SIEMENS

Application description • 01/2015

SINAMICS G/S: PROFINET connection with LabVIEW

Application to connect SINAMICS converters with LabVIEW via a standard Ethernet interface

http://support.automation.siemens.com/WW/view/en/99684399

Warranty and liability

Note

The application examples are not binding and do not claim to be complete regarding the circuits shown, equipping and any eventuality. These application examples do not represent specific customer solutions, but are only intended to provide support when it comes to typical tasks. You are responsible for the proper operation of the described products. These application examples do not relieve you of the responsibility in safely and professionally using, installing, operating and servicing equipment. By using these application examples, you agree that Siemens cannot be made liable for possible damage beyond the liability clause described. We reserve the right to make changes to these application examples at any time and without prior notice. If there are any differences between the suggestions made in these application examples and other Siemens publications, such as catalogs, the contents of the other document(s) take priority.

We give no guarantee that the information contained in this document is complete, accurate, or up-to-date.

We accept no liability for any damage or loss caused by the examples, information, programs, planning data, or performance data described in this application example, irrespective of the legal basis for claims arising from such damage or loss, unless liability is mandatory. For example, according to the product liability law, in cases of malfeasance, gross negligence, due to endangerment of life, body or health, due to assumption of a guarantee for the properties of a product, due to malicious concealment of a defect or due to violation of basic contractual obligations. However, claims for indemnification based on breach of contract shall be limited to liability for damages to the contract-specific, foreseeable damages, provided there is no mandatory liability for intent, acts of gross negligence, harm to the life, body and health of human beings. Any change to the burden of proof to your disadvantage is not covered hereby.

Security information

Siemens provides products and solutions with industrial security functions that support the secure operation of plants, solutions, machines, devices, and/or networks. They are important components in a holistic industrial security concept. With this in mind, Siemens products and solutions undergo continuous development. Siemens recommends strongly that you regularly check for product updates.

For the secure operation of Siemens products and solutions, it is necessary to take suitable preventive action (e.g. cell protection concept) and integrate each component into a holistic, state-of-the-art industrial security concept. Any third-party products that may be in use must also be taken into account. For more information about industrial security, visit http://www.siemens.com/industrialsecurity

To receive information about product updates on a regular basis, register for our product newsletter. For more information, visit http://support.automation.siemens.com.

List of contents

1 Task

1.1 Overview

Introduction

The application supports you when connecting SINAMICS converters with PROFINET interface to the LabVIEW software from National Instruments.

This facilitates cyclic I/O communication via any network interface of the PC being used. LabVIEW operates in this case as PROFINET controller.

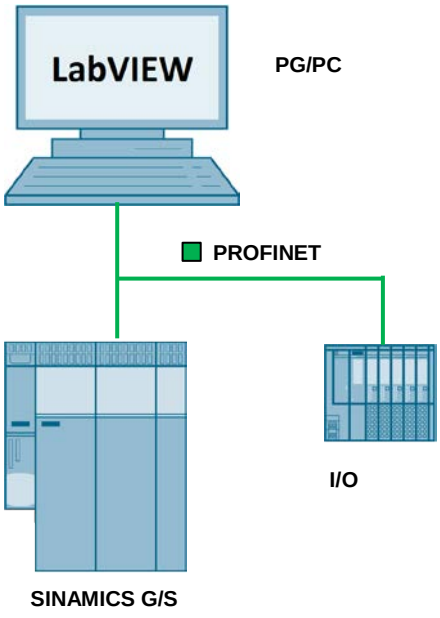
Non-cyclic communication via DPV1 is also possible.

Note All of the PROFINET I/O devices available in the hardware catalog of the TIA Portal can be Incorporated.
The application is primarily intended to control SINAMICS converters.

Overview of the automation task

The following diagram provides an overview of the connection.

Fig. 1-1: Labview as PROFINET controller



Description of the automation task

LabVIEW is a graphic programming system from National Instruments. LabVIEW programs are called Virtual Instruments or simply VIs. They comprise two components: The front panel accommodates the user interface, the block diagram shows the graphic program code.

LabVIEW is frequently used as calculation and simulation tool for tests stands. In order to integrate real hardware (e.g. converters) into the simulation or measurement, it is necessary that this hardware is directly connected to LabVIEW.

Frequency, it is sufficient to operate the SINAMICS drive with closed-loop speed or torque control – or to use the integrated basic positioner without placing any real-time demands on the communication.

PROFINET is one of the options available for establishing a connection between SINAMICS converters and LabVIEW via a standard Ethernet interface.

This application allows this communication path to be used.

2 Solution

2.1 Overview

Schematic

The application comprises a DLL, which is integrated in LabVIEW and which is executed there. This provides functions for communication with PROFINET I/O devices directly from LabVIEW.

Design

The DLL provided by the application is essentially based on the Siemens PROFINET driver for controllers.

An interface to the LabVIEW has been created via the IO base user programming interface.

For simpler handling, the application example VIs are provided for standard Siemens telegrams

- Standard telegram 1 (closed-loop speed control)
- Standard telegram 111 (EPOS operation)
- Free telegram configuration

their interface emulates 1:1 the SINAMICS telegram interface.

For non-cyclic communication, example VI is provided to read/write parameters; this can be expanded when required.

Advantages

The application described here offers you the following advantages

- Shorter time and lower costs when implementing the solution
- Simple integration of the communication driver
- Simple control of SINAMICS drives from LabVIEW
- Integration of additional PROFINET IO devices in LabVIEW
- Connection via standard Ethernet interface

Restriction

This application does not contain a description of:

- how to use LabVIEW
- commissioning SINAMICS drive systems
- using STEP 7 in the TIA Portal
- DPV1 parameter request and response

Knowledge required

It is assumed that readers have basic knowledge about LabVIEW. Examples showing the integration into LabVIEW are provided; however, the user is responsible for the integration in his particular application.

Further, it is assumed that readers are knowledgeable about configuring PROFINET nodes in the TIA Portal, as well as commissioning SINAMICS converters and STARTER / SCOUT.

