HIWIN MIKROSYSTEM

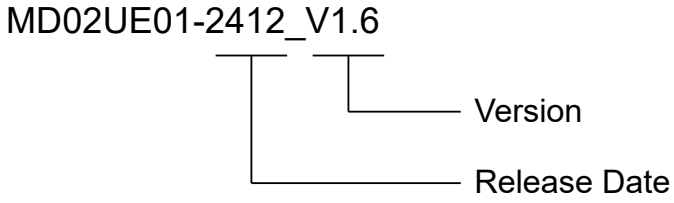
E Series Servo Drive

PROFINET Communication Command Manual

www.hiwinmikro.tw
MD02UE01-2412_V1.6

Revision History

The version of the manual is also indicated on the bottom of the front cover.



Release Date	Version	Applicable Product	Revision Contents
Dec. 20th, 2024	1.6	E1 PROFINET drive E2 PROFINET drive	1. Update section 5.1 **Fault number / code of drive alarm**. 2. Update section 6.12 **Traverse to fixed stop setting**.
Aug. 31st, 2024	1.5	E1 PROFINET drive E2 PROFINET drive	1. Delete section 1.3 **General precautions** and 1.4 **Safety precautions**. 2. Update section 2.2 **Communication specification**. 3. Update section 2.3 **Communication indicators**. 4. Update section 3.1 **IO data signals**. 5. Update section 3.2 **Supported telegrams**. 6. Update the titles of section 3.3.1, 3.4.1, 3.4.2, 3.5.1, 3.6.1, 3.6.2, and 3.6.3. 7. Add section 3.6.4 **Telegram 111**. 8. Update section 3.13 **Positioning control word 2 (POS_STW2)**. 9. Update section 3.15 **Positioning status word 2 (POS_ZSW2)**. 10. Update section 4.2 **PNU dictionary table**. 11. Update section 6.8 **JOG**. 12. Update section 6.11 **Reference torque setting**. 13. Update section 6.12 **Traverse to fixed stop setting**.
Mar. 15th, 2024	1.4	E1 PROFINET drive E2 PROFINET drive	1. Update manual's name. 2. Update section 2.2 **Communication specification**. 3. Update section 2.3 **Communication indicators**. 4. Add section 2.8 **PROFINET GSD files**. 5. Update section 3.1 **IO data signals**.

Release Date	Version	Applicable Product	Revision Contents
			6. Update section 3.2 **Supported telegrams**. 7. Update section 3.3 **Control word 1 (STW1)**. 8. Update section 3.4 **Control word 2 (STW2)**. 9. Update section 3.5 **Status word 1 (ZSW1)**. 10. Update section 3.6 **Status word 2 (ZSW2)**. 11. Update section 3.12 **Positioning control word 1 (POS_STW1)**. 12. Update section 3.14 **Positioning status word 1 (POS_ZSW1)**. 13. Update section 3.16's title: **Message word (MELDW)**. 14. Update section 4.2 **PNU dictionary table**. 15. Update section 5.1 **Fault number / code of drive alarm**. 16. Add section 6.10 **Digital outputs controlling / monitoring**. 17. Add section 6.11 **Reference torque setting**. 18. Update section 7.1 **Configure PROFINET communication by Thunder**.
Dec. 11st, 2023	1.3	E1 PROFINET drive	1. Update section 3.1 **IO data signals**. 2. Update section 3.2 **Supported telegrams**. 3. Update section 3.3.2 **Telegram 9, Telegram 111**. 4. Update section 4.2 **PNU dictionary table**. 5. Update section 6.1 **Velocity reference value setting**. 6. Update section 6.2 **Velocity limit setting**. 7. Update section 6.3 **Torque limit setting**. 8. Update section 6.4 **Quick stop**. 9. Update section 6.6.1 **MDI setpoints**. 10. Update section 6.8 **JOG**. 11. Add section 6.9 **Absolute encoder initialization**. 12. Update section 7.1 **Configure PROFINET communication by Thunder**.
Jan. 19th, 2023	1.2	E1 PROFINET drive	1. Section 2.2 **Communication specification**: Revise cable length and supported telegram. 2. Section 3.5.1 **Status word 1 (ZSW1) - Telegram 3**: Revise bit description. 3. Section 6.1 **Velocity reference value setting**: Revise the related Pt parameters. 4. Section 6.8 **JOG**: Revise the related Pt

Release Date	Version	Applicable Product	Revision Contents
			parameters. 5. Section 7.1 **Configure PROFINET communication by Thunder**: Revise "PROFINET setup" window in Thunder and its description.
Dec. 30th, 2021	1.1	E1 PROFINET drive	1. Add the information of HIWIN Telegram 111. 2. Revise the unit description of MDI_VELOCITY. 3. Add the description of JOG.
Jun. 18th, 2021	1.0	E1 PROFINET drive	First edition.

Related Documents

Through related documents, users can quickly understand the positioning of this manual and the correlation between manuals and products. Go to HIWIN MIKROSYSTEM's official website → Download → Manual Overview for details (https://www.hiwinmikro.tw/Downloads/ManualOverview_EN.htm).

Table of Contents

1. About this manual

1.1 Preface

PROFINET (as a portmanteau for Process Field Net) is an industry technical standard for data communication over Industrial Ethernet. The standard is maintained and supported by PROFIBUS & PROFINET International (PI), an umbrella organization headquartered in Karlsruhe, Germany. This manual mainly describes PROFINET communication and PROFIdrive profile applied to E series PROFINET drives. For a more complete understanding of E series servo drive, please refer to “E1 Series Servo Drive User Manual” and “E2 Series Servo Drive User Manual.”

1.2 Trademark

PROFINET ® is a registered trademark of PROFIBUS & PROFINET International (PI).

2. PROFINET communication

2.1 Brief introduction

PROFINET is a real-time fieldbus protocol based on Ethernet. It classifies all the devices into controllers, supervisors, and field devices.

- **Controller**
  A controller contains process IO image table and user program. A PLC is a typical controller which controls the whole application.
- **Supervisor**
  A supervisor can be a programming device (PG), personal computer (PC) or human machine interface (HMI) for commissioning or diagnosis purposes.
- **Field device**
  A field device is a communication slave controlled by the controller. A field device can transmit the processed data and system statuses (such as diagnoses and alarms) according to PROFINET protocol.

E series PROFINET drives are PROFINET field devices. A device description file, called GSD (**G**eneral **S**tation **D**escription) file, describes the functionality of E series PROFINET drives. A controller uses GSD file to identify and configure field devices. Please refer to section 2.8 **PROFINET GSD files** for the way of obtaining GSD files.

2.2 Communication specification

Table 2.2.1

PROFINET	Physical layer	100BASE-TX (IEEE 802.3)	
	Baud rate	100 Mbps	
	Cable	Ethernet Category 5 or higher (twisted-pair cable with double, aluminum tape and braided shielding)	
	Cable length	Max. 100 m (node to node)	
	Connector	RJ45	
	Communication service	Real-time communication (RT) Isochronous real-time communication (IRT)	
	Send clock	RT: 500 μs, 1 ms, 2 ms, 4 ms IRT: Min. 500 μs (500 μs increment)	
PROFIdrive	Supported telegram	Main telegram	Standard telegram 3 Standard telegram 5 Standard telegram 9 HIWIN telegram 102 HIWIN telegram 105 HIWIN telegram 111
		Supplementary telegram	HIWIN telegram 750
	Control mode	Speed mode, Position mode	

2.3 Communication indicators

■ **Panel configuration for ED1F drive**

Figure 2.3.1 is the panel of an E1 drive. 7-segment display will display drive status and the current alarm code. Table 2.3.1 describes the status of each LED, and Table 2.3.3 describes the status of 7-segment display. Other elements are not functional yet.

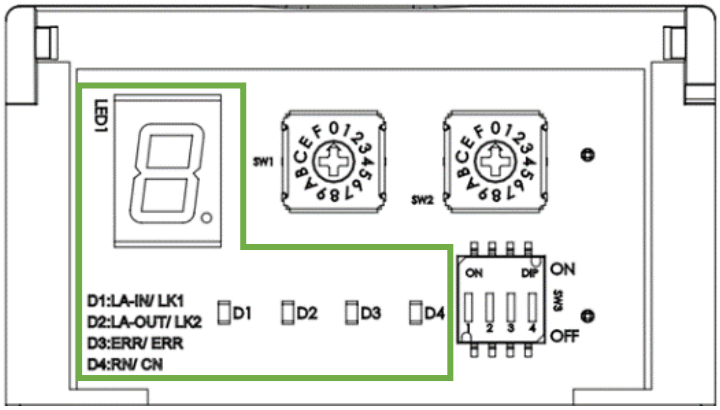


Figure 2.3.1

Table 2.3.1

LED	Color	Status	Description
LA-IN	Green	Solid green	The bus connector is connected and correctly wired.
		Off	The power or the bus connector is not connected.
LA-OUT	Green	Solid green	The bus connector is connected and correctly wired.
		Off	The power or the bus connector is not connected.
ERR	Red	Solid red	The communication is disconnected.
		Blinking	The communication is not established.
		Off	The power is not connected or the communication status is normal.
RN	Green	Solid green	PROFINET-IRT communication is established.
		Flickering	PROFINET-RT communication is established.
		Blinking	PLC program is not activated by the controller.
		Off	The power is not connected or the communication is establishing.

■ Panel configuration for ED2F drive

Figure 2.3.2 is the panel of an E2 drive. 7-segment display will display drive status and the current alarm code. Table 2.3.2 describes the status of each LED, and Table 2.3.3 describes the status of 7-segment display. Other elements are not functional yet.

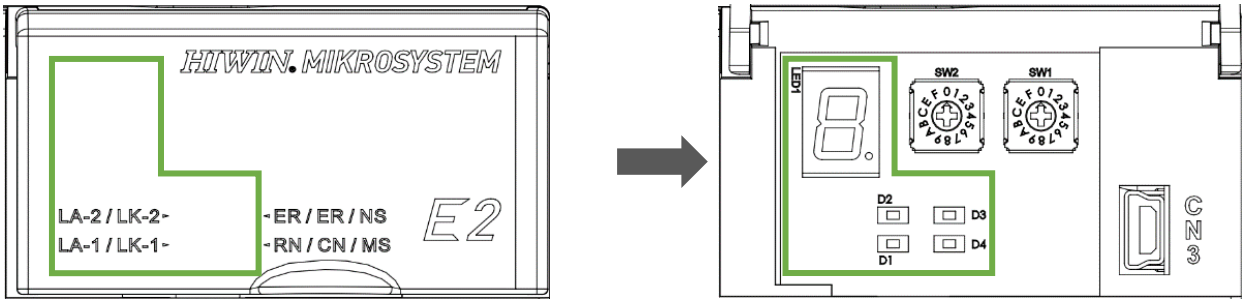


Figure 2.3.2

Table 2.3.2

LED	Color	Status	Description
LA-1 (IN)	Green	On	The bus connector is connected and correctly wired.
		Off	The power or the bus connector is not connected.
LA-2 (OUT)	Green	On	The bus connector is connected and correctly wired.
		Off	The power or the bus connector is not connected.
ER	Red	Solid red	The communication is disconnected.
		Blinking	The communication is not established.
		Off	The power is not connected or the communication status is normal.
RN	Green	Solid green	PROFINET-IRT communication is established.
		Flickering	PROFINET-RT communication is established.
		Blinking	PLC program is not activated by the controller.
		Off	The power is not connected or the communication is establishing.

Figure 2.3.3 displays the LED status.

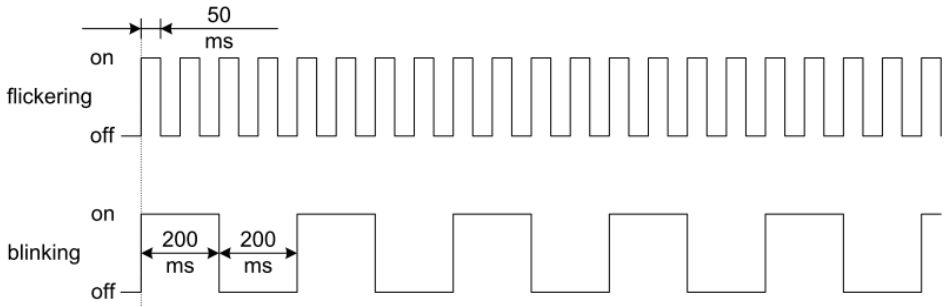
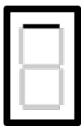
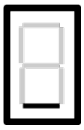
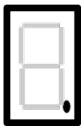


Figure 2.3.3

■ Status display of 7-segment display

Table 2.3.3

Display	Function Description
	Status of rotation detection output (TGON) signal Light up when the rotary velocity of the servo motor exceeds the setting value. (Set via Pt502 or Pt581. The default setting is 20 rpm or 20 mm/s.) Do not light up when the rotary velocity of the servo motor is below the setting value.
	Servo ready display Light up when servo OFF. Do not light up when servo ON.
	Display of command input Light up during command input.
	Display of connection Light up during connection.

2.4 PROFINET device model

PROFINET device model illustrates the relationship between controllers, supervisors, and field devices.

2.4.1 Slot, subslot and index

A device model is represented by DAP (**D**evice **A**ccess **P**oint) which defines modules to a particular device. It also addresses all IO signals in field devices, and this requires corresponding specifications to be made during data modeling. The addressing options are illustrated as follows.

