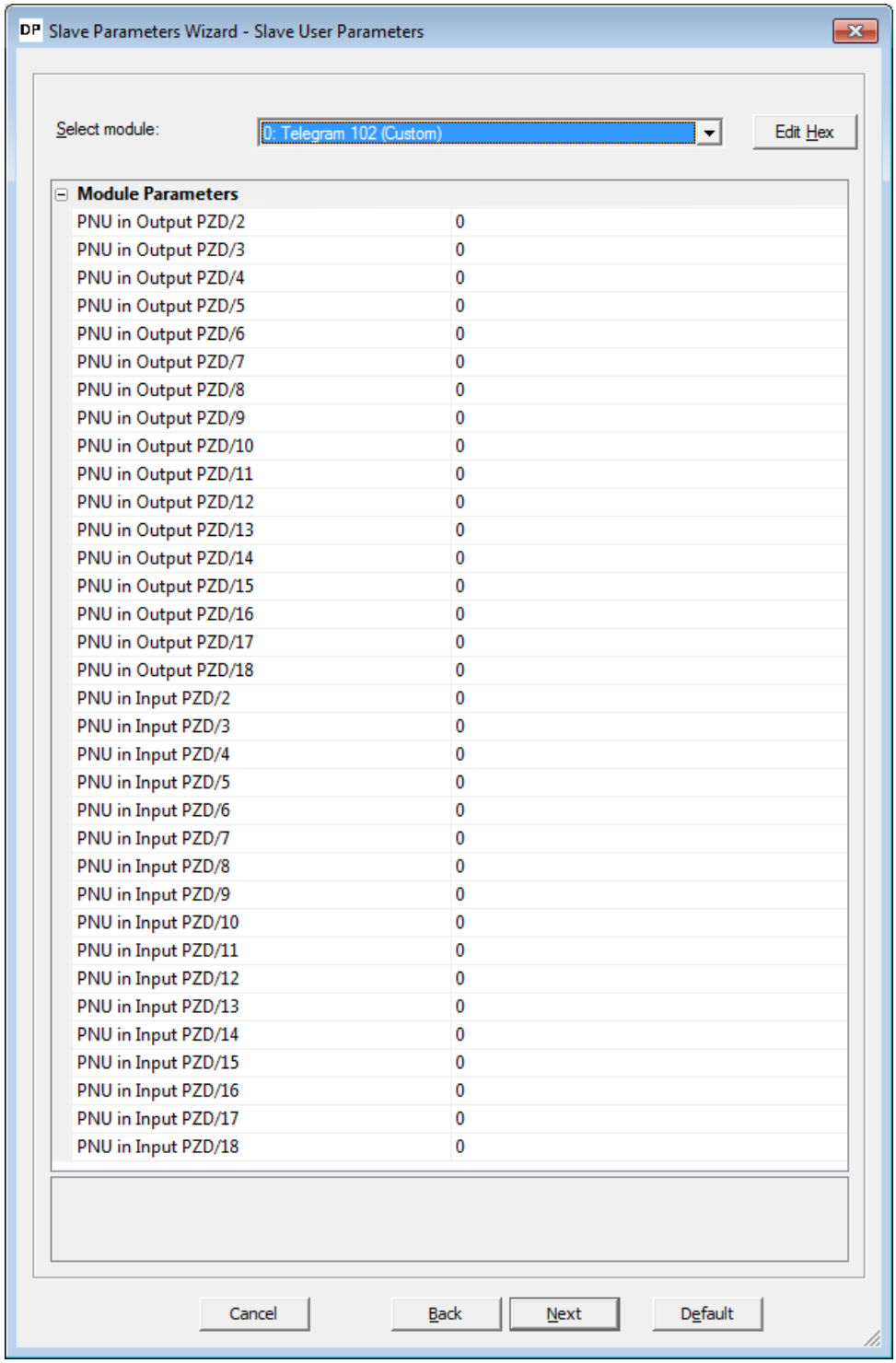
② Select “Standard Telegram 1” in the Project Slave Modules section. Click the red ‘x’ above this section to clear this slot.



③ From the Available Slave Modules, drag “Telegram 102 (Custom)” to the Project Slave Modules section. Click Next to proceed.

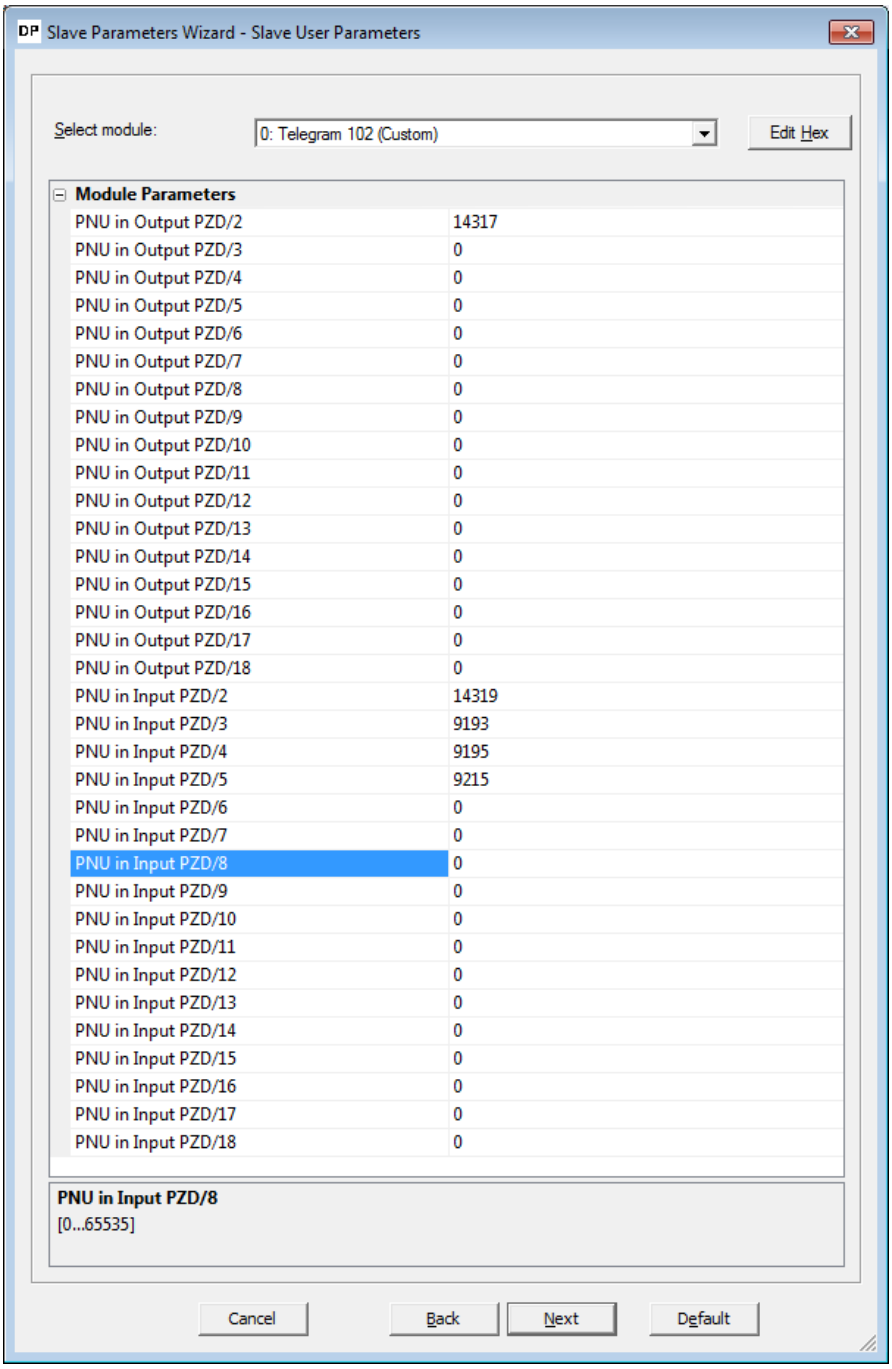


④ From the “select module” dropdown, select slot 0: Telegram 102 (Custom). It is now possible to setup the input and outputs of the telegram. There are 18 inputs and outputs used in this telegram, one of each is reserved for control/status word (leaving 17 inputs/outputs free for custom setup).

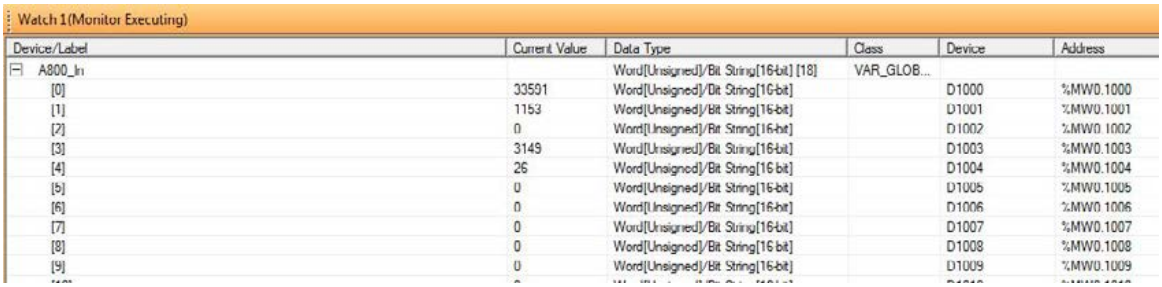


⑤ The setup of the telegram is done by entering signal/parameter numbers into the right column. All monitor data can be viewed as input, and some of the profile parameters as input/output. Please keep in mind, that not all parameters can be accessed via cyclic communication. The PROFIdrive parameter numbers (PNUs) available for use are listed in the Data Exchange subchapters 7.1 to 7.4. An example setup is provided:

Slot	Description	PNU
Output PZD/2	Speed set point	14317
Input PZD/2	Actual speed set point	14319
Input PZD/3	Output frequency	9193
Input PZD/4	Output voltage	9195
Input PZD/5	Actual operation time	9215



⑥ Click Next, and Finish to proceed. Complete the POU Generation and rename the global labels inside the library like before. Rebuild the project, and write it to the PLC. The additional monitor data can be viewed in the appropriate position of the A800_In array.


Watch 1(Monitor Executing)

Device/Label	Current Value	Data Type	Class	Device	Address
A800_In		Word[Unsigned]/Bit String[16-bit] [18]	VAR_GLOB...		
[0]	33591	Word[Unsigned]/Bit String[16-bit]		D1000	%MW0.1000
[1]	1153	Word[Unsigned]/Bit String[16-bit]		D1001	%MW0.1001
[2]	0	Word[Unsigned]/Bit String[16-bit]		D1002	%MW0.1002
[3]	3149	Word[Unsigned]/Bit String[16-bit]		D1003	%MW0.1003
[4]	26	Word[Unsigned]/Bit String[16-bit]		D1004	%MW0.1004
[5]	0	Word[Unsigned]/Bit String[16-bit]		D1005	%MW0.1005
[6]	0	Word[Unsigned]/Bit String[16-bit]		D1006	%MW0.1006
[7]	0	Word[Unsigned]/Bit String[16-bit]		D1007	%MW0.1007
[8]	0	Word[Unsigned]/Bit String[16-bit]		D1008	%MW0.1008
[9]	0	Word[Unsigned]/Bit String[16-bit]		D1009	%MW0.1009

3.12 GX Works (FX-CPU) Acyclic communication example

This chapter describes using acyclic communication through the A8NDPV1 Profibus option card using GX Works 2 with a FXCPU and FX3U-64DP-M Profibus Master. It contains examples of reading and writing individual parameters. First perform the setup using telegram 1.

The process of acyclic communication parameter read consists of a write request, and a response read. For details please refer to the Acyclic Data Exchange subchapter 7.7.

3.12.1 Reading a parameter (Sequence 1)

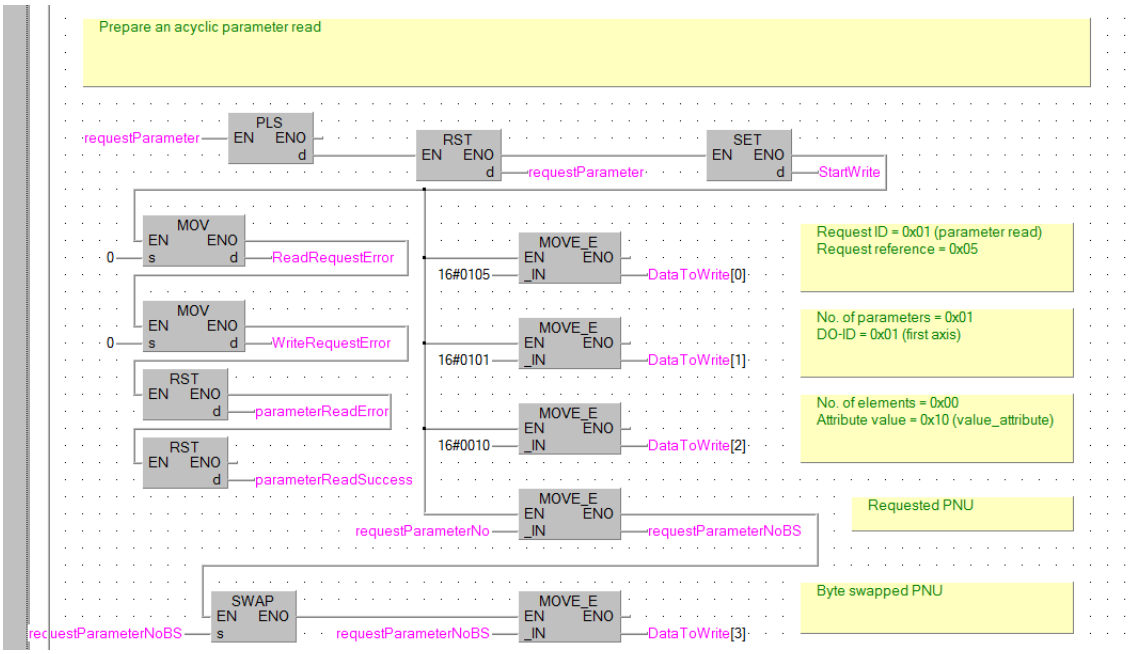
Creating the request

Start by preparing the content for the parameter read request. The data request has the following format:

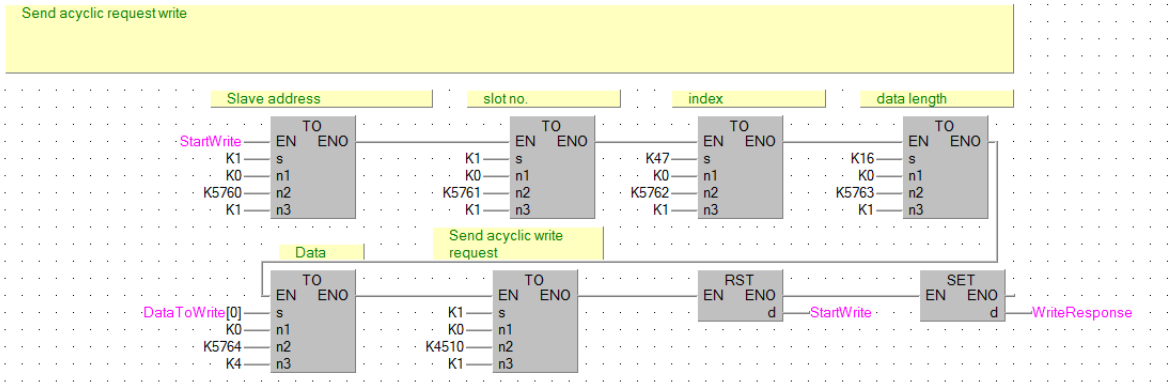
Byte no.	Description	Value
0	Request ID	0x01
1	Request reference	0x01-0xFF
2	No. of parameters	0x01
3	DO-ID	0x01
4	No. of elements	0x00
5	Attribute value	0x10 (value attribute)
6-7	Parameter number	Byte swapped PNU

Request reference is any valid number, this value is mirrored back in the response and can be used to distinguish multiple requests. The parameter number needs to be byte swapped(the SWAP instruction can be used to swap the lower and higher byte of a word).

In the code below *requestParameterNo* contains the unswapped parameter number, while *requestParameterNoBS* contains the byte swapped version. The content of the parameter request(*DataToWrite*), and the request to perform it (*StartWrite*), are shown in the screenshot below:



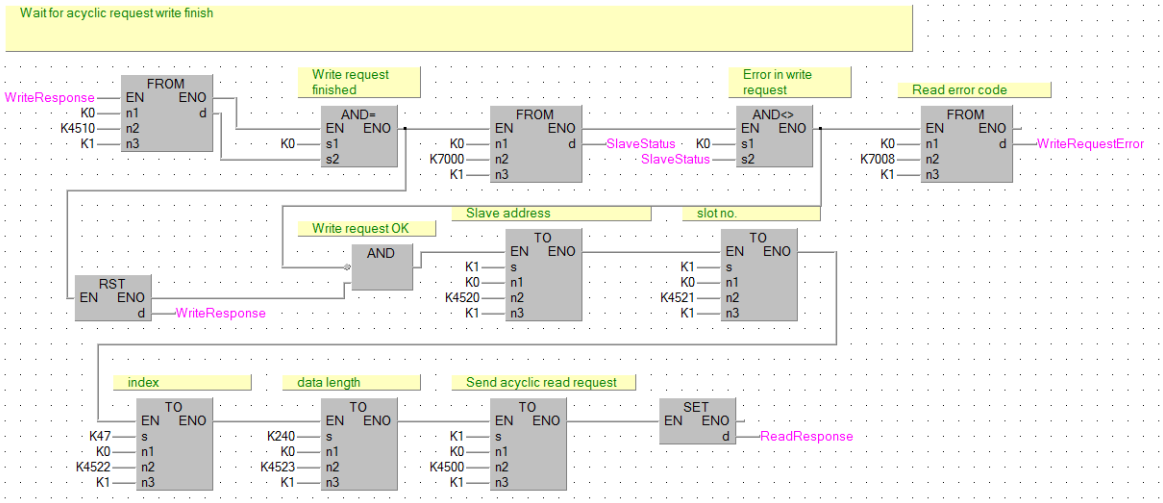
Proceed to issue the actual request by setting the appropriate BFMs. Set the slave address, slot number, target index, length of data and transfer the data of the request that was prepared above. Finally set the acyclic write request flag (BFM#4510).



Receiving the response

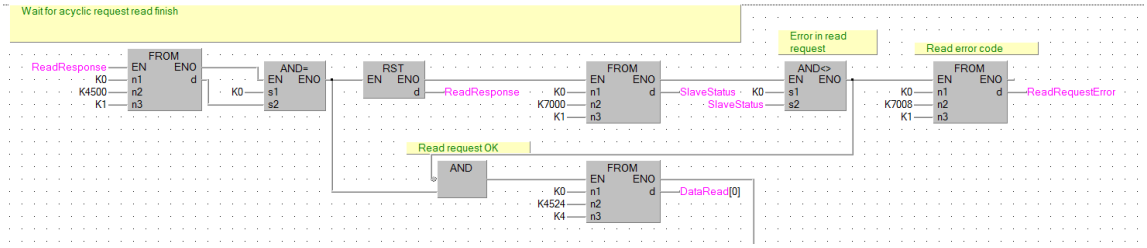
When the write request has been processed, BFM#4510 will be equal to 0. If the request is completed successfully (BFM#7000 is equal to 0), then perform a response read. In case of an error, read the error details from BFM#7008. For an explanation of the error codes refer to the FX3U-64DP-M user manual.

To issue the read request, set the slave address, slot number, target index, and length of data of the expected response. It should be noted, that 240 bytes can always be read. This is the maximum allowed length of a response. In case of a shorter response, only the available data will be read.



When the read request has been processed, BFM#4500 will be equal to 0. Similarly like before BFM#7000 will be checked for any errors.

If the read request was completed successfully, the requested data will be available starting from BFM#4524.



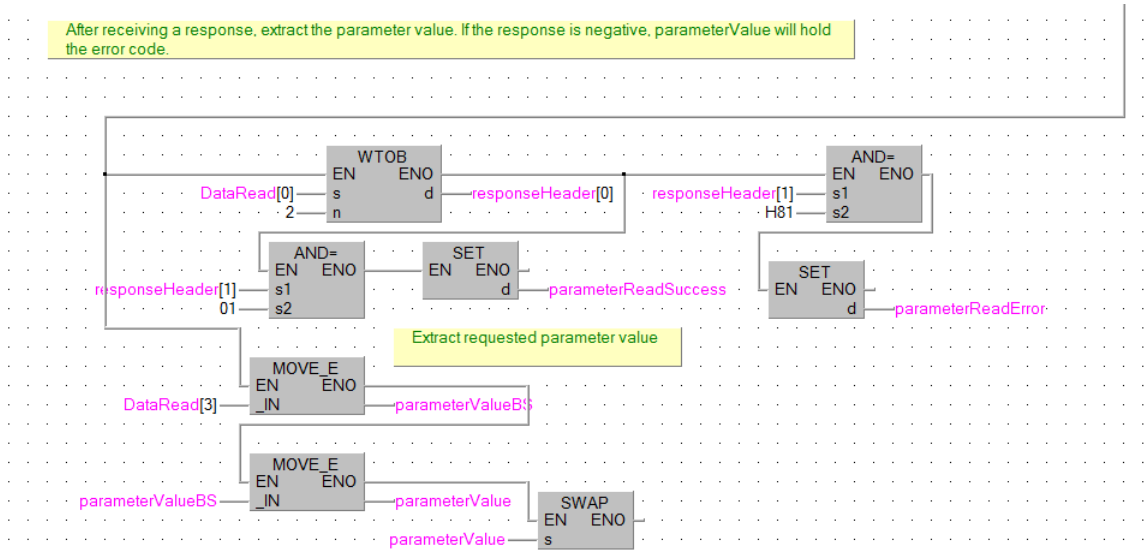
Proceed to process the received response. The format of a positive response is as follows:

Byte no.	Description	Value
0	Request ID	0x01
1	Request reference	0x01-0xFF (same as in request)
2	No. of parameters	0x01
3	DO-ID	0x01
4	No. of values	0x01
5	Format	See data format type table
6-7	Parameter value	Byte swapped parameter value

In case of a negative response, the format is as follows:

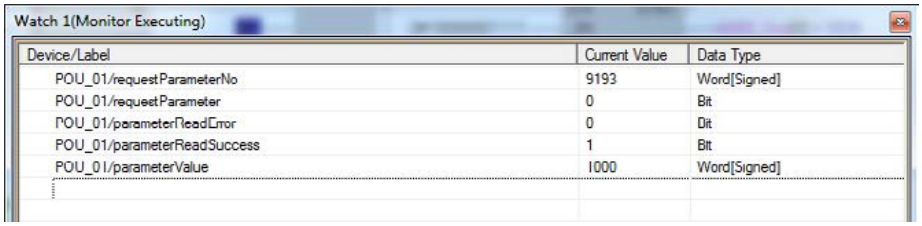
Byte no.	Description	Value
0	Request ID	0x81
1	Request reference	0x01-0xFF (same as in request)
2	No. of parameters	0x01
3	DO-ID	0x01
4	No. of values	0x01
5	Format	0x44 (error)
6-7	Error value	Byte swapped error value (see error table)

First check the received request ID to identify the type of received response (positive or negative). Set bit *parameterReadError* in case of a negative response, and write the error code to variable *parameterValue*. If a positive response was received, set bit *parameterReadSuccess* and write the read parameter value to *parameterValue*.