2.2 Hardware and software components

2.2.1 Validity

This application is valid for

- LabVIEW from version 2013 and higher
- SINAMICS drives
  - S120 CU320-2 PN, CU310-2 PN from firmware V4.5 and higher
  - S120 CU320-2 DP with CBE20 from firmware V4.5 and higher
  - S110 CU305 PN from firmware V4.4 and higher
  - G120 CU250S-2 PN, CU240E-2 PN, CU240E-2 PN-F from firmware V4.5 and higher
  - G120P CU230P-2 PN from firmware V4.5 and higher
  - G120C PN from firmware V4.5 and higher
- Step7 Professional from V12 (a license is not required)

2.2.2 Components used

The application was created with the following components:

Hardware components

Table 2-1

Component	Qty.	Article number	Note
SINAMICS demonstration case	1	6ZB2480-0CN00	The demonstration case comprises the following components: 6SL3040-1MA01-0AA0 6SL3054-0EG01-1BA0 6SL3130-6AE15-0AB1 6SL3120-2TE13-0AA4 6SL3055-0AA00-5BA3 1FK7022-5AK71-1LG0 1FK7022-5AK71-1AG3
ET200S	1		Any distributed I/O can be optionally integrated

Software components

Table 2-2

Component	Qty.	Article number	Note
LabVIEW 2013	1		
Siemens PROFINET driver runtime	1	6ES7195-3AA05-0XA0	Runtime license must be purchased
WinPCap v4.1.3	1		Freely available
VC++ 2010 SP1 redistributable	1		Freely available
STEP 7 Professional	1		A license is not required

Component	Qty.	Article number	Note
V12			Download
Sinamics STARTER v4.4	1		Download
PN driver HSP for TIA Portal V12	1		

Sample files and projects

The list below contains all the files and projects used in this example.

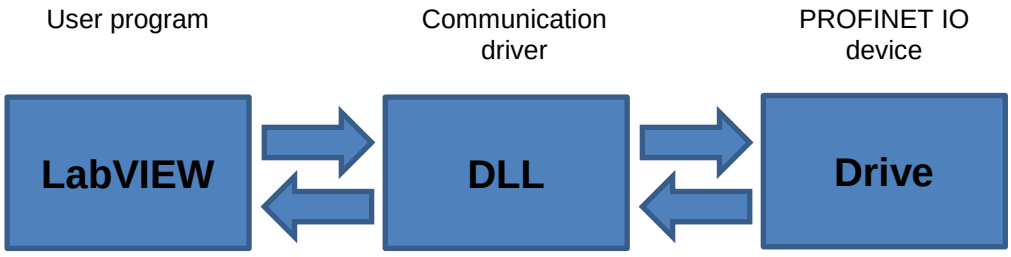
Table 2-3

Component	Note
99684399_PNIO_BIB_v20_EN.zip	This zipped file contains the LabVIEW library.
99684399_PN-Driver_HSP.zip	This zipped file contains the HSP for the TIA Portal V12.
99684399_TIA_Project.zip	This zipped file contains the TIA V12 project.
99684399_Starter_Project.zip	This zipped file contains the Starter project.
99684399_DOKU_v20_EN.pdf	This document.

3 Principle of operation

3.1 General overview

Fig. 3-1: Flowchart



The DLL serves as the interface between the LabVIEW user program and the PROFINET IO devices. It provides functions, which control the communication and allow cyclic IO data to be transferred.

3.2 Functionality of the basic functions

The complete functionality of the application is provided by the DLL, which is integrated into LabVIEW. The functions contained in the DLL control the initialization of the IO controller and the data transfer.

To use these functions, preconfigured VIs are available for LabVIEW; they correctly call the DLL functions, and can be integrated in the actual LabVIEW user program.

The function and interface of each VI are described in the following sections.

3.2.1 Program details about the block Start.vi

This function initializes the PROFINET controller. The Ethernet interface to be used is selected using the MAC address of the interface.

Table 3-1 - Interface of the VI:

Name	Type	Data type	Function
MAC address	Input	char[ ]	MAC address of the Ethernet interface
error_code	Output	uint_32	Error number

3.2.2 Program details about the block Stop.vi

This function stops the execution of the PROFINET controller. The Ethernet interface to be used is selected using the MAC address of the interface.

Table 3-2 - Interface of the VI:

Name	Type	Data type	Function
error_code	Output	uint_32	Error number

3.2.3 Program details about the block PNIO_STD_TG1.vi

This function is used for communication based on "Standard telegram 1".

Table 3-3 – Interface of the VI:

Name	Type	Data type	Function
input_address	Input	uint_32	Start address of the input memory area
output_address	Input	uint_32	Start address of the output memory area
STW1	Input	uint_16	Control word 1
NSOLL_A	Input	uint_16	Setpoint speed
ZSW1	Output	uint_16	Status word 1
NIST_A	Output	uint_16	Actual speed
state	Output	bool	Read/write status
error_read	Output	uint_32	Error has occurred when reading
error_write	Output	uint_32	Error has occurred when writing

3.2.4 Program details about block PNIO_STD_TG111.vi

This function is used for communication based on "Standard telegram 111".

Table 3-4 - Interface

Name	Type	Data type	Function
input_address	Input	uint_32	Start address of the input memory area
output_address	Input	uint_32	Start address of the output memory area
STW1	Input	uint_16	Control word 1
POS_STW1	Input	uint_16	Pos. control word 1
POS_STW2	Input	uint_16	Pos. control word 2
STW2	Input	uint_16	Control word 2
OVERRIDE	Input	uint_16	Scaling factor (100% = 0x4000h)
MDI_TARPOS	Input	uint_32	Setpoint position
MDI VELOCITY	Input	uint_32	Set velocity
MDI_ACC	Input	uint_16	Acceleration
MDI_DEC	Input	uint_16	Delay
user_write	Input	uint_16	Freely assignable word, user defined
ZSW1	Output	uint_16	Status word 1
POS_ZSW1	Output	uint_16	Pos. status word 1

Name	Type	Data type	Function
POS_ZSW2	Output	uint_16	Pos. status word 2
ZSW2	Output	uint_16	Status word 2
MELDW	Output	uint_16	Message word
XIST_A	Output	uint_32	Actual position
NIST_B	Output	uint_32	Actual speed
FAULT_CODE	Output	uint _16	Error code
WARN_CODE	Output	uint_16	Alarm code
user_read	Output	uint_16	Freely assignable word, user defined
state	Output	bool	Read/write status
error_read	Output	uint_32	Error has occurred when reading
error_write	Output	uint_32	Error has occurred when writing

3.2.5 Program details about block PNIO_FREE_TG.vi

This function is used for communication based on "Free telegram configuration with BICO".

Table 3-5 – Interface of the VI:

Name	Type	Data type	Function
input_address	Input	uint_32	Start address of the input memory area
output_address	Input	uint_32	Start address of the output memory area
number_bytes	Input	uint_32	Number of bytes to be written
data_write	Input	uint_8 [ ]	Data to be written
data_read	Output	uint_8 [ ]	Read data
state	Output	bool	Read/write status
error_read	Output	uint_32	Error has occurred when reading
error_write	Output	uint_32	Error has occurred when writing

3.2.6 Program details about block PNIO_READ.vi

This function is used to read from a PROFINET IO device.