- **Slot (Module)**
  Slot is the physical slot of an IO module of a field device. A module may contain one or more subslots for data exchange.
- **Subslot**
  Subslot is used as a communication object for IO data exchange, parameter access and alarm mechanism.
- **Index**
  Index specifies the accessible data in a slot/subslot. For example, parameters can be written to a module or read from a module by indexes.

Figure 2.4.1.1 shows a modular device model with a bus interface and three input/output modules.

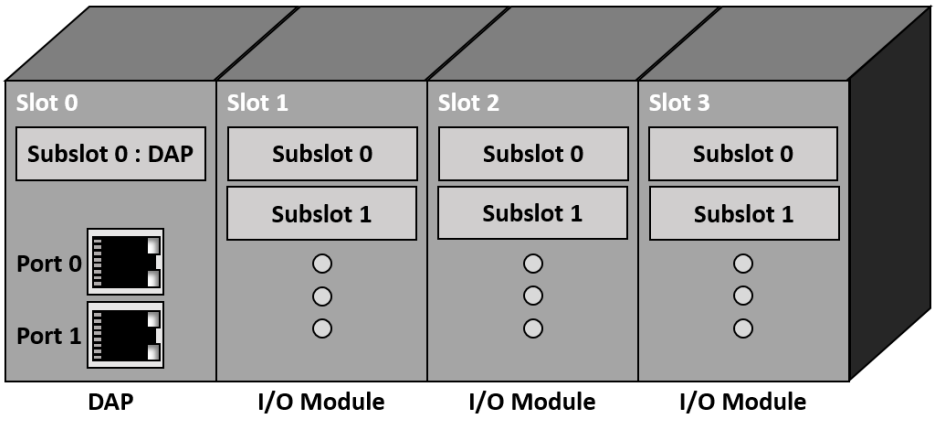


Figure 2.4.1.1

2.4.2 Application process identifier (API)

API (**A**pplication **P**rocess **I**dentifier) is the application profiles registered with PI. PROFIdrive DO is represented by a module with PROFIdrive application process (API=0x3A00).

2.4.3 Application relationship and communication relationship

Each data exchange between a controller/supervisor and a field device will be executed in the well-defined communication channels, which must be set up by the controller before data exchange. The following concept will provide an understanding of PROFINET device model and communication service.

- **Application relationship (AR)**

  Each data exchange is embedded in an application relationship (AR). AR is established between a controller and a field device. The following different ARs are defined in PROFINET communication.

Table 2.4.3.1

	Connect to API	Cyclic Data	Acyclic Data	Alarm	Write Access
IOC-AR	V	V	V	V	V
IOS-AR		V	V	V	V
IOS-DA			V		V
Implicit AR			V		

IOC-AR (**C**ontroller **AR**) defines the relationship between a controller and a field device. IOS-AR (**S**upervisor **AR**) defines the relationship between a supervisor and a field device. IOS-DA (**S**upervisor **D**ata **A**ccess) also defines the relationship between a supervisor and a field device, but this AR only supports acyclic data access. Finally, implicit AR is for read acyclic data between controller/supervisor and field device. This AR is always established and used by a controller. For E series PROFINET drives, IOC-AR is used to exchange cyclic data, acyclic data and alarms, and this AR is set as PROFIdrive API (0x3A00).

- **Communication relationship (CR)**

  Communication relationship (CR) for data exchange must be established within an AR, which specifies the explicit communication channel between a consumer and a provider. The following different CRs are defined in PROFINET communication.

Table 2.4.3.2

	Cyclic Data	Acyclic Data	Multicast
IO-CR	V		
Alarm-CR		V	
Record Data-CR		V	
MCR	V		V

IO-CR is defined for cyclic processed data exchange. Alarm-CR is defined for acyclic alarm transmission. Record Data-CR is defined for acyclic data exchange. Finally, MCR (**M**ulticast **C**ommunication **R**elationship) defines the communication between field devices. As E series PROFINET drive application, PROFINET defines IO-CR for cyclic data exchange and Record Data-CR for acyclic data exchange.

2.4.4 Relationship between device model and addressing

When configuring an automation system, engineers specify the data to be exchanged of the field device. Controller can set multiple IO-CRs for the field device. The actual applications in the field device are recognized based on the API. Figure 2.4.4.1 shows the relationship among IO-CRs, APIs, slots and subslots.

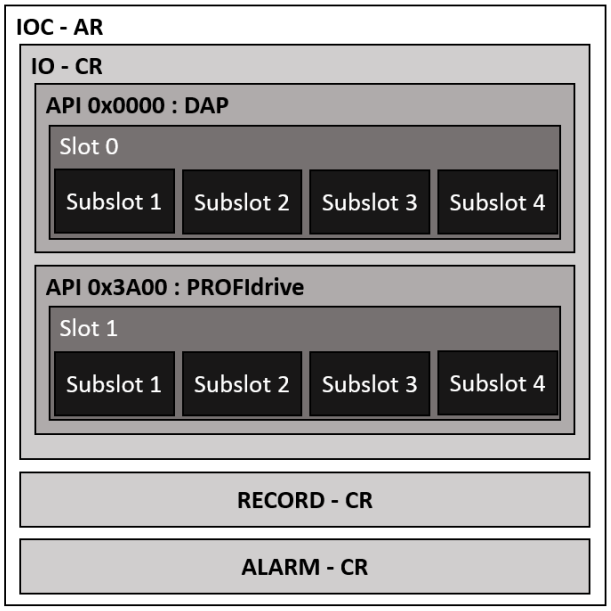


Figure 2.4.4.1

2.5 PROFINET communication services

PROFINET uses three different communication channels to exchange data. Standard Ethernet channel is used for parameterization and configuration of devices for acyclic operation. PROFINET RT (**R**eal-**T**ime) and PROFINET IRT (**I**sochronous **R**eal-**T**ime) are used for real-time application such as a motion control system.

2.5.1 PROFINET real-time class

In real-time communication, the response time must be within 5 to 10 ms. For this reason, it is necessary to add mechanisms to PROFINET that support both UDP/IP communications and offer an optimized communication path. PROFINET uses Ethertype 0x8892 (which has a higher priority than UDP/IP frames) and Frame_ID to address a particular communication channel. The following shows three kinds of real-time classes of PROFINET communication channels.

- **RT_CLASS_1**
  Non-synchronized RT communication within a subnet. No special addressing information is required for this communication. Standard switches suitable for industrial environments can be used in this RT class.

- **RT_CLASS_3**
  Synchronized communication within a subnet. During synchronized communication, the processed data is sent with the maximum precision in an exact order specified during system engineering. This optimized data transmission is called IRT functionality, and the following section will introduce PROFINET IRT communication.

- **RT_CLASS_UDP**
  Non-synchronized cross-subnet communication between different subnets requires addressing information about destination network. This variant is also called RT_CLASS_UDP, and it describes the properties of the local send list control in a device and the switching.

2.5.2 PROFINET acyclic data

Acyclic data exchange is used for parameterizing, configuring devices, and reading status information. This is accomplished with read/write frames via standard IT services by UDP/IP service. In addition to the data records available for use by device manufactures, the system data records are specially defined diagnostic information, error log entries, identification information, information function and IO data signals. The Ethernet frame of PROFINET UDP/IP service is shown as follows.

2 Bytes	28 Bytes	80 Bytes	2 Bytes	1~1364 Bytes	4 Bytes
Ethertype	UDP/IP	RPC	NDR	PROFINET data block	FCS

To an IPv4 data, Ethertype is 0x0800 and PROFINET data block is different from the transmitted type of PROFINET protocol. For example, parameterizing and configuring devices use the read request and write request in normal.

2.5.3 PROFINET cyclic data

After one AR and IO-CR are successfully created, cyclic data starts transmitting without acknowledgement. As previously mentioned, the processed data is assigned to subslots. Cyclic data defines IOPS (**IO P**rovider **S**tatus) and IOCS (**IO C**onsumer **S**tatus) for each subslot to specify data status more precisely. Between controller and field device, each input data or output data has its own IOPS and IOCS. Data transmitters transfer IOPS to data receivers, and data receivers transfer IOCS back to data transmitters. IOPS and IOCS will be “Good” in normal, but they must be set as “Bad” in the following conditions.

- **IOPS**
  - A submodule is not available for an established AR.
  - The application in controller detects the received submodule data is not valid, and IOPS of controller also informs field device about the validity of the output data from controller.
- **IOCS**
  - A submodule is not available for an established AR.
  - Device application cannot process the data.

The Ethernet frame of PROFINET cyclic data is shown as follows. There are two types, one is the frame from controller to field device, and the other is the frame from field device to controller.

4 Bytes	2 Bytes	2 Bytes	1 Byte	…	1 Byte	4 Bytes	4 Bytes
VLAN	Ethertype	Frame_ID	*IOCS	*Data	*IOPS	ADPU status	FCS

4 Bytes	2 Bytes	2 Bytes	1 Byte	…	1 Byte	4 Bytes	4 Bytes
VLAN	Ethertype	Frame_ID	*IOPS	*Data	*IOCS	ADPU status	FCS

Cyclic data has VLAN tag, and the Ethertype of PROFINET protocol is 0x8892. Each output data has an IOPS, and each input data has an IOCS. ADPU status defines the application protocol data unit status.

2.6 PROFINET IRT communication

To satisfy the requirement of the maximum performance and deterministic behavior, PROFINET defines synchronized PROFINET communication called IRT communication (**I**sochronous **R**eal-**T**ime **C**ommunication). The bus cycle of IRT communication is significantly less than 1 ms, and the maximum deviation from the start of the bus cycle is less than 1 us. To provide the maximum performance, PROFINET communication requires precise planning of communication paths in advance. Figure 2.6.1 shows a user scenario in which both the bus cycle and the specific application in the field devices are synchronized.

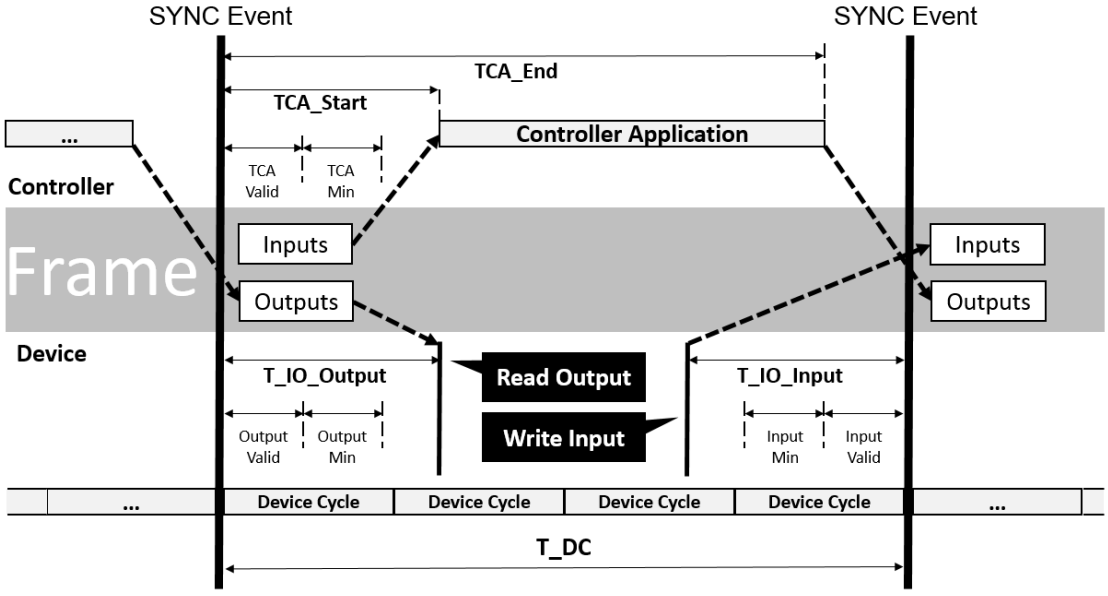


Figure 2.6.1

2.7 PROFINET system startup

Engineers need to set up PROFINET system step by step. The procedure can be separated into five steps, system engineering, download system information, address resolution, system startup and start to data exchange. These steps are explained in detail in the following sections.

2.7.1 System engineering

During system engineering, the GSD files of field devices are necessary. There is a need to map the modules/submodules defined in GSD files onto the real system and assign them to slots/subslots.

2.7.2 Download system information

After the completion of system engineering, engineers need to download the system data to the controller which also contains the system-specific application. In the end of this step, the controller has all the information needed for addressing the field devices and for data exchange.

2.7.3 Address resolution

Before it can perform data exchange with a field device, a controller must assign the field device an IP address before system startup. System startup refers to the start/restart of an automation system after power on or reset the system. The IP address is assigned within the subnet using PROFINET DCP protocol. If the field device is in a different subnet from that of controller, address resolution by a separate DHCP server will be offered.

2.7.4 System startup

A controller always initiates the system startup following start/restart based on the configuration data. This happens automatically from the perspective of the user. During system startup, a controller establishes ARs, CRs, configurations and IO data, then the PROFINET system is ready for data exchange.

2.7.5 Start to data exchange

After the successful completion of system startup, the controller and field devices exchange the processed data, alarms, and acyclic data. Figure 2.7.5.1 shows the sequence of startup after power on or reset the system.

