1 PROFIdrive controller application

The Goal of the PROFIdrive example application for controllers is to make it easier to create a user application for PROFIdrive drives connected via PROFINET, because all standard functionalities for PROFINET and PROFIdrive are provided by easy to use interface functions.

So, the creator of user application part doesn't need to know all details of PROFINET and especially PROFIdrive profile to make connected drives work in desired way. In addition to that it's independent from specific controller system and widely independent from operating systems requirements.

Table of contents

2 Description of application

Application layer structure

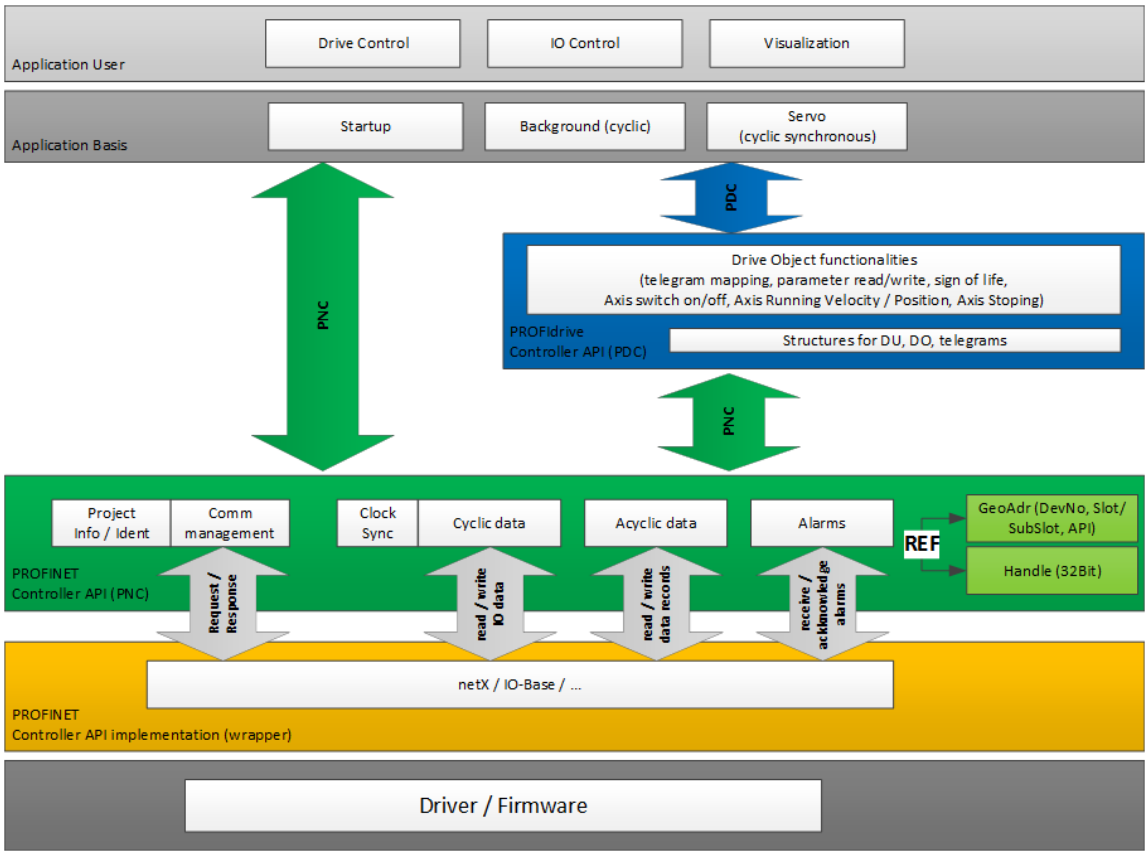
Inside PROFIdrive controller application example there are two interface layers to make usage of PROFIdrive devices, which are connected to a PROFINET controller system, easier.

The PNC API is used to provide PROFINET controller related functionalities like cyclic, acyclic and alarm data handling at application layer in a standardized way. So, it is possible to use the whole application example on different runtime systems by just implementing a proper wrapper for target system.

The PDC API is used to provide main PROFIdrive functionalities in a more abstract way. It provides functions to run connected axis with velocity or position setpoints with respect of chosen PROFIdrive application class and take over all necessary normalization, calculation and mapping in internal modules.

Both application interfaces provide standardized data types und functions to user application layer. That means public PNC function can be used in user application level as well as public PDC functions. By the way PDC is using PNC functionalities as well. In addition to that sources will include of course also internal functions and data types which are used by standard application layer (for basic controller application structure and elements).

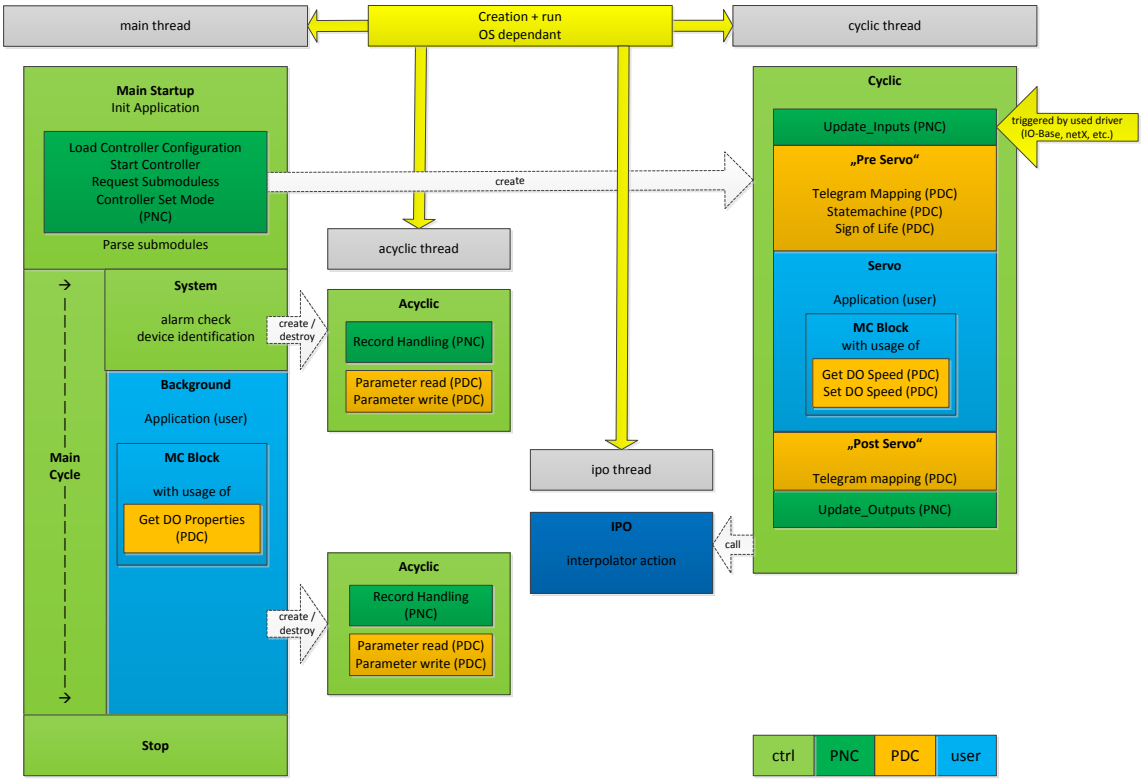
Figure 1 Overview of structure in PROFIdrive Controller example application with PNC and PDC API