Table 3-6 – Interface of the VI:

Name	Type	Data type	Function
input_address	Input	uint_32	Start address of the input memory area
number_bytes	Input	uint_32	Number of bytes to be read

Name	Type	Data type	Function
data_read	Output	uint_8 [ ]	Read data
state	Output	bool	Read status
error_write	Output	uint_32	Error has occurred when reading

3.2.7 Program details about block PNIO_WRITE.vi

This function is used to write to a PROFINET IO device.

Table 3-7 – Interface of the VI:

Name	Type	Data type	Function
output_address	Input	uint_32	Start address of the output memory area
number_bytes	Input	uint_32	Number of bytes to be written
data_write	Input	uint_8 [ ]	Data to be written
state	Output	bool	Write status
error_write	Output	uint_32	Error has occurred when writing

3.2.8 Program details about the block DESCALE_SGL.vi

This VI descales a word value (e.g. speed) to 0x4000h.

Table 3-8 - Interface of the VI:

Name	Type	Data type	Function
Input	Input	int16	Scaled value
Reference value	Input	float	Reference value, used as basis for scaling
Output	Output	float	Descaled value

3.2.9 Program details about the block DESCALE_DBL.vi

This VI descales a double word value (e.g. speed) to 0x40000000h.

Table 3-9 - Interface of the VI:

Name	Type	Data type	Function
Input	Input	int32	Scaled value
Reference value	Input	double	Reference value, used as basis for scaling
Output	Output	double	Descaled value

3.2.10 Program details about the block SCALE_SGL.vi

This VI scales a word value (e.g. speed) to 0x4000h.

Table 3-10 - Interface of the VI:

Name	Type	Data type	Function
Input	Input	uint_16	Unscaled value
Reference value	Input	float	Reference value, used as basis for scaling
Output	Output	uint_16	Scaled value

3.2.11 Configuring information

When compiling the communication VIs for the complete PROFINET network, it must be ensured that the PROFINET controller is first started, and this must be executed again before closing the application.

In between opening and closing the PROFINET controller, the communication VIs must be cyclically called (e.g. in a loop).

3.3 Functionality of the application example

The functionality and the interface of the application example are described in the following sections.

3.3.1 Program details about the block Example_S120_ET200S.vi

In the VI that is associated with the application example, communication is cyclic. As a first step, the PROFINET controller is initialized, then the communication functions cyclically run in a "while" loop until the user stops the execution. The PROFINET controller is terminated before exiting the VIs.

Communication is established to the following nodes/stations:

- S120 demonstration case – axis 1:
  communication using standard telegram 111 (control of EPOS)
- S120 demonstration case – axis 2:
  communication using standard telegram 1 (speed setpoint)
- ET200S – 1 input module and 1 output module (each 1 byte):
  communication via the VIs PNIO_READ.vi and PNIO_WRITE.vi

3.4 Functionality of the acyclic communication

In addition to cyclic PNIO communication, there is the option of reading and writing the parameter values of the SINAMICS converter.

Just the same as when communicating with a SIMATIC S7, a request block is required, that must be transferred to the SINAMICS drive.

The evaluation of the response is realized using a user event returned in LabVIEW.

Additional details regarding the structure of the DPV1 parameter task and the DPV1 parameter response can be reviewed in the "Function Manual Drive Functions".

The attached example VI is used to read or write a parameter. The data type of the read parameter is automatically identified, and the read value is converted into a 32-bit floating point number.

4 Installation and commissioning

The prerequisites and steps necessary to run the application are explained in this section.

4.1 Installing the hardware

The following diagram shows the hardware configuration of the application. The PC, SINAMICS S120 and ET200S (optional) components belonging to the application must be connected to via PROFINET.

Fig. 4-1

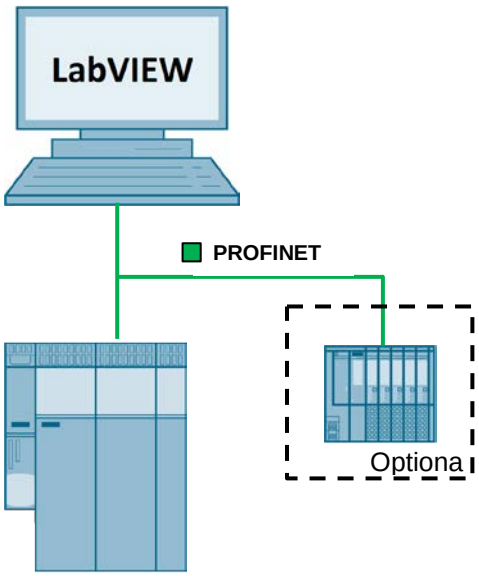


Table 4-1

No.	Action
1.	Connect the PC that you are using with any PROFINET interface of the CU320-2 PN of the SINAMICS demonstration case. Any Ethernet interface of the PC can be used.
2.	Connect the ET200S with any PROFINET interface of the CU320-2 of the SINAMICS demonstration case. Connecting the ET200S is optional, and is not required to operate the converter. This is not a component of the application example in LabVIEW.
3.	You require STEP7 Professional V13 on your engineering PC in order to configure the PC station. This should not correspond to the LabVIEW-PC, and must be connected in the PROFINET network in addition to the components mentioned above. There are 2 options here: - If an ET200S is being used, then connect the engineering PC with the free PROFINET interface of the ET200s - If an ET200S is not being used, then connect the engineering PC with the free PROFINET interface of the CU320-2 PN of the SINAMICS demonstration case. If you want to use one single PC station for the engineering and LabVIEW, then this PC station must have 2 Ethernet interfaces. In this case, connect the two interfaces with the PROFINET network.

4.2 Installing the software (download)

This chapter describes the steps required to install the code example.