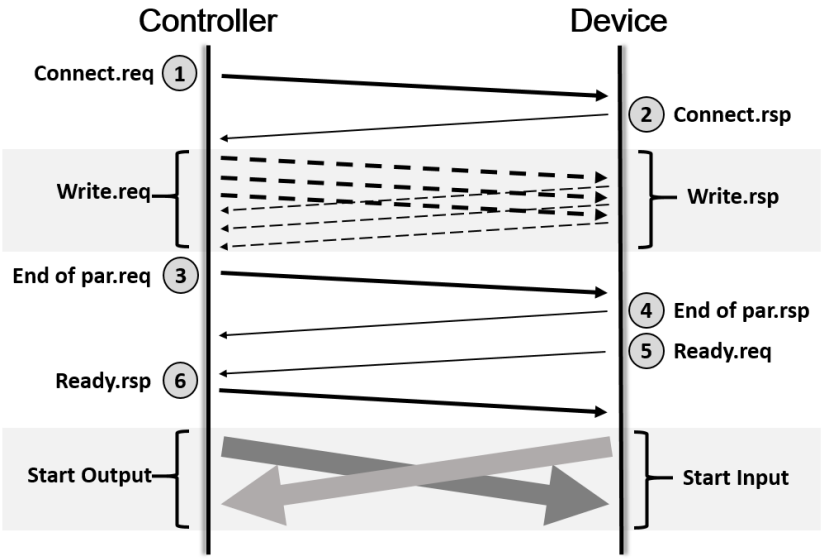
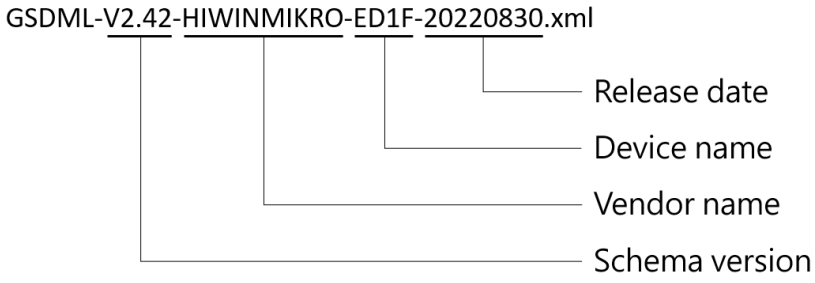


Figure 2.7.5.1

2.8 PROFINET GSD files

GSD files of E series PROFINET drives are maintained and released by HIWIN MIKROSYSTEM. After successfully installing Thunder software on a Windows system computer, obtain GSD files and release notes of GSD files through the path “C:\Thunder\doc\GSD Files\....”

- **Name explanation of GSD files**



- **Release notes of GSD files**

  Users can check the applicable firmware version and the updated information of GSD files. Please refer to “GSD Files Release Notes.pdf.”

(This page is intentionally left blank.)

3. Supported telegrams and IO data

3.1 IO data signals

Table 3.1.1 describes the IO data signals of E series servo drive telegrams.

Table 3.1.1

Signal	Description	Data Type	Access	Valid Range	Unit
STW1	Control word 1	Uint16	Receive	0 ~ 65535	-
STW2	Control word 2	Uint16	Receive	0 ~ 65535	-
ZSW1	Status word 1	Uint16	Send	0 ~ 65535	-
ZSW2	Status word 2	Uint16	Send	0 ~ 65535	-
NSOLL_B	Speed setpoint B (32 bit)*1	Int32	Receive	-2^{31} ~ (2^{31}-1)	$100/2^{30}$ %
NIST_B	Speed actual value B (32 bit) *1	Int32	Send	-2^{31} ~ (2^{31}-1)	$100/2^{30}$ %
G1_STW	Encoder 1 control word	Uint16	Receive	0 ~ 65535	-
G1_ZSW	Encoder 1 status word	Uint16	Send	0 ~ 65535	-
G1_XIST1	Encoder 1 actual position 1	Uint32	Send	-2^{31} ~ (2^{31}-1)	Control unit
G1_XIST2	Encoder 1 actual position 2	Uint32	Send	-2^{31} ~ (2^{31}-1)	Control unit
SATZANW	Position block selection	Uint16	Receive	0 ~ 65535	-
AKTSATZ	Selected position block	Uint16	Send	0 ~ 65535	-
XIST_A	Position actual value A	Int32	Send	-2^{31} ~ (2^{31}-1)	Control unit
MDI_TARPOS	MDI position	Int32	Receive	-2^{31} ~ (2^{31}-1)	Control unit
MDI_VELOCITY	MDI velocity	Int32	Receive	0 ~ (2^{31}-1)	1000control unit/min
MDI_ACC	MDI acceleration override*2	Int16	Receive	0 ~ 2^{14}	$100/2^{14}$ %
MDI_DEC	MDI deceleration override*2	Int16	Receive	0 ~ 2^{14}	$100/2^{14}$ %
MDI_MODE	Position MDI mode	Uint16	Receive	0 ~ 65535	-
POS_STW1	Positioning control word 1	Uint16	Receive	0 ~ 65535	-
POS_STW2	Positioning control word 2	Uint16	Receive	0 ~ 65535	-
POS_ZSW1	Positioning status word 1	Uint16	Send	0 ~ 65535	-
POS_ZSW2	Positioning status word 2	Uint16	Send	0 ~ 65535	-
MELDW	Message word	Uint16	Send	0 ~ 65535	-
OVERRIDE	Velocity override*3	Uint16	Receive	0 ~ (2^{15}-1)	$100/2^{14}$ %
FAULT_CODE	Fault code	Uint16	Send	0 ~ 65535	-
WARN_CODE	Warning code	Uint16	Send	0 ~ 65535	-
MOMRED	Torque reduction*4	Uint16	Receive	0 ~ 2^{14}	$100/2^{14}$ %
M_ACT	Actual torque	Int16	Send	-2^{15} ~ (2^{15}-1)	$100/2^{14}$ %
M_ADD1	Additional torque	Int16	Receive	-2^{15} ~ (2^{15}-1)	$100/2^{14}$ %

Signal	Description	Data Type	Access	Valid Range	Unit
M_LIMIT_POS	Positive torque limit	Int16	Receive	0 ~ (2^{15}-1)	$100/2^{14}$ %
M_LIMIT_NEG	Negative torque limit	Int16	Receive	-2^{15} ~ 0	$100/2^{14}$ %
KPC	Position controller gain factor	Int32	Receive	-2^{31} ~ (2^{31}-1)	(1/1000) s^{-1}
XERR	Position error	Int32	Receive	-2^{31} ~ (2^{31}-1)	Control unit

Note:

*1 Act on velocity reference value (Pt317 or Pt386), the setting ratio will be active based on actual command speed. For relative information, please refer to section 6.1 **Velocity reference value setting**.

*2 Act on acceleration / deceleration reference value (Pt534 / Pt537), the setting ratio will be active based on actual acceleration / deceleration. For relative information, please refer to section 6.6.1 **MDI setpoints**.

*3 Act on the command value of MDI velocity, the setting ratio will be active based on actual command speed. For relative information, please refer to section 6.6.1 **MDI setpoints**.

*4 Act on reference torque (Pt42A or Pt43A), the setting ratio will be active based on actual torque / force output. For relative information, please refer to section 6.11 **Reference torque setting**.

3.2 Supported telegrams

Table 3.2.1 shows the supported telegrams of E series PROFINET drives. “Received word” represents the processed data (PZD) sent from controller to field devices; while “Sent word” represents the processed data sent from field devices to controller. Table 3.2.2 is the frame of the supported telegrams, and Table 3.2.3 is the frame of the supported supplementary telegrams. Supplementary telegram cannot be operated independently; it should be used with main telegram.

Table 3.2.1

Telegram		Maximum number of PZD	
		Received word	Sent word
Main telegram	Standard telegram 3	5	9
	Standard telegram 5[*1]	9	9
	Standard telegram 9	10	5
	HIWIN telegram 102[*1]	6	10
	HIWIN telegram 105[*1]	10	10
	HIWIN telegram 111	12	12
Supplementary telegram	HIWIN telegram 750[*1]	3	1

Note:

[*1] For the firmware version that supports the telegram, please refer to section 2.8 **PROFINET GSD files**.

Table 3.2.2

	Telegram 3		Telegram 5		Telegram 102		Telegram 105	
Item	Received	Sent	Received	Sent	Received	Sent	Received	Sent
PZD1	STW1	ZSW1	STW1	ZSW1	STW1	ZSW1	STW1	ZSW1
PZD2	NSOLL_B	NIST_B	NSOLL_B	NIST_B	NSOLL_B	NIST_B	NSOLL_B	NIST_B
PZD3								
PZD4	STW2	ZSW2	STW2	ZSW2	STW2	ZSW2	STW2	ZSW2
PZD5	G1_STW	G1_ZSW	G1_STW	G1_ZSW	MOMRED	MELDW	MOMRED	MELDW
PZD6		G1_XIST1	XERR	G1_XIST1	G1_STW	G1_ZSW	G1_STW	G1_ZSW
PZD7						G1_XIST1	XERR	G1_XIST1
PZD8		G1_XIST2	KPC	G1_XIST2				
PZD9						G1_XIST2	KPC	G1_XIST2
PZD10								

	Telegram 9		Telegram 111	
Item	Received	Sent	Received	Sent
PZD1	STW1	ZSW1	STW1	ZSW1
PZD2	SATZANW	AKTSATZ	POS_STW1	POS_ZSW1
PZD3	STW2	ZSW2	POS_STW2	POS_ZSW2
PZD4	MDI_TARPOS	XIST_A	STW2	ZSW2
PZD5			OVERRIDE	MELDW
PZD6	MDI_VELOCITY		MDI_TARPOS	XIST_A
PZD7				
PZD8	MDI_ACC		MDI_VELOCITY	NIST_B
PZD9	MDI_DEC			
PZD10	MDI_MOD		MDI_ACC	FAULT_CODE
PZD11			MDI_DEC	WARN_CODE
PZD12				

Table 3.2.3

	Telegram 750	
Item	Received	Sent
PZD1	M_ADD1	M_ACT
PZD2	M_LIMIT_POS	
PZD3	M_LIMIT_NEG	

Note:

(1) 1 PZD = 1 word.

(2) For SATZANW, only MDI submode is functional.

3.3 Control word 1 (STW1)

3.3.1 Telegram 3, Telegram 5, Telegram 102, Telegram 105

Table 3.3.1.1

bit	Description	
0	1: ON	0: OFF
1	1: No Coast stop (No OFF2)	0: Coast stop (OFF2)
2	1: No Quick stop (No OFF3)	0: Quick stop (OFF3)
3	1: Enable operation	0: Disable operation
4	Reserved	
5	Reserved	
6	Reserved	
7	Fault acknowledge (rising edge active)	
8	Reserved	
9	Reserved	
10	1: Control by PLC	0: Control by Device
11	Reserved	
12~15	Reserved	

3.3.2 Telegram 9, Telegram 111

Table 3.3.2.1

bit	Description	
0	1: ON	0: OFF
1	1: No Coast stop (No OFF2)	0: Coast stop (OFF2)
2	1: No Quick stop (No OFF3)	0: Quick stop (OFF3)
3	1: Enable operation	0: Disable operation
4	1: Do Not Reject Traversing Task	0: Reject Traversing Task
5	1: No Intermediate Stop	0: Intermediate Stop
6	Activate Traversing Task (rising edge active)	
7	Fault acknowledge (rising edge active)	
8	1: JOG 1 ON (jog positive)*1	0: JOG 1 OFF
9	1: JOG 2 ON (jog negative)*1	0: JOG 2 OFF
10	1: Control by PLC	0: Control by Device
11	1: Start Homing Procedure	0: Stop Homing Procedure
12~15	Reserved	

Note:

*1 If Pt53A = 1 (which will reverse the jog direction of JOG 1 and JOG 2), the definition of JOG moving direction will be changed. For example, when Pt53A = 1, JOG 1 is negative; JOG 2 is positive; the default of Pt53A is 0.