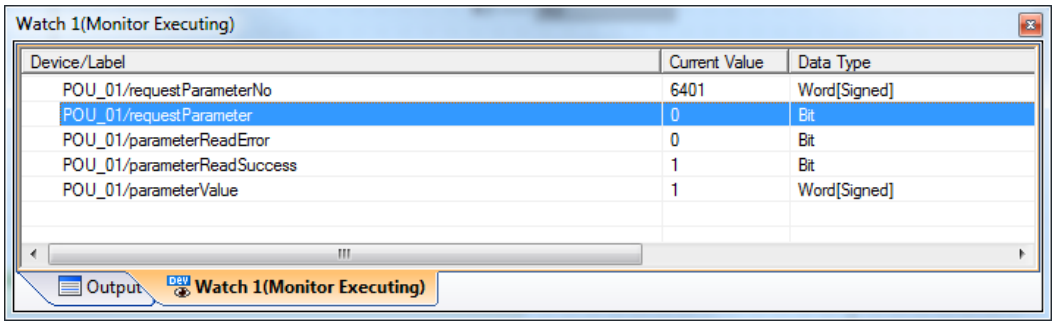


Executing example requests

After compiling the program, and writing it to the PLC, it is possible to execute parameter read requests. First, get the inverter running by setting bits M0 and M1. Now read the actual frequency by writing PNU 9193 to *requestParameterNo* and setting bit *requestParameter*. The result should be a value of 1000 in *parameterValue* which corresponds to 10Hz. The PROFIdrive parameter numbers (PNUs) available for use are listed in the Data Exchange subchapters 7.1 to 7.4.



Try reading an inverter parameter, for example the FDL address of the inverter (parameter 1305 + 5096 offset = PNU 6401). The expected result is 1. It is possible to read any inverter parameter in this way.



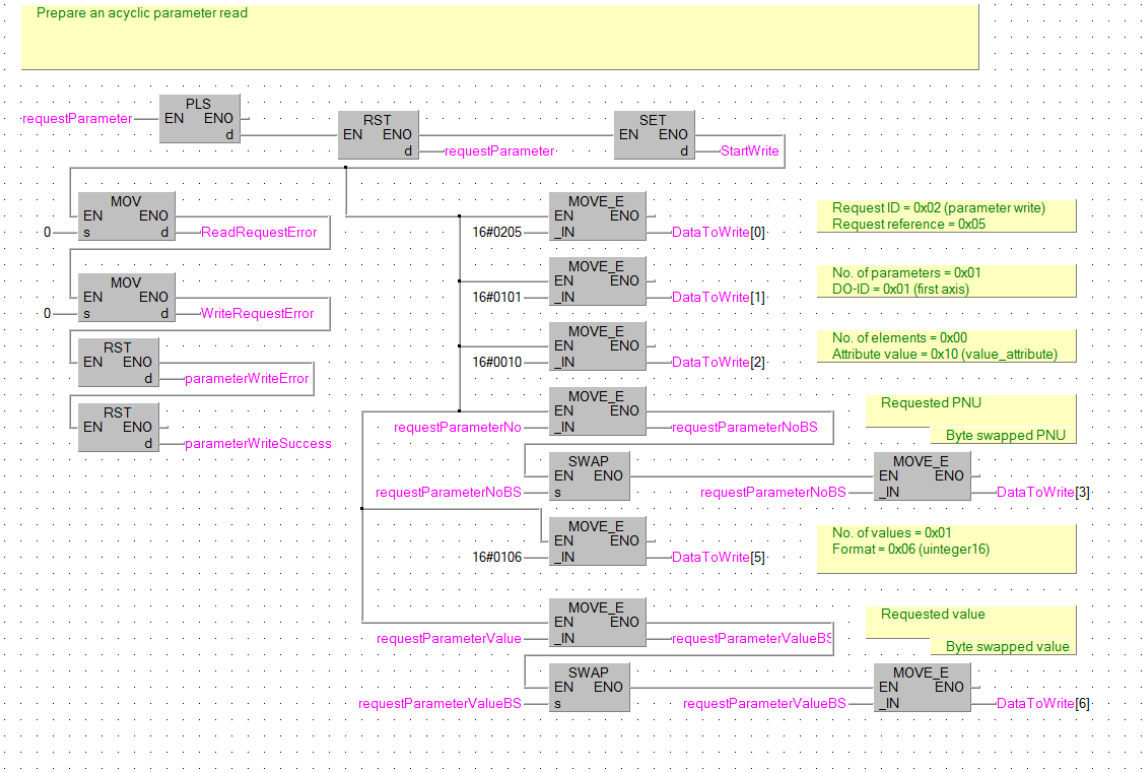
3.12.2 Changing parameters (Sequence 2)

Creating the request

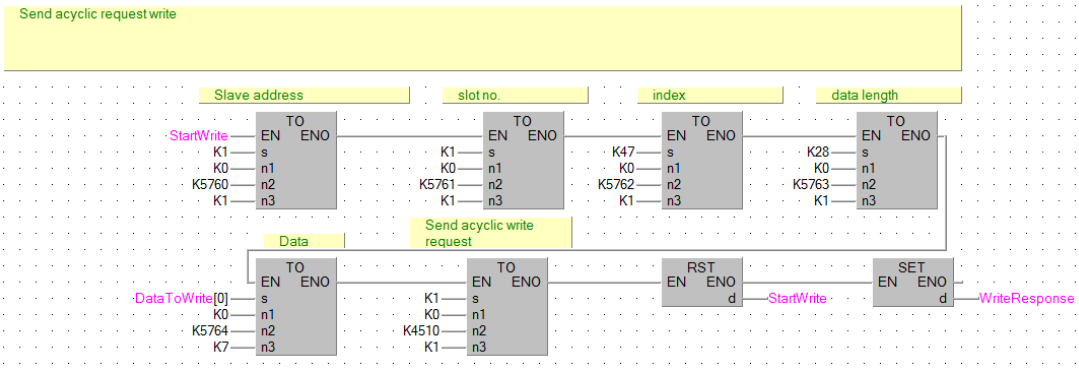
The format of the write request is as follows:

Byte no.	Description	Value
0	Request ID	0x02
1	Request reference	0x01-0xFF
2	No. of parameters	0x01
3	DO-ID	0x01
4	No. of elements	0x00
5	Attribute value	0x10 (value attribute)
6-7	Parameter number	Byte swapped PNU
8-9	Subindex (irrelevant)	0x00
10	No. of values	0x01
11	Format	0x06 (UINTEGER16 for all inverter parameters) See data format type table
12-13	Set value	Byte swapped set value

Modify the ladder block preparing the content of the request.



Change the actual length of the write request. Change the length to 28 bytes, and change the amount of words to transfer (7 words) starting from BFM#5764.



Reading the response

The length of the expected read response is up to 8 bytes. The ladder block responsible for waiting for completion of the write request, and issuing the read request does not change.

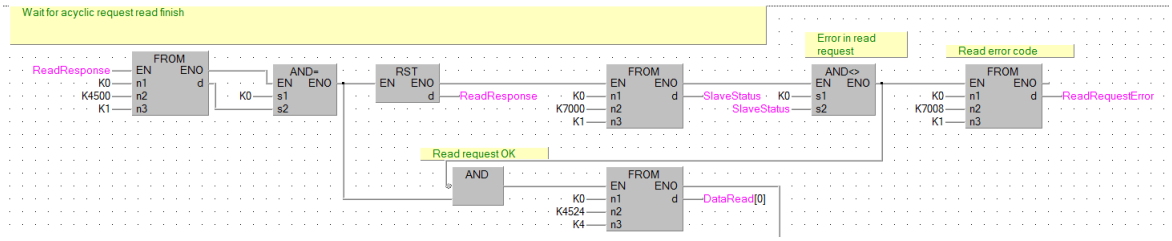
Finally modify the ladder block processing the response data. A positive response consists of just 4 bytes of data in the following format:

Byte no.	Description	Value
0	Request ID	0x02
1	Request reference	0x01-0xFF (same as in request)
2	No. of parameters	0x01
3	DO-ID	0x01

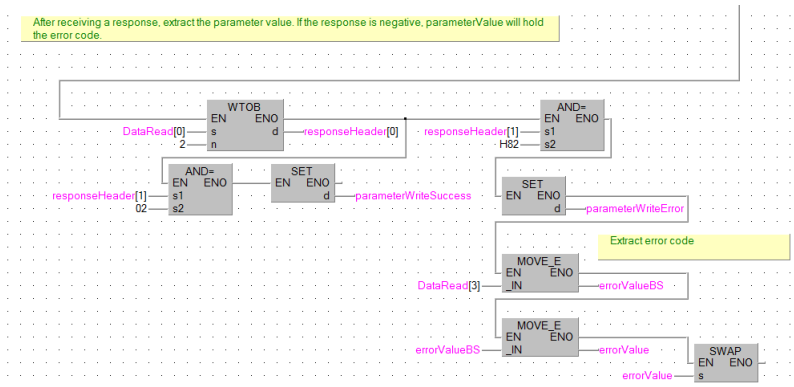
In case of a negative response, the format is as follows:

Byte no.	Description	Value
0	Request ID	0x82
1	Request reference	0x01-0xFF (same as in request)
2	No. of parameters	0x01
3	DO-ID	0x01
4	No. of values	0x01
5	Format	0x44 (error)
6-7	Error value	Byte swapped error value (see error table)

After a successful response is received (BFM#7000 equals 0), check the received request ID to determine, whether the parameter write request was processed successfully.

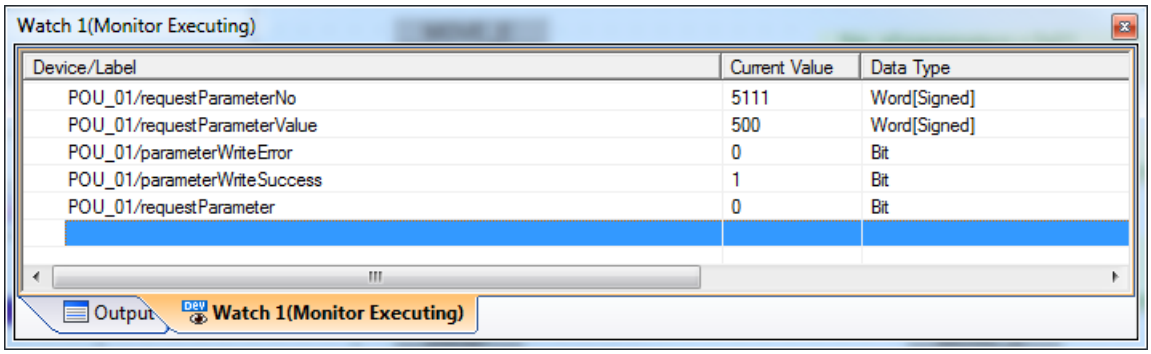


If the request ID is equal to H02, a positive response was received, and the parameter write was successful (bit *parameterWriteSuccess* is set). In case of a negative response (request ID is equal to H82), bit *parameterWriteError* is set and the error code is saved to variable *errorValue*.

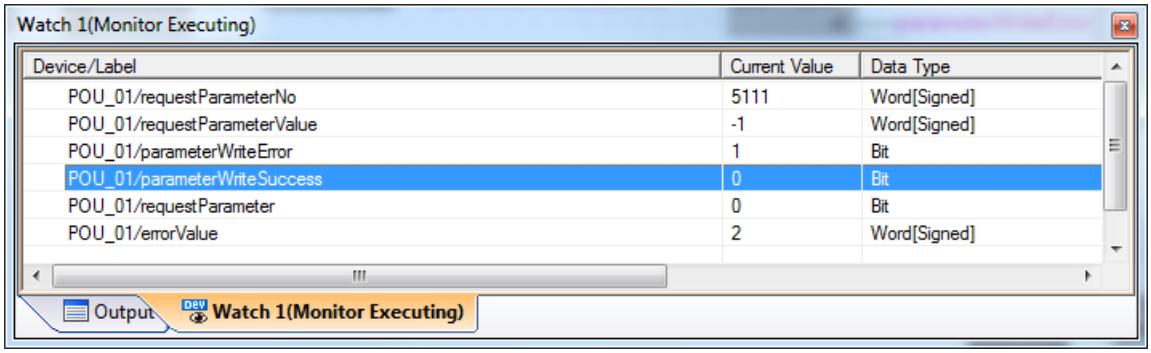


Executing example requests

To perform a parameter write, set the PNU of the parameter to change in *requestParameterNo*. As an example, to change parameter 15 (Jog frequency), add the offset 5096 and write the result to this variable. Set variable *requestParameterValue* to the requested value. The value that will be set, will be the output of multiplication of the set value, and the minimum setting increments, as according to the parameter list in the inverter manual. For parameter 15, the minimum setting increment is 0.01Hz. If a jog frequency of 5Hz is requested, set *requestParameterValue* to 500. Finally execute the request by setting bit *requestParameter* to ON. A successful write will result in setting of the *parameterWriteSuccess* bit.



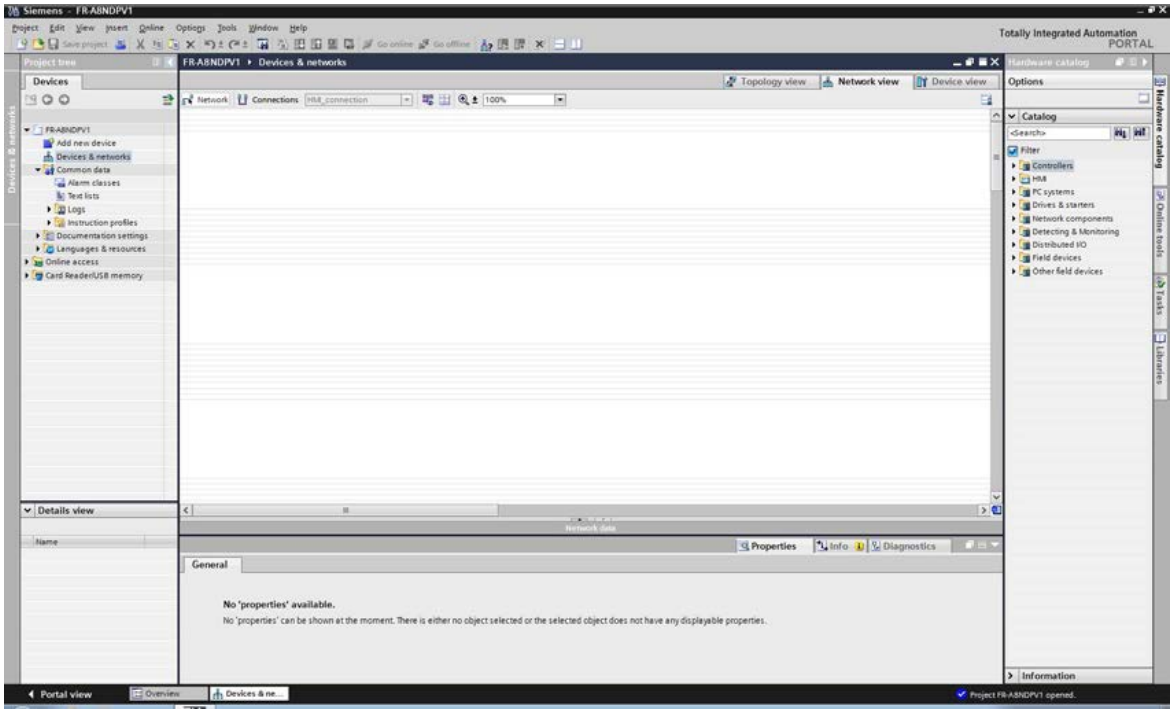
Test receiving a negative response, by trying to write an out of range value. After trying to write 0xFFFF as the value of parameter 15, a negative response with error code 0x02 (LOW_OR_HIGH_LIMIT_EXCEEDED) will be received, which is the expected behavior.



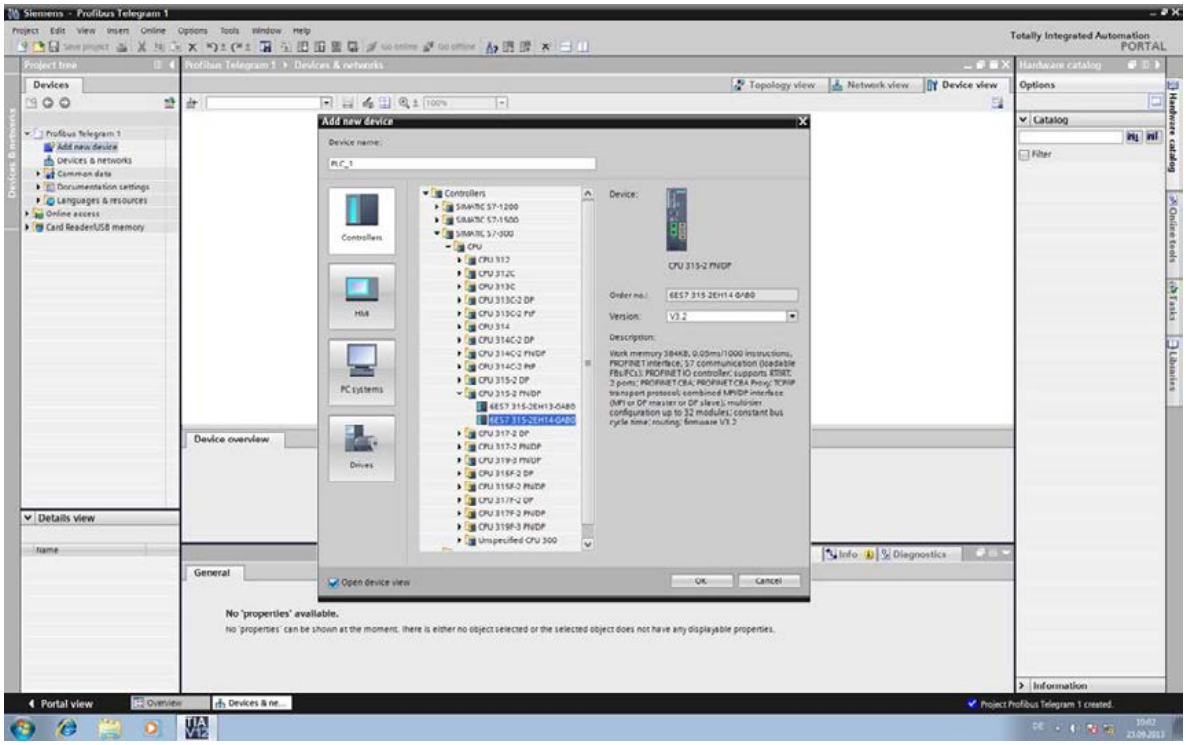
3.13 TIA Portal Telegram 1 example

This chapter describes running the inverter through the A8NDPV1 Profibus option card using TIA Portal.

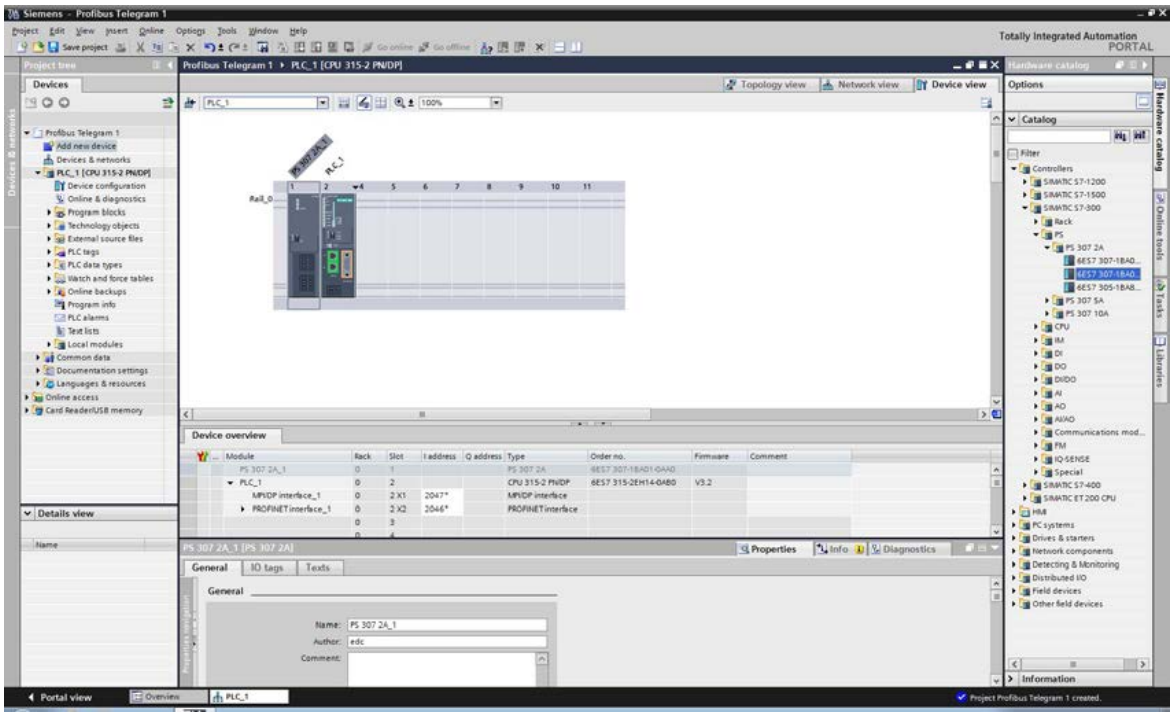
① Start with a new project.



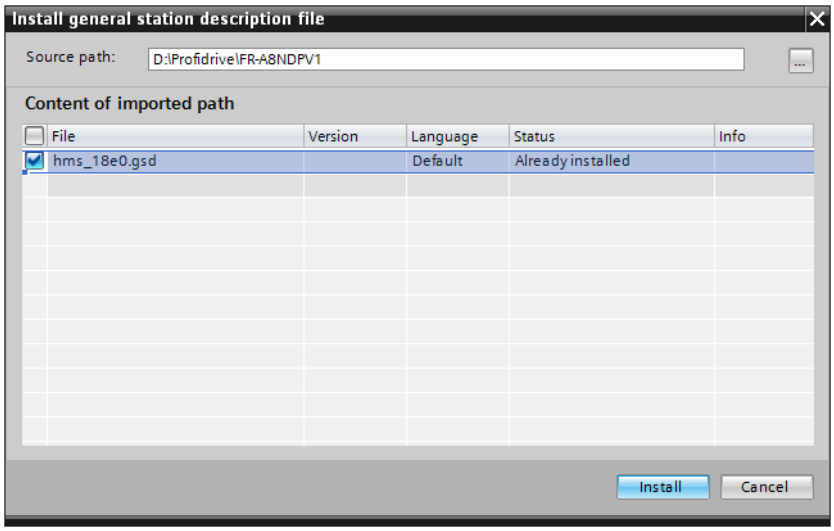
② Select Add new device from the Project tree section. In the new window select the PLC and click OK to add it to the project.



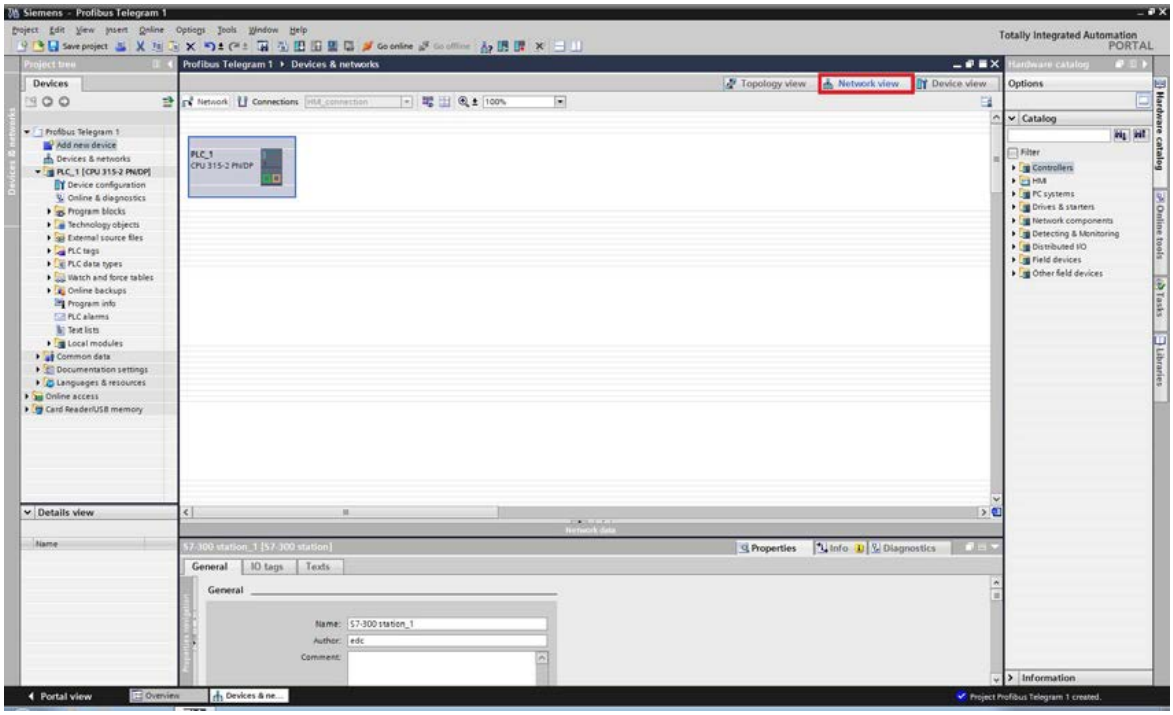
③ Now add the power supply. Select the power supply from the hardware catalog on the right side of the screen.