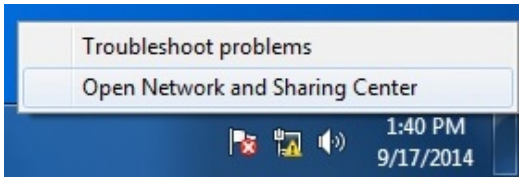
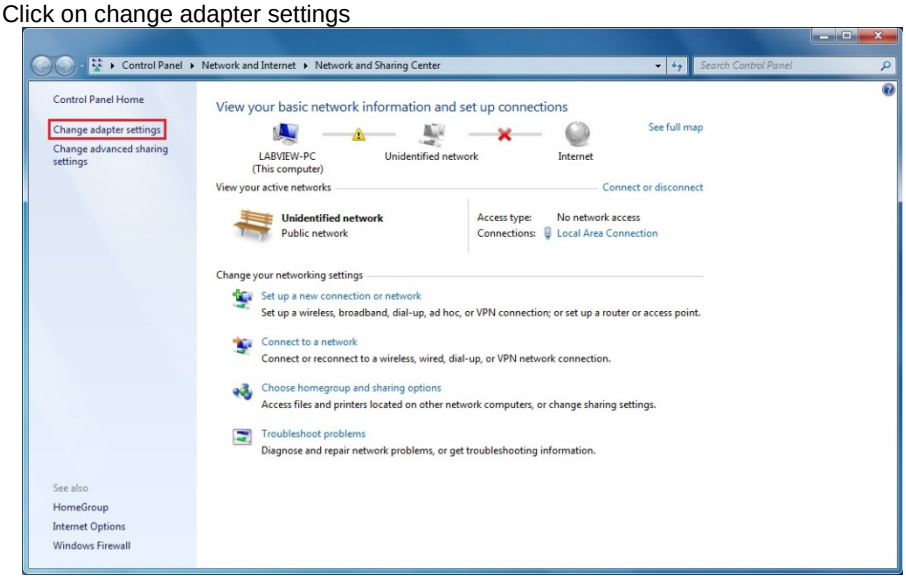
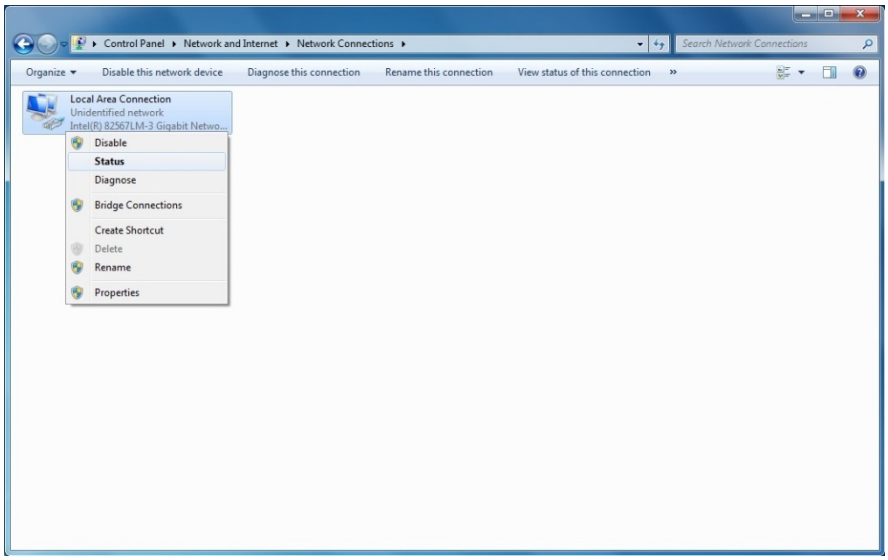
Note At the present time, the hardware support package required for the application is not available for STEP7 V13. STEP 7 Professional V12 is required to configure the PROFINET network.

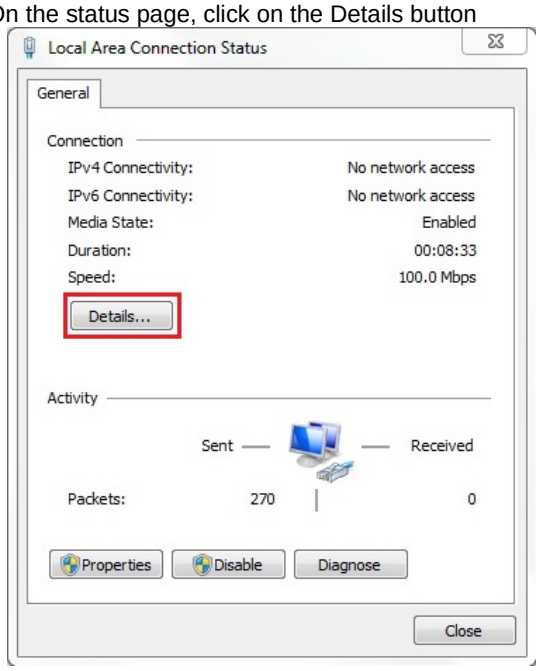
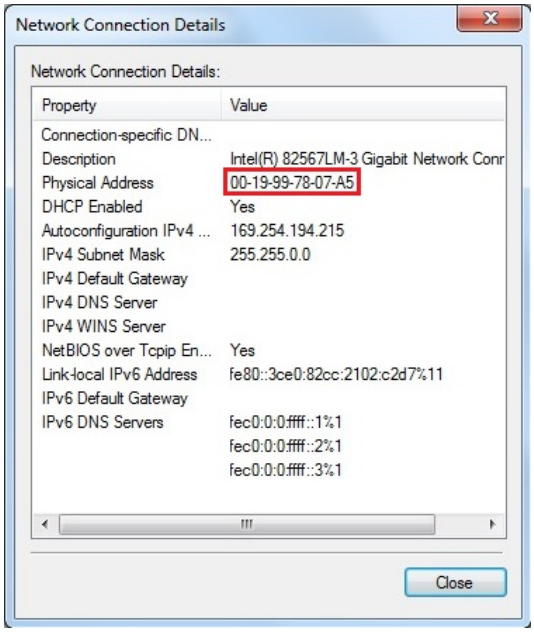
Table 4-2

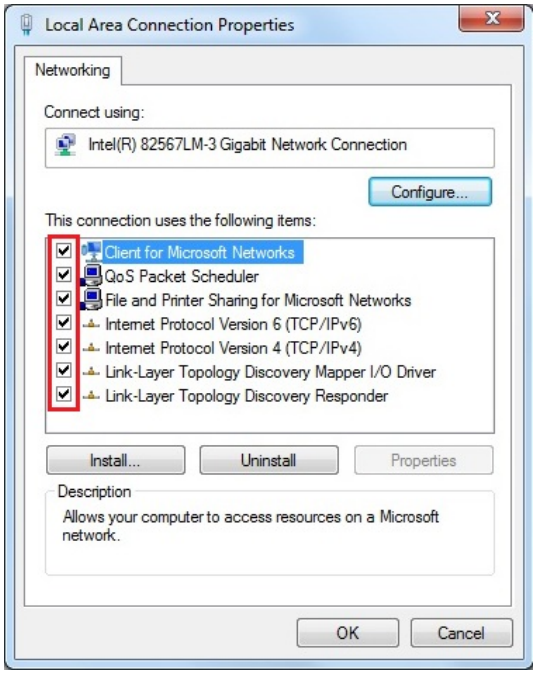
No.	Action
4.	Install LabVIEW in a version >2013 on the PC you are using.
5.	Install WinPCap with version 4.1.3 on the PC you are using. WinPCap can be downloaded from here: http://www.winpcap.org/
6.	Install Microsoft Visual C++ 2010 SP1 Redistributable Package (x86) on the PC you are using. This can be downloaded here: http://www.microsoft.com/en-US/download/details.aspx?id=8328
7.	Install STEP 7 Professional V12 on your engineering PC. This can be downloaded here: http://support.automation.siemens.com/WW/view/en/78793685
8.	Unzip the content of the 99684399_PN-Driver_HSP.zip to a temporary folder. To install the HSP, proceed as described in the following FAQ: http://support.automation.siemens.com/WW/view/en/57424910
9.	Unzip the content of the 99684399_PNIO_BIB_v20.zip to folder "user.lib" in the LabVIEW installation directory. As default, this is as follows: „C:/Program Files/National Instruments/LabVIEW 2013/user.lib/“