Pt No.	Pt53A	PNU Number	0x253A		
Data Type	Uint16	Setting Range	0~1	Default	0
Name	PROFIdrive JOG mode moving direction inverse setting	Unit	-	Applicable Motor	All
Effective	Immediately	Attribute	Setup	Applicable Telegram	9, 111

3.4 Control word 2 (STW2)

3.4.1 Telegram 3, Telegram 5, Telegram 9

Table 3.4.1.1

bit	Description
0~11	Reserved
12~15	Controller Sign-of-Life

3.4.2 Telegram 102, Telegram 105, Telegram 111

Table 3.4.2.1

bit	Description
0~7	Reserved
8	1: Activate Traverse to fixed stop
9~11	Reserved
12~15	Controller Sign-of-Life

3.5 Status word 1 (ZSW1)

3.5.1 Telegram 3, Telegram 5, Telegram 102, Telegram 105

Table 3.5.1.1

bit	Description	
0	1: Ready to switch on	0: Not ready to switch on
1	1: Ready to operation	0: Not ready to operation
2	1: Operation enable	0: Operation disable
3	1: Fault	0: No fault
4	1: No Coast stop act (No OFF2)	0: Coast stop act (OFF2)
5	1: No Quick stop act (No OFF3)	0: Quick stop act (OFF3)
6	1: Switch on inhibited	0: Switch on not inhibited
7	1: Warning present	0: No warning
8	1: Speed error within tolerance range	0: Speed error out of tolerance range
9	1: Control requested	0: No Control requested
10	1: Speed is reached	0: Speed is not reached
11	Reserved	
12~15	Reserved	

3.5.2 Telegram 9, Telegram 111

Table 3.5.2.1

bit	Description	
0	1: Ready to switch on	0: Not ready to switch on
1	1: Ready to operation	0: Not ready to operation
2	1: Operation enable	0: Operation disable
3	1: Fault	0: No fault
4	1: No Coast stop act (No OFF2)	0: Coast stop act (OFF2)
5	1: No Quick stop act (No OFF3)	0: Quick stop act (OFF3)
6	1: Switch on inhibited	0: Switch on not inhibited
7	1: Warning present	0: No warning
8	1: Following error in tolerance range	0: Following error out of tolerance range
9	1: Control requested	0: No Control requested
10	1: Target position reached	0: Not at target position
11	1: Home position set	0: Home position not yet set
12	Traversing task acknowledgement (rising edge active)	
13	1: Motor stopped	0: Motor moving
14~15	Reserved	

3.6 Status word 2 (ZSW2)

3.6.1 Telegram 3, Telegram 5

Table 3.6.1.1

bit	Description
0~11	Reserved
12~15	Drive Sign-of-Life

3.6.2 Telegram 9

Table 3.6.2.1

bit	Description	
0~10	Reserved	
11	1: Pulses enabled	0: Pulses disabled
12~15	Drive Sign-of-Life	

3.6.3 Telegram 102, Telegram 105

Table 3.6.3.1

bit	Description
0~7	Reserved
8	1: Traverse to fixed stop activated
9~11	Reserved
12~15	Drive Sign-of-Life

3.6.4 Telegram 111

Table 3.6.4.1

bit	Description	
0~7	Reserved	
8	1: Traverse to fixed stop activated	
9~10	Reserved	
11	1: Pulses enabled	0: Pulses disabled
12~15	Drive Sign-of-Life	

3.7 Encoder 1 control word (G1_STW)

Table 3.7.1

bit	Description	
0	Function 1 (Reference mark 1)	
1	Function 2 (Reference mark 2)	
2	Function 3 (Reference mark 3)	
3	Function 4 (Reference mark 4)	
4	0: No function 1: Activate functions 2: Read value 3: Cancel functions Other: Reserved	
5		
6		
7	1: Reserved	0: Reference mark search
8	Reserved	
9	Reserved	
10	Reserved	
11	Reserved	
12	Reserved	
13	Reserved	
14	1: Activate parking sensor	
15	1: Acknowledging a sensor error	

3.8 Encoder 1 status word (G1_ZSW)

Table 3.8.1

bit	Description
0	Function 1 (Reference mark 1)
1	Function 2 (Reference mark 2)
2	Function 3 (Reference mark 3)
3	Function 4 (Reference mark 4)
4	Value 1 (Reference mark 1)
5	Value 2 (Reference mark 2)
6	Value 3 (Reference mark 3)
7	Value 4 (Reference mark 4)
8	Reserved
9	Reserved
10	Reserved
11	Error acknowledgement in process
12	Reserved
13	Reserved
14	1: Parking sensor is activated
15	1: Sensor error

3.9 Position block selection (SATZANW)

Table 3.9.1

bit	Description	
0~5	Reserved	
6~14	Reserved	
15	1: Activate MDI submode	0: Deactivate MDI submode

3.10 Selected position block (AKTSATZ)

Table 3.10.1

bit	Description	
0~14	Reserved	
15	1: Activate MDI submode	0: Deactivate MDI submode

3.11 Position MDI mode (MDI_MODE)

Table 3.11.1

bit	Description	
0	1: Absolute positioning	0: Relative positioning
1	Reserved	
2	Reserved	
3~15	Reserved	

3.12 Positioning control word 1 (POS_STW1)

Table 3.12.1

bit	Description	
0~7	Reserved	
8	1: Absolute positioning selected	0: Relative positioning selected
9	0: MDI setting-up mode standstill 1: Positive direction for MDI setting-up mode 2: Negative direction for MDI setting-up mode 3: MDI setting-up mode standstill	
10		
11~13	Reserved	
14	1: Select MDI setting-up mode*1	0: Select MDI positioning mode*2
15	1: Activate MDI submode	0: Deactivate MDI submode

Note:

*1 On MDI setting-up mode, the functions of bit 9 and bit 10 are active. The constant velocity motion will be executed based on MDI velocity (MDI_VELOCITY).

*2 On MDI positioning mode, the functions of bit 8 are active. The positioning motion will be executed based on MDI position (MDI_TARPOS).

3.13 Positioning control word 2 (POS_STW2)

Table 3.13.1

bit	Description	
0	Reserved	
1	1: Set current position as reference point	
2~4	Reserved	
5	1: Select JOG incremental mode*1	0: Select JOG velocity mode*2
6~15	Reserved	

Note:

*1 On JOG incremental mode, relative moving motion will be executed based on program P2P velocity (Pt533 or Pt585) and program P2P relative travel distance (Pt539).

*2 On JOG velocity mode, constant velocity motion will be executed based on program P2P velocity (Pt533 or Pt585).

3.14 Positioning status word 1 (POS_ZSW1)

Table 3.14.1

bit	Description
0~7	Reserved
8	1: Negative overtravel is active
9	1: Positive overtravel is active
10	1: Jog is active
11	1: Homing procedure is active
12~13	Reserved
14	1: MDI setting-up mode is selected
15	1: MDI is active

3.15 Positioning status word 2 (POS_ZSW2)

Table 3.15.1

bit	Description
0~1	Reserved
2	1: Setpoint is available
3	Reserved
4	1: Axis moves forwards
5	1: Axis moves backwards
6~11	Reserved
12	1: Fixed stop is reached
13	1: Fixed stop clamping torque is reached
14	1: Traverse to fixed stop is activated
15	1: Traversing command is active

3.16 Message word (MELDW)

Table 3.16.1

bit	Description
0	Reserved
1	1: Torque limit is not active
2~5	Reserved
6	1: No motor overload warning
7	1: No I^2T warning
8	1: Speed deviation is in tolerance (not functional, always is 1)
9~10	Reserved
11	1: Controller is enabled
12	1: Drive ready
13	1: Pulses are enabled
14	Reserved
15	Reserved

(This page is intentionally left blank.)

4. Parameters

4.1 PROFIdrive parameters

Table 4.1.1 describes the supported PROFIdrive parameters.

Table 4.1.1

<table>
<tr><th>PNU (dec)</th><th>Read / Write</th><th>Data Type</th><th>Significance</th><th>Default</th></tr>
<tr><td rowspan="2">922</td><td>Read</td><td>Uint16</td><td>Telegram selection</td><td>0</td></tr>
<tr><td colspan="4">This parameter displays the current activated telegram number.</td></tr>
<tr><td rowspan="2">925</td><td>Read / Write</td><td>Uint16</td><td>Tolerance of the number of Controller Sign-of-Life failures</td><td>5</td></tr>
<tr><td colspan="4">The setting tolerance of Controller Sign-of-Life failures.</td></tr>
<tr><td rowspan="2">930</td><td>Read</td><td>Uint16</td><td>Operation mode</td><td>-</td></tr>
<tr><td colspan="4"><table><tr><th>Value</th><th>Description</th></tr><tr><td>1</td><td>Speed mode with RFG functionality</td></tr><tr><td>2</td><td>Position mode</td></tr><tr><td>3</td><td>Speed mode without RGF functionality</td></tr></table></td></tr>
<tr><td rowspan="2">944</td><td>Read</td><td>Uint16</td><td>Fault message counter</td><td>-</td></tr>
<tr><td colspan="4">The fault message counter is incremented each time the fault buffer changes.</td></tr>
<tr><td rowspan="2">945</td><td>Read</td><td>Uint16 Array[64]</td><td>Fault code</td><td>-</td></tr>
<tr><td colspan="4">The fault codes of alarms.
Refer to table 5.1.1 for the fault codes of drive alarms.</td></tr>
<tr><td rowspan="2">947</td><td>Read</td><td>Uint16 Array[64]</td><td>Fault number</td><td>-</td></tr>
<tr><td colspan="4">The internal fault numbers of alarms.
Refer to table 5.1.1 for the fault numbers of drive alarms.</td></tr>
<tr><td rowspan="2">950</td><td>Read</td><td>Uint16 Array[2]</td><td>Scaling of the fault buffer</td><td>-</td></tr>
<tr><td colspan="4">This parameter defines the number of fault situations (Subindex 0) and the number of fault messages in a fault situation (Subindex 1) of the fault buffer.</td></tr>
<tr><td rowspan="2">952</td><td>Read / Write</td><td>Uint16</td><td>Fault situation counter</td><td>-</td></tr>
<tr><td colspan="4">This parameter specifies the number of fault situations.
If this parameter is set to 0, the complete fault buffer is deleted.</td></tr>
</table>

<table>
<tr><th>PNU (dec)</th><th>Read / Write</th><th>Data Type</th><th>Significance</th><th>Default</th></tr>
<tr><td rowspan="2">964</td><td>Read</td><td>Uint16 Array[7]</td><td>Drive unit identification</td><td>-</td></tr>
<tr><td colspan="4">Data for the drive identification.
<table>
<tr><th>Subindex</th><th>Significance</th><th>Description</th></tr>
<tr><td>0</td><td>Manufacturer</td><td>Fix 0xAAA</td></tr>
<tr><td>1</td><td>Drive unit type</td><td>Fix 0x05</td></tr>
<tr><td>2</td><td>Firmware version</td><td>xxyy (decimal)
Example: Version 2.1 results in 0201.</td></tr>
<tr><td>3</td><td>Firmware data (year)</td><td>yyyy (decimal)</td></tr>
<tr><td>4</td><td>Firmware data (day/month)</td><td>ddmm (decimal)</td></tr>
<tr><td>5</td><td>Number of DO</td><td>Fix 1</td></tr>
<tr><td>6</td><td>Minor Firmware Version</td><td>-</td></tr>
</table></td></tr>
<tr><td rowspan="2">979</td><td>Read</td><td>Uint32 Array[31]</td><td>Encoder format</td><td>-</td></tr>
<tr><td colspan="4">
<table>
<tr><th>Subindex</th><th>Significance</th><th>Description</th></tr>
<tr><td>0</td><td>Header</td><td>-</td></tr>
<tr><td>1</td><td>Encoder type</td><td>-</td></tr>
<tr><td>2</td><td>Encoder resolution</td><td>-</td></tr>
<tr><td>3</td><td>Shift factor for G1_XIST1</td><td>-</td></tr>
<tr><td>4</td><td>Shift factor for G1_XIST2</td><td>-</td></tr>
<tr><td>5</td><td>Determinable revolutions</td><td>-</td></tr>
<tr><td>6~30</td><td>Reserved</td><td>-</td></tr>
</table></td></tr>
</table>

4.2 PNU dictionary table

Table 4.2.1

<table>
<tr><th>PNU (hex)</th><th>Sub-Index</th><th>Name</th><th>Data Type</th><th>Read / Write Attribute</th><th>Applicable Telegram</th><th>Valid Value</th><th>Unit</th></tr>
<tr><td>2XXXh</td><td>00h</td><td colspan="6">The 2000h series objects are from servo Pt parameters. Please refer to the chapter “List of parameters” in each servo drive user manual. The mapping relationship between servo Pt parameter numbers and object indexes is as follows:
Object index = 2000h + servo Pt parameter number
Example: Servo drive’s parameter Pt100 is “Velocity loop gain”, and its corresponding object is 2100h.</td></tr>
<tr><td rowspan="2">3000h</td><td rowspan="2">00h</td><td>Motor type</td><td>U16</td><td>ro</td><td>All</td><td>0 ~ 2</td><td>-</td></tr>
<tr><td colspan="6">Motor type used with the drive
0: Linear motor (LM)
1: Direct drive motor / Torque motor (DM / TM)
2: AC servo motor (AC)</td></tr>
<tr><td rowspan="2">3001h</td><td rowspan="2">00h</td><td>Inner encoder resolution</td><td>I32</td><td>ro</td><td>All</td><td>-2147483648 ~ 2147483647</td><td>-</td></tr>
<tr><td colspan="6">Encoder resolution for internal loop</td></tr>
<tr><td rowspan="2">3044h</td><td rowspan="2">00h</td><td>Digital outputs (one-way)*1</td><td>U16</td><td>rw</td><td>All</td><td>0 ~ 0x1F</td><td>-</td></tr>
<tr><td colspan="6">Digital outputs controlling</td></tr>
<tr><td rowspan="2">3056h</td><td rowspan="2">00h</td><td>Software state[12]</td><td>U16</td><td>ro</td><td>All</td><td>0 ~ 0xFFFF</td><td>-</td></tr>
<tr><td colspan="6">Software state table. The state corresponding to each bit is described as follows.
<table>
<tr><th>Bit</th><th>State Name</th><th>State Definition</th></tr>
<tr><td>0</td><td>Reserved</td><td>N/A</td></tr>
<tr><td>1</td><td>Reserved</td><td>N/A</td></tr>
<tr><td>2</td><td>Reserved</td><td>N/A</td></tr>
<tr><td>3</td><td>Homing state</td><td>0: Homing is not executed
1: Homing is in process</td></tr>
<tr><td>4</td><td>Position trigger function state</td><td>0: Position trigger function is not enabled
1: Position trigger function is enabled</td></tr>
<tr><td>5</td><td>Communication state of gantry control system</td><td>0: Communication for gantry control system is not executed
1: Communication for gantry control system is normal</td></tr>
<tr><td>6</td><td>Motor power state of gantry yaw axis</td><td>0: Motor for gantry yaw axis is unpowered
1: Motor for gantry yaw axis is powered</td></tr>
<tr><td>7</td><td>Alarm state of gantry yaw axis</td><td>0: No alarm is in gantry yaw axis
1: An alarm occurs in gantry yaw axis</td></tr>
<tr><td>8</td><td>Activated state of gantry control system</td><td>0: Gantry control system is not activated
1: Gantry control system is activated</td></tr>
<tr><td>9</td><td>Homing state of gantry yaw axis</td><td>0: Homing for gantry yaw axis is not completed
1: Homing for gantry yaw axis is completed</td></tr>
<tr><td>10</td><td>Near home sensor state of gantry yaw axis</td><td>0: Gantry yaw axis is not in the range of near home sensor
1: Gantry yaw axis is in the range of near home sensor</td></tr>
<tr><td>11</td><td>Regulating state of gantry yaw axis</td><td>0: Gantry yaw axis regulating is incompleted
1: Gantry yaw axis regulating is completed</td></tr>
<tr><td>12</td><td>In-position state of gantry yaw axis</td><td>0: Gantry yaw axis is not in-position
1: Gantry yaw axis is in-position</td></tr>
<tr><td>13</td><td>Ready state of gantry yaw axis</td><td>0: Drive for gantry yaw axis is not ready
1: Drive for gantry yaw axis is ready without triggering STO</td></tr>
<tr><td>14</td><td>Reserved</td><td>N/A</td></tr>
<tr><td>15</td><td>Reserved</td><td>N/A</td></tr>
</table></td></tr>
<tr><td rowspan="2">3057h</td><td rowspan="2">00h</td><td>Application mode of gantry system</td><td>U16</td><td>rw</td><td>All</td><td>1, 2, 11</td><td>-</td></tr>
<tr><td colspan="6">Application mode setting of gantry control system. The applicable modes are as follows.
Please refer to “E Series Servo Drive Gantry Control System User Manual” for detailed settings.</td></tr>
</table>