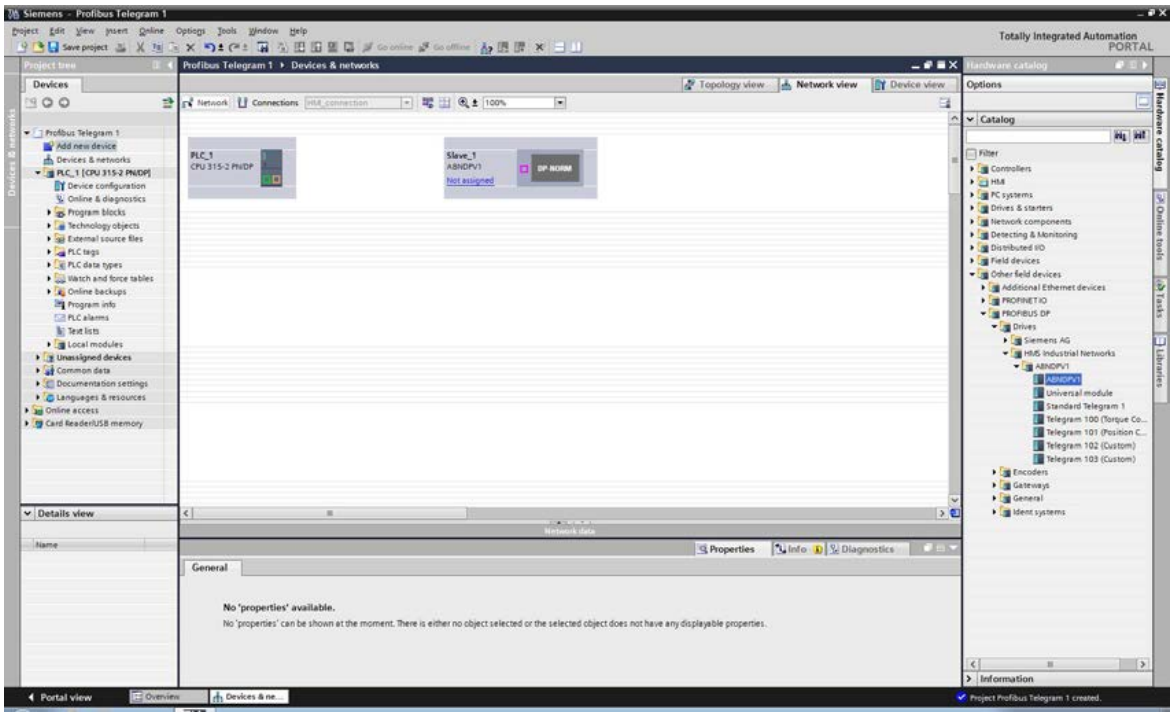
④ From the top menu select Options and Install general station description file (GSD). Search for the GSD file supplied with this manual, click install and follow instructions to add the option card to the hardware library.



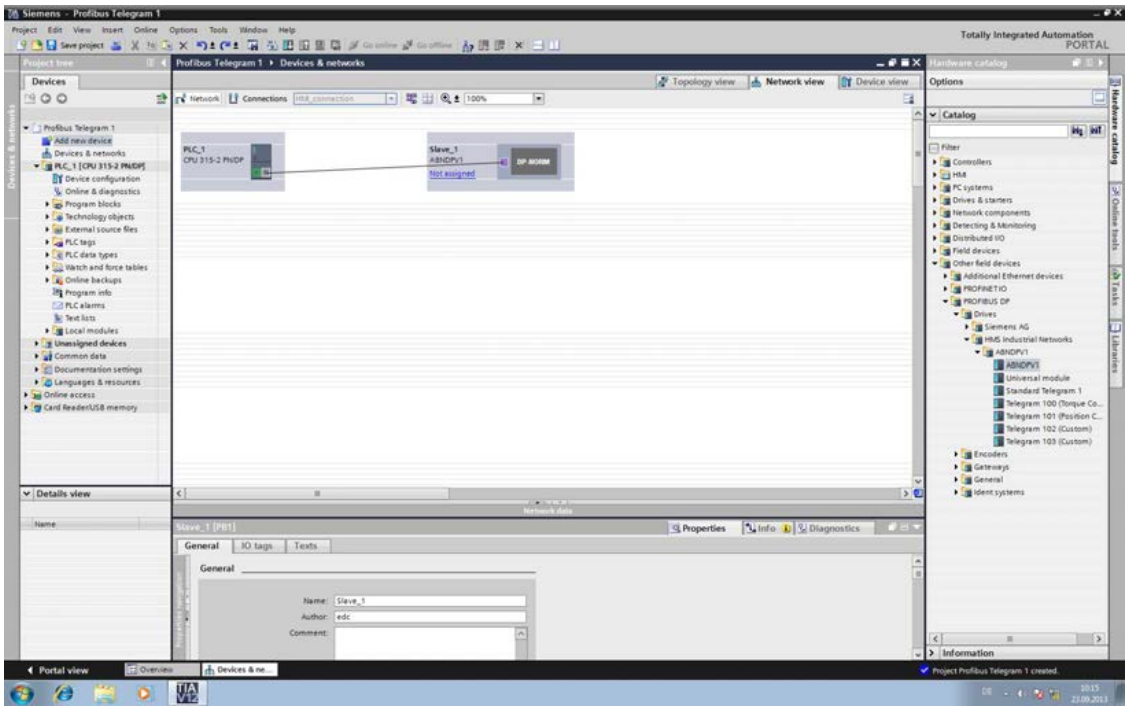
⑤ After finishing installation and returning to the main window, switch to Network view.



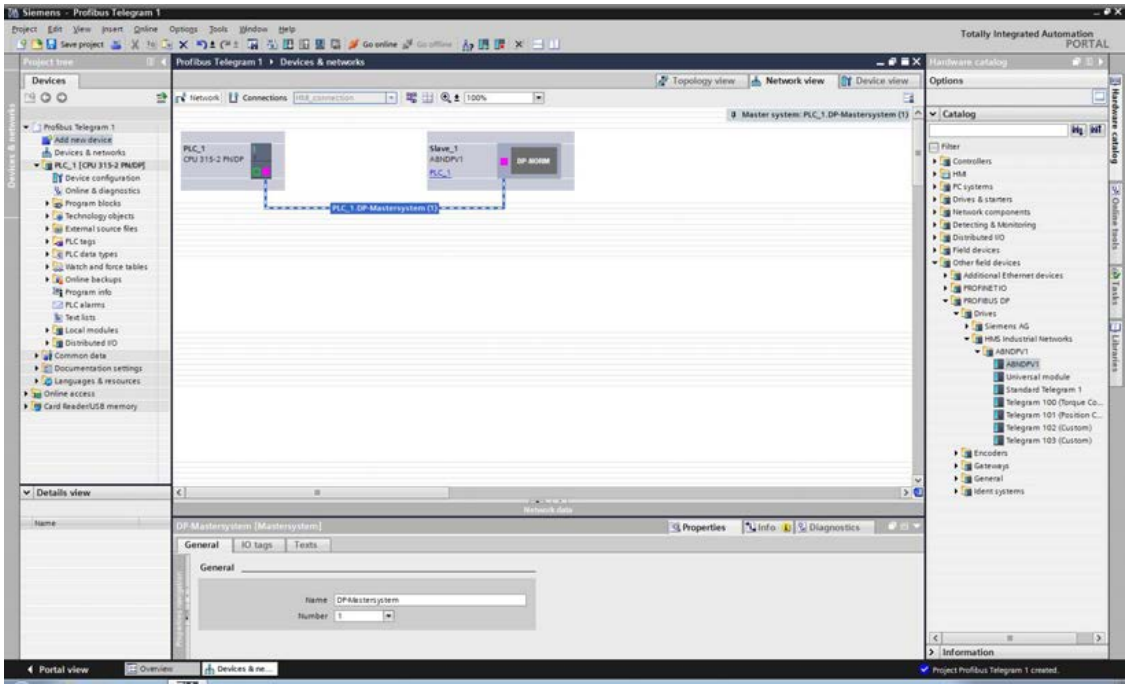
⑥ From the hardware catalog on the right side of the screen, expand Other field devices => Profibus DP => Drives => HMS Industrial Networks => A8NDPV1 and add A8NDPV1 to the project by dragging it to an empty space.



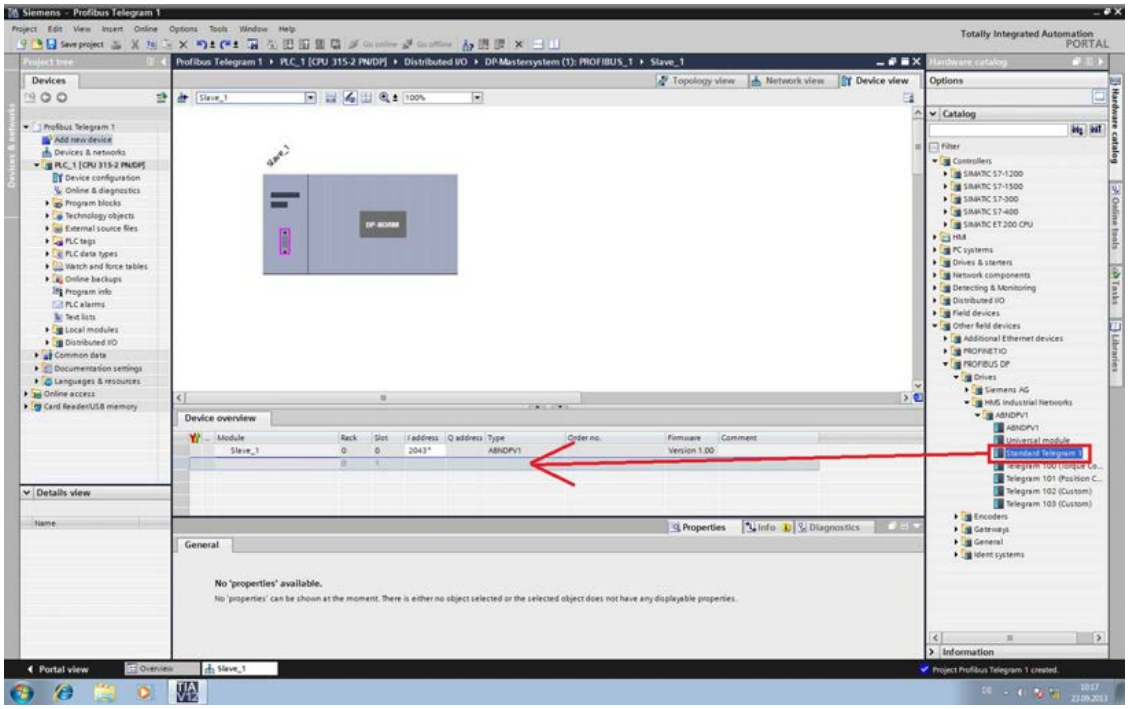
⑦ Drag a line from the slave connection node to the PLC connection node to make the PROFIBUS connection.



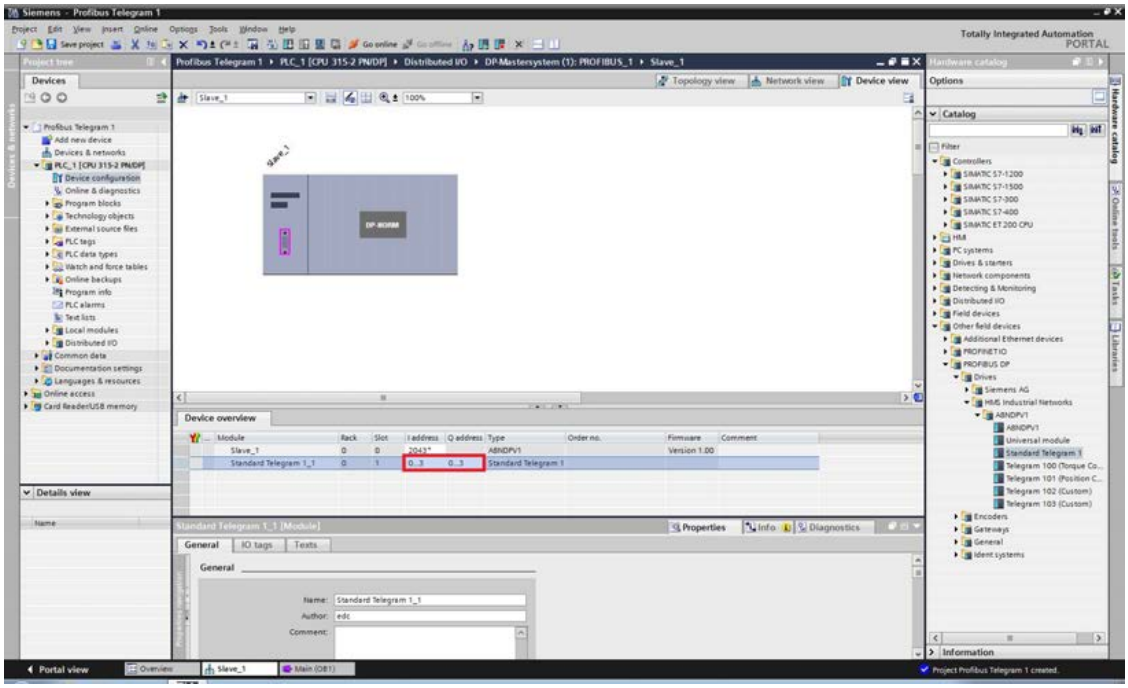
The result should look like below:



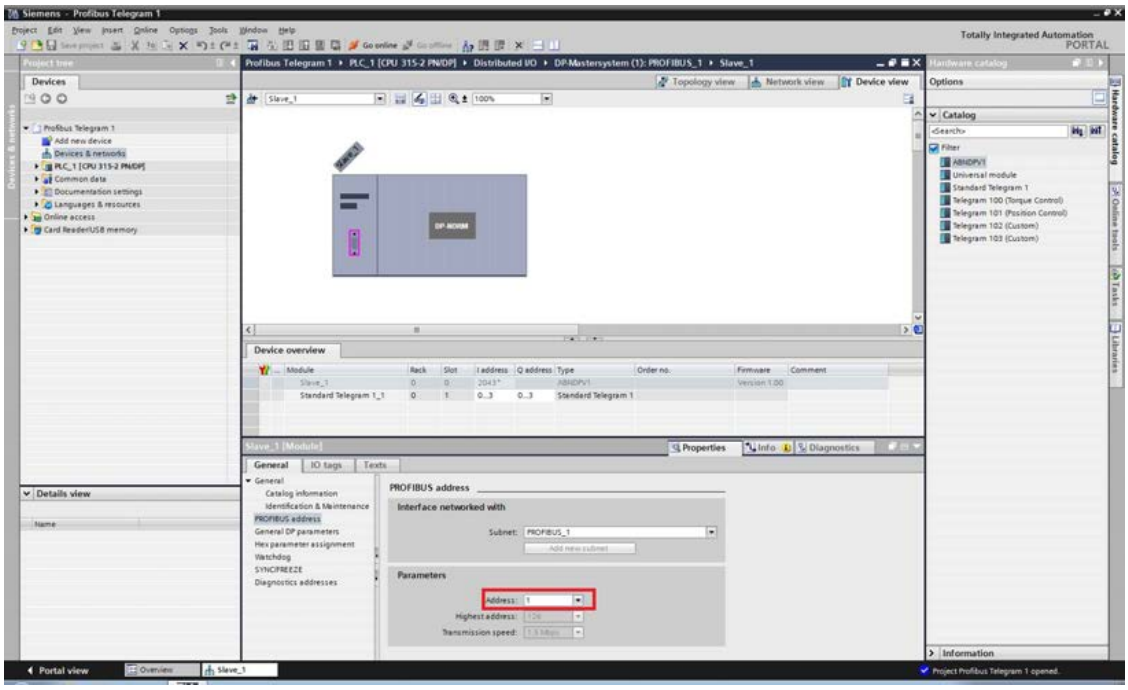
⑧ Double click on the slave to enter device view. From the hardware catalog expand Other field devices => Profibus DP => Drives => HMS Industrial Networks => A8NDPV1 and drag Standard Telegram 1 to the available slot.



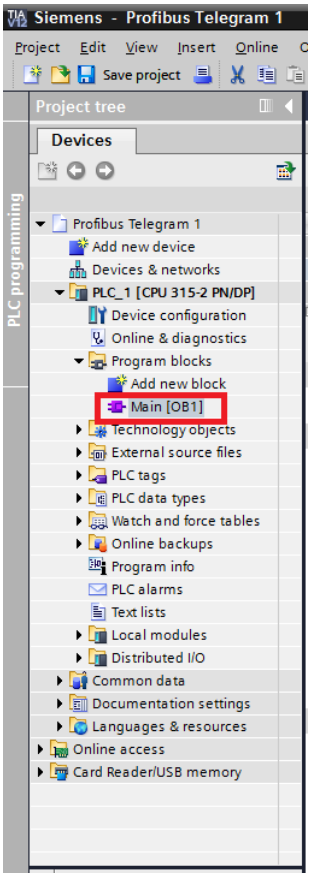
⑨ Change the I address and Q address to start at 0.



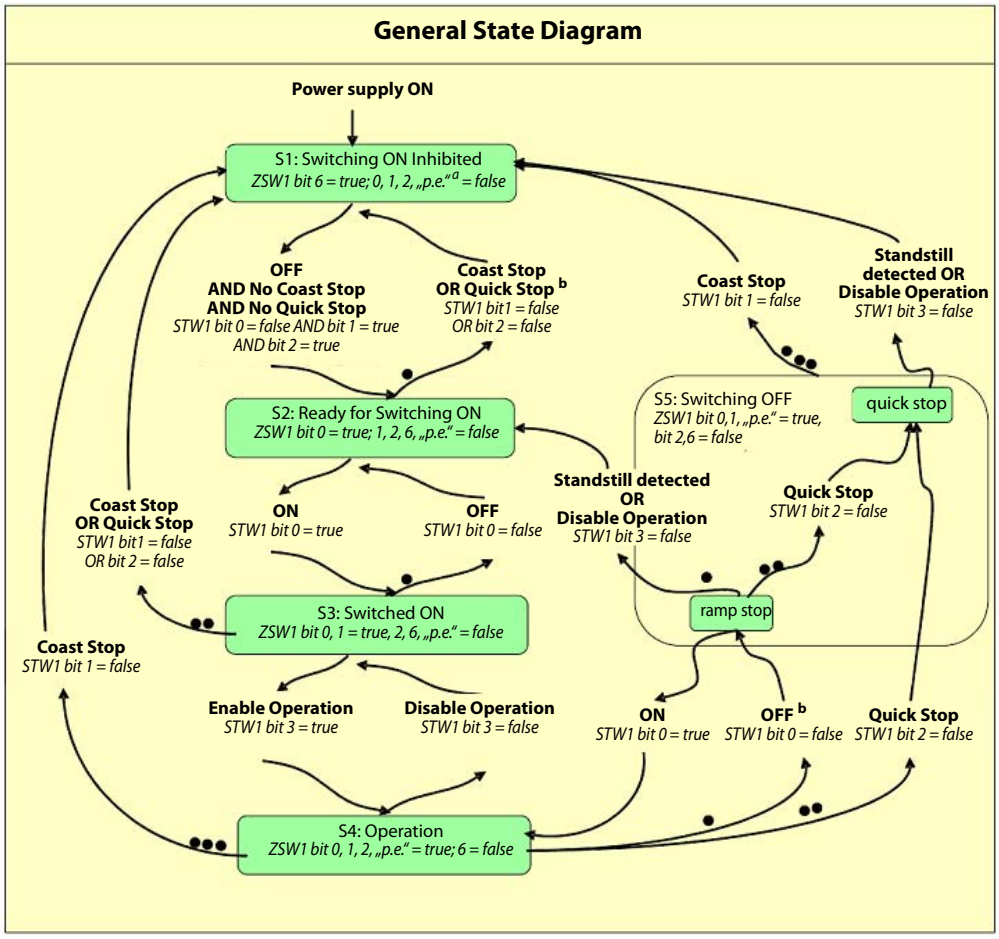
⑩ Double click on Slave_1 from the device view to bring up the Properties panel. Select PROFIBUS Address and enter 1 in the address field.



⑪ From the project tree select Program blocks and double click on Main to start editing the program. With this setup STW1 is available under QW0, NSOLL_A under QW2, ZSW1 under IW0, and NIST_A under IW2.



⑫ After powering up the inverter and Profibus master, connection will be established. Expect to see bit 6 (Switching on inhibited) set in ZSW1. Below is a simplified state diagram, dependent on control word 1 (STW1).

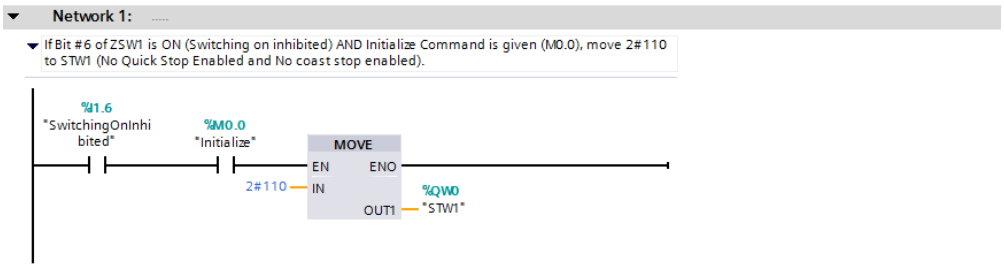


Control Word 1 (STW1) description

State / Bit combination	Bit 10	Bit 7	Bit 3	Bit 2	Bit 1	Bit 0
Description	**Control By PLC**	**Fault acknowl-edge**	**Enable operation**	**No quick stop**	**No coast stop**	**On**
Switching on Inhibited	-	-	-	-	0	-
	-	-	-	0	-	-
Ready To Switch On	-	-	-	1	1	-
Switched On	-	-	-	1	1	1
Operation	-	-	1	1	1	1
Rotation	1	-	1	1	1	1
Fault reset	-	0 =>1	-	-	-	-

Both No Coast Stop and No Quick Stop need to be set to reset the Switching On Inhibited bit in ZSW1. To do this set bits 1 and 2 in STW1, that is bit 1 and 2 in QW0.

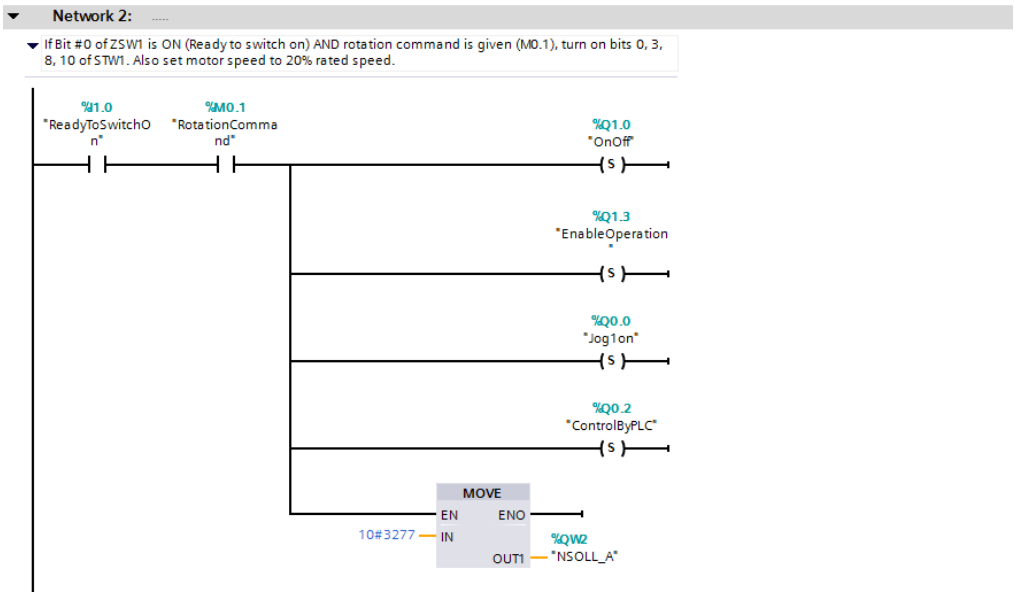
Input the following ladder block to allow the inverter to enter “Ready For Switching On” status, after connection is established, and initialization command is given (M0.0).