Runtime structure

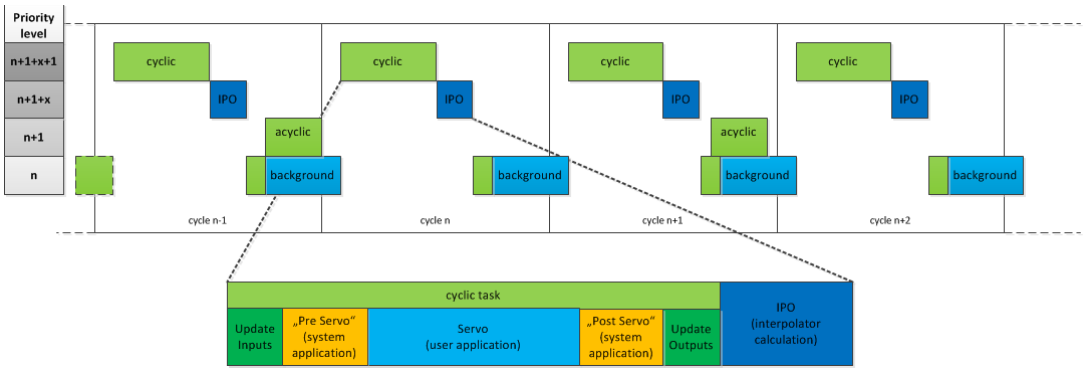
After startup phase the example application runs in a main cycle divided into system and application part. In system part all internal, not time critical tasks are executed. In application part the user can implement any Background tasks by use of appropriate PNC and PDC functions. If there are configured and connected devices with cyclic data exchange the cyclic part is separated into an ordered sequence of PROFINET and appropriate profile parts. The internal system part handles reading and writing of PROFINET input and output data (of submodules) and embeds the internal cyclic PROFIdrive related parts, like mapping raw data to used PROFIdrive telegrams. The user then will be able to implement cyclic related programs in Servo part.

Figure 2 runtime structure of PROFIdrive Controller example application



While running, the application will follow a fixed procedure in execute its parts described also by their priority order.

Figure 3 application procedure call structure



PD controller application

Source structure

The following picture and table describes all source structure elements and the corresponding content referring to the functionality inside the PROFIdrive Controller Application.

Figure 4 source code parts relating to application layer

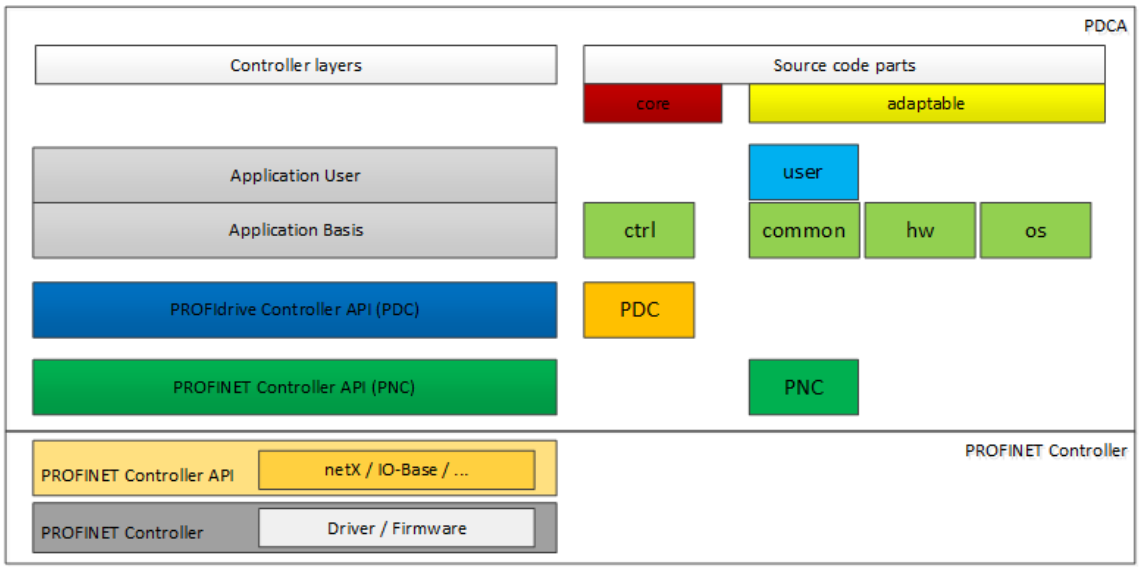


Table 1 source structure

Folder	File	Description
common	common.h	common includes and definitions (**mandatory**) additional common includes and definitions (optional) (for example <math.h>)
	trace.cpp trace.h	Trace modules (**mandatory**) header includes macro definition for trace endpoint (optional) (for example “printf” on a console)
	utils.cpp utils.h	header includes common macros (**mandatory**), implementation of macro content is optional PNC wrapper specific utilities (optional) System specific common functionalities (optional)
ctrl	ctrl_acyc_data.cpp ctrl_acyc_data.h	Processing of data records
	ctrl_alrm_data.cpp ctrl_alrm_data.h	Handling of alarm data
	ctrl_cyc_data.cpp ctrl_cyc_data.h	Handling of cyclic update of data (called / triggered by PNC API) Handling of change of data status Implementation for cyclic thread (for RT only configurations)
	ctrl_cycle.cpp	Cyclic operation of the controller (“System” and “Background”, Figure 2Figure 2) and includes functionality for create reference numbers and device identification
	ctrl_main.cpp	main function of application triggers startup and cyclic operation phase enables stop functionality for controller (PNC API)