<table>
<tr><th>PNU (hex)</th><th>Sub-Index</th><th>Name</th><th>Data Type</th><th>Read / Write Attribute</th><th>Applicable Telegram</th><th>Valid Value</th><th>Unit</th></tr>
<tr><td></td><td></td><td colspan="6">1: Activate gantry control system
2: Deactivate gantry control system
11: Execute yaw axis regulating</td></tr>
<tr><td rowspan="2">3058h</td><td rowspan="2">00h</td><td>Yaw target position</td><td>I32</td><td>rw</td><td>All</td><td>-2147483648 ~ 2147483647</td><td>control unit</td></tr>
<tr><td colspan="6">Target position for gantry yaw axis</td></tr>
<tr><td rowspan="2">3059h</td><td rowspan="2">00h</td><td>Yaw feedback position</td><td>I32</td><td>ro</td><td>All</td><td>-2147483648 ~ 2147483647</td><td>control unit</td></tr>
<tr><td colspan="6">Feedback position for gantry yaw axis</td></tr>
<tr><td rowspan="2">3060h</td><td rowspan="2">00h</td><td>Use reference mark enable specific function</td><td>U16</td><td>rw</td><td>All</td><td>0 ~ 3</td><td>-</td></tr>
<tr><td colspan="6">Enable specific function with Reference mark.
<table>
<tr><th>Bit</th><th>Function</th><th>Definition</th></tr>
<tr><td>0</td><td>Error map</td><td>0: Do not use Reference mark to enable error map.
1: Use Reference mark to enable error map.</td></tr>
<tr><td>1</td><td>Position trigger function</td><td>(Before using this function, set Pt00E = t.1□□□.)
0: Do not use Reference mark to enable position trigger function.
1: Use Reference mark to enable position trigger function.</td></tr>
<tr><td>2~15</td><td>Reserved</td><td>N/A</td></tr>
</table>
For the details of error map and position trigger function, please refer to each servo drive user manual.
Reference mark corresponds to descriptions regarding touch probe function.</td></tr>
<tr><td rowspan="2">3061h</td><td rowspan="2">00h</td><td>Enable position trigger function</td><td>U16</td><td>rw</td><td>All</td><td>0 ~ 1</td><td>-</td></tr>
<tr><td colspan="6">Enable position trigger function.
For the details of position trigger function, please refer to each servo drive user manual.
0: Disable position trigger function
1: Enable position trigger function</td></tr>
<tr><td rowspan="2">3062h</td><td rowspan="2">00h</td><td>Overtravel stop mode selection</td><td>U16</td><td>rw</td><td>All</td><td>0 ~ 1</td><td>-</td></tr>
<tr><td colspan="6">Reserved.</td></tr>
<tr><td rowspan="2">3063h</td><td rowspan="2">00h</td><td>Velocity analog input voltage*2</td><td>I16</td><td>ro</td><td>All</td><td>-10000 ~ 10000</td><td>mV</td></tr>
<tr><td colspan="6">Control signal's velocity analog input (V_REF) (applicable to E2 series servo drive)
Formula: Object 3063h = Actual voltage - Object 3064h</td></tr>
<tr><td rowspan="2">3064h</td><td rowspan="2">00h</td><td>Velocity analog input voltage offset*2</td><td>I16</td><td>rw</td><td>All</td><td>-10000 ~ 10000</td><td>mV</td></tr>
<tr><td colspan="6">Velocity analog input's offset (applicable to E2 series servo drive)</td></tr>
<tr><td rowspan="2">3065h</td><td rowspan="2">00h</td><td>Torque analog input voltage*2</td><td>I16</td><td>ro</td><td>All</td><td>-10000 ~ 10000</td><td>mV</td></tr>
<tr><td colspan="6">Control signal's torque analog input (T_REF) (applicable to E2 series servo drive)
Formula: Object 3065h = Actual voltage - Object 3066h</td></tr>
<tr><td rowspan="2">3066h</td><td rowspan="2">00h</td><td>Torque analog input voltage offset*2</td><td>I16</td><td>rw</td><td>All</td><td>-10000 ~ 10000</td><td>mV</td></tr>
<tr><td colspan="6">Torque analog input's offset (applicable to E2 series servo drive)</td></tr>
<tr><td rowspan="2">3067h</td><td rowspan="2">00h</td><td>Analog output 1 voltage</td><td>I16</td><td>rw</td><td>All</td><td>-10000 ~ 10000</td><td>mV</td></tr>
<tr><td colspan="6">Control signal's analog output 1 (AO1)
When Pt006 = t.□□17 is set, users can control analog output 1 with this object.</td></tr>
<tr><td rowspan="2">3068h</td><td rowspan="2">00h</td><td>Analog output 2 voltage</td><td>I16</td><td>rw</td><td>All</td><td>-10000 ~ 10000</td><td>mV</td></tr>
<tr><td colspan="6">Control signal's analog output 2 (AO2)
When Pt007 = t.□□17 is set, users can control analog output 2 with this object.</td></tr>
<tr><td rowspan="2">3069h</td><td rowspan="2">00h</td><td>Position trigger array value</td><td>I32</td><td>rw</td><td>All</td><td>-2147483648 ~ 2147483647</td><td>inc</td></tr>
<tr><td colspan="6">Position trigger array's value</td></tr>
<tr><td rowspan="2">306Ah</td><td rowspan="2">00h</td><td>Position trigger array index</td><td>U16</td><td>rw</td><td>All</td><td>0 ~ 255</td><td>-</td></tr>
<tr><td colspan="6">Position trigger array's index value</td></tr>
<tr><td rowspan="2">306Bh</td><td rowspan="2">00h</td><td>Position trigger array control object</td><td>U16</td><td>rw</td><td>All</td><td>0 ~ 65535</td><td>-</td></tr>
<tr><td colspan="6">Writing procedure of operating position trigger array.</td></tr>
</table>

<table>
<tr><th>PNU (hex)</th><th>Sub-Index</th><th>Name</th><th>Data Type</th><th>Read / Write Attribute</th><th>Applicable Telegram</th><th>Valid Value</th><th>Unit</th></tr>
<tr><td></td><td></td><td colspan="6">Set 0x0001~0x0080 to select the writing procedure. The writing result will be displayed by 0x1000~0x2000.
<table>
<tr><th>Value</th><th>Definition</th><th>Category</th></tr>
<tr><td>0x0001</td><td>Write the value of object 3069h to the “position array” corresponding to object 306Ah.
(At this time, object 306Ah cannot exceed 255.)</td><td rowspan="4">Command</td></tr>
<tr><td>0x0008</td><td>Set all the values in the “position array” to 0.</td></tr>
<tr><td>0x0010</td><td>Write the value of object 3069h to the “status array” corresponding to object 306Ah.
(At this time, object 306Ah cannot exceed 7.)</td></tr>
<tr><td>0x0080</td><td>Set all the values in the “status array” to 0.</td></tr>
<tr><td>0x1000</td><td>The writing succeeds.</td><td rowspan="2">Result</td></tr>
<tr><td>0x2000</td><td>The writing fails. Refer to object 306Ch for the causes.</td></tr>
</table></td></tr>
<tr><td rowspan="2">306Ch</td><td rowspan="2">00h</td><td>Position trigger function error code</td><td>U16</td><td>ro</td><td>All</td><td>0 ~ 65535</td><td>-</td></tr>
<tr><td colspan="6">The reasons that the writing of position trigger array or the enabling of position trigger function fails.
<table>
<tr><th>Bit</th><th>Definition</th></tr>
<tr><td colspan="2">※ The reasons that the writing of position trigger array fails</td></tr>
<tr><td>0</td><td>Fixed interval PT mode does not support the writing of position trigger array.</td></tr>
<tr><td>1</td><td>Wrong index value of array (object 306Ah)</td></tr>
<tr><td>2</td><td>Undefined command (object 306Bh)</td></tr>
<tr><td>3~7</td><td>Reserved</td></tr>
<tr><td colspan="2">※ The reasons that the enabling of position trigger function fails</td></tr>
<tr><td>8</td><td>The encoder does not support position trigger function.</td></tr>
<tr><td>9</td><td>Homing is not executed.</td></tr>
<tr><td>10</td><td>The parameter setting of Pt00E or Pt230~Pt232 is wrong.</td></tr>
<tr><td>11</td><td>The current motor position exceeds the end position set by Pt232 (fixed interval PT mode Pt00E = t.□□1□).</td></tr>
<tr><td>12~15</td><td>Reserved</td></tr>
</table></td></tr>
<tr><td rowspan="2">306Dh</td><td rowspan="2">00h</td><td>Position trigger function status</td><td>I16</td><td>ro</td><td>All</td><td>0 ~ 32767</td><td>-</td></tr>
<tr><td colspan="6">Status of position trigger function.
<table>
<tr><th>Value</th><th>Definition</th></tr>
<tr><td>0</td><td>Position trigger function is not enabled.</td></tr>
<tr><td>3</td><td>Fixed interval position trigger function is executing (trigger direction: position decreasing).</td></tr>
<tr><td>4</td><td>Fixed interval position trigger function is executing (trigger direction: position increasing).</td></tr>
<tr><td>13</td><td>Random interval position trigger function is executing (trigger direction: index value decreasing).</td></tr>
<tr><td>14</td><td>Random interval position trigger function is executing (trigger direction: index value increasing).</td></tr>
<tr><td>20</td><td>Wait until it goes back to the first set trigger position (if repeat mode is enabled Pt012 = t.□□□1).</td></tr>
<tr><td>99</td><td>Position trigger function is invalid (Pt00E = t.□□□0).</td></tr>
</table></td></tr>
<tr><td rowspan="2">306Eh</td><td rowspan="2">00h</td><td>Expected total number of position trigger</td><td>U16</td><td>ro</td><td>All</td><td>0 ~ 65535</td><td>-</td></tr>
<tr><td colspan="6">Expected total number of position trigger</td></tr>
<tr><td rowspan="2">306Fh</td><td rowspan="2">00h</td><td>Triggered number of position trigger</td><td>U16</td><td>ro</td><td>All</td><td>0 ~ 65535</td><td>-</td></tr>
<tr><td colspan="6">Triggered number of position trigger</td></tr>
</table>