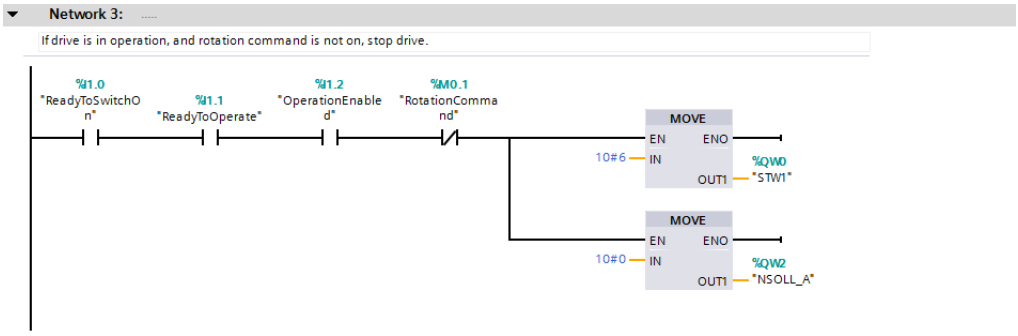
⑬ This initialization will result in setting bit 0 in ZSW1 (Ready To Switch On). It is now possible to switch on the inverter, and start operation. To do this, enable bits 0 (ON/Off), 3 (enable operation), and 10 (Control By PLC); while leaving bits 1 and 2 enabled.

⑭ At this moment it is also possible to set the desired rotation frequency. Assuming the motor is rated for 50 Hz (inverter parameter 3) and a frequency of 10 Hz is requested, set 20% of the rated motor speed to achieve this frequency. The value in Profidrive to allow full power is 16384, so setting 20% of this value (3276,8 round to 3277) will give the expected result. This value shall be set in NSOLL_A.

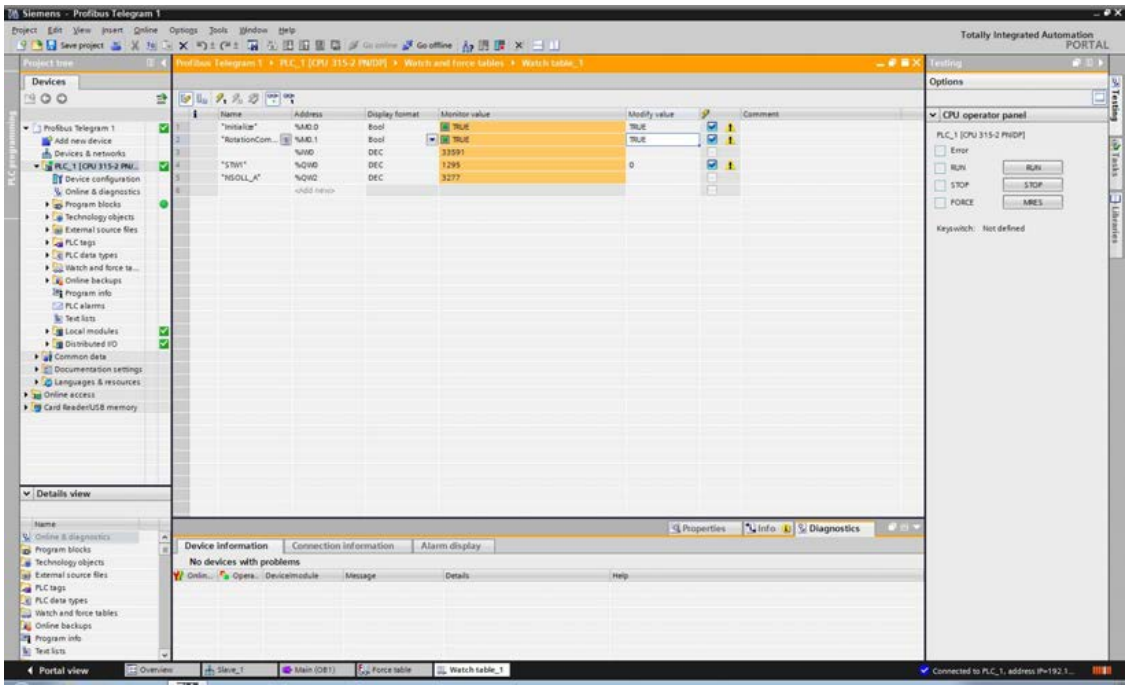
Add the following ladder block to enable rotation command after initialization, and giving rotation command (M0.1).



⑮ Finally add the code to stop the drive, when M0.1 is reset. In a new ladder block, check whether the first 3 bits of ZSW1 are ON. This condition means that the drive is in operation mode. If this condition is met, and M1 is not ON, reset the set point speed to zero, and set STW1 as 6.



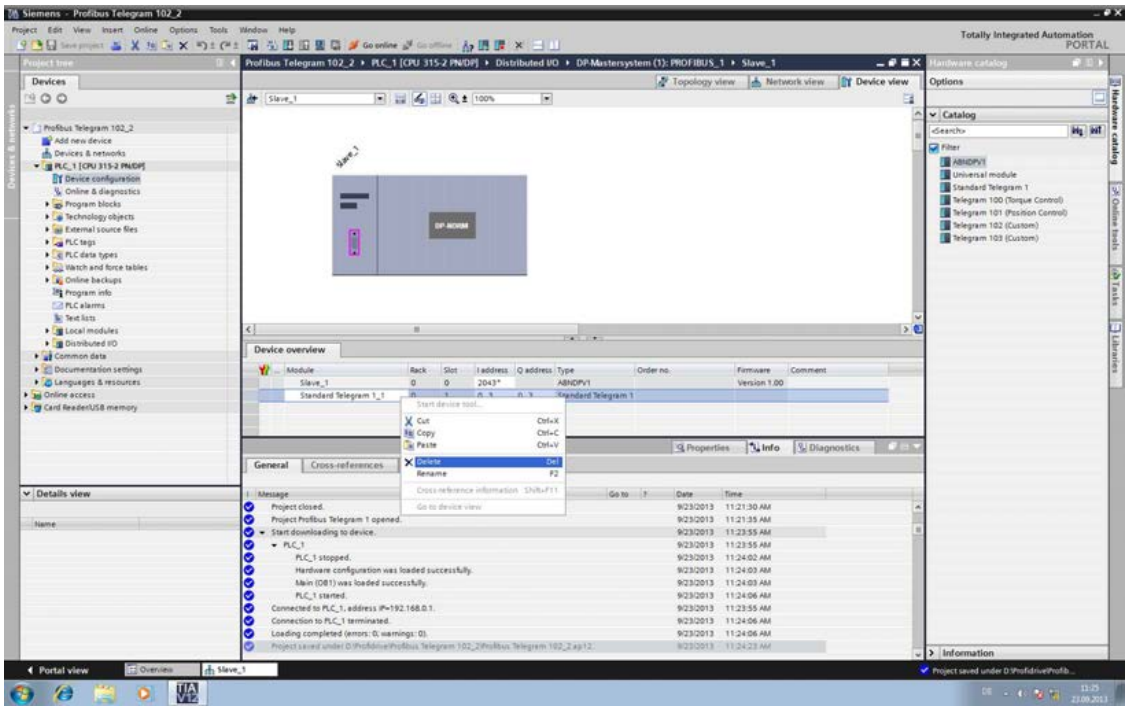
⑯ Compile, and write the program and parameters to the PLC. After resetting the PLC, and powering up the inverter set bits M0.0 and M0.1 to get the inverter running with the requested set speed.



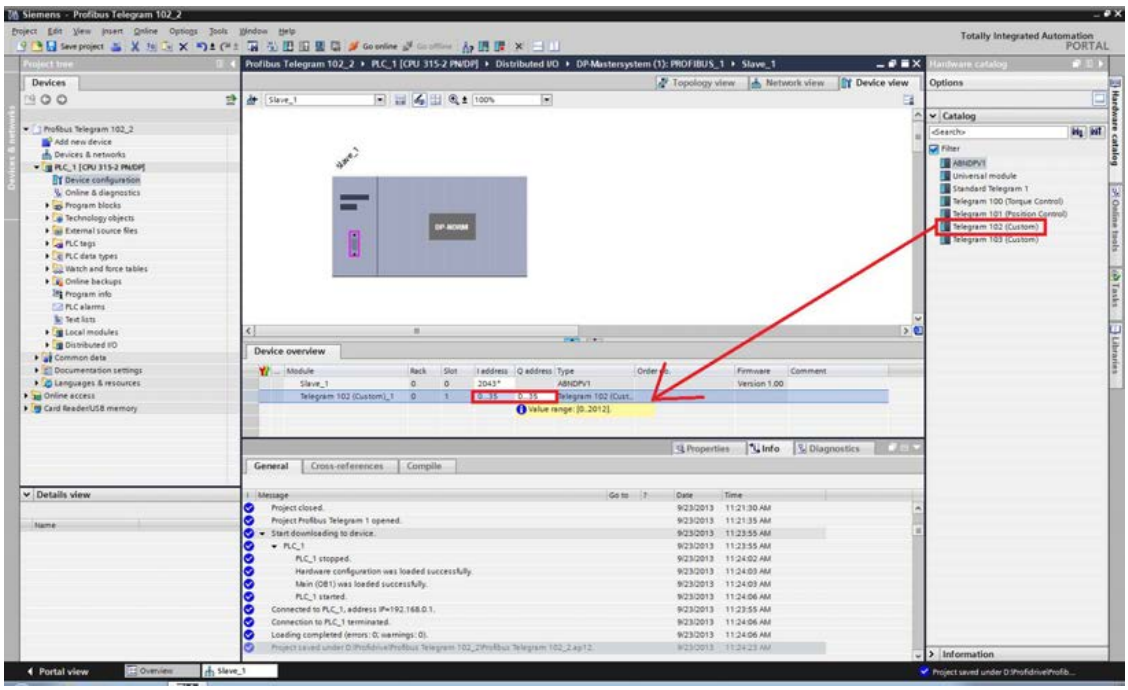
3.14 TIA Portal Telegram 102 example

This chapter describes running the inverter through the A8NDPV1 Profibus option card using Siemens TIA. First perform the setup using telegram 1.

① Enter the device view of the PROFIBUS option card slave and delete Standard Telegram 1 from the device overview.



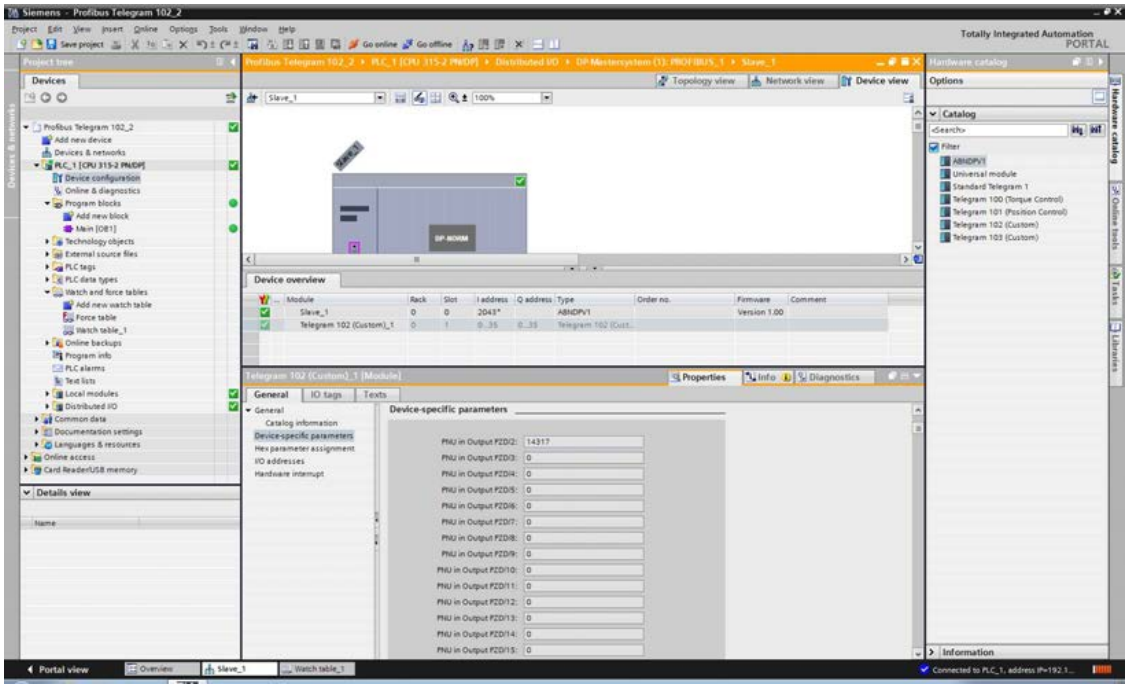
② Now drag Telegram 102 (Custom) from the Hardware Catalog to the empty slot. Change the starting I and Q address to 0.

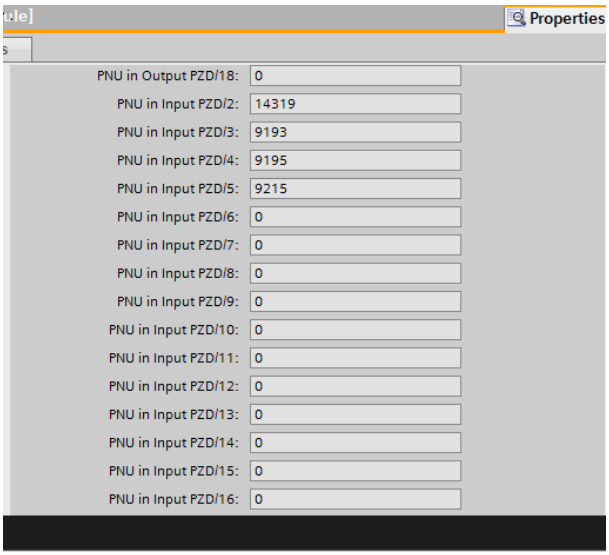


③ After double clicking on the telegram, setup the input and outputs of the telegram. There are 18 inputs and outputs used in this telegram, one of each is reserved for control/status word (leaving 17 inputs/outputs free for custom setup).

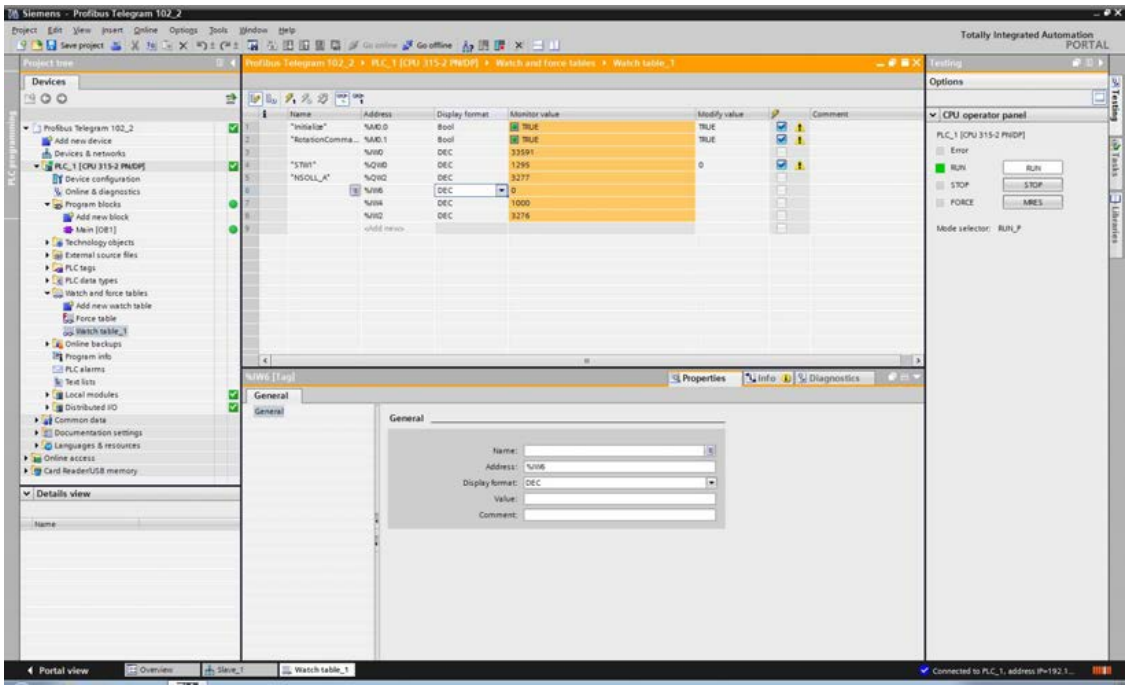
The setup of the telegram is done by entering signal/parameter numbers into the right column. All monitor data can be viewed as input, and some of the profile parameters as input/output. Please keep in mind, that not all parameters can be accessed via cyclic communication. The PROFIdrive parameter numbers (PNUs) available for use are listed in the Data Exchange subchapters 7.1to 7.4. An example setup is provided below:

Slot	Description	PNU
Output PZD/2	Speed set point	14317
Input PZD/2	Actual speed set point	14319
Input PZD/3	Output frequency	9193
Input PZD/4	Output voltage	9195
Input PZD/5	Actual operation time	9215





④ Compile the project and download it to the PLC. The additional monitor data can be viewed in the appropriate IW address.



3.15 TIA Portal Acyclic communication example

This chapter describes using acyclic communication through the A8NDPV1 Profibus option card using TIA Portal. It contains examples of reading and writing individual parameters. First perform the setup using telegram 1.

The process of acyclic communication parameter read consists of a write request, and a response read, performed using instructions *WRREC* and *RDREC*. For details please refer to the Acyclic Data Exchange subchapter 7.7.

3.15.1 Reading a parameter (Sequence 1)

Preparing structure of write request, and read response

Start by preparing the structure for the write request. The data request has the following format:

Byte no.	Description	Value
0	Request ID	0x01
1	Request reference	0x01-0xFF
2	No. of parameters	0x01
3	DO-ID	0x01
4	No. of elements	0x00
5	Attribute value	0x10 (value attribute)
6-7	Parameter number	Byte swapped PNU

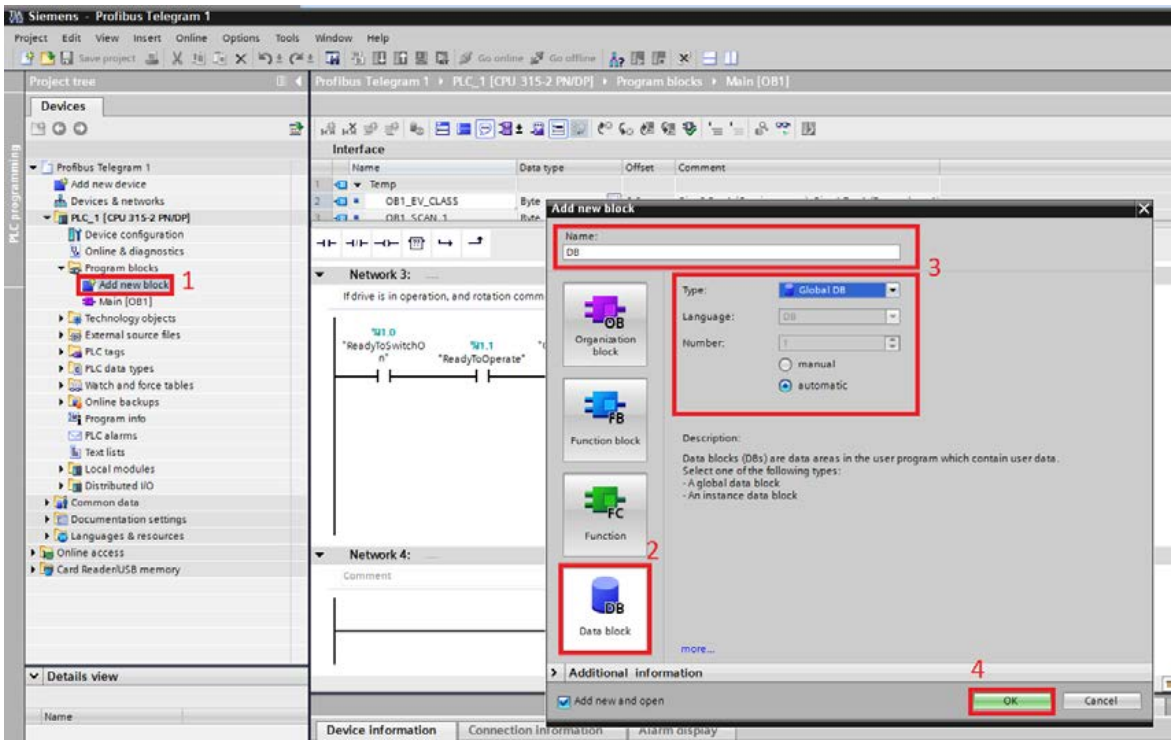
Request reference is any valid number, this value is mirrored back in the response and can be used to distinguish multiple requests. Once this request is processed successfully, a read request can be issued. The response of this request will contain the actual parameter value. The format of a positive response is as follows:

Byte no.	Description	Value
0	Request ID	0x01
1	Request reference	0x01-0xFF (same as in request)
2	No. of parameters	0x01
3	DO-ID	0x01
4	No. of values	0x01
5	Format	See data format type table
6-7	Parameter value	Byte swapped parameter value

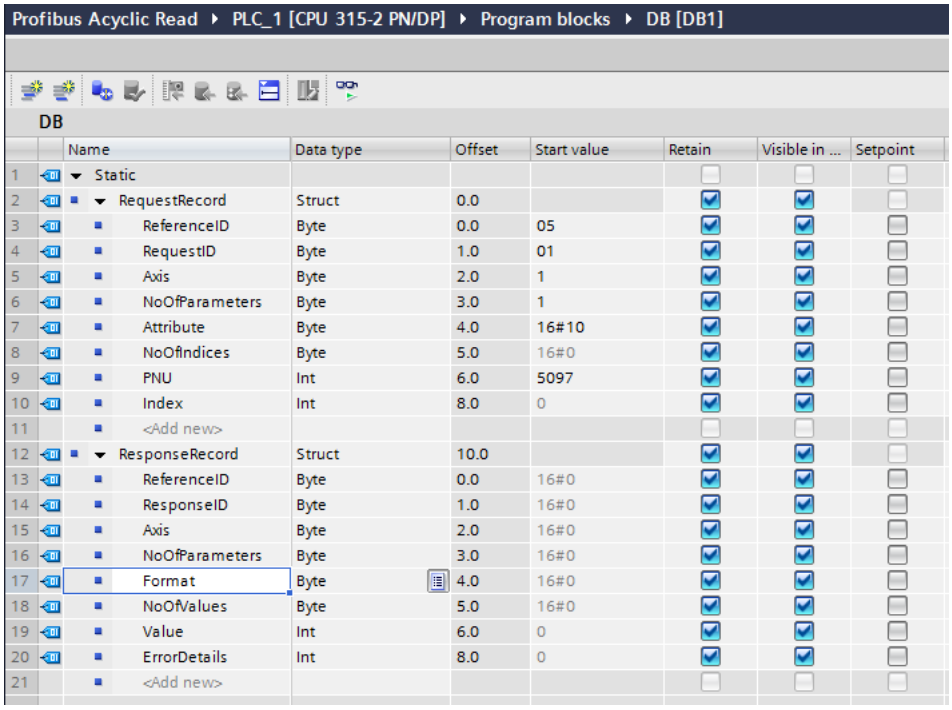
In case of a negative response, the format is:

Byte no.	Description	Value
0	Request ID	0x81
1	Request reference	0x01-0xFF (same as in request)
2	No. of parameters	0x01
3	DO-ID	0x01
4	No. of values	0x01
5	Format	0x44 (error)
6-7	Error value	Byte swapped error value (see error table)

The content of the data request will be contained in a data block structure. Expand the PLC in the project tree and add a new block.