PD controller application

Folder	File	Description
	ctrl_startup.cpp	Handling of startup phase Initialization of Application (user modules in “background”, Figure 2) Loading of controller configuration, start controller, readout submodule configuration, controller mode change (PNC API) Creation of cyclic (RT only configuration) and interpolator thread
	ctrl.h	Common header for the controller (basic structures for controller and device(s))
hw_xxx *hw_template*	hw_led.cpp hw_led.h	Handling of LED or other hardware related functionalities
os_template *os_template*	os_sem.cpp os_sem.h	adaption for semaphore handling
	os_sys.cpp os_sys.h	adaption for system functions (e.g. timers)
	os_task.cpp os_task.h	adaption for thread handling
pdc	pdc_acyc.cpp pdc_acyc.h	Handling of PROFIdrive Parameter functionalities
	pdc_api.cpp pdc_api.h	PROFIdrive controller application interface Implementation of PDC functions and data types
	pdc_cyc.cpp pdc_cyc.h	Handling of cyclic PROFIdrive related data Implementation of state machine, fault handling, sign of life, telegram data mapping, interpolator and position controller Implementation for interpolator thread
	pdc_int.h	type and structure definitions for PROFIdrive related content
	pdc_main.cpp	Main implementations of PROFIdrive related content
pnc	pnc_api.cpp pnc_api.h	PROFINET controller application interface Implementation of PNC functions and data types (calls system specific functions of PNC wrapper)
	pnc_wrapper.cpp pnc_wrapper.h	Implementation of wrapper functions **(mandatory)** Content depends completely on used system and is connected to defined functions in PNC
user	background.cpp background.h	User background module (part of cyclic operation)
	servo.cpp servo.h	User Servo module (part of cyclic operation)

Adaptation

The following table list files and content to be maybe modified for used operating systems and PROFINET controller APIs.

Table 2

Folder	File	Usage	Description
common	common.h	mandatory	should be used for all additional general includes and definitions
	trace.cpp trace.h	mandatory	definition of macro “PRINT” decides trace output target working of trace module is enabled / disabled with definition of “TRACE_ENABLED” (makefile) trace with output of source location is enabled / disabled with definition of “TRACE_LOC_ON” (makefile)
	utils.cpp utils.h	optional	common usable functionalities (can be extended with further functionalities)
hw_xxx *hw_template*	hw_led.cpp hw_led.h	optional	content of functions can be filled with hardware related implementations
os_xxx *os_template*	os_sem.cpp os_sem.h	mandatory	must be filled with os dependent implementation for semaphore type and functionalities
	os_sys.cpp os_sys.h	mandatory	content of functions can be filled with operating system related implementations (can be extended with further functionalities)
	os_task.cpp os_task.h	mandatory	must be filled with os dependent implementation for thread type and functionalities
PNC	pnc_wrapper.cpp pnc.wrapper.h	mandatory	will have to be create / modified referring to used system (PROFINET controller hardware driver)

Used parameters

The following table gives an overview about (PROFIdrive) parameters that are used inside application. These shall be implemented in used drives to guarantee proper functionality.

Table 3

Parameter	Usage	Implementation
p964	drive identification	mandatory
p978	drive identification	mandatory
p975	drive identification	mandatory
p922	drive identification	mandatory
r61000	drive (device) identification	optional
r61001	drive (device) identification	optional
r60000	drive (axis) identification	mandatory
p979	drive (axis) encoder identification	mandatory

3 PDC

In PDC user related part it is included just necessary and easy to use functions and data types for handling a PROFIdrive device in a controller application.

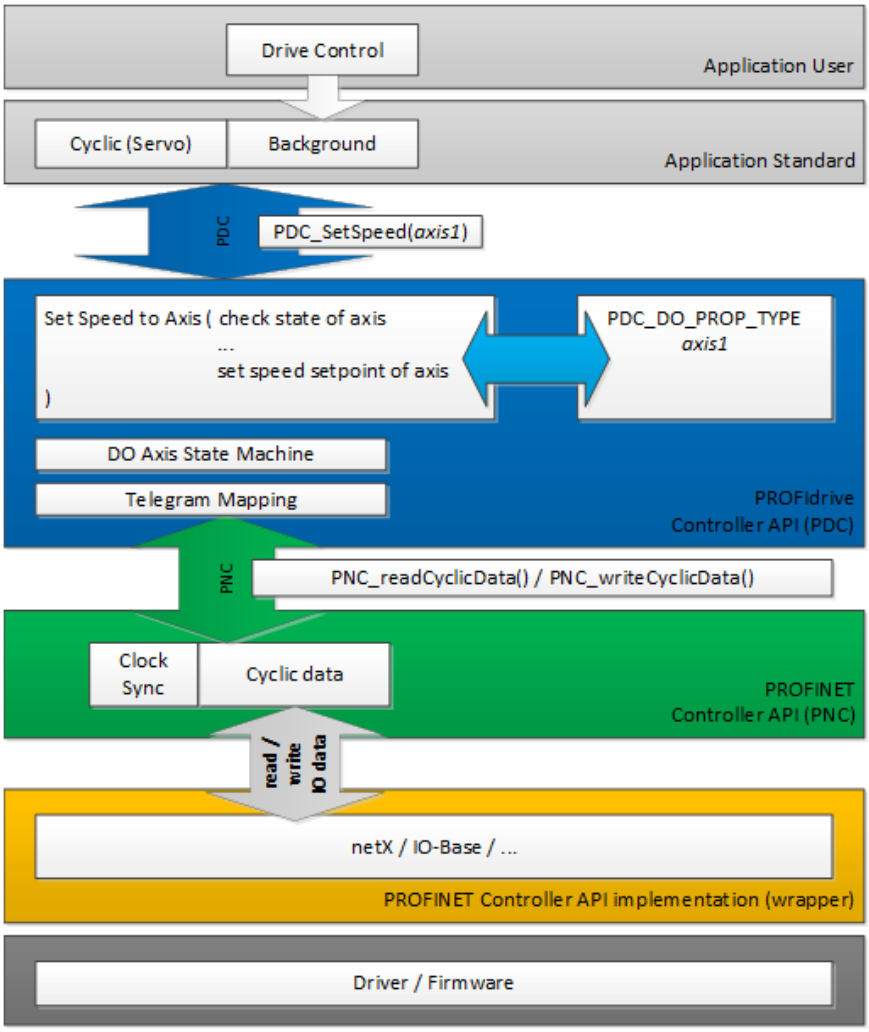
The internal handling of PROFIdrive controller in connection with PROFINET controller interface is implemented by standard application and internal PDC parts. It handles also all functions and data from user related part. The sources for the internal part are also visible / present but they shouldn't be known or modified by user, because it shouldn't be relevant for user.