<table>
<tr><th>PNU (hex)</th><th>Sub-Index</th><th>Name</th><th>Data Type</th><th>Read / Write Attribute</th><th>Applicable Telegram</th><th>Valid Value</th><th>Unit</th></tr>
<tr><td>3070h</td><td>00h</td><td>Remaining number of position trigger</td><td>U16</td><td>ro</td><td>All</td><td>0 ~ 65535</td><td>-</td></tr>
<tr><td colspan="8">Remaining number of position trigger</td></tr>
<tr><td rowspan="2">3080h</td><td rowspan="2">00h</td><td>Gantry control: index</td><td>U16</td><td>rw</td><td>All</td><td>0x2000 ~ 0x4FFF</td><td>-</td></tr>
<tr><td colspan="6">The index value of the operation object for gantry slave axis parameter.
Example: If this object is set to 0x2100, it indicates that index 2100h of gantry slave axis parameter is designated.</td></tr>
<tr><td rowspan="2">3081h</td><td rowspan="2">00h</td><td>Gantry control: subindex</td><td>U16</td><td>rw</td><td>All</td><td>0</td><td>-</td></tr>
<tr><td colspan="6">The subindex value of the operation object for gantry slave axis parameter.
The current version only supports the object with subindex value being 0.</td></tr>
<tr><td rowspan="2">3082h</td><td rowspan="2">00h</td><td>Gantry control: data type of selected object</td><td>I16</td><td>ro</td><td>All</td><td>-3 ~ 8</td><td>-</td></tr>
<tr><td colspan="6">The data type of the gantry slave axis parameter designated by object 3080h.
Different data type has different input / output register, the corresponding register is described as follows:
<table>
<tr><th>Value</th><th>Definition</th><th>Corresponding Input / Output Register</th></tr>
<tr><td>1</td><td>The data type of the designated object is BOOL.</td><td rowspan="7">3085h / 3086h (DINT)</td></tr>
<tr><td>2</td><td>The data type of the designated object is I8.</td></tr>
<tr><td>3</td><td>The data type of the designated object is I16.</td></tr>
<tr><td>4</td><td>The data type of the designated object is I32.</td></tr>
<tr><td>5</td><td>The data type of the designated object is U8.</td></tr>
<tr><td>6</td><td>The data type of the designated object is U16.</td></tr>
<tr><td>7</td><td>The data type of the designated object is U32.</td></tr>
<tr><td>8</td><td>The data type of the designated object is F32.</td><td>3087h / 3088h (REAL)</td></tr>
<tr><td>-1</td><td>The index value cannot be operated.</td><td rowspan="3">N/A</td></tr>
<tr><td>-2</td><td>The designated index object does not exist.</td></tr>
<tr><td>-3</td><td>The designated subindex object does not exist.</td></tr>
</table>
Note: When object 3084h = -1, this object is not applicable.</td></tr>
<tr><td rowspan="2">3083h</td><td rowspan="2">00h</td><td>Gantry control: command</td><td>U16</td><td>rw</td><td>All</td><td>0 ~ 3</td><td>-</td></tr>
<tr><td colspan="6">The operation command of gantry slave axis parameter. The function of each command is described as follows:
<table>
<tr><th>Value</th><th>Definition</th><th>Description</th></tr>
<tr><td>0</td><td>Idle / Reset state</td><td>Idle / Reset state.</td></tr>
<tr><td>1</td><td>Writing command</td><td>The command will be triggered (positive edge) when this object is switched from 0 to 1. When the command is triggered, the value of the input register will be written to the designated object (3080h).
Note: If the command is given during data processing (object 3084h is 1), it will be invalid.</td></tr>
<tr><td>2</td><td>Single reading command</td><td>The command will be triggered (positive edge) when this object is switched from 0 to 2. When the command is triggered, the value of the designated object (3080h) will be put into the corresponding output register.
Note: If the command is given during data processing (object 3084h is 1), it will be invalid.</td></tr>
<tr><td>3</td><td>Continuous reading command</td><td>The values of the designated object (3080h) will be continuously put into the corresponding output register.
Note: Continuous reading command is not periodically updated.</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>PNU (hex)</th><th>Sub-Index</th><th>Name</th><th>Data Type</th><th>Read / Write Attribute</th><th>Applicable Telegram</th><th>Valid Value</th><th>Unit</th></tr>
<tr><td rowspan="2">3084h</td><td rowspan="2">00h</td><td>Gantry control: status</td><td>I16</td><td>ro</td><td>All</td><td>-6 ~ 2</td><td>-</td></tr>
<tr><td colspan="6">The operation status of gantry slave axis parameter. The definition is as follows:
<table>
<tr><th>Value</th><th>Definition</th></tr>
<tr><td>0</td><td>Not in operation.</td></tr>
<tr><td>1</td><td>Data is being processed.</td></tr>
<tr><td>2</td><td>Data processing succeeds.</td></tr>
<tr><td>-1</td><td>The operation function of gantry slave axis parameter cannot be operated. Check if the firmware versions of master axis and slave axis are the same and the gantry control system is activated.</td></tr>
<tr><td>-2</td><td>The designated object (3080h) cannot be operated.</td></tr>
<tr><td>-3</td><td>The value of the input register exceeds the upper limit of the designated object (3080h)’s data type.</td></tr>
<tr><td>-4</td><td>The writing command is executed to read-only object.</td></tr>
<tr><td>-5</td><td>The unsupported operation command is used in object 3083h.</td></tr>
<tr><td>-6</td><td>Data processing timeout.</td></tr>
</table></td></tr>
<tr><td rowspan="2">3085h</td><td rowspan="2">00h</td><td>Gantry control: input register of DINT</td><td>I32</td><td>rw</td><td>All</td><td>-2147483648 ~ 2147483647</td><td>-</td></tr>
<tr><td colspan="6">Input register for data type being BOOL, I8, I16, I32, U8, U16 or U32</td></tr>
<tr><td rowspan="2">3086h</td><td rowspan="2">00h</td><td>Gantry control: output register of DINT</td><td>I32</td><td>ro</td><td>All</td><td>-2147483648 ~ 2147483647</td><td>-</td></tr>
<tr><td colspan="6">Output register for data type being BOOL, I8, I16, I32, U8, U16 or U32</td></tr>
<tr><td rowspan="2">3087h</td><td rowspan="2">00h</td><td>Gantry control: input register of REAL</td><td>F32</td><td>rw</td><td>All</td><td>-3.40282e+38 ~ 3.40282e+38</td><td>-</td></tr>
<tr><td colspan="6">Input register for data type being F32</td></tr>
<tr><td rowspan="2">3088h</td><td rowspan="2">00h</td><td>Gantry control: output register of REAL</td><td>F32</td><td>ro</td><td>All</td><td>-3.40282e+38 ~ 3.40282e+38</td><td>-</td></tr>
<tr><td colspan="6">Output register for data type being F32</td></tr>
<tr><td>3100h
|
3104h</td><td>N/A</td><td colspan="6">This section is about alarm state table, and it is not supported yet.
Use object 4095h (error code) to check the contents.</td></tr>
<tr><td rowspan="2">3110h</td><td rowspan="2">00h</td><td>Drive warning events 1</td><td>U16</td><td>ro</td><td>All</td><td>0 ~ 0xFFFF</td><td>-</td></tr>
<tr><td colspan="6">Warning state table 1. The warning corresponding to each bit is described as follows.
It is recommended to replace this object with the object 4096h (warning code).
<table>
<tr><th>Bit</th><th>Waning No.</th><th>Warning Name</th></tr>
<tr><td>0</td><td>AL.900</td><td>Position deviation overflow</td></tr>
<tr><td>1</td><td>AL.901</td><td><Not supported></td></tr>
<tr><td>2</td><td>AL.910</td><td>Overload</td></tr>
<tr><td>3</td><td>AL.911</td><td><Not supported></td></tr>
<tr><td>4</td><td>AL.912</td><td><Not supported></td></tr>
<tr><td>5</td><td>AL.920</td><td><Not supported></td></tr>
<tr><td>6</td><td>AL.921</td><td><Not supported></td></tr>
<tr><td>7</td><td>AL.923</td><td>Internal fan stop</td></tr>
<tr><td>8</td><td>AL.930</td><td>Encoder battery malfunction</td></tr>
<tr><td>9</td><td>AL.941</td><td>Parameter or function that goes into effect after saving or power off has been modified</td></tr>
<tr><td>10</td><td>AL.971</td><td>Undervoltage</td></tr>
<tr><td>11</td><td>AL.9A0</td><td>Overtravel detected when servo ON (P-OT or N-OT signal is received)</td></tr>
<tr><td>12</td><td>AL.9A1</td><td>P-OT signal is received</td></tr>
<tr><td>13</td><td>AL.9A2</td><td>N-OT signal is received</td></tr>
<tr><td>14</td><td>AL.9AA</td><td><Not supported></td></tr>
<tr><td>15</td><td>AL.9Ab</td><td><Not supported></td></tr>
</table>
When the value of the bit is 1, the warning occurs.</td></tr>
<tr><td>3111h</td><td>00h</td><td>Drive warning events 2</td><td>U16</td><td>ro</td><td>All</td><td>0 ~ 0xFFFF</td><td>-</td></tr>
</table>

<table>
<tr><th>PNU (hex)</th><th>Sub-Index</th><th>Name</th><th>Data Type</th><th>Read / Write Attribute</th><th>Applicable Telegram</th><th>Valid Value</th><th>Unit</th></tr>
<tr><td></td><td></td><td colspan="6">Warning state table 2. The warning corresponding to each bit is described as follows.
It is recommended to replace this object with the object 4096h (warning code).

<table>
<tr><th>Bit</th><th>Waning No.</th><th>Warning Name</th></tr>
<tr><td>0</td><td>AL.9F0</td><td>Servo voltage too big</td></tr>
<tr><td>1</td><td>AL.943</td><td>Fieldbus synchronous cycle time warning</td></tr>
<tr><td>2</td><td>AL.944</td><td>System warning</td></tr>
<tr><td>3</td><td>AL.945</td><td>Torque limit warning</td></tr>
<tr><td>4</td><td>AL.946</td><td>Encoder communication warning</td></tr>
<tr><td>5</td><td>AL.947</td><td>Multi-motion malfunction warning</td></tr>
<tr><td>6</td><td>AL.924</td><td>I^2T</td></tr>
</table>
When the value of the bit is 1, the warning occurs.</td></tr>
<tr><td rowspan="2">3200h</td><td rowspan="2">00h</td><td>Absolute encoder initialization*2</td><td>I32</td><td>rw</td><td>All</td><td>0 ~ 1</td><td>-</td></tr>
<tr><td colspan="6">Initialize absolute encoder. When it is set to 1, the multi-turn data of motor will be cleared. Keep servo off during the execution. The object will set the value according to the execution state:

<table>
<tr><th>Value</th><th>Definition</th></tr>
<tr><td>0</td><td>Not in operation.</td></tr>
<tr><td>1</td><td>Send the command of clearing multi-turn data.</td></tr>
<tr><td>2</td><td>The command of clearing multi-turn data is being executed.</td></tr>
<tr><td>4</td><td>The command of clearing multi-turn data is successfully executed.</td></tr>
<tr><td>16</td><td>Do not clear multi-turn data when the motor is enabled.
Please disable the motor before issuing the command again.</td></tr>
<tr><td>32</td><td>Fail to execute the command of clearing multi-turn data.</td></tr>
</table></td></tr>
<tr><td rowspan="2">3201h</td><td rowspan="2">00h</td><td>General object i1</td><td>I32</td><td>rw</td><td>All</td><td>-2147483648 ~ 2147483647</td><td>-</td></tr>
<tr><td colspan="6">Self-defined object with data type of DINT (1)</td></tr>
<tr><td rowspan="2">3202h</td><td rowspan="2">00h</td><td>General object i2</td><td>I32</td><td>rw</td><td>All</td><td>-2147483648 ~ 2147483647</td><td>-</td></tr>
<tr><td colspan="6">Self-defined object with data type of DINT (2)</td></tr>
<tr><td rowspan="2">3203h</td><td rowspan="2">00h</td><td>General object i3</td><td>I32</td><td>rw</td><td>All</td><td>-2147483648 ~ 2147483647</td><td>-</td></tr>
<tr><td colspan="6">Self-defined object with data type of DINT (3)</td></tr>
<tr><td rowspan="2">3204h</td><td rowspan="2">00h</td><td>General object i4</td><td>I32</td><td>rw</td><td>All</td><td>-2147483648 ~ 2147483647</td><td>-</td></tr>
<tr><td colspan="6">Self-defined object with data type of DINT (4)</td></tr>
<tr><td rowspan="2">3205h</td><td rowspan="2">00h</td><td>General object i5</td><td>I32</td><td>rw</td><td>All</td><td>-2147483648 ~ 2147483647</td><td>-</td></tr>
<tr><td colspan="6">Self-defined object with data type of DINT (5)</td></tr>
<tr><td rowspan="2">3206h</td><td rowspan="2">00h</td><td>General object i6</td><td>I32</td><td>rw</td><td>All</td><td>-2147483648 ~ 2147483647</td><td>-</td></tr>
<tr><td colspan="6">Self-defined object with data type of DINT (6)</td></tr>
<tr><td rowspan="2">3207h</td><td rowspan="2">00h</td><td>General object i7</td><td>I32</td><td>rw</td><td>All</td><td>-2147483648 ~ 2147483647</td><td>-</td></tr>
<tr><td colspan="6">Self-defined object with data type of DINT (7)</td></tr>
<tr><td rowspan="2">3208h</td><td rowspan="2">00h</td><td>General object i8</td><td>I32</td><td>rw</td><td>All</td><td>-2147483648 ~ 2147483647</td><td>-</td></tr>
<tr><td colspan="6">Self-defined object with data type of DINT (8)</td></tr>
<tr><td rowspan="2">3209h</td><td rowspan="2">00h</td><td>General object i9</td><td>I32</td><td>rw</td><td>All</td><td>-2147483648 ~ 2147483647</td><td>-</td></tr>
<tr><td colspan="6">Self-defined object with data type of DINT (9)</td></tr>
<tr><td rowspan="2">3210h</td><td rowspan="2">00h</td><td>General object f0</td><td>F32</td><td>rw</td><td>All</td><td>-3.40282e+38 ~ 3.40282e+38</td><td>-</td></tr>
<tr><td colspan="6">Self-defined object with data type of REAL (0)</td></tr>
</table>

PNU (hex)	Sub-Index	Name	Data Type	Read / Write Attribute	Applicable Telegram	Valid Value	Unit
3211h	00h	General object f1	F32	rw	All	-3.40282e+38 ~ 3.40282e+38	-
		Self-defined object with data type of REAL (1)					
3212h	00h	General object f2	F32	rw	All	-3.40282e+38 ~ 3.40282e+38	-
		Self-defined object with data type of REAL (2)					
3213h	00h	General object f3	F32	rw	All	-3.40282e+38 ~ 3.40282e+38	-
		Self-defined object with data type of REAL (3)					
3214h	00h	General object f4	F32	rw	All	-3.40282e+38 ~ 3.40282e+38	-
		Self-defined object with data type of REAL (4)					
3215h	00h	Reset drive	I16	rw	All	0 ~ 1	-
		Reset the drive. When it is set to 1, the drive will be reset. After it is done, the object will be automatically set to 0.					
3216h	00h	Send parameter to flash	I16	rw	All	0 ~ 1	-
		Save parameters to drive. When it is set to 1, the current drive parameters will be saved. After it is done, the object will be automatically set to 0.					
4XXXh	00h	The 4000h series objects are from servo Ut parameters. Users can read more information of servo drive from this series of objects. Please refer to the chapter “List of panel monitoring parameters” in each servo drive user manual. The mapping relationship between servo Ut parameter numbers and object indexes is as follows: Object index = 4000h + servo Ut parameter number Example: Servo drive’s panel monitoring parameter Ut095 is “Alarm code”, and its corresponding object is 4095h.					