The data block will be added to the program blocks with the specified name. Add two structures (*RequestRecord* and *ResponseRecord)* to the data block. Create the structure of the records according to the specification, like on the screenshot below. Enter the start value for the variables so that the fields don't need to be initialized in the user program.



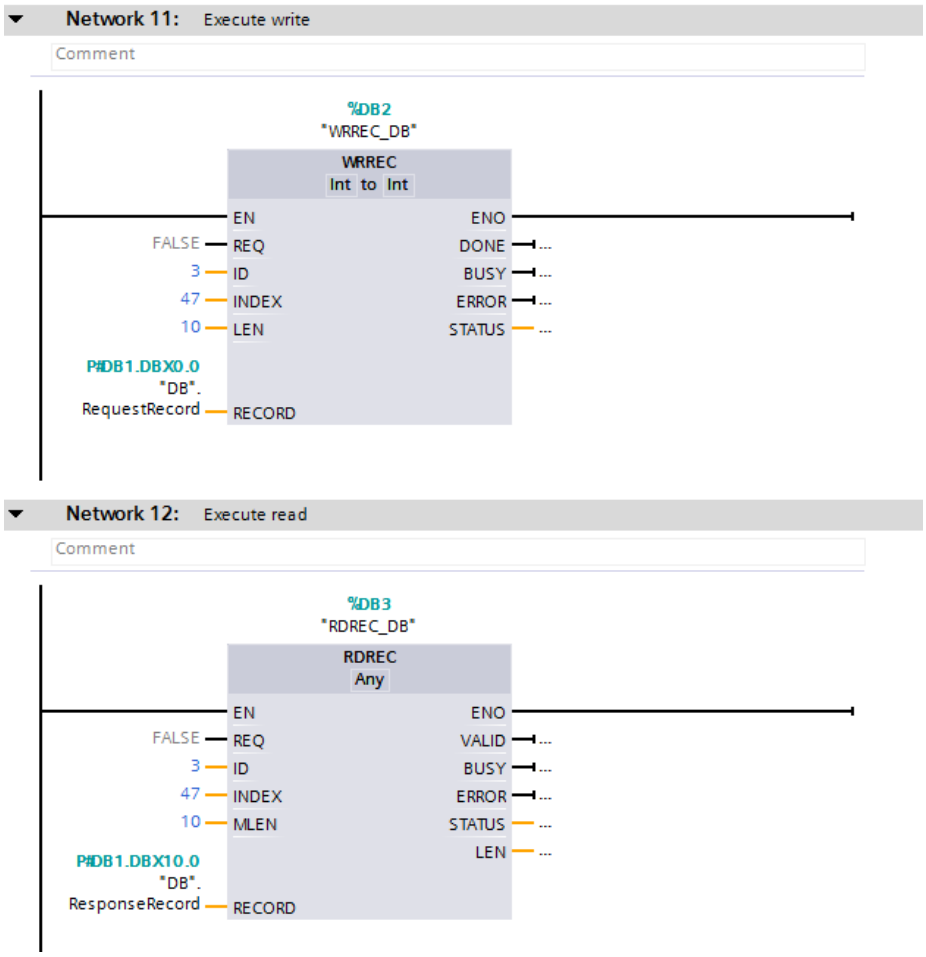
	Name	Data type	Offset	Start value	Retain	Visible in ...	Setpoint
1	Static						
2	RequestRecord	Struct	0.0		☑	☑	
3	ReferenceID	Byte	0.0	05	☑	☑	
4	RequestID	Byte	1.0	01	☑	☑	
5	Axis	Byte	2.0	1	☑	☑	
6	NoOfParameters	Byte	3.0	1	☑	☑	
7	Attribute	Byte	4.0	16#10	☑	☑	
8	NoOfIndices	Byte	5.0	16#0	☑	☑	
9	PNU	Int	6.0	5097	☑	☑	
10	Index	Int	8.0	0	☑	☑	
11	<Add new>						
12	ResponseRecord	Struct	10.0		☑	☑	
13	ReferenceID	Byte	0.0	16#0	☑	☑	
14	ResponseID	Byte	1.0	16#0	☑	☑	
15	Axis	Byte	2.0	16#0	☑	☑	
16	NoOfParameters	Byte	3.0	16#0	☑	☑	
17	Format	Byte	4.0	16#0	☑	☑	
18	NoOfValues	Byte	5.0	16#0	☑	☑	
19	Value	Int	6.0	0	☑	☑	
20	ErrorDetails	Int	8.0	0	☑	☑	
21	<Add new>						

Creating the request

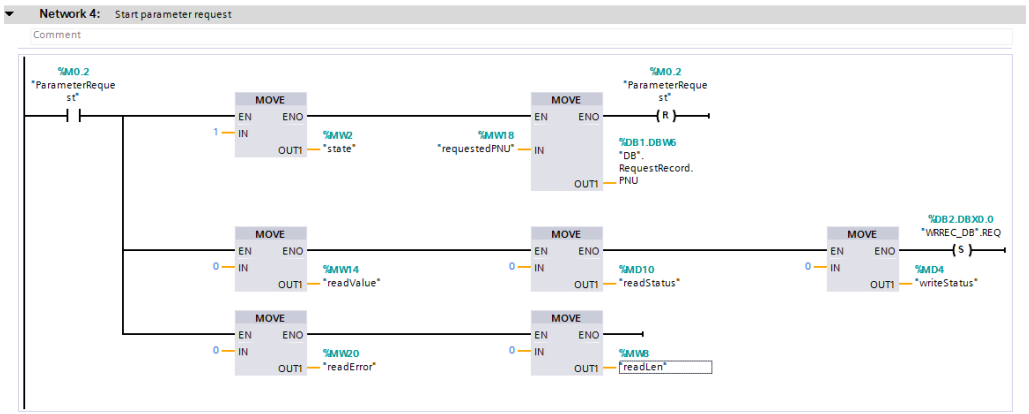
Proceed to create the user program for reading the parameter value. A state machine architecture is used for the process of creating, receiving and processing acyclic communication. The current state of the state machine is held in PLC tag *state*. A description of each state used in this example is provided below:

State	Description
1	Send write request, wait for BUSY = 0
2	Write request sent, read results
3	Write request sent successfully, send read request
4	Wait for BUSY = 0 (Read request)
5	Read request sent, read results
6	Results read
7	Successfully read requested PNU value
10	Write request error
11	Read request error
12	Parameter read error (negative response)

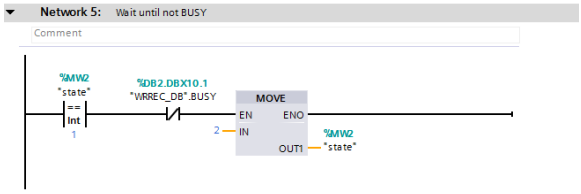
Start by adding a new network that will contain the instructions to issue the actual write and read requests.



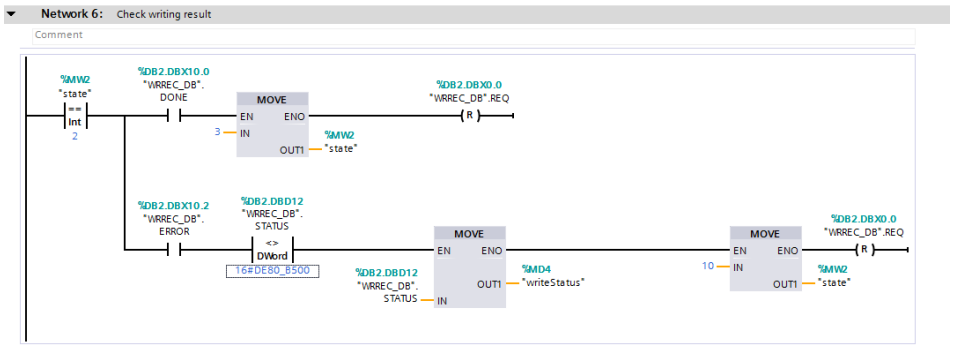
It will now be possible to reference the inputs and outputs of the function block instances, making it easier to create the program. Proceed to prepare the sequence. Add a new network in the user program that will start the state machine sequence. Setting *ParameterRequest* will start the state machine from state 1, initialize all used variables (*readStatus, readValue, writeStatus)*, copy the requested PNU from tag *requestedPNU* to the data structure, and trigger the write request (by setting the *REQ* input of the *WRREC* function block).



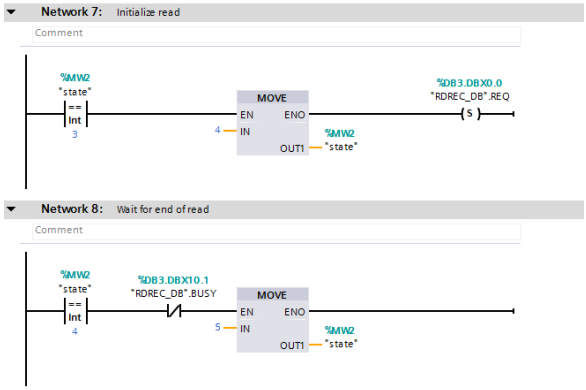
The *BUSY* output of *WRREC* will immediately turn ON. Once it turns OFF, proceed to reading the result of the write request (state 2).



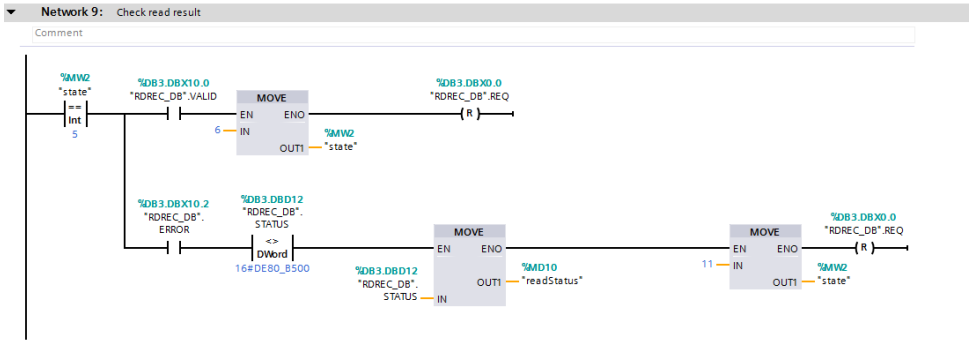
Expect to see either output *DONE* or *ERROR* of *WRREC* in an ON state. If *DONE* is ON, the write request was performed successfully, reset the *REQ* input, and proceed to creating the read request (state 3). If the *ERROR* output is ON, check the *STATUS* output to read the error. If the status is equal to 0xDE80B500, the result is not yet ready. In this case, do nothing and wait for the actual result. In every other case, save the status to tag *writeStatus* and enter state 10, indicating that a write request error occurred.



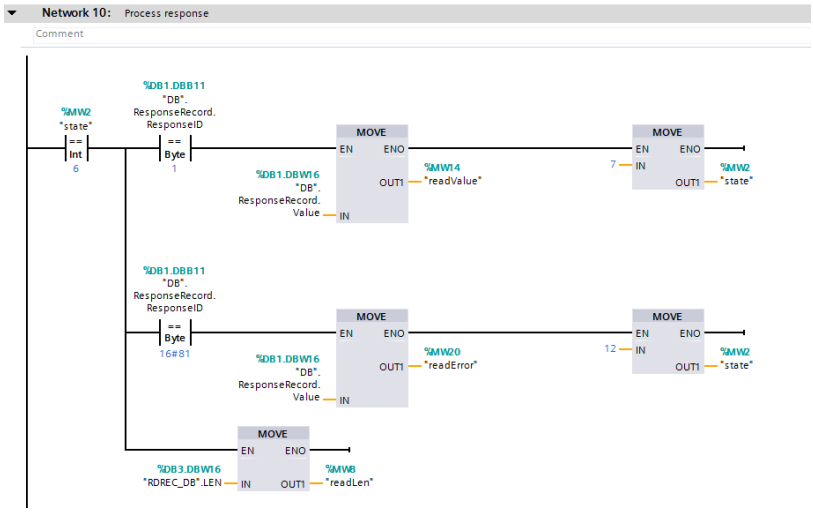
To initiate the read request, set input *REQ* of function block *RDREC*. Enter state 4, and wait for the *BUSY* output to turn off. Once this condition is met, enter state 5 to check the results of the read request.



Expect to see either output *DONE* or *ERROR* of *RDREC* in an ON state. If *DONE* is ON, the read request was performed successfully, reset the *REQ* input, enter state 6 to process the response. If the *ERROR* output is ON, check the *STATUS* output to read the error. If the status is equal to 0xDF80B500, the result is not yet ready. In this case, do nothing and wait for the actual result. In every other case, save the status to tag *readStatus* and enter state 11, indicating that a read request error occurred.

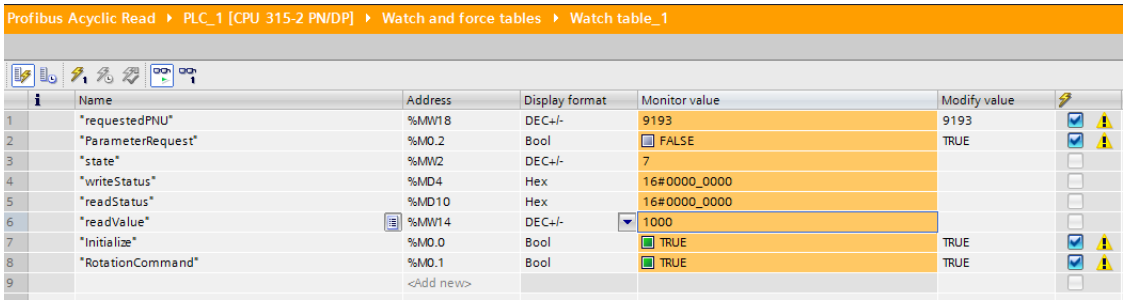


Finally process the received response. Check the *ResponseID* field of the *ResponseRecord*. A value of 1 indicates that the request was performed successfully. In this case move the *Value* field of this structure to the *readValue* tag. Enter state 7 to indicate a successful parameter read. If the *responseID* is equal to 0x81, then a negative response was received. Save the error value (also the *Value* field of the structure) to the *errorValue* tag. Enter state 12 to signal a negative response.



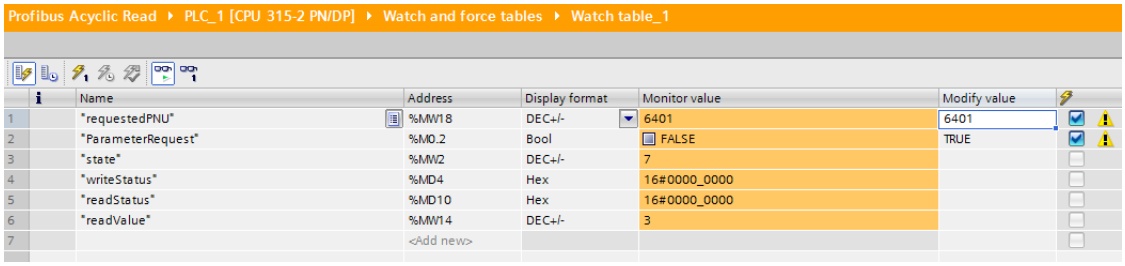
Executing example requests

After compiling the program, and writing it to the PLC, it is possible to execute parameter read requests. First, get the inverter running by setting bits M0 and M1. Now read the actual frequency by writing PNU 9193 to *requestPNU* and setting bit *ParameterRequest*. The result should be a value of 1000 in *readValue,* which corresponds to 10Hz. The PROFIdrive parameter numbers (PNUs) available for use are listed in the Data Exchange subchapters 7.1to 7.4.



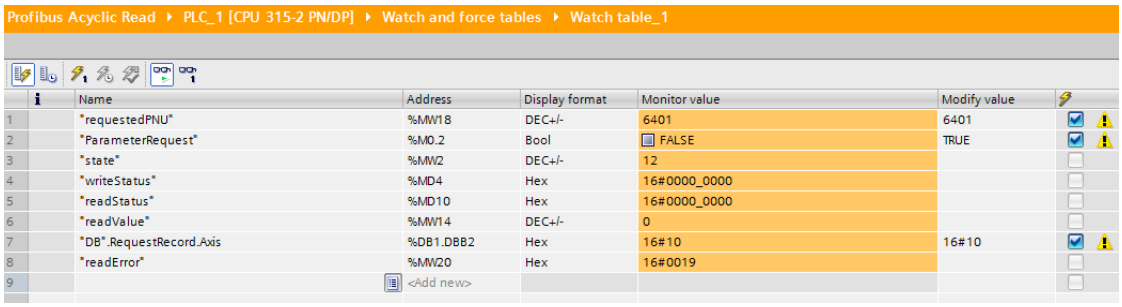
	Name	Address	Display format	Monitor value	Modify value
1	"requestedPNU"	%MW18	DEC+/-	9193	9193
2	"ParameterRequest"	%M0.2	Bool	FALSE	TRUE
3	"state"	%MW2	DEC+/-	7	
4	"writeStatus"	%MD4	Hex	16#0000_0000	
5	"readStatus"	%MD10	Hex	16#0000_0000	
6	"readValue"	%MW14	DEC+/-	1000	
7	"Initialize"	%M0.0	Bool	TRUE	TRUE
8	"RotationCommand"	%M0.1	Bool	TRUE	TRUE
9		<Add new>			

Try reading an inverter parameter, for example the FDL address of the inverter (parameter 1305 + 5096 offset = PNU 6401). The expected result is 1. It is possible to read any inverter parameter in this way.



	Name	Address	Display format	Monitor value	Modify value
1	"requestedPNU"	%MW18	DEC+/-	6401	6401
2	"ParameterRequest"	%M0.2	Bool	FALSE	TRUE
3	"state"	%MW2	DEC+/-	7	
4	"writeStatus"	%MD4	Hex	16#0000_0000	
5	"readStatus"	%MD10	Hex	16#0000_0000	
6	"readValue"	%MW14	DEC+/-	3	
7		<Add new>			

Try testing a negative response by changing the axis number to 0x10. Notice, that after executing the sequence, state 12 is active. An error code of 0x19 (AXIS_DO_NONEXISTENT) is expected.



	Name	Address	Display format	Monitor value	Modify value
1	"requestedPNU"	%MW18	DEC+/-	6401	6401
2	"ParameterRequest"	%M0.2	Bool	FALSE	TRUE
3	"state"	%MW2	DEC+/-	12	
4	"writeStatus"	%MD4	Hex	16#0000_0000	
5	"readStatus"	%MD10	Hex	16#0000_0000	
6	"readValue"	%MW14	DEC+/-	0	
7	"DB".RequestRecord.Axis	%DB1.DBB2	Hex	16#10	16#10
8	"readError"	%MW20	Hex	16#0019	
9			<Add new>		

3.15.2 Changing parameters (Sequence 2)

This example will be based off of the previous example (Reading a parameter – Sequence 1), as the changes needed are minimal.

Preparing structure of write request, and read response

Start by preparing the structure for the write request. The data request has the following format:

Byte no.	Description	Value
0	Request ID	0x02
1	Request reference	0x01-0xFF
2	No. of parameters	0x01
3	DO-ID	0x01
4	No. of elements	0x00
5	Attribute value	0x10 (value attribute)
6-7	Parameter number	PNU to modify
8-9	Subindex (irrelevant)	0x00
10	No. of values	0x01
11	Format	0x06 (UINTEGER16 for all inverter parameters) See data format type table
12-13	Set value	PNU value to write

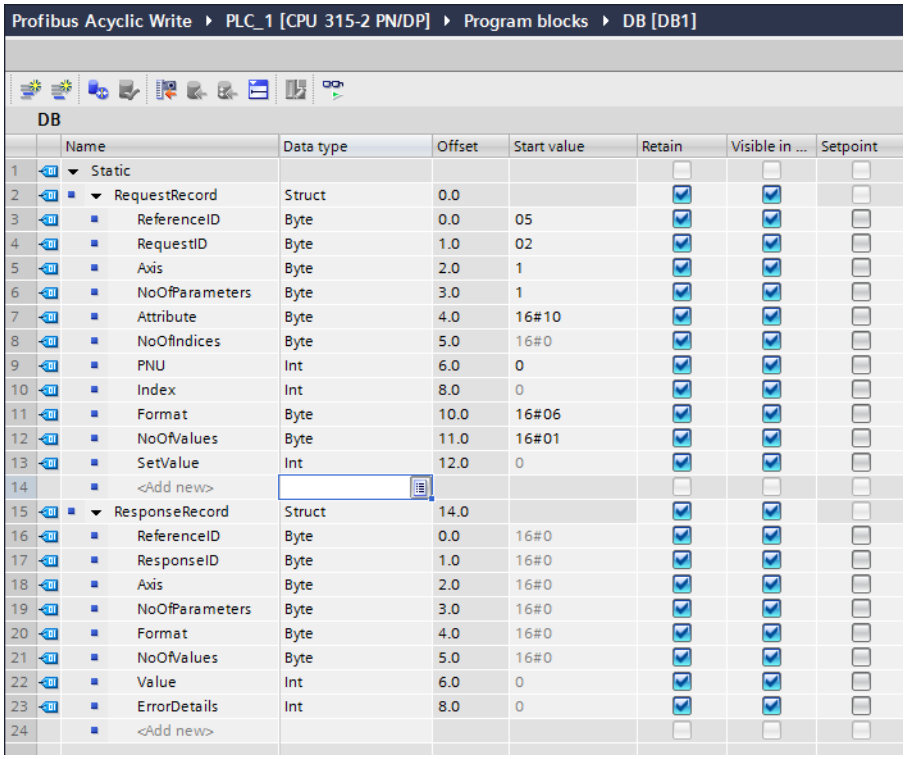
Once this request is processed successfully, a read request can be issued. The response of this request will determine whether the parameter write was successful. The format of a positive response is as follows:

Byte no.	Description	Value
0	Request ID	0x02
1	Request reference	0x01-0xFF (same as in request)
2	No. of parameters	0x01
3	DO-ID	0x01

In case of a negative response, the format is as follows:

Byte no.	Description	Value
0	Request ID	0x82
1	Request reference	0x01-0xFF (same as in request)
2	No. of parameters	0x01
3	DO-ID	0x01
4	No. of values	0x01
5	Format	0x44 (error)
6-7	Error value	Error value (see error table)

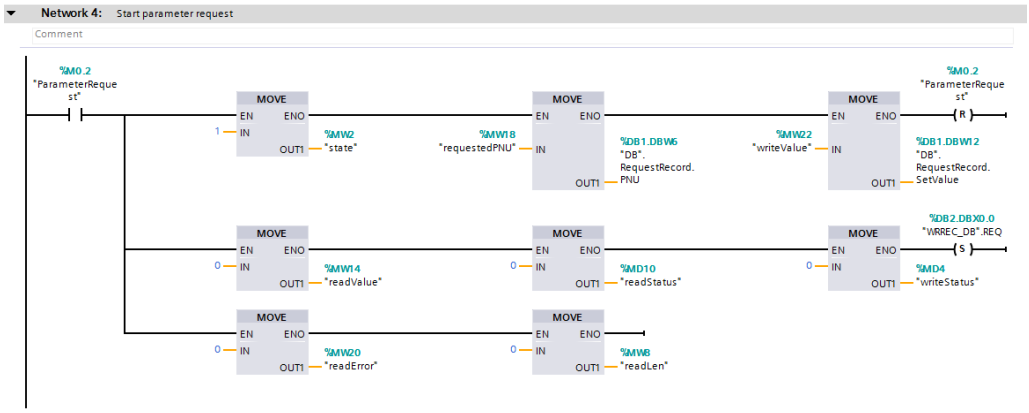
Modify the data structures created in the previous example to account for the changes (note the changes in start value fields):