Example

In following example, a speed set point command is set by user application part, which is processed in standard application cycle part "Background".

The used function "PDC_SetSpeed" with corresponding ID of drive object (i.e. axis) will be handled internally by checking actual state of axis and other necessary things with help of stored internal structure data.

Figure 5 example use of PDC API with function PDC_SetSpeed()



3.1 Data types

3.1.1 General

PROFIdrive Data Types

Data types of PROFIdrive specification (PROFIdrive_3172_V42_Oct15.pdf) are used mainly for acyclic data (parameter access).

The data type definitions are placed in header file of PDC API function interface.

3.1.2 Configuration

Drive Object Reference

Reference to identify a drive object out of all available drive objects.

Table 4

PDC_DO_REF	Description
uint16_t	unique identifier for an axis drive object

Drive Object Properties

This structure represents all available properties as set by PROFIdrive parameters in the DO.

Table 5

PDC_DO_PROP_TYPE	Description	
Reference	PDC_DO_REF	Identification of drive object (PDC_DO_REF)
TelegramNo	uint16_t	Telegram number (p922)
DOtype	uint16_t	(p975.1) Manufacturer specific
DOtypeClass	uint16_t	(p975.5) CU, AXIS, INFEED
HWversion	uint16_t	Hardware version(s) (p975)
SWversion	uint16_t	Software version(s) (p975)
SpeedNormVal	uint16_t	Speed normalization value
SpeedMax	float	Speed maximum
SpeedRes	uint32_t	Speed resolution
G1MeasType	uint32_t	Measurement type of G1 (p979)
G1Res	uint32_t	Resolution of G1 (p979)
G1RefStratType	uint32_t	Reference strategy type of G1 (p979)
SolTolErr	uint32_t	Sign of life tolerated errors (p925)
FaultBufSize	uint32_t	Fault buffer size (p944-p952)

NOTE It may be used to set / write (some) properties of the DOs in future versions.

Drive Object Status

This structure represents actual status values of a drive object.

Table 6

PDC_DO_STAT_TYPE	Description	
InOp	bool	communication with related device is established / brocken
InError	bool	error is present
InPositioning	bool	in positioning task
InVelocity	bool	in velocity task
referenced	bool	axis is referenced

Telegram content

The telegram structure consists of all possible telegram elements, which can appear in implemented telegrams.
The elements are mapped to input / output data by internal functions.

Table 7

PDC_DO_TELEGRAM_TYPE	Description	
Tel_num	uint16_t	Number of telegram
STW1	uint16_t	Control word 1
STW2	uint16_t	Control Word 2
N_SOLL_A	uint16_t	Speed Set point value (16 Bit)
N_SOLL_B	uint32_t	Speed Set point value (32Bit)
G1_STW	uint16_t	Sensor 1 control word
G2_STW	uint16_t	Sensor 2 control word
XERR	uint32_t	Sytem deviation
KPC	uint32_t	Positon controller, gain factor
ZSW1	uint16_t	Status word 1
ZSW2	uint16_t	Status word 2
NIST_A	uint16_t	Actual Speed value (16Bit)
NIST_B	uint32_t	Actual Speed value (32Bit)
G1_ZSW	uint16_t	Sensor 1 Status word
G1_XIST1	uint32_t	Sensor 1 Position actual value 1
G1_XIST2	uint32_t	Sensor 1 Position actual value 2
G2_ZSW	uint16_t	Sensor 2 Status word
G2_XIST1	uint32_t	Sensor 2 Position actual value 1
G2_XIST2	uint32_t	Sensor 2 Position actual value 2
…		
space for manufacturer specific telegrams		
…		

3.1.3 Parameter handling

For each element of a Parameter a corresponding structure is defined.

Parameter value

Table 8

PD_PAR_VALUE	Description
ParamNo	Number of parameter
ParamSubIdx	sub index of parameter
NoOfElements	count of parameter (single or sub index)
NoOfValues	count of values (sub index)
nDataLen	length of data
Format	data format
nFormatLen	Length of data format
i8Value … f32Value	Reference to parameter values (3.1.1)

Parameter description

Table 9

PD_PAR_DESC	Description
ParamDesc	Description (PROFIdrive 4.2 chap. 6.2.1.3)

Parameter text

Table 10

PD_PAR_TEXT	Description
aParamText[16]	character array with parameter text

3.1.4 Fault handling

Faults / Warnings

Tabelle 11

PDC_ERROR_TYPE	Description	
FaultEntryNo	uint32_t	Fault error entry number
FaultNo	uint32_t	Fault error code (p944)

3.1.5 Function return values (status)

PDC function result

Reference to identify a drive object out of all available drive objects.

Table 12

PDC_RESULT	Description
uint32_t	return value typpe for PDC functions

Besides function related return values „status" is describing the actual state of execution of particular function.

All possible returned values are grouped in separate ranges to classify the status more quickly.

Tabelle 13

Return values	Description
0x00000000	Function executed successfully (without any error)
0x0001 – 0x0FFF	Function returns specific status
0x1000 – 0x10FF	Error at xxx
…	…
0x8000 – 0x8FFF	Error at internal execution
…	…
0xFFFFFFFF	Global error (not specified in detail)

3.2 Functions

3.2.1 Configuration

PDC_GetDOProperties

Read properties of one certain drive object.

Table 14

Input	Description	Output	Description
DOref	reference to Drive Object	Status	function status (3.1.5)
		Properties	DO properties (3.1.2)

Table 15

Return value	description
PDC_OK	no error
PDC_ERR_DO_REF_NOT_FOUND	the DO reference was not found in actual / present configuration

PDC_GetAllDOProperties

Read list of all available DOs with their responding properties.