4.3 Commissioning

Table 4-3

No.	Action
10.	In the first step, the MAC address of the Ethernet interface selected for PROFINET communication must be obtained. To do this, the PC must be connected to the CU320-2 and the operating voltage connected to the SINAMICS Control Unit. The MAC address of the interface can be identified as follows: - Right-click on the network symbol in the taskbar and you go to the Network and Sharing Center  - Click on change adapter settings  - Right-click on the network connection and then open the network connection status. 

No.	Action
	- On the status page, click on the Details button  - The MAC address of the interface can be taken from the window that then opens. This is subsequently required for communication with the PROFINET I/O devices from LabVIEW. 
11.	Please deactivate all of the elements used by the network connection. To do this, proceed as follows: - As described in the previous step, return to network connections. - Right-click on the network connection and then open the network connection properties.

No.	Action
	- Deactivate all elements: 
12.	Set the IP address of the Ethernet interface of the engineering PC as follows: - Go to the properties of the Ethernet interface of the engineering PC analogously to the previous description. - Select element "Internet protocol Version 4 (TCP/Ipv4)“ and open the properties. - Here, enter the following addresses: IP address: 192.168.0.100 Subnet mask: 255.255.255.0 - Save the changes.
13.	Unzip the archive that has been made available [Entry-ID]_TIA_Project.zip. Start the TIA Portal and open the project. The components required for the example are already preconfigured in the project. The following settings are relevant for the communication: - IP addresses: LabVIEW PC: 192.168.0.1 SINAMICS S120: 192.168.0.10 - PROFINET name: LabVIEW PC: labview SINAMICS S120: s120 After the configuration has been completed, the LabVIEW PC station must be compiled. In the "Devices & networks" view, select the station and in the Edit menu, select the Compile function. After compilation has been completed, the configuration XML required for LabVIEW is provided in the following directory: "<Project directory>/AdditionalFiles/PNDriver/" Please copy the .XML file generated in this directory to directory ".../user.lib/PNIO/" inserted under Point 9.
14.	Then assign the PROFINET IO devices their PROFINET names and IP addresses.

No.	Action
15.	Unzip the 99684399_Starter_Project.zip archive provided. Start STARTER / SCOUT and open the project. Then transfer the project to the SINAMICS converter.

After these points have been completed, the components have been setup and the application can be used in LabVIEW. In the next chapter we will tell you how the supplied LabVIEW library is structured and how the application can be used.

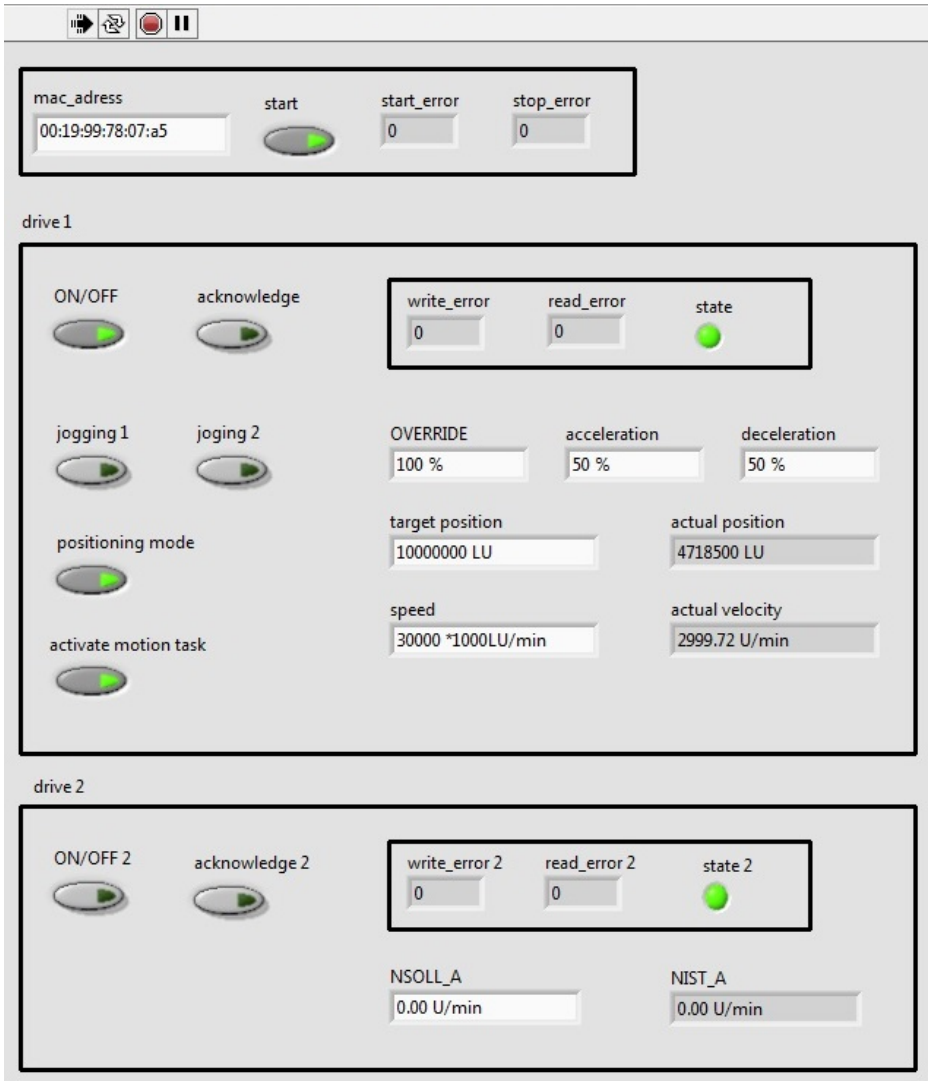
5 Using the application

5.1 Overview

The structure of the application example is explained in this section and you are shown how to operate the LabVIEW user-interface.