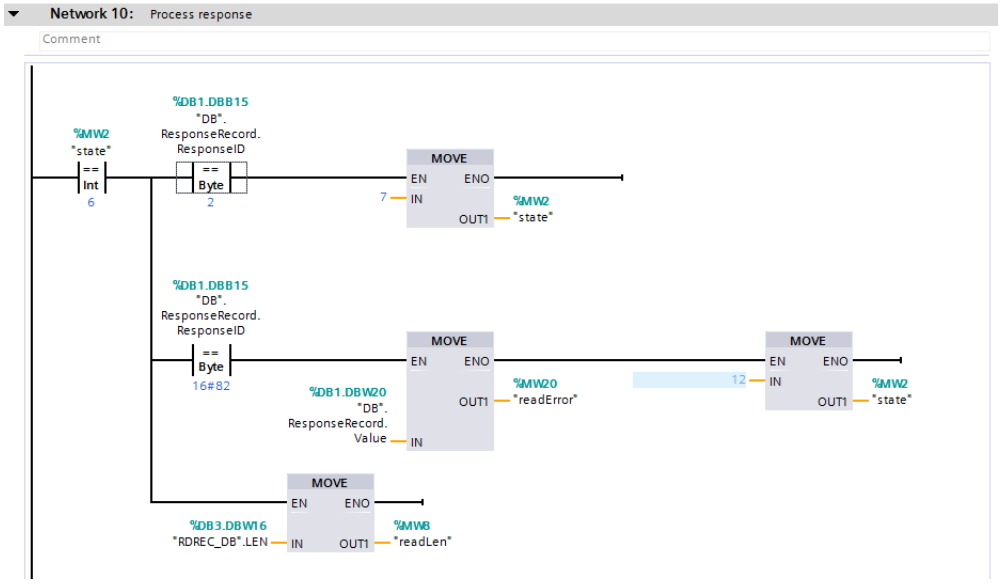
	Name	Data type	Offset	Start value	Retain	Visible in ...	Setpoint
1	Static				☐	☐	☐
2	RequestRecord	Struct	0.0		☑	☑	☐
3	ReferenceID	Byte	0.0	05	☑	☑	☐
4	RequestID	Byte	1.0	02	☑	☑	☐
5	Axis	Byte	2.0	1	☑	☑	☐
6	NoOfParameters	Byte	3.0	1	☑	☑	☐
7	Attribute	Byte	4.0	16#10	☑	☑	☐
8	NoOfIndices	Byte	5.0	16#0	☑	☑	☐
9	PNU	Int	6.0	0	☑	☑	☐
10	Index	Int	8.0	0	☑	☑	☐
11	Format	Byte	10.0	16#06	☑	☑	☐
12	NoOfValues	Byte	11.0	16#01	☑	☑	☐
13	SetValue	Int	12.0	0	☑	☑	☐
14	<Add new>				☐	☐	☐
15	ResponseRecord	Struct	14.0		☑	☑	☐
16	ReferenceID	Byte	0.0	16#0	☑	☑	☐
17	ResponseID	Byte	1.0	16#0	☑	☑	☐
18	Axis	Byte	2.0	16#0	☑	☑	☐
19	NoOfParameters	Byte	3.0	16#0	☑	☑	☐
20	Format	Byte	4.0	16#0	☑	☑	☐
21	NoOfValues	Byte	5.0	16#0	☑	☑	☐
22	Value	Int	6.0	0	☑	☑	☐
23	ErrorDetails	Int	8.0	0	☑	☑	☐
24	<Add new>				☐	☐	☐

Creating the request

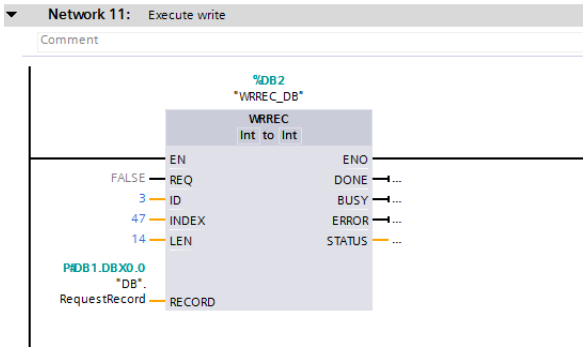
Modify the ladder network initiating the sequence, to set the value of field *SetValue* in the *RequestRecord* structure. This is the value that will be written to the requested PNU.



Modify the network responsible for processing of the read request response. A positive response will have ID 0x02, while a negative response ID 0x82.

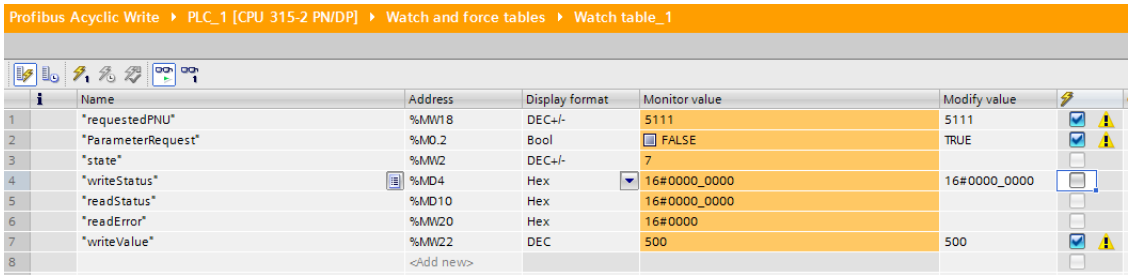


Finally change the length of the write request to 14 bytes.



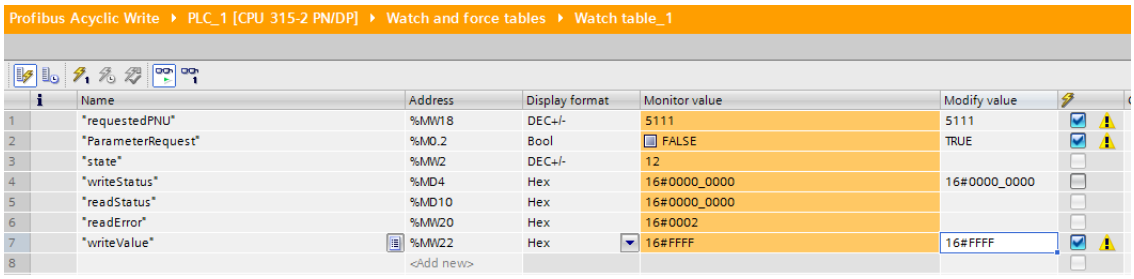
Executing example requests

To perform a parameter write, set the PNU of the parameter to change in *requestedPNU*. As an example, to change parameter 15 (Jog frequency), add the offset 5096 and write the result (5111) to variable *requestedPNU*. The value that will be set, will be the output of multiplication of the set value, and the minimum setting increments, as according to the parameter list in the inverter manual. For parameter 15, the minimum setting increment is 0.01Hz. If a jog frequency of 5Hz is requested, set *writeValue* to 500. Finally execute the request by setting bit *ParameterRequest* to ON. A successful write will result in entering state 7.



	Name	Address	Display format	Monitor value	Modify value
1	"requestedPNU"	%MW18	DEC+/-	5111	5111
2	"ParameterRequest"	%M0.2	Bool	FALSE	TRUE
3	"state"	%MW2	DEC+/-	7	
4	"writeStatus"	%MD4	Hex	16#0000_0000	16#0000_0000
5	"readStatus"	%MD10	Hex	16#0000_0000	
6	"readError"	%MW20	Hex	16#0000	
7	"writeValue"	%MW22	DEC	500	500
8		<Add new>			

Test receiving a negative response, by trying to write an out of range value. After trying to write 0xFFFF as the value of parameter 15, state 12 will be entered indicating a negative response with error code 0x02 (LOW_OR_HIGH_LIMIT_EXCEEDED) stored in *readError* tag, which is the expected behavior.



	Name	Address	Display format	Monitor value	Modify value
1	"requestedPNU"	%MW18	DEC+/-	5111	5111
2	"ParameterRequest"	%M0.2	Bool	FALSE	TRUE
3	"state"	%MW2	DEC+/-	12	
4	"writeStatus"	%MD4	Hex	16#0000_0000	16#0000_0000
5	"readStatus"	%MD10	Hex	16#0000_0000	
6	"readError"	%MW20	Hex	16#0002	
7	"writeValue"	%MW22	Hex	16#FFFF	16#FFFF
8		<Add new>			

3.16 SIMATIC STEP7 example

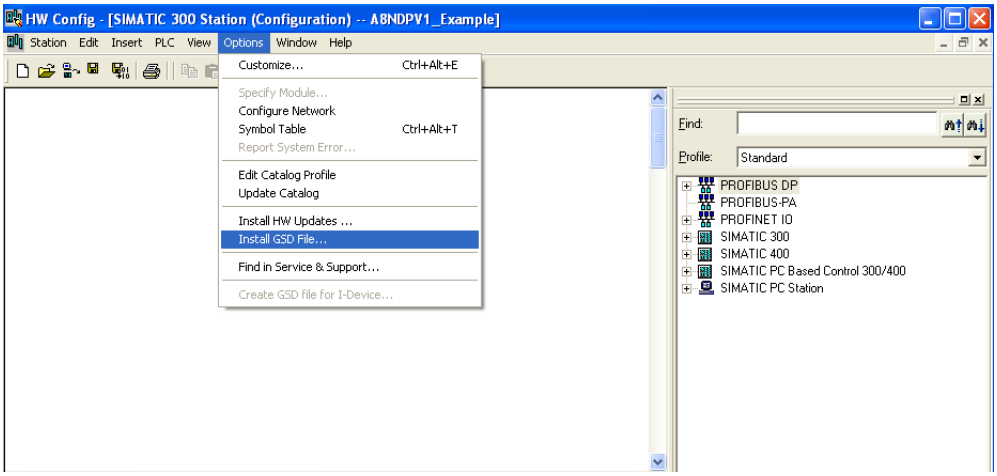
3.16.1 Creating a Configuration

This section describes a simple example on how to configure a PROFIBUS network including an A8NDPV1 PROFIBUS Option Board using the SIMATIC STEP7 configuration tool. Please consult the SIMATIC STEP7 for further information on how to use the tool.

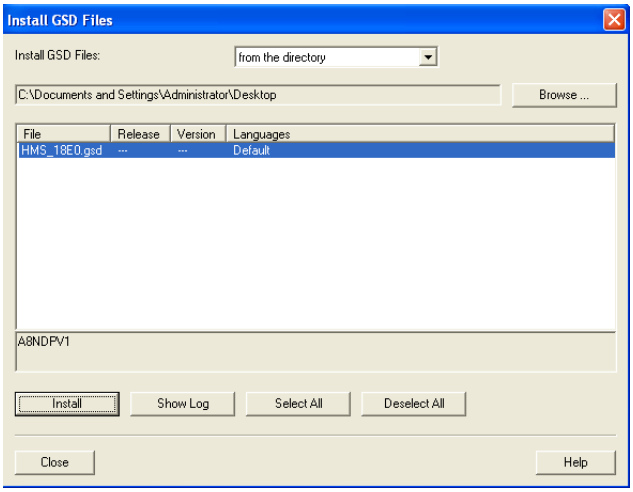
The configuration is created offline and downloaded to the network when it is finished.

① Open SIMATIC Step7.

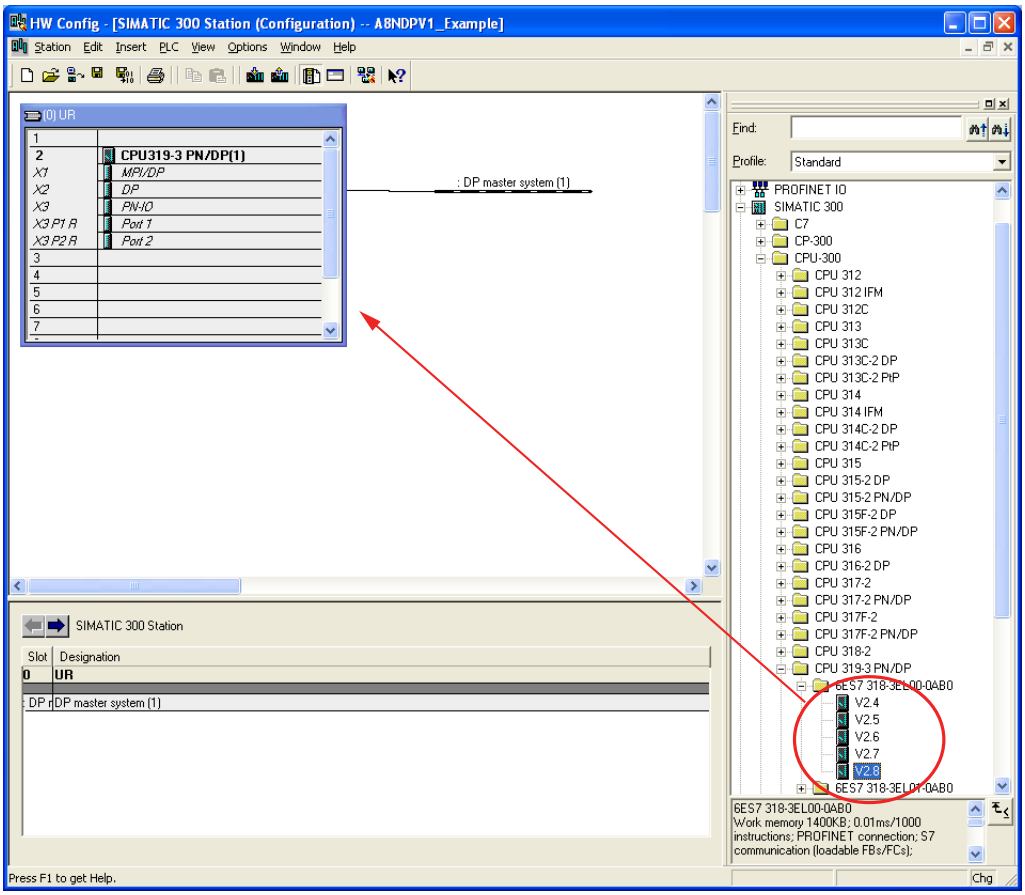
② Select Options - Install GSD.



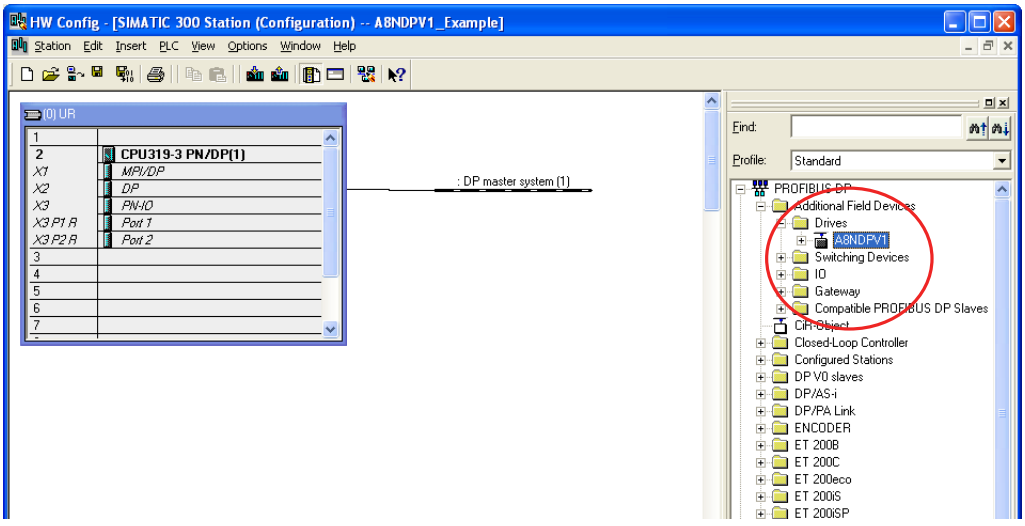
③ Browse for the correct GSD-file to use with the option board. Select file and click Install.



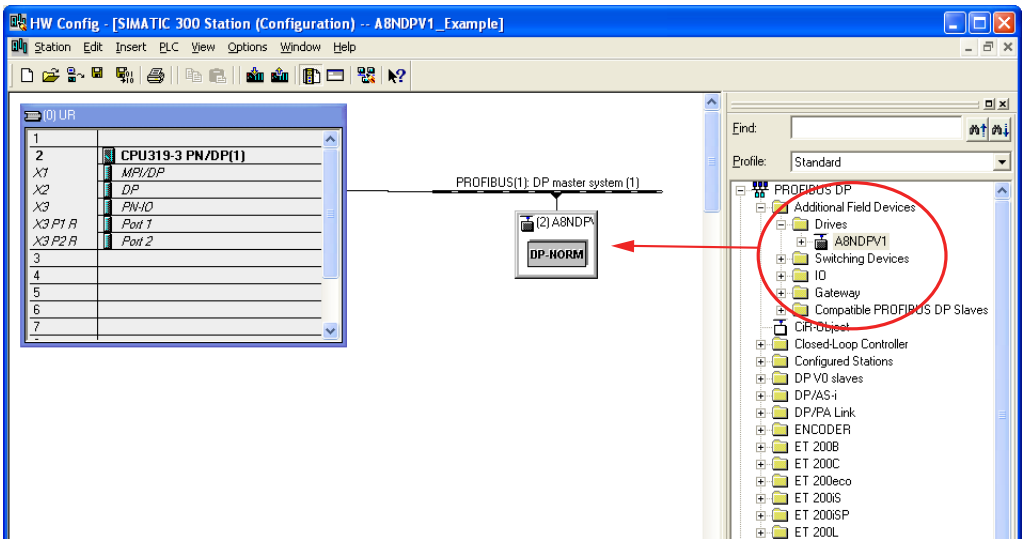
④ Select PLC and drag an instance of the selected PLC into the configuration window.



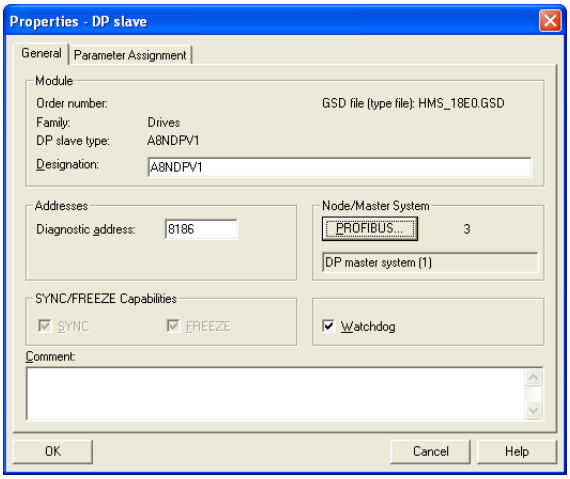
⑤ Locate the option board in the HW catalog (PROFIBUS DP -> Additional Field Devices-Drives).

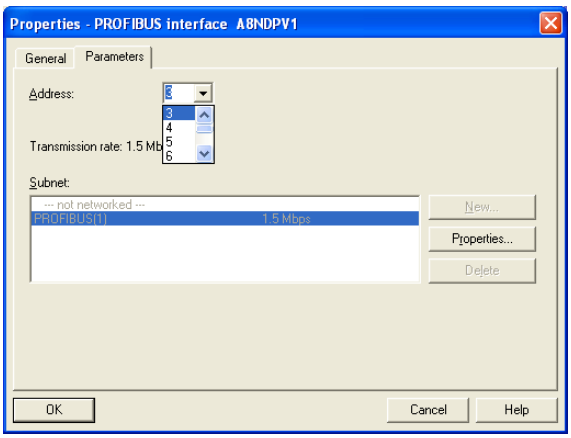


⑥ Drag and drop an instance of the option board to the configuration.

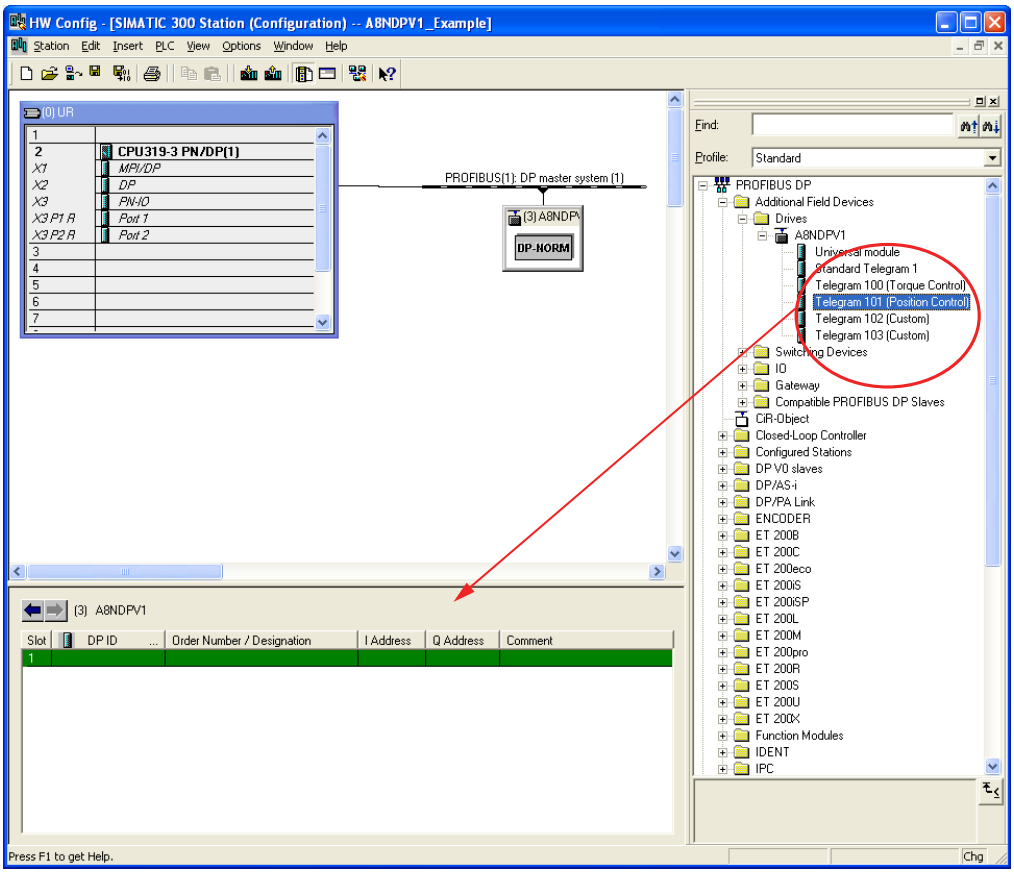


⑦ With the option board selected, right click on it and open Object Properties from the menu in the pop-up window. Select the General tab. Click the PROFIBUS button and then enter the node address under the Parameters tab.