Table 16

Input	Description	Output	Description
		Status	function status (3.1.5)
		AllProperties	Array of DO properties (3.1.2)

Table 17

Return value	description
PDC_OK	no error
PDC_ERR_DO_REF_PART_NOT_FOUND	at least one DO reference was not found in actual / present configuration

PDC_GetTelegram

Read telegram number of one certain drive object.

Table 18

Input	Description	Output	Description
DOref	reference to Drive Object	Status	function status (3.1.5)
		TelNo	Telegram number (3.1.2)

Table 19

Return value	description
PDC_OK	no error
PDC_ERR_DO_REF_NOT_FOUND	the DO reference was not found in actual / present configuration

PDC_GetCacf

Read Controller application cycle factor (CACF) from corresponding drive device, where drive object is present.

Table 20

Input	Description	Output	Description
DOref	reference to Drive Object	Status	function status (3.1.5)
		cacf	periodicy of controller application process in multiple of data exchange cycle time

Table 21

Return value	description
PDC_OK	no error
PDC_ERR_DO_REF_NOT_FOUND	the DO reference was not found in actual / present configuration

NOTE Application is working with CACF 1.

PDC_GetStatus

Read drive object status.

Table 22

Input	Description	Output	Description
DOref	reference to Drive Object	PDC_RESULT	function status (3.1.5)
		Status	DO status (3.1.2)

Table 23

Return value	description
PDC_OK	no error
PDC_ERR_DO_REF_NOT_FOUND	the DO reference was not found in actual / present configuration
PDC_ERR_SM_DO_NOT_AXIS	DO is not an axis drive object

3.2.2 Basic state machine

PDC_Precharge

Switch state from S2 to S3 of one certain drive object.

Table 24

Input	Description	Output	Description
DOref	Reference to DO (axis)	Status	(3.1.5)

Table 25

Return value	description
PDC_OK	no error
PDC_ERR_DO_REF_NOT_FOUND	the DO reference was not found in actual / present configuration
PDC_ERR_SM_DO_NOT_AXIS	DO is not an axis drive object
PDC_ERR_SM_PRECOND	DO is not in expected initial state

PDC_PrechargeDisable

Switch state from S3 to S2 of one certain drive object.

Table 26

Input	Description	Output	Description
DOref	Reference to DO (axis)	Status	(3.1.5)

Table 27

Return value	description
PDC_OK	no error
PDC_ERR_DO_REF_NOT_FOUND	the DO reference was not found in actual / present configuration
PDC_ERR_SM_DO_NOT_AXIS	DO is not an axis drive object
PDC_ERR_SM_PRECOND	DO is not in expected initial state

PDC_PulseEnable

Switch state from S3 to S4 of one certain drive object.

Table 28

Input	Description	Output	Description
DOref	Reference to DO (axis)	Status	(3.1.5)

Table 29

Return value	description
PDC_OK	no error
PDC_ERR_DO_REF_NOT_FOUND	the DO reference was not found in actual / present configuration
PDC_ERR_SM_DO_NOT_AXIS	DO is not an axis drive object
PDC_ERR_SM_PRECOND	DO is not in expected initial state

PDC_PulseInhibit

Switch state from S4 to S3 of one certain drive object.

Table 30

Input	Description	Output	Description
DOref	Reference to DO (axis)	Status	(3.1.5)

Table 31

Return value	description
PDC_OK	no error
PDC_ERR_DO_REF_NOT_FOUND	the DO reference was not found in actual / present configuration
PDC_ERR_SM_DO_NOT_AXIS	DO is not an axis drive object
PDC_ERR_SM_PRECOND	DO is not in expected initial state

PDC_CoastStop

Switch on AUS 2 (Coast down) of one certain drive object.

Table 32

Input	Description	Output	Description
DOref	Reference to DO (axis)	Status	(3.1.5)

Table 33

Return value	description
PDC_OK	no error
PDC_ERR_DO_REF_NOT_FOUND	the DO reference was not found in actual / present configuration
PDC_ERR_SM_DO_NOT_AXIS	DO is not an axis drive object
PDC_ERR_SM_PRECOND	DO is already in CoastStop

PDC_NoCoastStop

Switch off AUS 2 (Coast down) of one certain drive object.

Table 34

Input	Description	Output	Description
DOref	Reference to DO (axis)	Status	(3.1.5)

Table 35

Return value	description
PDC_OK	no error
PDC_ERR_DO_REF_NOT_FOUND	the DO reference was not found in actual / present configuration
PDC_ERR_SM_DO_NOT_AXIS	DO is not an axis drive object
PDC_ERR_SM_PRECOND	DO is already in NoCoastStop

PDC_Quickstop

Switch on AUS 3 (Quick Stop) of one certain drive object.

Table 36

Input	Description	Output	Description
DOref	Reference to DO (axis)	Status	(3.1.5)

Table 37

Return value	description
PDC_OK	no error
PDC_ERR_DO_REF_NOT_FOUND	the DO reference was not found in actual / present configuration
PDC_ERR_SM_DO_NOT_AXIS	DO is not an axis drive object
PDC_ERR_SM_PRECOND	DO is already in QuickStop

PDC_NoQuickstop

Switch off AUS 3 (Quick Stop) of one certain drive object.

Table 38

Input	Description	Output	Description
DOref	Reference to DO (axis)	Status	(3.1.5)

Table 39

Return value	description
PDC_OK	no error
PDC_ERR_DO_REF_NOT_FOUND	the DO reference was not found in actual / present configuration
PDC_ERR_SM_DO_NOT_AXIS	DO is not an axis drive object
PDC_ERR_SM_PRECOND	DO is already in NoQuickStop

3.2.3 Speed

PDC_SetSpeed

Run axis with velocity / revolution preset of one certain drive object.

Table 40

Input	Description	Output	Description
DOref	Reference to DO (axis)	Status	(3.1.5)
velocity	Velocity setpoint		

Table 41

Return value	description
PDC_OK	no error
PDC_ERR_DO_REF_NOT_FOUND	the DO reference was not found in actual / present configuration
PDC_ERR_SC_DO_NOT_AXIS	DO is not an axis drive object
PDC_ERRSC_DO_ALRDY_IN_USE	*DO is already in use*

PDC_GetSpeed

Read axis velocity / revolution of one certain drive object.