Overview and description of the user interface

Fig. 5-1 - User interface of the application example

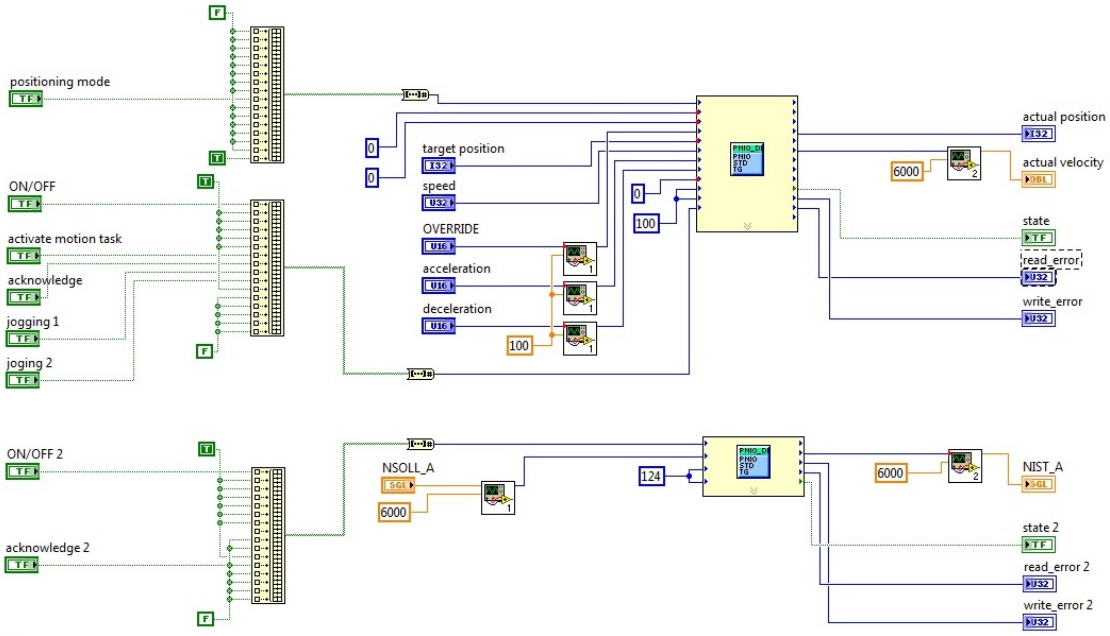


As shown in the diagram, the user interface comprises 3 components:

- The first component is used to operate the PROFINET controller.
- The second component is used to operate the basic positioner via direct setpoint input/MDI (upper axis of the S120 demonstration case).
- The third component is used to operate the speed-controlled axis (lower axis of the S120 demonstration case)

A section of the program code behind the user interface – the LabVIEW block diagram – is shown in the following figure:

Fig. 5-2 – Section of the LabVIEW block diagram of the application example



The calls of the two VIs "PNIO_STD_TG111.vi" and "PNIO_STD_TG111.vi" can be identified in the block diagram. These make up the composition of the telegram and transfer the data to the DLL. The entry fields with a red border are used to enter the I/O addresses, set in the TIA Portal for the I/O devices. These do not have to be changed to use the application example.

The individual steps that are required to use the application example are described in the following sections.

5.2 Starting the PROFINET controller

The following steps are required to activate the PROFINET controller (corresponds to the RUN state of an S7-PLC):

Table 7-1

No.	Action	Remark
1.	Enter the MAC address of the network interface, read out under Point 4.3, Step 10, which should be used for the PROFINET communication, in the field MAC address of the 1st component of the LabVIEW user interface of the application exampleVIs	The MAC address must be entered in the following format: 00:ab:00:ab:00:ab The individual groups are separated by a colon, lowercase letters are used.
2.	Set switch "Start" of the 1st component of the LabVIEW user interface to "ON"	
3.	Start the execution of the VIs	If all of the components have been correctly connected and commissioned, a value of 0 is displayed in field "start_error". The status fields "state" and "state2" of the drives are green, and the fault fields of the drives also indicate a value of 0x0h. (It can take some time to establish communications to the drive device. During this time, fault 0x301h may be displayed. This disappears as soon as the connection has been established to the IO device).

5.3 Operating the basic positioner of the upper drive

The application example involves operating the basic positioner with the most necessary operating functions via direct setpoint input/MDI. Other functions can also be used; however, in this case, a dedicated virtual instrument must be programmed.

The following table lists the various ways in which the application example can be operated:

Table 7-2

No.	Action	Remark
1.	Press the "ON/OFF" button	Switch on and switch off the drive object using OFF1.
2.	Press the "Acknowledge" button	Acknowledge active faults in the drive object
3.	Press the "Jog 1" button	Jogging in direction 1 with the jog velocity set in the drive object
4.	Press the "Jog 2" button	Jogging in direction 2 with the jog velocity set in the drive object
5.	Press the "Positioning type" button	Changing the positioning type "ON" = absolute positioning
6.	Press the "Activate traversing task" button	Starting a traversing task
7.	"OVERRIDE" entry field	Setting the velocity override
8.	"Acceleration" entry field	Scaling factor of the acceleration set in the drive object
9.	"Deceleration" entry field	Scaling factor of the deceleration set in the drive object
10.	"Target position" entry field	Entry of the target position in LU
11.	"Velocity" entry field	Enter the required traversing velocity in 1000 LU/min

5.4 Operating the speed setpoint of the lower drive

The application example involves operating the lower drive by entering a speed setpoint. The following table lists the various ways in which the application example can be operated:

Table 7-3

No.	Action	Remark
1.	Press the "ON/OFF" button	Switch on and switch off the drive object using OFF1.
2.	Press the "Acknowledge" button	Acknowledge active faults in the drive object
3.	"Setpoint speed" entry field	Enter the required setpoint speed in rpm