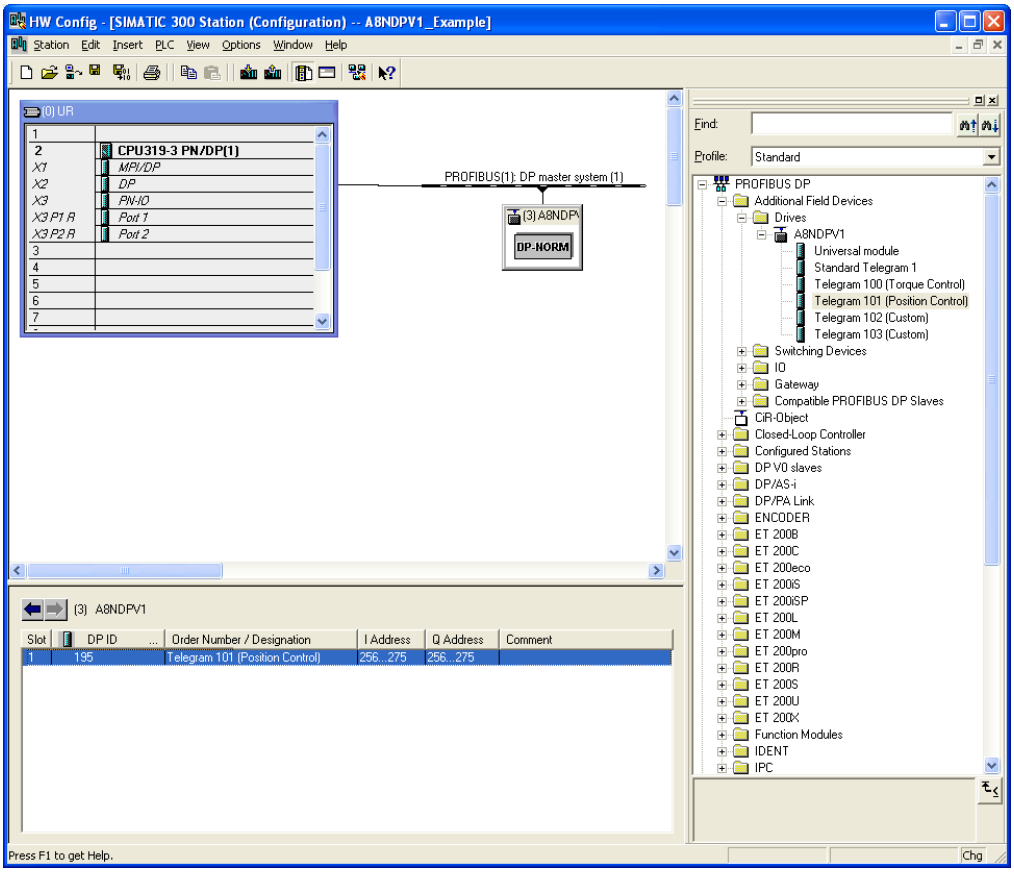


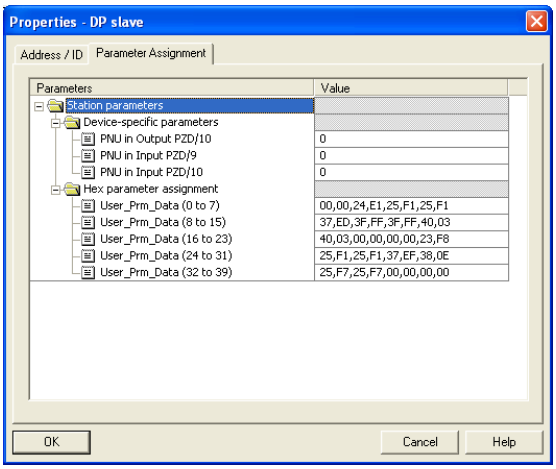


⑧ Select and drag and drop telegram to option board. The telegrams are found in the HW catalog with the option board.



⑨ With the new telegram instance selected, right click and open the telegram Properties window from the pop up menu. Enter a name for the telegram and map the cyclic data in the telegram.





3.16.2 Download Configuration

The configuration of the PROFIBUS network can be built offline. To download the configuration to the physical network, follow the steps below:

① Power up network.

② Set node address for each individual module using the parameter unit.[1]

③ Download the configuration to the master. How this is done depends on which configuration tool is used.

④ If errors occur during the connection phase, scan the network and use the live list to find the slaves. Make sure that the node addresses are correct, otherwise they can be changed to match the configuration.

3.16.3 Run

Set the PLC in RUN mode.

3.17 Parameter Settings

Parameters can be set at runtime using the hand-held parameter unit or the setup software MELSOFT FR-Configurator[2]. The values of some of parameters will not be valid until the module has been restarted.

For more information, please consult the manual for the FR-A800 series frequency inverter.

[1] Node addresses can also be set from a master or a configuration tool. See the manual or online help for the configuration tool you are using. The PNU ident number of the A8NDPV1 PROFIBUS Option Board is 18E0h.

[2] See the product pages at https://eu3a.mitsubishielectric.com.

4. Inverter Settings

4.1 Inverter Parameters

The inverter parameters are critical for overall operation of the end-to-end communication system. Some of these parameters must be set to specific values, and some may have multiple allowable settings depending on the desired operation of the overall application. It is important to understand the manner in which the parameters will impact successful communications with, and control of the inverter.

The inverter parameters can be changed via the handheld parameter unit or using the setup software MELSOFT FR-Configurator[1]. The handheld parameter unit is described in the user manual for the FR-A800 series.

4.2 Option Board Parameters

The option board parameters are stored in the inverter. At startup they are transferred from the inverter to the option board and can be changed using the parameter unit and the setup software MELSOFT FR-Configurator[1].

The option board parameters are stored in the main unit. At startup they are transferred from the inverter to the option board.

No (dec)	No (hex)	Parameter Name.	Description
1300	514h	Option parameter 1	General settings, see "Parameter 1300 (514h, General Settings)" on page 118
1301	515h	Option parameter 2	N/A
1302	516h	Option parameter 3	Network Type[1]
1303	517h	Option parameter 4	Serial number (low word)[2]
1304	518h	Option parameter 5	Serial number (high word)[2]
1305	519h	Option parameter 6	Network specific 1, see "Parameter 1305 (519h, Device address, node address)" on page 118
1306	51Ah	Option parameter 7	Network specific 2, see "Parameter 1306 (51Ah, Actual device address, node address)" on page 118
1307	51Bh	Option parameter 8	Network specific 3, see "Parameter 1307 (51Bh, Option board settings priority)" on page 119
1308 - 1343	51Ch-53Fh	Option parameter 9 - 44	N/A

[1] During startup the option board will verify value of this parameter. If it differs from the network type of the option board, the option board will change the parameter value to the actual network type and clear all other parameters.

[2] During startup the option board will verify the value of this parameter. If it differs from the serial number of the option board, the option will board will change the parameter value to the actual serial number and also set the option board host name to [main-unit-name]-[serial number]

The option board and the inverter start up with default standard settings. Any changes of the parameter values during runtime, will not be available until the inverter has been restarted.

The application can more often than not be run on standard settings.

[1] See http://eu3a.mitsubishielectric.com/fa/en/products/drv/inv/fr_configurator/

Parameter 1300 (514h, General Settings)

Only read by the option board during startup.

Bit	Name	Default	Description
0 - 3	(reserved)	0	N/A
4	Clear all Option Parameters	0	Will clear all option parameters and set the default values when set to 1. Resets itself to False (0) after completion.
5 - 15	(reserved)	0	N/A

Parameter 1305 (519h, Device address, node address)

Only read by the option board during startup. The lower byte can be written by using the hand-held parameter unit.

Note: For these settings to be used, parameter 1307 (51Bh) has to be set to 1.

Bit	Name	Default	Description
0 - 7	Configured device address[1]	-	This device address (node address) is set via the parameter unit. The module must have a unique node address (device address) in order to be able to communicate on the PROFIBUS network. Value — Description 0 - 125 — Values in this range are considered as a local address setting by the inverter (set by the parameter unit e.g.) and will be used as the new node address after restart. Set Slave Address (SSA)[2] functionality is disabled. 126 - 254 — Use previous valid setting received from network. Any value in this span will default to 126. Option Parameter 7 (parameter 51Ah) reflects the current node address setting. 255 — A factory default reset is performed. The device will default to node address 126 and wait for an SSA[2] command. Option Parameter 7 (parameter 51Ah) reflects the current node address setting.
8 - 15	(reserved)	0	N/A

[1] Parameter 51Ah (option board parameter 7) reflects the current setting.

[2] For more information, see “Set Slave Address” on page 121.

When read by the parameter unit, an address in the span 0-125 indicates that a node address has been set by the unit. If the value is 126, address setting has been managed via SSA functionality, see “Set Slave Address” on page 121.

Parameter 1306 (51Ah, Actual device address, node address)

Only read from the option board during startup. The parameter cannot be changed at runtime.

Bit	Name	Default	Description
0 - 7	Actual device address	-	This device address reflects the current setting. Valid settings range from 0 - 126.
8 - 15	(reserved)	0	N/A

Parameter 1307 (51Bh, Option board settings priority)

Note:	If the settings of parameter 1305 are to be used by the option card, this parameter must be set to 1.

Bit	Name	Default	Description
0 - 7	Communication settings	-	This parameter decides the priority of setting.Valid settings range from 0 - 1. Value Description 0 Option board settings are used for network settings. 1 Option parameters are used for network settings.
8 - 15	(reserved)	0	N/A

4.3 Operation Mode Setting

Please refer to the FR-A800 instruction manual for information.

5. Identifying Option Board

A device on a PROFIBUS network is defined by its node address. Valid settings range from 0…125. The node address for an A8NDPV1 PROFIBUS Option Board is set using the parameter held unit.

At startup, the master will address the option board using this node address.

5.1 Set Slave Address

The option board supports SSA (Set Slave Address), which allows a PROFIBUS network master or a configuration tool to set the node address from the network. Thus it is possible to scan for slaves via the live list, that is available, and also change the node addresses.

This service features a flag, referred to as "No Add Change", which specifies whether or not it is allowed to change the device address from the network again at a later stage. If the service is accepted, the module saves the value of this flag in non-volatile memory. The only way to restore it again is by entering device address 255 in option parameter 6 and setting option parameter 8 to 1, which will cause a factory default of the Option Board.

The module will accept new settings received via this service under the following conditions:

- The 'Configured device address'-attribute in option parameter 6 is set to a value higher than 125 or option parameter 8 is set to 0.
- The module is not in exchanging data.
- The module is addressed with the correct Ident Number.
- No previous 'Set Slave Address'-request with the flag “No Add Change” set, prevents the module from accepting the new settings

6. PROFIBUS DP-V1 Implementation

6.1 General

The A8NDPV1 PROFIBUS Option Board complies to PROFIBUS specification, IEC 61158 Type 3. Drive operation complies to PROFIdrive V4.1. Supports Application class 1 functionality.

Performance:

- Up to 12 Mbit, detected automatically by the module.

Device Model

- The communication relationship between the controller and master is represented by MS0 AR plus MS1 AR. Relationship between a DP-master class2 (Supervisor) is represented by MS2 AR.
- Cyclic communication uses the MS0 channel.
- Two different channels exists for acyclic communication the MS1 and MS2.
- Data access via DP-V1 read/write services (via slot/index)

Slots

One slot available for a selection of telegram types.

I/O Data

The amount of I/O data that can be transferred is determined by the main unit interface, or a maximum of 64 bytes of I/O data in each direction.

6.2 Electronic Data Sheet (GSD)

A standard GSD file is available for download, see “Download” on page 1.

6.3 DAP

DAP is implemented for backwards compatibility.

6.4 I&M

Identification & Maintenance (I&M) provides a standard way of gathering information about an I/O device. The I&M information can be accessed by the I/O Controller by means of acyclic Record Data Read/Write services.

The option board provides support for I&M 0-4 data.

7. Data Exchange

7.1 General Information

Inverter parameters can be accessed acyclically by means of the PROFIdrive Parameter Protocol. The master issues a parameter request to the option board. The option board processes the request. Meanwhile the master repeatedly tries to read the parameter response, but an error message will be returned until the request is fully processed by the module, after which the master can read the response.

Translation of signal numbers

Signal numbers used in the inverter do not directly translate to signal numbers (PNUs) used on PROFIBUS. An offset is added to avoid ambiguous numbering on PROFIBUS, where parameters and monitor data have different PNU numbers.

The array below presents a summary of the signal number translation. For a complete presentation of how to translate the signal numbers for the option board drive profile parameters, see "Translation of Signal Numbers" on page 145.

Signal Type	Signal No, Main Unit		Offset	PNU No, PROFIdrive (Signal No. + Offset)	Acyclic Data Exchange	Cyclic Data Exchange
	Hexadecimal	Decimal				
Parameter	0h…54Fh	0…1359d	5096d	5096d…6455d	Yes	No
Monitor Data	000h…3FFh	0d…1023d	9192d	9192d…10215d	Yes	Yes
Option Board Drive Profile Parameters	400h…411h	1024d…1041d	13288d	14312d…14329d	Yes	Yes/no[1]
	412h, 415h…41Ch[2]	1042d, 1045d…1052d	13288d[2]	14330d, 14333d…14340d	Yes	No
			15336d[2]	16378d, 16381d…16388d	Yes	Yes
	413h…414h	1043d…1044d	13288d	14331d…14332d	Yes	No
	41Dh…431h	1053d…1073d	13288d	14341d…14361d	Yes	Yes/no[1]

[1] Some of these parameters can be mapped for cyclic data exchange, see "Translation of Signal Numbers" on page 145.

[2] These parameters can either be accessed (as acyclic data) as an array, with offset 13288d, or can each entry in the array be mapped as a separate parameter with offset 15336d (as acyclic and/or cyclic data).

7.2 Inverter parameters (Acyclic Data Exchange)

Inverter parameters are available only via acyclic communication. They can**not** be used in cyclic data exchange, for example it is **not** possible to map them as parameters for telegram 102. The offset used for inverter parameters is 5096, i.e. inverter parameter no. 1 (maximum frequency) shall be addressed as PNU 5097. An example of reading writing inverter parameters is shown in "Reading a parameter (Sequence 1)" on page 34.

All of the inverter parameters are unsigned 16bit integers, with the exception of the following:

Parameter	PNU	Description	Data type
7	5103	Acceleration time	Unsigned 32 bit integer
8	5104	Deceleration time	
16	5112	Jog acceleration/deceleration time	
44	5140	Second acceleration/deceleration time	
110	5206	Third acceleration/deceleration time	
111	5207	Third deceleration time	
264	5360	Power-failure deceleration time 1	
265	5361	Power-failure deceleration time 2	
791	5887	Acceleration time in low-speed range	
792	5888	Deceleration time in low-speed range	
All other	-	All inverter parameters NOT listed in this table	Unsigned 16bit integer

7.3 Monitor Data (Acyclic and Cyclic Data Exchange)

This chapter contains a table of monitor data available via both cyclic and acyclic data exchange. The values are available as 16 and 32 bit integers. To calculate the actual monitor data value, the returned integer should be multiplied by the unit specified for that parameter. For example if PNU 9193 (output frequency) is requested and a value of 500 is returned, this means that the output frequency is equal to 5 Hz (500 × 0.01 Hz).

PNU No	Internal No	Description	Unit	Type	Access type
		16 bit			
9193	1	Output frequency	0.01Hz	unsigned	R
9194	2	Output current	0.01A/0.1A	unsigned	R
9195	3	Output voltage	0.1V	unsigned	R
9196	4	reserved	-	-	-
9197	5	Frequency setting value	0.01Hz	unsigned	R
9198	6	Motor speed	1 rpm	unsigned	R
9199	7	Motor torque	0.1%	unsigned	R
9200	8	Converter output voltage	0.1V	unsigned	R
9201	9	Regenerative brake duty	0.1%	unsigned	R
9202	10	Electric thermal relay function	0.1%	unsigned	R
9203	11	Output current peak value	0.01A/0.1A	unsigned	R
9204	12	Converter output voltage peak value	0.1V	unsigned	R
9205	13	Input power	0.01/ 0.1kWh	unsigned	R
9206	14	Output power	0.01/ 0.1kWh	unsigned	R
9207	15	Input terminal status *1	-	-	R
9208	16	Output terminal status *2	-	-	R

PNU No	Internal No	Description	Unit	Type	Access type
16 bit					
9209	17	Load meter	0.1%	unsigned	R
9210	18	Motor excitation current	0.01A/0.1A	unsigned	R
9211	19	Position pulse	1	unsigned	R/W
9212	20	Cumulative energization time	1h	unsigned	R
9213	21	reserved	-	-	-
9214	22	Orientation status	1	unsigned	R
9215	23	Actual operation time	1h	unsigned	R
9216	24	Motor load factor	0.1%	unsigned	R
9217	25	Cumulative power	1kWh	unsigned	R
9218-9223	26-31	reserved	-	-	-
9224	32	Torque order	0.1%	unsigned	R
9225	33	Torque current order	0.1%	unsigned	R
9226	34	Motor output	0.1kW	unsigned	R
9227	35	Feedback pulse	1	unsigned	R
9228-9237	36-45	reserved	-	-	-
9238	46	Motor temperature			R
9239-9241	47-49	reserved	-	-	-
9242	50	Power saving effect	-	unsigned	R
9243	51	Cumulative saving power	-	unsigned	R
9244	52	PID set point	0.1%	unsigned	R/W
9245	53	PID measured value	0.1%	unsigned	R/W
9246	54	PID deviation	0.1%	unsigned	R/W
9250	58	Option input terminal status1 *3	-	-	R
9251	59	Option input terminal status2 *3	-	-	R
9252	60	Option output terminal status *4	-	-	R
9253	61	Motor thermal load factor	0.1%	unsigned	R
9254	62	Transistor thermal load factor	0.1%	unsigned	R
9256	64	PTC thermistor resistance	ohm	unsigned	R
9257	65	Output power			R
9192		(with regenerative display)			
9258	66	Cumulative regenerative power			R
9260	68	2nd PID set point	0.1%	unsigned	R/W
9261	69	2nd PID measured value	0.1%	unsigned	R/W
9262	70	2nd PID deviation	0.1%	unsigned	R/W
9272	80	Integrated power on time			R
9273	81	Running time			R
9274	82	Saving energy monitor			R
9276 - 9283	84 - 91	Fault code (1) - (8)	-	-	R
9441	249	Run command *5	-	-	R/W
32 bit					
9705	513	Output frequency	0.01Hz	signed	R
9707	515	Setting frequency	0.01Hz	signed	R
9709	517	Motor rotation	0.1rpm	signed	R
9711	519	Load meter	0.1%	signed	R
9713	521	Positioning pulse	1	signed	R/W
9715	523	Watt-hour meter(1kWh step)	1kWh	unsigned	R
9717	525	Watt-hour meter(0.1/0.01kWh step)	0.1/ 0.01kWh	unsigned	R
9719	527	Position error	1	signed	R

*1 Input terminal status details (ex. A800)

b15															b0
-	-	-	-	CS	RES	STOP	MRS	JOG	RH	RM	RL	RT	AU	STR	STF

This data indicates status of main unit input terminal.
The number of terminal and names depends on main unit specification.

*2 Output terminal status details (ex. A800)

b15															b0
-	-	-	-	-	-	-	-	SO	ABC2	ABC1	FU	OL	IPF	SU	RUN

This data indicates status of main unit output terminal.
The number of terminal and names depends on main unit specification.

*3 Option input terminal status1,2 (ex. A800 and FR-A8AX)

Option input terminal status1

b15															b0
X15	X14	X13	X12	X11	X10	X9	X8	X7	X6	X5	X4	X3	X2	X1	X0

Option input terminal status2

b15															b0
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	DY

This data indicates status of FR-A8AX terminal.

*4 Option output terminal status (ex. A800 and A8AY/A8AR)

b15															b0
-	-	-	-	-	-	RA3	RA2	RA1	Y6	Y5	Y4	Y3	Y2	Y1	DY

This data indicates status of FR-A8AY(Y0-Y6), FR-A8AR(RA1-RA3) terminal.

*5 Run command (ex. A800 and A8AY/A8AR)

Users can specify the terminal function using this data.

b15															b0
-	-	-	-	RES	STOP	CS	JOG	MRS	RT	RH	RM	RL	-	-	AU

These bits function is depending on VFD parameter setting.

7.4 Drive Profile Parameters (Acyclic Data Exchange)

Inverter parameters are, when possible, mapped to PROFIdrive parameters. The remaining parameters can be accessed as vendor specific parameters.

It is recommended not to use the standard PROFIdrive parameters as vendor specific parameters.

7.4.1 PROFIdrive Parameters

The following parameters are implemented in the option board:

Parameter	Definition	R/W	Data Type	Value/Description
P915	Selection switch Setpoint telegram	R	Array [n] Unsigned16	Holds the current configuration of the Setpoint telegram.
P916	Selection switch Actual value telegram	R	Array [n] Unsigned16	Holds the current configuration of the Actual value telegram.
P918	Node Address	R	Unsigned 16	Current node address
P922	Telegram Selection	R	Unsigned 16	Reflects the latest accepted configuration data from the master. Value: Telegram: 0 Telegram 1 (Speed Control) or telegram 100 (Torque Control), depending on which telegram is configurated. 101 Telegram 101 (Position Control) 102 Telegram 102 103 Telegram 103 .. ..
P923	List of all parameters for signals	R	Array[60000] Unsigned16	All parameters that are possible to map to process data (i.e. is defined as PROFIdrive signals) are listed here; subindex = signal number. If a parameter connected to the specific signal is possible to map, the PNU number is returned. If the parameter is not possible to map, 0 is returned.
P924	Status word bit Pulses Enabled	R	Array[2] Unsigned16	Subindex 0: 2 (Signal number for ZSW1) Subindex 1: 15 (Bit position)
P930	Operating mode	R/W	Unsigned16	Always 8001h (Vendor specific mode).
P944	Fault message counter	R	Unsigned16	Incremented by one each time the fault buffer (P947) changes.
P947	Fault numbers	R	Array[8] Unsigned16	Subindex 0: Active fault situation (if the drive is in Faulted state). Subindex 1-7 Fault history, where subindex 1 holds the most recent fixed fault situation. Fault number is the corresponding PROFIdrive fault class.
P950	Scaling of fault buffer	R	Array[2] Unsigned16	Subindex 0: 8 Subindex 1: 1 Defines the number of fault situations (subindex 0) and the number of fault messages (subindex 1) for each situation that the fault buffer can hold.

Parameter	Definition	R/W	Data Type	Value/Description
P951	Fault number list	R	Array[255] Unsigned16	Holds descriptive text for each fault that is supported by the option board. Each subindex corresponds to the event code number, which means that the accessible subindices are limited. The fault description is accessed as an additional text array.
P964	Device identification	R	Array[5] Unsigned16	Manufacturer ID: 010Ch (HMS) Drive Unit Type: 0 Version (software) xxyy (decimal) Firmware date, year yyyy (decimal) Firmware date, day/month ddmm (decimal)
P965	Profile number	R	OctetString 2	Byte 0: 3 (PROFIdrive profile) Byte 1: 41 (Version 4.1)
P967	STW1	R	V2	Last control word received from the controller.
P968	ZSW	R	V2	Current status word from the drive.
P972	Drive reset	R/W	Unsigned16	1: Power-on reset 2: Prepare power-on reset
P980	List of parameters	R	Array[n] Unsigned16	Parameter numbers of all existing parameters are saved in the subindices (profile parameters and regular drive parameters). The array is assigned in increasing sequence and consecutively. The end of the list of defined parameters is marked by a subindex with the value 0.

7.4.2 Setpoint- and Actual Value (P915/P916)

P915 and P916 reflect the configuration of the PROFIBUS telegrams that are used for process data. The parameters are read only and consist of arrays, where the number of elements corresponds to the number of words in the IO DATA. Each element holds the Parameter Number (PNU) of the parameter mapped to that particular IO DATA word.

The parameters reflect the latest accepted parameterization data.

7.4.3 Signal List (P923)

The signal list is not saved. It is created at the time of request only for those elements (subindices) that are requested. The possibility to map vendor specific parameters is examined by asking for the Descriptor attribute of the specific parameter. It is not possible to map other PROFIdrive standard parameters than P967 and P968 (Control and Status words).