Note:

*1 For relative information, please refer to section 6.10 **Digital outputs controlling / monitoring**.

*2 The parameter is only applicable to E2 series servo drive.

*3 For relative information, please refer to section 6.9 **Absolute encoder initialization**.

4.3 Parameter access

E series PROFINET drives only support the single parameter request. Figure 4.3.1 describes the data flow of a parameter access.

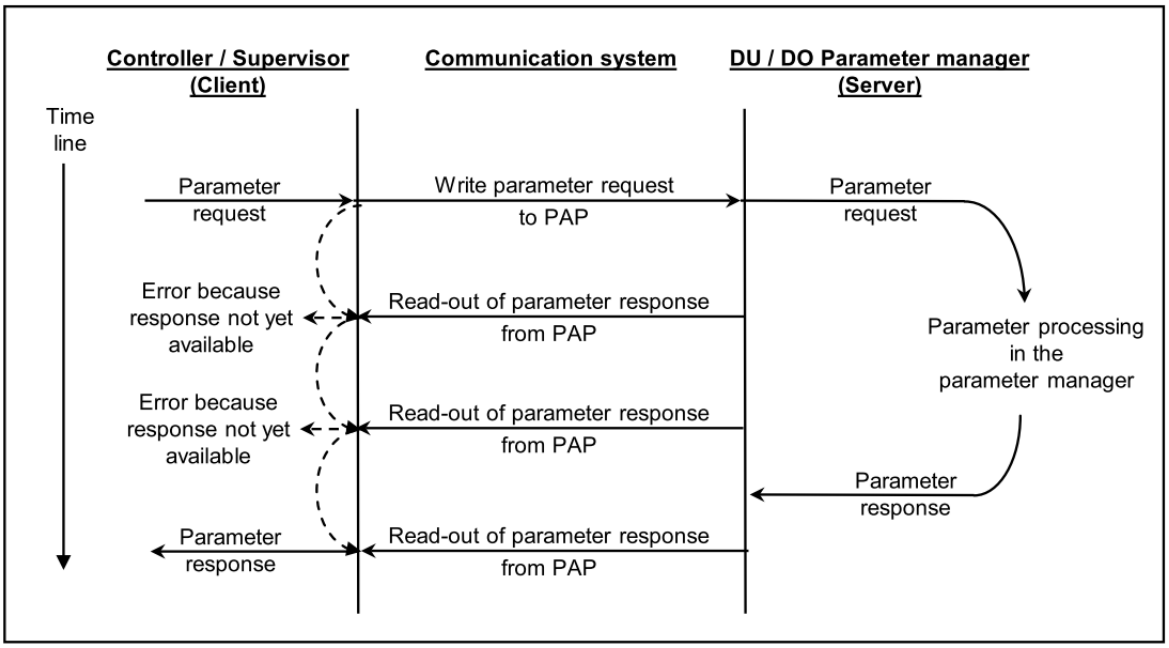


Figure 4.3.1

4.3.1 Struct of read a value

Table 4.3.1.1 Parameter request

Byte	Block Definition	Field	Value
0	Request header	Request reference	0x01~0xFF (by master)
1		Request ID	0x01
2		DO-ID	0
3		Number of parameters	1
4	Parameter address	Attribute	0x10 (Value)
5		Number of elements	0
6		Parameter number	PNU number
7			
8		Subindex (irrelevant)	0
9			

Table 4.3.1.2 Parameter response positive

Byte	Block Definition	Field	Value
0	Response header	Request reference mirrored	-
1		Response ID	0x01
2		DO-ID mirrored	-
3		Number of parameters	1
4	Parameter value	Format	Refer to table 4.3.5.1
5		Number of values	1
6		Value	data
7			
8			
9			

Table 4.3.1.3 Parameter response negative

Byte	Block Definition	Field	Value
0	Response header	Request reference mirrored	-
1		Response ID	0x81
2		DO-ID mirrored	-
3		Number of parameters	1
4	Parameter value	Format	0x44
5		Number of values	1
6		Error value	Refer to table 4.3.6.1
7			

4.3.2 Struct of write a value

Table 4.3.2.1 Parameter request

Byte	Block Definition	Field	Value
0	Request header	Request reference	0x01~0xFF (by master)
1		Request ID	0x02
2		DO-ID	0
3		Number of parameters	1
4	Parameter address	Attribute	0x10 (Value)
5		Number of elements	0
6		Parameter number	PNU number
7			
8		Subindex (irrelevant)	0
9			
10	Parameter value	Format	Refer to table 4.3.5.1
11		Number of values	1
12		Value	data
13			
14			
15			

Table 4.3.2.2 Parameter response positive

Byte	Block Definition	Field	Value
0	Response header	Request reference mirrored	-
1		Response ID	0x02
2		DO-ID mirrored	-
3		Number of parameters	1

Table 4.3.2.3 Parameter response negative

Byte	Block Definition	Field	Value
0	Response header	Request reference mirrored	-
1		Response ID	0x82
2		DO-ID mirrored	-
3		Number of parameters	1
4	Parameter value	Format	0x44
5		Number of values	1
6		Error value	Refer to table 4.3.6.1
7			

4.3.3 Struct of read array elements

Table 4.3.3.1 Parameter request

Byte	Block Definition	Field	Value
0	Request header	Request reference	0x01~0xFF (by master)
1		Request ID	0x01
2		DO-ID	0
3		Number of parameters	1
4	Parameter address	Attribute	0x10 (Value)
5		Number of elements	n
6		Parameter number	PNU number
7			
8		Subindex	0
9			

Table 4.3.3.2 Parameter response positive

Byte	Block Definition	Field	Value
0	Response header	Request reference mirrored	-
1		Response ID	0x01
2		DO-ID mirrored	-
3		Number of parameters	1
4	Parameter value	Format	Refer to table 4.3.5.1
5		Number of values	n
6		Value	data 1 to n
7			
8			
9			
…			

Table 4.3.3.3 Parameter response negative

Byte	Block Definition	Field	Value
0	Response header	Request reference mirrored	-
1		Response ID	0x81
2		DO-ID mirrored	-
3		Number of parameters	1
4	Parameter value	Format	0x44
5		Number of values	1
6		Error value	Refer to table 4.3.6.1
7			

4.3.4 Struct of write array elements

Table 4.3.4.1 Parameter request

Byte	Block Definition	Field	Value
0	Request header	Request reference	0x01~0xFF (by master)
1		Request ID	0x02
2		DO-ID	0
3		Number of parameters	1
4	Parameter address	Attribute	0x10 (Value)
5		Number of elements	n
6		Parameter number	PNU number
7			
8		Subindex	Subindex value
9			
10	Parameter value	Format	Refer to table 4.3.5.1
11		Number of values	n
12		Value	data 1 to n
13			
14			
15			
…			

Table 4.3.4.2 Parameter response positive

Byte	Block Definition	Field	Value
0	Response header	Request reference mirrored	-
1		Response ID	0x02
2		DO-ID mirrored	-
3		Number of parameters	1

Table 4.3.4.3 Parameter response negative

Byte	Block Definition	Field	Value
0	Response header	Request reference mirrored	-
1		Response ID	0x82
2		DO-ID mirrored	-
3		Number of parameters	1
4	Parameter value	Format	0x44
5		Number of values	1
6		Error value	Refer to table 4.3.6.1
7			

4.3.5 Struct information

Table 4.3.5.1

Field	Data Type	Value	Note
Format	Uint8	0x00: Reserved 0x01: Boolean 0x02: int8 0x03: int16 0x04: int32 0x05: Uint8 0x06: Uint16 0x07: Uint32 0x08: Float point 32 0x41: Byte 0x42: Word 0x43: Double word 0x44: Error Other: Reserved	-

4.3.6 Error number in parameter response

Table 4.3.6.1

Error No.	Error Name	Description
0x00	Impermissible parameter number	Access to an unavailable parameter.
0x01	Parameter value cannot be changed	Change access to a parameter value that cannot be changed.
0x02	Low or high limit exceeded	Change access with the value out of the value limits.
0x03	Faulty subindex	Access to an unavailable subindex of array or string parameter.
0x04	No array	Access with subindex to non-indexed parameter.
0x05	Incorrect data type	Change access with the value that does not match the data type of the parameter.
0x06	Setting not permitted (may only be reset)	Change access with the value unequal to 0. This is not permitted.
0x07	Description element cannot be changed	Change access to a description element that cannot be changed.
0x08	Reserved	-
0x09	Unavailable description data	Access to an unavailable description data.
0x0A	Reserved	-
0x0B	No operation priority	Change access without rights to change parameters.
0x0C	Reserved	-
0x0D		
0x0E		
0x0F	Unavailable text array	Access to an unavailable text array.
0x10	Reserved	-
0x11	Request cannot be executed because of operating state	Access is temporarily not possible for reasons that are not specified in detail.
0x12	Reserved	-
0x13		
0x14	Impermissible value	Change access with the value within the value limits, but not permissible for other long-term reasons (parameter with defined single values).
0x15	Response too long	The length of the current response exceeds the maximum transmittable length of the response transport block. In case of a multi parameter request, the response block was shortened by omitting the parameter requests.
0x16	Impermissible parameter address	Illegal value (reserved) or value which is not supported for

Error No.	Error Name	Description
		the attribute, illegal or not supported number of elements, illegal parameter number or illegal subindex or a combination.
0x17	Illegal format	Write request: Illegal format or format of the parameter data not supported.
0x18	Number of values are not consistent	Write request: Number of values of the parameter data does not match number of elements in parameter address.
0x19	Nonexistent Axis/DO	Access to an Axis/DO which does not exist.
0x1A~0x1F	Reserved	-
0x20	Parameter text element cannot be changed	Change access to a parameter text element that cannot be changed.
0x21	Service not supported	Illegal or unknown Request ID. (Response ID = 0x80)
0x22	Too much parameter requests	Multi parameter request: The response block does not contain all parameter responses since the maximum number of the supported parameter requests per multi parameter request exceeded.
0x23	Multi parameter access not supported	Device parameter manager does not support multi parameter requests. Request is discarded.
0x24~0xFF	Reserved	-

(This page is intentionally left blank.)

5. Diagnostics

5.1 Fault number / code of drive alarm

Table 5.1.1

Fault Number (dec)	Fault Code (hex)	Alarm Name	Alarm No.
4	024	System alarm 1	AL.024
5	025	System alarm 2	AL.025
6	030	Main circuit malfunction	AL.030
7	040	Parameter setting error	AL.040
11	050	Combination error	AL.050
12	070	Motor change detected	AL.070
14	0b0	Invalid servo on command	AL.0b0
15	100	Overcurrent detected	AL.100
16	320	Regenerative energy overflow	AL.320
17	400	Overvoltage	AL.400
18	410	Undervoltage	AL.410
19	510	Overspeed	AL.510
20	511	Encoder pulse output overspeed	AL.511
24	710	Overload (instantaneous maximum load)	AL.710
25	720	Overload (continuous maximum load)	AL.720
29	7A1	Drive overload	AL.7A1
30	7A2	Power board temperature error	AL.7A2
33	800	Encoder absolute position lost	AL.800
34	810	Encoder battery undervoltage	AL.810
35	820	Encoder communication error	AL.820
36	830	Encoder data error	AL.830
37	840	Encoder communication crc error	AL.840
38	850	Encoder counting error	AL.850
39	860	Encoder data writing error	AL.860
40	870	Encoder overheating	AL.870
41	880	Incremental encoder signal phase order error	AL.880
42	890	Excellent Smart Cube (ESC) - incremental encoder disconnection	AL.890
43	8A0	First set of encoder - Excellent Smart Cube (ESC) signal error	AL.8A0
44	8b0	First set of encoder - encoder signal error	AL.8b0
45	8C0	Second set of encoder - Excellent Smart Cube (ESC) signal	AL.8C0

Fault Number (dec)	Fault Code (hex)	Alarm Name	Alarm No.
		error	
46	8d0	Second set of encoder - encoder signal error	AL.8d0
47	8E0	Digital encoder disconnection	AL.8E0
48	8F0	Excellent Smart Cube (ESC) internal error	AL.8F0
49	861	Motor overheating	AL.861
50	b10	Velocity command A/D converter error	AL.b10
52	b20	Torque command A/D converter error	AL.b20
53	b33	Current detection malfunction	AL.b33
54	C10	Motor out of control	AL.C10
55	C20	Phase detection error	AL.C20
56	C21	Hall sensor error	AL.C21
58	C50	Electrical angle detection failure	AL.C50
59	C51	Overtravel detected during electrical angle detection	AL.C51
60	C52	Electrical angle detection incomplete	AL.C52
62	d00	Position deviation overflow	AL.d00
65	d10	Motor-load position deviation overflow	AL.d10
66	Eb0	Safety function alarm	AL.Eb0
67	Eb1	Safety function signal input timing error	AL.Eb1
68	Eb2	Safety function module error	AL.Eb2
69	F10	Power cable open phase	AL.F10
70	F50	Motor main circuit cable disconnection	AL.F50
71	FA0	Encoder power error	AL.FA0
72	FB0	Fieldbus communication hardware malfunction	AL.FB0
73	FB1	Fieldbus communication error	AL.FB1
74	FC0	Group control system communication error	AL.FC0
75	FC1	Slave axis error in group control system	AL.FC1
76	891	Incremental encoder signal error	AL.891
77	FB2	Fieldbus communication setup error	AL.FB2
78	EE0	Fixed stop application alarm AL.EE0	
79	Fd0	Electronic cam control system alarm	AL.Fd0
80	EF9	Multi-motion alarm	AL.EF9

(This page is intentionally left blank.)