Table 42

Input	Description	Output	Description
DOref	Reference to DO (axis)	Status	(3.1.5)
		velocity	Velocity actual value

Table 43

Return value	description
PDC_OK	no error
PDC_ERR_DO_REF_NOT_FOUND	the DO reference was not found in actual / present configuration
PDC_ERR_SC_DO_NOT_AXIS	DO is not an axis drive object

3.2.4 Position

PDC_SetPosition

Set position set point value of one certain drive object.

Table 44

Input	Description	Output	Description
DOref	Reference to DO (axis)	Status	(3.1.5)
position	position set point value		

Table 45

Return value	description
PDC_OK	no error
PDC_ERR_DO_REF_NOT_FOUND	the DO reference was not found in actual / present configuration
PDC_ERR_PC_DO_NOT_AXIS	DO is not an axis drive object

NOTE PDC_setPosition() have to be executed in Servo cycle.

PDC_GetPosition

Read actual axis position of one certain drive object.

Table 46

Input	Description	Output	Description
DOref	Reference to DO (axis)	Status	(3.1.5)
		position	position actual value

Table 47

Return value	description
PDC_OK	no error
PDC_ERR_DO_REF_NOT_FOUND	the DO reference was not found in actual / present configuration
PDC_ERR_PC_DO_NOT_AXIS	DO is not an axis drive object

PDC_SetKv

Sets position control gain of one certain drive object.

Table 48

Input	Description	Output	Description
DOref	Reference to DO (axis)	Status	(3.1.5)
Kv	Position control gain		

Table 49

Return value	description
PDC_OK	no error
PDC_ERR_DO_REF_NOT_FOUND	the DO reference was not found in actual / present configuration
PDC_ERR_PC_DO_NOT_AXIS	DO is not an axis drive object

PDC_SetDSC

Enable / disable DSC mode of one certain drive object.

Table 50

Input	Description	Output	Description
DOref	Reference to DO (axis)	Status	(3.1.5)
OffOn	Disables / enables DSC		

Table 51

Return value	description
PDC_OK	no error
PDC_ERR_DO_REF_NOT_FOUND	the DO reference was not found in actual / present configuration
PDC_ERR_PC_DO_NOT_AXIS	DO is not an axis drive object

PDC_RefLogicControl

Enables referencing of one certain drive object

Table 52

Input	Description	Output	Description
DOref	Reference to DO (axis)	Status	(3.1.5)

Table 53

Return value	description
PDC_OK	no error
PDC_ERR_DO_REF_NOT_FOUND	the DO reference was not found in actual / present configuration
PDC_ERR_PC_DO_NOT_AXIS	DO is not an axis drive object
PDC_ERR_SM_PRECOND	DO is not in expected initial state
PDC_REF_SEARCH_ACITVE	Incremental: Reference mark search active Absolute: reference mark is set (offset is stored)
PDC_REF_SEARCH_FINISHED	value can be read out via PDC_RefLogicValues

PDC_RefLogicValues

Reads referencing values of one certain drive object

Table 54

Input	Description	Output	Description
DOref	Reference to DO (axis)	Status	(3.1.5)
		position	reference position value

Table 55

Return value	description
PDC_OK	no error
PDC_ERR_DO_REF_NOT_FOUND	the DO reference was not found in actual / present configuration
PDC_ERR_RL_DO_NOT_AXIS	DO is not an axis drive object

3.2.5 Parameter handling

PDC_ReadParameterValue

Read Parameter (single or array) value of one certain drive object.

Table 56

Input	Beschreibung	Output	Beschreibung
DOref	Reference to DO	Status	function status (3.1.5)
Mode	BMPA (local, global)		
ParCnt	number of parameter		
ParamValueData	Parameter data type (3.1.3)	ParamValueData	Parameter value data type (3.1.3)

Table 57

Return value	description
PDC_OK	no error
PDC_ERR_DO_REF_NOT_FOUND	the DO reference was not found in actual / present configuration
PDC_ERR_PAR_READ	error at writing of parameter (see detailed error values in parameter value data parameter)
PDC_ERR_PAR_INT	error at internal execution (see detailed error in trace output)

PDC_WriteParameterValue

Write Parameter (single / multi / array) value of one certain drive object.

Table 58

Input	Beschreibung	Output	Beschreibung
DOref	Reference to DO	Status	function status (3.1.5)
Mode	BMPA (local, global)		
ParCnt	number of parameter		
ParamValueData	Parameter data type (3.1.3)	ParamValueData	Parameter value data type (3.1.3)

Table 59

Return value	description
PDC_OK	no error
PDC_ERR_DO_REF_NOT_FOUND	the DO reference was not found in actual / present configuration
PDC_ERR_PAR_WRITE	error at writing of parameter (see detailed error values in parameter value data parameter)
PDC_ERR_PAR_INT	error at internal execution (see detailed error in trace output)

PDC_ReadParameterDescription

Read Parameter description, which contains relevant information about respective parameter of one certain drive object.

Table 60

Input	Beschreibung	Output	Beschreibung
DOref	Reference to DO	Status	function status (3.1.5)
Mode	BMPA (local, global)	ParamDesc	Parameter description data type (3.1.3)
ParamNo	parameter number		

Table 61

Return value	description
PDC_OK	no error
PDC_ERR_DO_REF_NOT_FOUND	the DO reference was not found in actual / present configuration
PDC_ERR_PAR_DESCRIPTION	error at reading parameter desscription
PDC_ERR_PAR_INT	error at internal execution (see detailed error in trace output)

PDC_ReadParameterText

Read PROFIdrive parameter text (general description of parameter function or value for visualization purposes) of one certain drive object.

Table 62

Input	Beschreibung	Output	Beschreibung
DOref	Reference to DO	Status	function status (3.1.5)
Mode	BMPA (local, global)	ParamText	Parameter text data type (3.1.3)
ParamNo	parameter number		

Table 63

Return value	description
PDC_OK	no error
PDC_ERR_DO_REF_NOT_FOUND	the DO reference was not found in actual / present configuration
PDC_ERR_PAR_TEXT	error at reading parameter text
PDC_ERR_PAR_INT	error at internal execution (see detailed error in trace output)

3.2.6 Fault handling

PDC_FaultAcknowledge

Acknowledge of faults of one certain drive object.