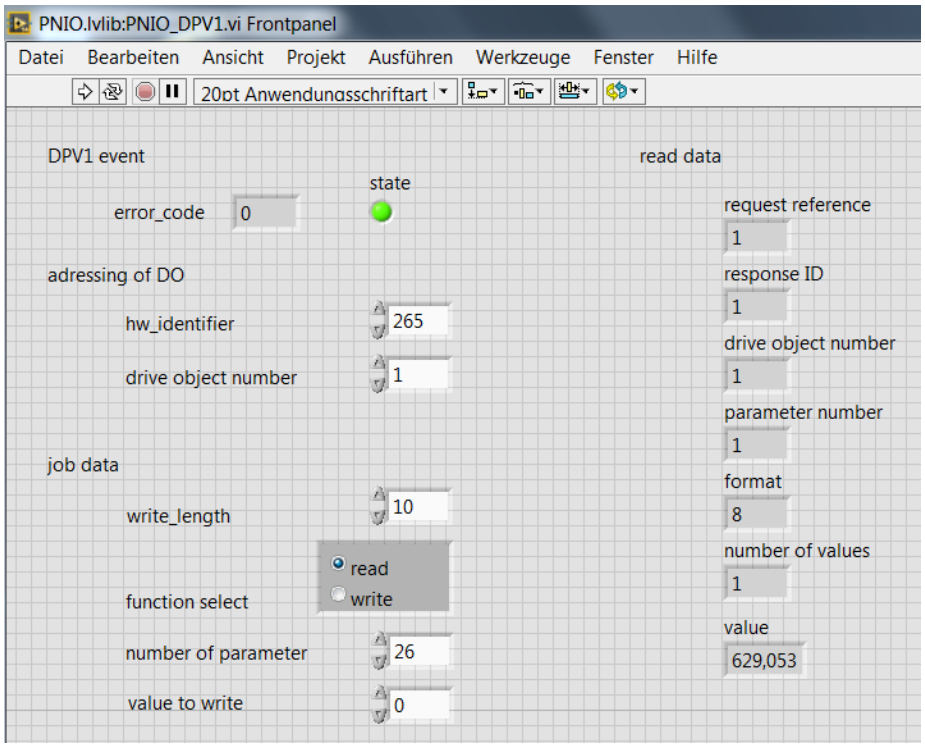
6 Acyclically reading and writing parameters

6.1 Overview

The structure of the VI example "PNIO_DPV1.vi" for acyclic communication is shown in this section, and its operation explained.

Overview and description of the user interface

Fig. 6-1 - User interface of the application example



As shown in the diagram, the following information is required for acyclically reading and writing parameters.

- HW identifier

  The HW identifier is used to identify the drive object. The HW identifier of the ModulAccessPoints of the drive object to be addressed is required here.
- Drive object number

  Number assigned in the drive for the drive object.
- Write length

  The data block length for the parameter task must be specified here
- Function selection

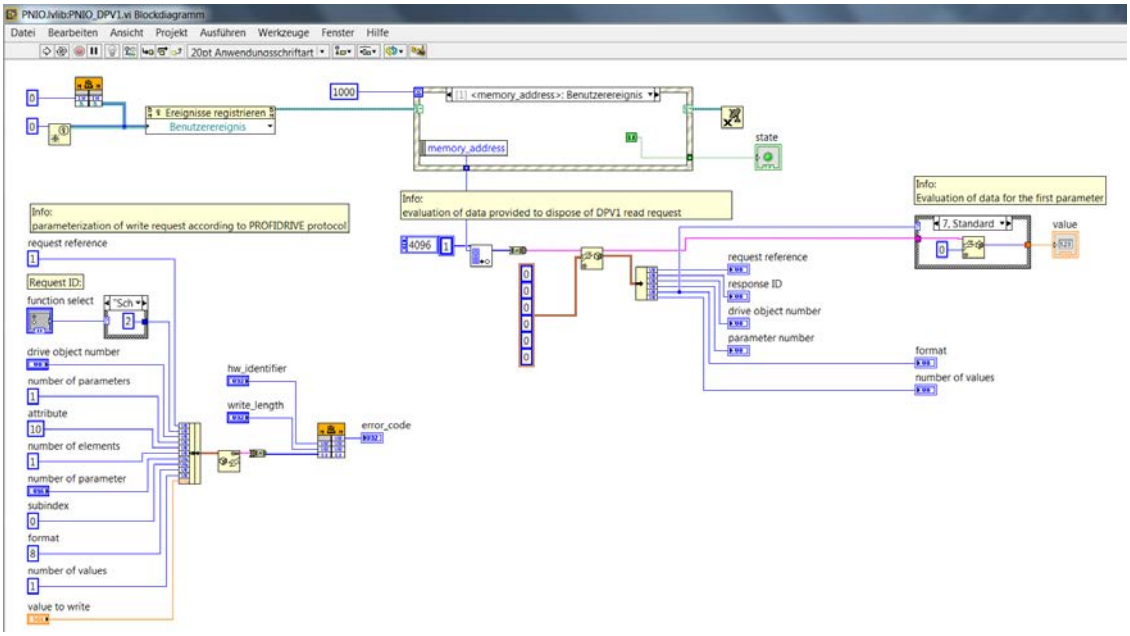
  Selection whether to be read or written
- Parameter number

  Number of the parameter to be accessed

- Value to be written
  Value to be written to the parameter

A section of the program code behind the user interface – the LabVIEW block diagram – is shown in the following figure:

Fig. 6-2 – Section of the LabVIEW block diagram in the VI example



6.2 Starting the DPV1 task

The read/write task is started with the specified data by executing the VI example.

The read data are evaluated and displayed.

All of the data required for assembling the data block and the data types used can be taken from the "Sinamics S120 Function Manual Drive Functions" documentation that was mentioned previously.

7 Error codes

The possible error codes of the communication driver are listed in the following table.