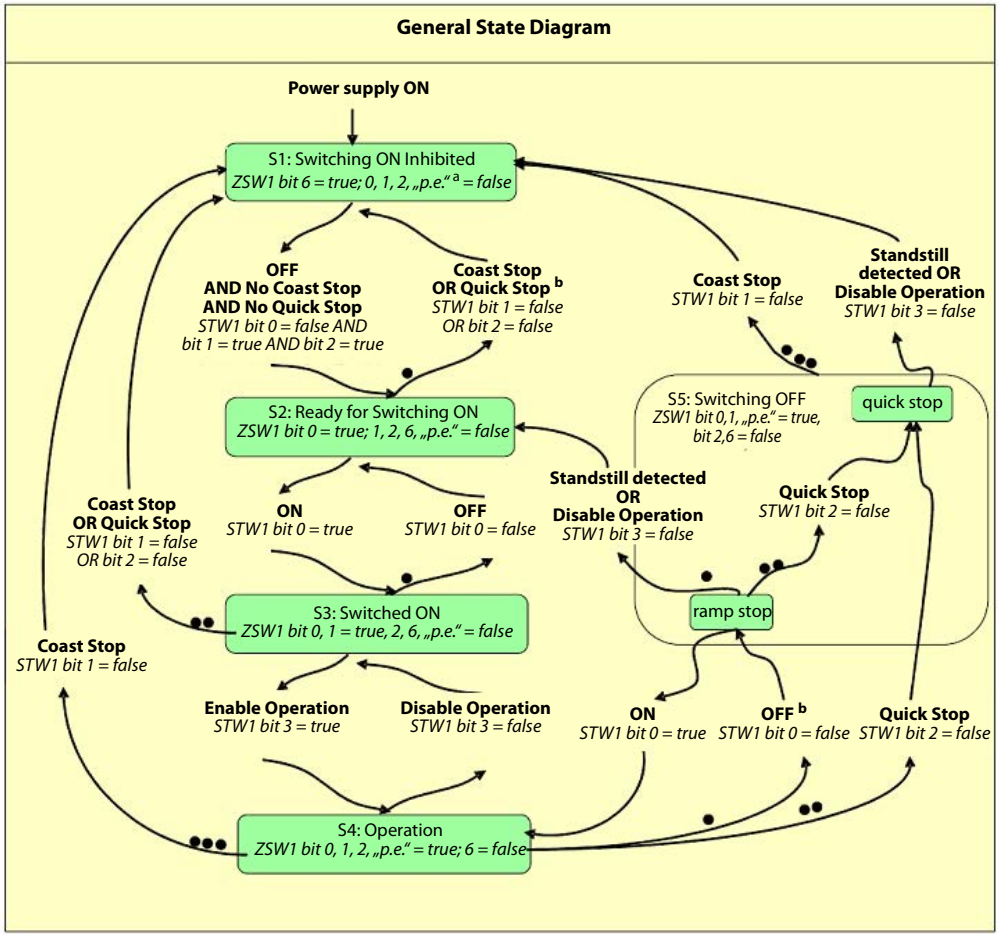
7.4.4 Drive Reset (P972)

The PROFIBUS master can send a reset request to the application by writing to this parameter. If the request is rejected an error is returned.

Error Code	Description
0	Illegal parameter number (Drive reset not supported).
1	Request cannot be executed due to operation status.
7	Request cannot be executed due to operation status.

7.5 General State Diagram

This general state diagram shows how the inverter behaves based on Control Word 1 (STW1). For detailed state diagrams for various modes, please consult ProfiDrive technical specification manual.



7.6 Process Data (Cyclic Data Exchange)

7.6.1 General

Drive control is per default performed through the cyclic data exchange channel. DriveControl, DriveStatus, Setpoint and Actual Values are sent as IO DATA telegrams on PROFIBUS. These telegrams are transformed and mapped to the main unit parameters by the option board.

The Process Data map is based entirely on the requirements of PROFIdrive and the end user by means of User Parameterization Data. The active process data is mapped in the configuration tool, where PROFIdrive profile specific signals are specified using signal numbers 1 - 1000 and vendor specific signals are specified using signal numbers 1001…59999.

For all configurations, the default for the first IO_DATA items in each direction will be DriveControl and DriveStatus respectively.

7.6.2 Signals

Each IO DATA channel is assigned a signal which is mapped to an inverter parameter. Thus PROFIdrive parameters that are to be exchanged as IO DATA (i.e. Process Data) must be assigned a unique signal number.

Signal numbers 1…1000 are reserved for standard Drive Profile parameters, while signals 1001…59999 are used for vendor specific parameters.

The table below shows the implementation of the PROFIdrive signal list.

Signal	Abbreviation	PROFIdrive Parameter	Main unit Parameter
1	STW1	Control word 1 (P967)	DriveControl
2	ZSW1	Status word 1 (P968)	DriveStatus
-	-	-	(not assigned)
5	NSOLL_A	Speed setpoint A	SetpointSpeed
6	NIST_A	Speed actual value A	ActualSpeed
-	-	-	(not assigned)
1001…59999	-	Vendor specific	Mapped to Process Data (if possible)

Below is a brief explanation of the format of the used signals. For a detailed explanation, including control and status words specific to various operation modes, please consult ProfiDrive specification manual.

Control Word STW1

Bit	Contents
0	ON
1	No Coast Stop
2	No Quick Stop
3	Enable Operation
4	Enable Ramp Generator
5	Unfreeze Ramp Generator
6	Enable Setpoint
7	Fault Acknowledge
8	Jog 1 ON (not used)
9	Jog 2 ON (not used)
10	Control By PLC
11-15	-

Status Word ZSW1

Bit	Contents
0	Ready To Switch ON
1	Ready To Operate
2	Operation Enabled
3	Fault Present
4	Coast Stop Not Activated
5	Quick Stop Not Activated
6	Switching On Inhibited
7	Warning Present
8	Speed Error Within Tolerance
9	Control Requested
10-14	-
15	Pulses Enabled

Speed setpoint A (NSOLL_A)

The output frequency set by NSOLL_A is calculated relative to the rated frequency parameter (inverter parameter no. 3).

$$f = \frac{NSOLL_A}{0x4000} \times f_r$$

f: output frequency [Hz]
f_r : motor rated frequency [Hz]

Speed actual value A (NIST_A)

Speed actual value A is calculated in the same way as NSOLL_A.

7.6.3 Telegram Types

The Option Board supports PROFIdrive Standard Telegram 1. If additional parameters are to be mapped to process data, one of the additional product-specific telegrams (100-103) must be used, depending on the size of the data in question.

Telegram	Description	Size (words)
1	Standard Telegram 1	2
100	Telegram 100 (Torque control)	2
101	Telegram 101 (Position control)	10
102	Telegram 102 (Custom)	18
103	Telegram 103 (Custom)	32

The user specifies which telegram to use when configuring the option board.

Note: Only one telegram module can be used at a time.

Setpoint Telegram

Setpoint telegrams are used for data from the master to the main unit. Depending on control mode, the contents of the telegram may differ, see table below for configuration.

Telegram	IO DATA (PROFIdrive)		Read Process Data (Main unit)	
	Word	Signal	Octet	Process data Parameter
Speed Control (SC) Torque Control (TC) Position Control (PC)	IO DATA 1	1 (STW1)	0…1	DriveControl (0402h)
Speed Control (SC)	IO DATA 2	5 (NSOLL_A)	2…3	SetPoint Speed (0405h)
Torque Control (TC)		14349		TargetTorque (0425h)
Position Control (PC)		9441		Run command (00F9h)
Position Control (PC)	IO DATA 3	9713	4…7	Positioning pulse (0209h…020Ah)
Position Control (PC)	IO DATA 4			
Position Control (PC)	IO DATA 5	5 (NSOLL_A)	8…9	SetPointSpeed (0405h)
Position Control (PC)	IO DATA 6	16383	10…13	AccelerationDeltaTime (0417h…0418h)
Position Control (PC)	IO DATA 7			
Position Control (PC)	IO DATA 8	16387	14…17	DecelerationDelta Time (014Bh…041Ch)
Position Control (PC)	IO DATA 9			

Actual Value Telegram

Actual value telegrams are used for data from the main unit to the master. Depending on control mode, the contents of the telegram may differ, see table below for configuration.

Telegram	IO DATA (PROFIdrive)		Write Process Data (Main unit)	
	Word	Signal	Octet	Process data Parameter
Speed Control (SC) Torque Control (TC) Position Control (PC)	IO DATA 1	2 (ZSW1)	0…1	DriveStatus (0403h…0404h)
Speed Control (SC)	IO DATA 2	6 (NIST_A)	2…3	ActualSpeed (0407h)
Torque Control (TC)		14350		ActualTorque (0426h)
Position Control (PC)		9208		Output terminal (0010h)
Position Control (PC)	IO DATA 3	9713	4…7	Positioning pulse (0209h…020Ah)
Position Control (PC)	IO DATA 4			
Position Control (PC)	IO DATA 5	6 (NIST_A)	8…9	ActualSpeed (0407h)
Position Control (PC)	IO DATA 6	14350	10…11	ActualTorque (0426h)
Position Control (PC)	IO DATA 7			
Position Control (PC)	IO DATA 8	(9719)	12…15	PositionError (020Fh…0210h)
Position Control (PC)	IO DATA 9			

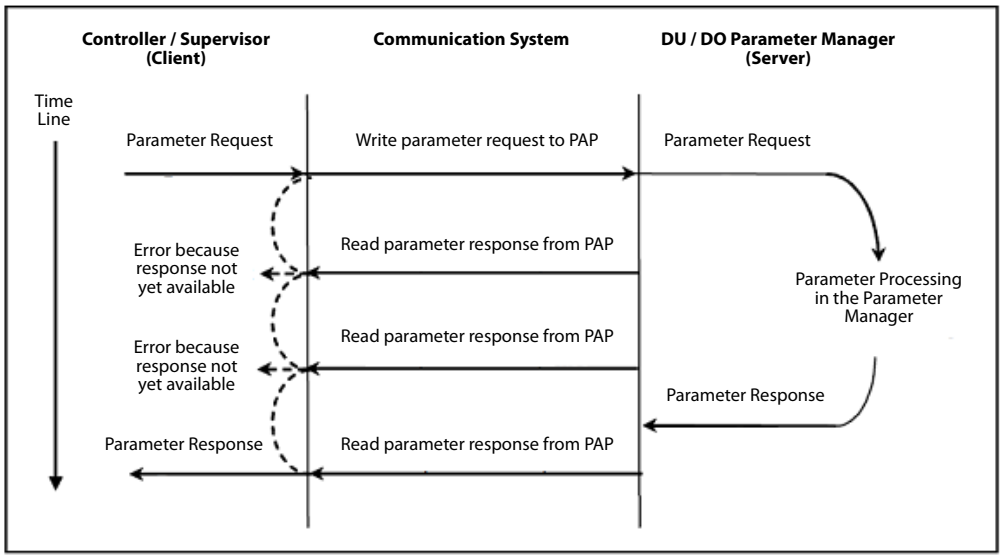
7.6.4 Vendor Specific Access to Parameters

Vendor specific parameters 1001… 59999 can be defined via the PROFIBUS network master. Parameter range below 1001 can not be accessed as vendor specific as they may be reserved for other uses.

7.7 Acyclic Data Exchange

This subchapter describes some of the basic sequences used in acyclic communication. All available sequences are described in the ProfiDrive Specification manual (chapter 6.2.3.6 Telegram sequences for Parameter Access).

The data flow for acyclic communication always consists of writing of the parameter request, and reading the parameter response. The parameter response read may fail, if the data requested is not yet ready. The option card should be asked for a response again, until it is ready. This is presented on the figure below:



7.7.1 Explanation of fields used in requests

This table explains the fields used in various sequences. Words sent in requests should have their most significant byte transmitted first (Big endian encoding). This is also the format in which incoming words will be transmitted.

Field	Data Type	Values	Comment
Request reference	Unsigned8	0x01 – 0xFF	
Request ID	Unsigned8	0x01 Request parameter 0x02 Change parameter	
Response ID	Unsigned8	0x01 Request parameter (+) 0x02 Change parameter (+) 0x81 Request parameter (-) 0x82 Change parameter (-)	
Axis / DO-ID	Unsigned8	0x01	
No. of Parameters	Unsigned8	0x01 – 0x27 Quantity	
Attribute	Unsigned8	0x10 Value 0x20 Description 0x30 Text	
No. of elements	Unsigned8	0x01 – 0xEA Quantity	
Parameter number	Unsigned16	0x0001 – 0xFFFF	
Subindex	Unsigned16	0x0001 – 0xFFFF	
Format	Unsigned8	0x01 – 0x36 Data types 0x40 – 0x44 Data types	See data format type table
No. of values	Unsigned8	0x00 – 0xEA Quantity	
Error number	Unsigned16	0x0000 – 0x00FF Error numbers	See error table

7.7.2 Data format type table

Data type	Value
BOOLEAN	0x01
INTEGER8	0x02
INTEGER16	0x03
INTEGER32	0x04
UINTEGER8	0x05
UINTEGER16	0x06
UINTEGER32	0x07
VISIBLESTRING	0x09
OCTETSTRING	0x0a
ZERO	0x40
ERROR	0x44
N2	0x71
N4	0x72
V2	0x73

7.7.3 Error table

Error description	Value
IMPERMISSIBLE_PARAMETER_NUMBER	0x00
PARAMETER_VALUE_CANNOT_BE_CHANGED	0x01
LOW_OR_HIGH_LIMIT_EXCEEDED	0x02
FAULTY_SUBINDEX	0x03
NO_ARRAY	0x04
INCORRECT_DATA_TYPE	0x05
SETTING_NOT_PERMITTED	0x06
DESCRIPTION_ELEMENT_CANNOT_BE_CHANGED	0x07
NO_DESCRIPTION_DATA_AVAILABLE	0x09
NO_OPERATION_PRIORITY	0x0b
NO_TEXT_ARRAY_AVAILABLE	0x0f
REQUEST_CANNOT_BE_EXECUTED_BECAUSE_OF_OPERATING_STATE	0x11
VALUE_IMPERMISSIBLE	0x14
RESPONSE_TOO_LONG	0x15
PARAMETER_ADDRESS_IMPERMISSIBLE	0x16
ILLEGAL_FORMAT	0x17
NUMBER_OF_VALUES_ARE_NOT_CONSISTENT	0x18
AXIS_DO_NONEXISTENT	0x19
PARAMETER_TEXT_ELEMENT_CANNOT_BE_CHANGED	0x20

7.7.4 Sequence 1: Request parameter value, single

Byte no.	Field
0	Request ID
1	Request reference
2	No. of parameters
3	DO-ID
4	No. of elements
5	Attribute value
6-7	Parameter number
8-9	Subindex (irrelevant for non-arrays)

7.7.5 Sequence 1: Parameter response positive

Byte no.	Field
0	Request ID
1	Request reference
2	No. of parameters
3	DO-ID
4	No. of values
5	Format
6-7	Parameter value
8-9	Parameter value (only for 32 bit data)

7.7.6 Sequence 1: Parameter response negative

Byte no.	Field
0	Request ID
1	Request reference
2	No. of parameters
3	DO-ID
4	No. of values
5	Format
6-7	Error value

7.7.7 Sequence 2: Change parameter value

Byte no.	Field
0	Request ID
1	Request reference
2	No. of parameters
3	DO-ID
4	No. of elements
5	Attribute value
6-7	Parameter number
8-9	Subindex (irrelevant for non-arrays)
10	No. of values
11	Format
12-13	Set value

7.7.8 Sequence 2: Parameter response positive

Byte no.	Field
0	Request ID
1	Request reference
2	No. of parameters
3	DO-ID

7.7.9 Sequence 2: Parameter response negative

Byte no.	Field
0	Request ID
1	Request reference
2	No. of parameters
3	DO-ID
4	No. of values
5	Format
6-7	Error value

7.7.10 Sequence 3: Request parameter value, several array elements

Byte no.	Field
0	Request ID
1	Request reference
2	No. of parameters
3	DO-ID
4	No. of elements (n)
5	Attribute value
6-7	Parameter number
8-9	Subindex

7.7.11 Sequence 3: Parameter response positive

Byte no.	Field
0	Request ID
1	Request reference
2	No. of parameters
3	DO-ID
4	No. of values
5	Format
6-7	Parameter value 1
8-9	Parameter value 2
…	…
(4+2n) – (5+2n)	Parameter value n

7.7.12 Sequence 3: Parameter response negative

Byte no.	Field
0	Request ID
1	Request reference
2	No. of parameters
3	DO-ID
4	No. of values
5	Format
6-7	Error value

8. Diagnostics

Conversion of diagnostic fault codes

Main Unit Fault Code	Main Unit Fault Name	Main Unit Description	PROFIdrive Fault	PROFIdrive Fault Code
10h	E.OC1	OC During Acc	Motor Overload	08
11h	E.OC2	Steady spd OC	Motor Overload	08
12h	E.OC3	OC During Dec	Motor Overload	08
20h	E.OV1	OV During Acc	DC Link Overvoltage	04
21h	E.OV2	Steady spd OV	DC Link Overvoltage	04
22h	E.OV3	OV During Dec	DC Link Overvoltage	04
30h	E.THT	Inv. Ovrload	Overtemperature Electronic Device	06
31h	E.THM	Motor Ovrload	Motor Overload	08
40h	E.FIN	H/Sink O/Temp	Overtemperature Electronic Device	06
50h	E.IPF	Inst. Pwr. Loss	Mains Supply	02
52h	E.ILF	Input phase loss	Mains Supply	02
60h	E.OLT	Stall Prev STP	Motor Overload	08
70h	E.BE	Br. Cct. Fault	Brake Resistor	14
80h	E.GF	Ground Fault	Earth/Ground Fault	07
81h	E.LF	Output phase loss	Power Electronics	05
90h	E.OHT	OH Fault	Motor Overload	08
91h	E.PTC	PTC Activated	External	16
A0h	E.OPT	Option fault	Engineering	18
A1h	E.OP1	Option1 fault	Internal Communication	12
A2h	E.OP2	Option2 fault	Internal Communication	12
A3h	E.OP3	Option3 fault	Internal Communication	12
B0h	E.PE	Corrupt memory	Microcontroller Hardware or Software	01
B1h	E.PUE	PU Leave out	Internal Communication	12
B2h	E.RET	Retry No Over	Technology	17
B3h	E.PE2	PR Storage Alarm	Microcontroller Hardware or Software	01
C0h	E.CPU	CPU Fault	Microcontroller Hardware or Software	01
C1h	E.CTE	PU Short cct	Other	19
C2h	E.P24	24VDC short cct	Other	19
C4h	E.CDO	OC Detect level	Technology	17
C5h	E.IOH	Inrush overheat	Mains Supply	02
C6h	E.SER	VFD Comm error	Internal Communication	12
C7h	E.AIE	Analog in error	External	16
C8h	E.USB	USB Comm error	Internal Communication	12
C9h	E.SAF	Safety cct fault	Technology	17
D0h	E.OS	Overspeed	Technology	17
D1h	E.OSD	Excess spd deviation	Technology	17
D2h	E.ECT	Enc. Signal loss	Feedback	11
D3h	E.OD	Excess pos fault	Technology	17
D5h	E.MB1	Brake seq fault	Other	19
D6h	E.MB2	Brake seq fault	Other	19
D7h	E.MB3	Brake seq fault	Other	19
D8h	E.MB4	Brake seq fault	Other	19

Main Unit Fault Code	Main Unit Fault Name	Main Unit Description	PROFIdrive Fault	PROFIdrive Fault Code
D9h	E.MB5	Brake seq fault	Other	19
DAh	E.MB6	Brake seq fault	Other	19
DBh	E.MB7	Brake seq fault	Other	19
DCh	E.EP	Enc. Phase Fault	Feedback	11
F1h	E.1	Fault 1 (opt slot 1)	Internal Communication	12
F2h	E.2	Fault 2 (opt slot 2)	Internal Communication	12
F3h	E.3	Fault 3 (opt slot 3)	Internal Communication	12
F5h	E.5	Fault 5	Microcontroller Hardware or Software	01
F6h	E.6	Fault 6	Microcontroller Hardware or Software	01
F7h	E.7	Fault 7	Microcontroller Hardware or Software	01
FBh	E.11	Fault 11	Technology	17
FDh	E.13	Fault 13	Power Electronics	05
51h	E.UVT	Under Voltage	Mains Supply	02

9. Troubleshooting

The table shows tips on troubleshooting.

Error Event	Possible Cause(s)	Corrective Action

A. Translation of Signal Numbers

Signal numbers used in the main unit do not directly translate to signal numbers (PNUs) used on PROFIBUS. An offset is added to avoid ambiguous numbering on PROFIBUS, where parameters and monitor data have different PNU numbers.

Every signal number corresponds to to 16 bits. A 32-bit parameter e.g., occupies two numbers, but is addressed by the lower number only.

Name	Signal No, Main Unit		Offset (decimal)	PNU No, PROFIdrive (Signal No. + Offset)	Acyclic Data Exchange	Cyclic Data Exchange
	Hexadec-imal	Decimal				
DriveControlMaskWrite	400	1024	13288	14312	Yes	Yes
DriveControl	402	1026	13288	14314	Yes	Yes
DriveStatus	403	1027	13288	14315	Yes	Yes
SetpointSpeed	405	1029	13288	14317	Yes	Yes
CommandSpeed	406	1030	13288	14318	Yes	Yes
ActualSpeed	407	1031	13288	14319	Yes	Yes
SpeedScaleNumerator	408	1032	13288	14320	Yes	No
SpeedScaleDenominator	40A	1034	13288	14322	Yes	No
RatedSpeed	40C	1036	13288	14324	Yes	No
PoleCount	40D	1037	13288	14325	Yes	No
RatedCurrent	40E	1038	13288	14326	Yes	No
RatedVoltage	410	1040	13288	14328	Yes	No
MotorType	411	1041	13288	14329	Yes	No
DriveMode[1]	412	1042	13288	14330	Yes	No
			15336	16378	Yes	Yes
SupportedModes	413	1043	13288	14331	Yes	No
AccelerationDeltaSpeed[1]	415	1045	13288	14333	Yes	No
			15336	16381	Yes	Yes
AccelerationDeltaTime[1]	417	1047	13288	14335	Yes	No
			15336	16383	Yes	Yes
DecelerationDeltaSpeed/ Time[1]	419	1049	13288	14337	Yes	No
			15336	16385	Yes	Yes
QuickDecelerationDelta-Speed	41D	1053	13288	14341	Yes	No
QuickDecelerationDeltaTime	41F	1055	13288	14343	Yes	No
MaxSpeed	421	1057	13288	14345	Yes	No
MinSpeed	423	1059	13288	14347	Yes	No
TargetTorque	425	1061	13288	14349	Yes	Yes
ActualTorque	426	1062	13288	14350	Yes	Yes
TorqueSlope	427	1063	13288	14351	Yes	No
TorqueProfileType	429	1065	13288	14353	Yes	No
RatedTorque	42A	1066	13288	14354	Yes	No
TorqueScaleNumerator	42C	1068	13288	14356	Yes	No
TorqueScaleDenominator	42E	1070	13288	14358	Yes	No
DisableOptionCode	430	1072	13288	14360	Yes	No
ShutdownOptionCode	431	1073	13288	14361	Yes	No

[1] These parameters can either be mapped (as acyclic data) as an array, with offset 13288d, or can each entry in the array be mapped as a separate parameter with offset 15366d (as acyclic and/or cyclic data).

Index

A

Actual device address (Parameter 1306) 118
Acyclic Data Exchange 136

C

Communication settings 119
Components
  Overview 6
Control Word STW1 133

D

Data Exchange 125
  Drive Profile Parameters 125
  Monitor Data 125
  Parameter 125
Device address (Parameter1305) 118
Drive Profile Parameters 129
Drive Reset 130

E

Environment
  Specifications 7

G

General Settings (Parameter 1300) 118
General State Diagram 131
GX Works (FX-CPU)
  Acyclic communication 82
  Telegram 1 65
  Telegram 102 77
GX Works (Q-CPU)
  Acyclic communication 34
  Simple Ladder Acyclic communication 54
  Simple Ladder Telegram 1 42
  Simple Ladder Telegram 102 50
  Telegram 1 16
  Telegram 102 29

I

Inverter parameters 126

L

LED Indicators 13
  Error 13
  Module status 13
  Network status 13

M

Monitor Data 126

N

Network Connector 12
Node address (Parameter 1305) 118

O

Overview
  Components 6

P

Parameters
  1300 (General Settings) 118
  1305 (Device address, node address) 118
  1306 (Actual device address, node address) .118
  1307 (Option board settings priority) 119
  Option board 117
Process Data 132
PROFIdrive Parameters 129

S

SIMATIC STEP7 111
Specifications
  Environment 7
Speed actual value A (NIST_A) 133
Speed setpoint A (NSOLL_A) 133
Status Word ZSW1 133

T

Telegram Types 134
TIA Portal
  Acyclic communication 101
  Telegram 1 89
  Telegram 102 98