Table 64

Input	Description	Output	Description
DOref	Reference to DO (axis)	Status	(3.1.5)

Table 65

Return value	description
PDC_OK	no error
PDC_ERR_DO_REF_NOT_FOUND	the DO reference was not found in actual / present configuration
PDC_ERR_FH_NO_FAULT_PRESENT	no fault present

4 PNC

In PNC the PROFINET functionalities of a Controller are represent by functions and data types for the user application part, which are not related to a specific driver or platform solution. These functions and data types are oriented to PROFINET standards for cyclic, acyclic and alarm channel application relations.

The implementation of these standardized functions then represents the usage of the target PROFINET controller platform with their specific functionalities. Therefore for PNC it will be necessary to write a proper wrapper file for used controller type to adapt it to used platform solution (i.e. Siemens IO Base or Hilscher netX).

So all application content will be usable independent from used controller platform.

4.1 Data Types

4.1.1 Reference

Submodule

Table 66

PNC_GEO_ADDR_SUBMOD	Data type	Description
devNo	uint32_t	device / station number
API	uint32_t	application identifier
slot	uint32_t	slot number
subSlot	uint32_t	subslot number

Table 67

PNC_SUBMOD_REF	Data type	Description
SmGeoAddr	PNC_GEO_ADDR_SUBMOD (Table 66)	geographical address (submodule) according to PROFINET standard
handleId	uint32_t	handle for identification of communication object

Device

Table 68

PNC_GEO_ADDR_DEV	Data type	Description
devNo	uint32_t	device / station number

Table 69

PNC_DEV_REF	Data type	Description
geoAddrDev	PNC_GEO_ADDR_DEV (Table 68)	geographical address (device) according to PROFINET standard
handleID	uint32_t	handle for identification of communication object

4.1.2 Function return values

Table 70

Name	Data type	Description
PNC_RESULT	uint32_t	
Define	**Value**	
PNC_OK	0x00000000	Function executed successfully (without any error)
PNC_ERR	((PNC_OK) – 1)	Error at internal Execution
PNC_xyz		specific error description depending on returning function

4.1.3 Project Ident

PN IO types

Table 71

PNC_IO_TYPE (enum)	Value	Description
PNC_IO_IN	0	input data type
PNC_IO_OUT	1	output data type

PN data types

Table 72

PNC_PN_DATA_TYPE (enum)	Value	Description
PNC_DATA_RT	0	RealTime data
PNC_DATA_IRT	1	isochronous RealTime data

Submodule list

Table 73

PNC_SUBMOD_LIST		Data type	Description
SmRef		PNC_SUBMOD_REF (Table 67)	submodule reference
DATA_PROP			
	dataType	PNC_IO_TYPE (Table 71)	type of data (input/output)
	lenIn	uint32_t	length of input data
	lenOut	uint32_t	length of output data
COM_PROP			
	dataType	PNC_PN_DATA_TYPE (Table 72)	RT/IRT data
	cycleTime	uint32_t	Cycle time in us
	cacf	uint32_t	CACF controller application cycle factor
	redFactor	uint32_t	Reduction Factor
	phase	uint32_t	Phase
	ti	uint32_t	Ti time in us
	to	uint32_t	To time in us
PN_PROP			

PNC_SUBMOD_LIST		Data type	Description
	modId	uint32_t	Module ID
	submodId	uint32_t	Submodule ID

Controller Information

Table 74

PNC_CTRL_INFO	Data type	Description
CtrlName	Array of uint8_t [PNC_PN_NAME_LEN]	controller name (a.k.a. PN name)
CtrlIp	Array of uint8_t [PNC_IP_LEN]	controller IP address
SubnetMask	Array of uint_t [PNC_IP_LEN]]	controller Subnet Mask
GatewayIp	Array of uint8_t [PNC_IP_LEN]]	controller Gateway IP address (a.k.a. Router IP)

Controller Cycle Information

Table 75

PNC_CTRL_CYCLE_INFO	Data type	Description
tcaStart	uint32_t	time controller application start value (referring to cycle start) in us
tcaEnd	uint32_t	time controller application end value (referring to cycle start) in us

PN Controller Configuration

Table 76

PNC_CONFIG	Data type	Description
pConfig	uint8_t *	Pointer to configuration data
configLen	uint32_t	Length of configuration data
pRema	uint8_t *	Pointer to remanent data
remaLen	uint32_t	Length of remanent data

Device Information

Table 77

PNC_DEV_LIST	Data type	Description
DevRef	PNC_DEV_REF (Table 69)	
struct deviceIfProp		
DevName	Array of uint8_t [PNC_PN_NAME_LEN]	device name (a.k.a. PN name)
DevIP	Array of uint8_t [PNC_IP_LEN]	device IP address
SubnetMask	Array of uint_t [PNC_IP_LEN]]	device Subnet Mask
GatewayIP	Array of uint8_t [PNC_IP_LEN]]	device Gateway IP address (a.k.a. Router IP)

4.1.4 Communication Management

Controller Mode

Table 78

PNC_CTRL_MODE (enum)	Value	Description
PNC_PNIO_GOOD	0	offline
PNC_PNIO_BAD	1	operate

4.1.5 Cyclic Communication

PN data status

Table 79

PNC_PN_IOXS_TYPE (enum)	Value	Description
PNC_PNIO_GOOD	0	status ok
PNC_PNIO_BAD	1	status not ok

4.1.6 Acyclic Communication

PN Error Status

Table 80

PDC_PN_ERR_STAT_TYPE	Data type	Description
ErrCode	uint8_t	ErrorCode: Most significant word, most significant byte of PNIO Status
ErrDecode	uint8_t	ErrorDecode: Most significant word, least significant byte of PNIO Status
ErrCode1	uint8_t	ErrorDecode: Least significant word, most significant byte of PNIO Status
ErrCode2	uint8_t	ErrorCode2: Least significant word, least significant byte of PNIO Status
AddValue1	uint16_t	additional information 1
AddValue2	uint16_t	additional information 2

4.1.7 Alarms

PN Alarm Types

Table 81

PNC_PN_ALARM_PRIO_TYPE (enum)	Value	Description
PNC_PNIO_APRIO_LOW	0	Alarm priority low
PNC_PNIO_APRIO_HIGH	1	Alarm priority high