Table 6-1

Error name	Error code	Remark
PNIO_OK	0x00000000	success
PNIO_WARN_IRT_INCONSISTENT	0x00000010	IRT Data may be inconsistent
PNIO_WARN_NO_SUBMODULES	0x00000011	no submodules to be updated
PNIO_WARN_LOCAL_STATE_BAD	0x00000012	data was written with local state PNIO_S_BAD, because not all components of splitted module have local state PNIO_S_GOOD
PNIO_ERR_PRM_HND	0x00000101	parameter Handle is illegal
PNIO_ERR_PRM_BUF	0x00000102	parameter buffer is NULL-Ptr
PNIO_ERR_PRM_LEN	0x00000103	parameter length is wrong
PNIO_ERR_PRM_ADD	0x00000104	parameter address is wrong
PNIO_ERR_PRM_RSTATE	0x00000105	parameter remote state is NULL-Ptr
PNIO_ERR_PRM_CALLBACK	0x00000106	parameter cbf is illegal
PNIO_ERR_PRM_TYPE	0x00000107	parameter type has no valid value
PNIO_ERR_PRM_EXT_PAR	0x00000108	parameter ExtPar has no valid value
PNIO_ERR_PRM_IO_TYPE	0x00000109	parameter PNIO_ADDR::IODataType is wrong
PNIO_ERR_PRM_CP_ID	0x0000010A	parameter CpIndex is wrong, probably driver is not loaded
PNIO_ERR_PRM_LOC_STATE	0x0000010B	parameter IOlocState has no valid value
PNIO_ERR_PRM_REC_INDEX	0x0000010C	parameter RecordIndex has no valid value
PNIO_ERR_PRM_TIMEOUT	0x0000010D	parameter timeout has no valid value
PNIO_ERR_PRM_DEV_ANNOTATION	0x0000010E	parameter annotation has no valid value
PNIO_ERR_PRM_DEV_STATE	0x0000010F	parameter state has no valid value
PNIO_ERR_PRM_PCBF	0x00000110	parameter pCbf has no valid value
PNIO_ERR_PRM_MAX_AR_VALUE	0x00000111	parameter MaxAR has no valid value
PNIO_ERR_PRM_ACCESS_TYPE	0x00000112	parameter AccessType has no valid value
PNIO_ERR_PRM_POINTER	0x00000113	an invalid pointer was passed
PNIO_ERR_PRM_INVALIDARG	0x00000114	an invalid argument was passed
PNIO_ERR_PRM_MEASURE_NUMBER	0x00000115	wrong Measure No in cycle statistics, must be -1 (actual measure) up to 49
PNIO_ERR_PRM_CYCLE_OFFSET	0x00000116	wrong Offset for cycle info buffer (must be 0 to 19)
PNIO_ERR_PRM_ROUTER_ADD	0x00000117	address used by io router

PNIO_ERR_WRONG_HND	0x00000201	unknown handle
PNIO_ERR_MAX_REACHED	0x00000202	maximal number of opens reached; close unused applications
PNIO_ERR_CREATE_INSTANCE	0x00000203	fatal error, reboot your system
PNIO_ERR_MODE_VALUE	0x00000204	parameter mode has no valid value
PNIO_ERR_OPFAULT_NOT_REG	0x00000205	register OPFAULT callback before register STARTOP callback
PNIO_ERR_NEWCYCLE_SEQUENCE_REG	0x00000206	register NEWCYCLE callback before register STARTOP callback
PNIO_ERR_NETWORK_PROT_NOT_AVAILABLE	0x00000207	network protocol not available, check card configuration
PNIO_ERR_NO_CONNECTION	0x00000301	device data not available, because device is not connected to controller
PNIO_ERR_OS_RES	0x00000302	fatal error, no more operation system resources available
PNIO_ERR_ALREADY_DONE	0x00000303	action was already performed
PNIO_ERR_ALLREADY_DONE		
PNIO_ERR_NO_CONFIG	0x00000304	no configuration for this index available
PNIO_ERR_SET_MODE_NOT_ALLOWED	0x00000305	PNIO_set_mode not allowed, use PNIO_CEP_MODE_CTRL by PNIO_controller_open
PNIO_ERR_DEV_ACT_NOT_ALLOWED	0x00000306	PNIO_device_activate not allowed, use PNIO_CEP_MODE_CTRL by PNIO_controller_open
PNIO_ERR_NO_LIC_SERVER	0x00000307	license server not running, check your installation
PNIO_ERR_VALUE_LEN	0x00000308	wrong length value
PNIO_ERR_SEQUENCE	0x00000309	wrong calling sequence
PNIO_ERR_INVALID_CONFIG	0x0000030A	invalid configuration, check your configuration
PNIO_ERR_UNKNOWN_ADDR	0x0000030B	address unknown in configuration, check your configuration
PNIO_ERR_NO_RESOURCE	0x0000030C	no resource too many requests been processed
PNIO_ERR_CONFIG_IN_UPDATE	0x0000030D	configuration update is in progress or CP is in STOP state, try again later
PNIO_ERR_NO_FW_COMMUNICATION	0x0000030E	no communication with firmware, reset cp or try again later
PNIO_ERR_STARTOP_NOT_REGISTERED	0x0000030F	no synchronous function allowed, use PNIO_CEP_SYNC_MODE by PNIO_controller_open or PNIO_device_open
PNIO_ERR_OWNED	0x00000310	interface-submodule cannot be removed because it is owned by an AR

PNIO_ERR_START_THREAD_FAILED	0x00000311	failed to start thread, probably by lack of pthread resources
PNIO_ERR_START_RT_THREAD_FAILED	0x00000312	failed to start realtime thread, probably you need root capability to do it
PNIO_ERR_DRIVER_IOCTL_FAILED	0x00000313	failed to ioctl driver, probably API version mismatch
PNIO_ERR_AFTER_EXCEPTION	0x00000314	exception occurred, save exception info (see manual) and reset cp
PNIO_ERR_NO_CYCLE_INFO_DATA	0x00000315	no cycle data available
PNIO_ERR_SESSION	0x00000316	request belongs to an old session
PNIO_ERR_ALARM_DATA_FORMAT	0x00000317	wrong format of alarm data
PNIO_ERR_ABORT	0x00000318	operation was aborted
PNIO_ERR_CORRUPTED_DATA	0x00000319	data are corrupter or have wrong format
PNIO_ERR_FLASH_ACCESS	0x0000031A	error by flash operations
PNIO_ERR_WRONG_RQB_LEN	0x0000031B	wrong length of request block at firmware interface, firmware not compatible to host sw
PNIO_ERR_NO_RESET_VERIFICATION	0x0000031C	reset request was sent to firmware, but firmware rut up can't be verified
PNIO_ERR_INTERNAL	0x000003FF	fatal error, contact SIEMENS hotline
PNIO_ERR_MAC_ADRESS_NOT_FOUND	0x00000401	the entered mac-address could not be found
PNIO_ERR_XML_NOT_FOUND	0x00000402	the XML-File can not be opened

8 References

Table 9-1

	Topic	Title
\1\	Siemens Industry Online Support	http://support.automation.siemens.com
\2\	SINAMICS S120 Function Manual Drive Functions	http://support.automation.siemens.com/WW/view/en/49084671
\3\	Download page of the article	http://support.automation.siemens.com/WW/view/en/99684399
\4\	SINAMICS application examples	http://siemens.com/sinamics-applications

9 Contact person

Siemens AG

Industry Sector
I DT MC PMA APC
Frauenauracher Strasse 80
D - 91056 Erlangen, Germany
E-mail: tech.team.motioncontrol@siemens.com

10 History

Table 11-1

Version	Date	Revision
V1.0	08/2014	First Edition
V2.0	01/2015	Examples of acyclic communication