Table 82

PNC_PN_ALARM_TYPE (enum)	Value	Description
PDC_PNIO_ALARM_DIAGNOSTIC	0x01	diagnostic
PNC_PNIO_ALARM_PROCESS	0x02	process
PNC_PNIO_ALARM_PULL	0x03	
PNC_PNIO_ALARM_PLUG	0x04	
PNC_PNIO_ALARM_STATUS	0x05	
PNC_PNIO_ALARM_UPDATE	0x06	
PNC_PNIO_ALARM_REDUNDANCY	0x07	
PNC_PNIO_ALARM_CONTROLLED_BY_SUPERVISOR	0x08	
PNC_PNIO_ALARM_RELEASED_BY_SUPERVISOR	0x09	
PNC_PNIO_ALARM_PLUG_WRONG	0x0A	
PNC_PNIO_ALARM_RETURN_OF_SUBMODULE	0x0B	
PNC_PNIO_ALARM_DIAGNOSTIC_DISAPPEARS	0x0C	
PNC_PNIO_ALARM_MCR_MISMATCH	0x0D	
PNC_PNIO_ALARM_PORT_DATA_CHANGED	0x0E	
PNC_PNIO_ALARM_SYNC_DATA_CHANGED	0x0F	
PNC_PNIO_ALARM_ISOCHRONE_MODE_PROBLEM	0x10	
PNC_PNIO_ALARM_NETWORK_COMPONENT_PROBLEM	0x11	
PNC_PNIO_ALARM_TIME_DATA_CHANGED	0x12	
PNC_PNIO_ALARM_UPLOAD_AND_STORAGE	0x1E	
PNC_PNIO_ALARM_PULL_MODULE	0x1F	
PDC_PNIO_ALARM_DEV_FAILURE	0x00010000	device failure
PDC_PNIO_ALARM_DEV_RETURN	0x00010001	device return

PN Alarm data

Table 83

PNC_CTRL_ALARM_DATA	Data type	Description
alarmType	PNC_PN_ALARM_TYPE (Table 82)	original alarm-identifier from alarm-data-block
alarmPrio	PNC_PN_ALARM_PRIO (Table 81)	alarm priority for net, high or low -> see cm defines
devNum	uint32_t	station number
slotNum	uint32_t	slot-no form alarm-data-block or 0
subslotNum	uint32_t	subslot-no from alarm-data-block or 0

4.2 Functions

4.2.1 Project Ident

PNC_ctrl_load_config

Table 84

Input	Description	Output	Description
pCfgFilename	name of file which includes controller configuration (4.1.3)	Result	function status (4.1.2)

Table 85

Return	Value	Description
PNC_ERR_CFG_LOAD	0x0000_0000	load of configuration fail

PNC_submodList_read_req

Request actual submodule list

Table 86

Input	Description	Output	Description
		Result	function status (4.1.2)

PNC_submodList_read_res

Respond actual submodule list (callback)

Table 87

Input	Description	Output	Description
		Result	function status (4.1.2)
		ppSmLst (4.1.3)	submodule list data (4.1.1)
		pSmLstLen	submodule list count

PNC_deviceList_read_req

Request actual device list

Table 88

Input	Description	Output	Description
		Result	function status (4.1.2)

PNC_deviceList_read_rsp

Respond actual device list (callback)

Table 89

Input	Description	Output	Description
		Result	function status (4.1.2)
		ppDevLst (4.1.3)	device list data (4.1.1)
		pDevLstLen	device list count

4.2.2 Communication Management

PNC_ctrl_start

Start Controller

Table 90

Input	Description	Output	Description
		Result	function status (4.1.2)

PNC_ctrl_stop

Stop Controller

Table 91

Input	Description	Output	Description
		Result	function status (4.1.2)

PNC_ctrl_set_mode

Set Controller Mode (STOP. RUN)

Table 92

Input	Description	Output	Description
CtrlMode	New mode for controller (4.1.4)	Result	function status (4.1.2)

PNC_dev_act

Activate Device

Table 93

Input	Description	Output	Description
DevRef	activation of a device (4.1.1)	Result	function status (4.1.2)

PNC_dev_deact

Deactivate Device

Table 94

Input	Description	Output	Description
DevRef	Deactivation of a device (4.1.1)	Result	function status (4.1.2)

4.2.3 Cyclic Communication

PNC_cycle_ind

Trigger update of cyclic data

Table 95

Input	Description	Output	Description
		Result	function status (4.1.2)

PNC_cycData_read

Read cyclic data

Table 96

Input	Description	Output	Description
SmRef	submodule reference (4.1.1)	Result	function status (4.1.2)
pData	read data	pCycInf	cycle timing information (4.1.3)
dataLen	read data length	pIopsRem	remote provider status (4.1.5)
		pIocsRem	remote consumer status (4.1.5)

PNC_cycData_write

Write cyclic data

Table 97

Input	Description	Output	Description
SmRef	submodule reference (4.1.1)	Result	function status (4.1.2)
pData	read data	pCycInf	cycle timing information (4.1.3)
dataLen	read data length		
pIopsLoc	local provider status (4.1.5)		
pIocsLoc	local consumer status (4.1.5)		

4.2.4 Acyclic Communication

PNC_rec_write_req

Request to write record data set

Table 98

Input	Description	Output	Description
SmRef	submodule reference (4.1.1)	Result	function status (4.1.2)
recIdx	record index		
recLen	record data length		
pData	record write data		
reqRef	user application reference		

PNC_rec_write_rsp

Response to write record data set

Table 99

Input	Description	Output	Description
		Result	function status (4.1.2)
		SmRef	submodule reference (4.1.1)
		pErrStat	PNIO status (4.1.6)
		dataLen	record data length
		reqRef	user application reference

PNC_rec_read_req

Request to read record data set

Table 100

Input	Description	Output	Description
SmRef	submodule reference (4.1.1)	Result	function status (4.1.2)
recIdx	record index		
recLen	record data length		
pData	record write data		
reqRef	user application reference		

PNC_rec_read_rsp

Response to read record data set

Table 101

Input	Description	Output	Description
		Result	function status (4.1.2)
		SmRef	submodule reference (4.1.1)
		pErrStat	PNIO status (4.1.6)
		dataLen	record data length
		pData	record read data
		reqRef	user application reference

4.2.5 Alarms

PNC_alarm_ind

Receive alarm data

Table 102

Input	Description	Output	Description
		Result	function status (4.1.2)
		SmRef	submodule reference (4.1.1)
		pData	record read data (4.1.7)

5 History

Table 103 History

Version	Date	Changes
V1.0	07/2019	First public version