6. Function descriptions

6.1 Velocity reference value setting

The velocity reference value is used to calculate normalized speed setpoints (NSOLL_A, NSOLL_B) and speed actual values (NIST_A, NIST_B). For NSOLL_A and NIST_A, 0x4000 is 100% of the velocity reference value; for NSOLL_B and NIST_B, 0x40000000 is 100% of the velocity reference value. The relationship among velocity reference value, normalized speed setpoints and speed actual values is shown as follows.

$$Actual\ Command\ speed = \frac{NSOLL\_A}{4000h} \times (velocity\ reference\ value)$$

$$Actual\ Command\ speed = \frac{NSOLL\_B}{40000000h} \times (velocity\ reference\ value)$$

$$NIST\_A = \frac{(actual\ speed)}{(velocity\ reference\ value)} \times 4000h$$

$$NIST\_B = \frac{(actual\ speed)}{(velocity\ reference\ value)} \times 40000000h$$

To change the velocity reference value, set Pt317 for rotary motors, and set Pt386 for linear motors.

Pt No.	Pt317	PNU Number	0x2317		
Data Type	Uint16	Setting Range	1~65535	Default	3000
Name	Motor reference velocity (rotary servo motor)[*1]	Unit	1 rpm	Applicable Motor	Rotary
Effective	Immediately	Attribute	Value	Applicable Telegram	3, 9, 102, 111

Pt No.	Pt386	PNU Number	0x2386		
Data Type	Uint16	Setting Range	1~100	Default	20
Name	Motor reference velocity (linear servo motor)[*1]	Unit	100 mm/s	Applicable Motor	Linear
Effective	Immediately	Attribute	Value	Applicable Telegram	3, 9, 102, 111

Note:

[*1] The default value of Pt317 is 3000 and Pt386 is 20. These are the velocity commands which 100% correspond to the controller commands.

6.2 Velocity limit setting

To change the velocity limit, set Pt316 for rotary motors, and set Pt385 for linear motors.

Pt No.	Pt316	PNU Number	0x2316		
Data Type	Uint16	Setting Range	0~65535	Default	10000
Name	Maximum motor velocity (rotary servo motor)	Unit	1 rpm	Applicable Motor	Rotary
Effective	After power on	Attribute	Value	Applicable Telegram	3, 9, 102, 111

Pt No.	Pt385	PNU Number	0x2385		
Data Type	Uint16	Setting Range	0~100	Default	50
Name	Maximum motor velocity (linear servo motor)	Unit	100 mm/s	Applicable Motor	Linear
Effective	After power on	Attribute	Value	Applicable Telegram	3, 9, 102, 111

6.3 Torque limit setting

To change the torque limit, set Pt402 and Pt403 for rotary motors, and set Pt483 and Pt484 for linear motors.

Pt No.	Pt402	PNU Number	0x2402		
Data Type	Uint16	Setting Range	0~800	Default	800
Name	Forward torque limit	Unit	1%	Applicable Motor	Rotary
Effective	Immediately	Attribute	Value	Applicable Telegram	3, 9, 102, 111

Pt No.	Pt403	PNU Number	0x2403		
Data Type	Uint16	Setting Range	0~800	Default	800
Name	Reverse torque limit	Unit	1%	Applicable Motor	Rotary
Effective	Immediately	Attribute	Value	Applicable Telegram	3, 9, 102, 111

Pt No.	Pt483	PNU Number	0x2483		
Data Type	Uint16	Setting Range	0~800	Default	30
Name	Forward force limit value for internal force limit (linear servo motor)	Unit	1%	Applicable Motor	Linear
Effective	Immediately	Attribute	Value	Applicable Telegram	3, 9, 102, 111

Pt No.	Pt484	PNU Number	0x2484		
Data Type	Uint16	Setting Range	0~800	Default	30
Name	Reverse force limit value for internal force limit (linear servo motor)	Unit	1%	Applicable Motor	Linear
Effective	Immediately	Attribute	Value	Applicable Telegram	3, 9, 102, 111

6.4 Quick stop

To change the quick stop deceleration time, set Pt30A for speed mode, and set Pt538 for position mode.

Pt No.	Pt30A	PNU Number	0x230A		
Data Type	Uint16	Setting Range	0~10000	Default	0
Name	Deceleration time for servo off and forced stop	Unit	1 ms	Applicable Motor	All
Effective	Immediately	Attribute	Value	Applicable Telegram	3, 102

Pt No.	Pt538	PNU Number	0x2538		
Data Type	Uint16	Setting Range	2~10000	Default	10
Name	Program P2P emergency deceleration time	Unit	1 ms	Applicable Motor	All
Effective	Immediately	Attribute	Value	Applicable Telegram	9, 111

6.5 Coast stop

The coast stop method could be set by Pt001 = t.□□□X.

Parameter		Servo Motor Stopping Method	Servo Motor State After Stop	Effective
Pt001	t.□□□0 (default)	Dynamic brake	Dynamic brake	After power on
	t.□□□1		Free run	
	t.□□□2	Free run	Free run	

6.6 MDI submode

6.6.1 MDI setpoints

When working with MDI submode, MDI setpoints should be set as follows. For the details of control unit, refer to section 6.11 **Electronic gear ratio** in “E1 Series Servo Drive User Manual” and “E2 Series Servo Drive User Manual.”

MDI setpoint	
MDI_TARPOS	1h = 1 control unit
MDI_VELOCITY	1h = 1000 control unit/min
MDI_ACC	4000h = 100%
MDI_DEC	4000h = 100%
OVERRIDE	4000h = 100%

The acceleration reference value could be set by Pt534.

Pt No.	Pt534	PNU Number	0x2534		
Data Type	Uint16	Setting Range	2~10000	Default	100
Name	Program P2P acceleration time	Unit	1 ms	Applicable Motor	All
Effective	Immediately	Attribute	Value	Applicable Telegram	9, 111

The deceleration reference value could be set by Pt537.

Pt No.	Pt537	PNU Number	0x2537		
Data Type	Uint16	Setting Range	2~10000	Default	100
Name	Program P2P deceleration time	Unit	1 ms	Applicable Motor	All
Effective	Immediately	Attribute	Value	Applicable Telegram	9, 111

6.7 Homing

Only Telegram 9 and Telegram 111 can apply internal homing of drive. For the details of homing methods and parameters setting, refer to section 8.11 **Internal homing** in “E1 Series Servo Drive User Manual” and “E2 Series Servo Drive User Manual.”

6.8 JOG

Only Telegram 9 and Telegram 111 can apply jog. Set STW1.8 = 1 to jog positive; set STW1.9 = 1 to jog negative. To change the jog velocity, set Pt533 for rotary motors, and set Pt585 for linear motors. The jog velocity is also affected by velocity override (PZD5 in Telegram 111).

Pt No.	Pt533	PNU Number	0x2533		
Data Type	Uint16	Setting Range	1~10000	Default	600[*1]
Name	Program P2P velocity	Unit	1 rpm	Applicable Motor	Rotary
Effective	Immediately	Attribute	Value	Applicable Telegram	9, 111

Note: [*1] If direct drive motor is used, the default value of Pt533 is 60 rpm.

Pt No.	Pt585	PNU Number	0x2585		
Data Type	Uint16	Setting Range	1~10000	Default	50
Name	Program P2P velocity (linear servo motor)	Unit	1 mm/s	Applicable Motor	Linear
Effective	Immediately	Attribute	Value	Applicable Telegram	9, 111

The acceleration reference value could be set by Pt534.

Pt No.	Pt534	PNU Number	0x2534		
Data Type	Uint16	Setting Range	2~10000	Default	100
Name	Program P2P acceleration time	Unit	1 ms	Applicable Motor	All
Effective	Immediately	Attribute	Value	Applicable Telegram	9, 111

The deceleration reference value could be set by Pt537.

Pt No.	Pt537	PNU Number	0x2537		
Data Type	Uint16	Setting Range	2~10000	Default	100
Name	Program P2P deceleration time	Unit	1 ms	Applicable Motor	All
Effective	Immediately	Attribute	Value	Applicable Telegram	9, 111

The moving distance of JOG incremental mode could be set by Pt539.

Pt No.	Pt539	PNU Number	0x2539		
Data Type	Int32	Setting Range	1~1073741824	Default	32768
Name	Program P2P relative travel distance	Unit	Control unit	Applicable Motor	All
Effective	Immediately	Attribute	Value	Applicable Telegram	9, 111

The jog direction of STW1.8 and STW1.9 can be reversed by setting Pt53A = 1 and thus to swap the direction definition of theirs.

Pt No.	Pt53A	PNU Number	0x253A		
Data Type	Uint16	Setting Range	0~1	Default	0
Name	PROFIdrive JOG mode moving direction inverse setting	Unit	-	Applicable Motor	All
Effective	Immediately	Attribute	Setup	Applicable Telegram	9, 111

6.9 Absolute encoder initialization

When using a rotary absolute encoder, it is necessary to clear multi-turn data at the first start up after installing the battery. There are two types of data in a rotary absolute encoder: single-turn data and multi-turn data. The single-turn data shows the position of the motor's rotation within a single turn. The multi-turn data counts the number of the turns, and the backup is stored by the battery.

■ Method 1

For relative information of function blocks usage, please refer to section 3.4 **Initialize absolute encoder** in "Function Blocks Application Manual E1 PROFINET Drive with Siemens TIA Portal."

■ Method 2

Adopt the following steps to clear multi-turn data:

Step 1. Disable the motor.
Step 2. Set parameter 3200h to 1.
Step 3. Wait until parameter 3200h changes to 4 (the command is successfully executed).
Step 4. Reset the drive (set parameter 3215h to 1).

Note:
For the parameter attribute of 3200h and 3215h, please refer to section 4.2 **PNU dictionary table**.

■ Definition of parameter 3200h

Table 6.9.1

Value	Definition
0	Not in operation.
1	Send the command of clearing multi-turn data.
2	The command of clearing multi-turn data is being executed.
4	The command of clearing multi-turn data is successfully executed.
16	Do not clear multi-turn data when the motor is enabled. Please disable the motor again before issuing the command again.
32	Fail to execute the command of clearing multi-turn data.

Note:
(1) This function is only supported in firmware version 2.8.16 or above.
(2) Users can directly download the attachment of "Function Blocks Application Manual E1 PROFINET Drive with Siemens TIA Portal" for usage.

6.10 Digital outputs controlling / monitoring

E series servo drives support 5 sets of pins for digital output signal. When the drive leaves the factory, each pin has the default digital output signal. Users can choose to directly use the factory default settings or allocate digital output signals and set pin polarity by themselves. Users can also perform signal status controlling / monitoring with PNU parameters.

- **Allocation of digital output signal**

  For relative information of digital output signal usage, please refer to section 8.1.2 **Digital output signal allocation** in “E1 Series Servo Drive User Manual” and “E2 Series Servo Drive User Manual.”

- **Signal status controlling**

  Based on the allocated digital output signal, PNU parameter 3044h allows users to control the digital output signal status in the logic of union. If the allocation of digital output signal is set to “Not configure”, PNU parameter 3044h will fully take control.

- **Signal status monitoring**

  With PNU parameter 4005h, users can monitor the digital output signal status.

- **Definition of parameter 3044h**

Table 6.10.1

bit	15 ... 5	4	3	2	1	0
Signal	Reserved	O5	O4	O3	O2	O1
Description	Reserved	0: When the output condition is satisfied, signal status is ON. 1: Whether the output condition is satisfied or not, signal status is ON.				

- **Definition of parameter 4005h**

Table 6.10.2

bit	15 ... 5	4	3	2	1	0
Signal	Reserved	O5	O4	O3	O2	O1
Pin	Reserved	CN6-19/20	CN6-17/18	CN6-15/16	CN6-13/14	CN6-11/12
Description	Reserved	0: Signal status is OFF. (The polarity will determine whether to output the signal.) 1: Signal status is ON. (The polarity will determine whether to output the signal.)				

Note:

(1) This function is only supported in firmware version 2.8.16 or above.

(2) For the parameter attribute of 3044h and 4005h, please refer to section 4.2 **PNU dictionary table**.

6.11 Reference torque setting

The reference torque acts on the normalized processing data exchanged between the drive and the controller, including torque reduction (MOMRED) and actual torque (M_ACT), or additional torque (M_ADD1) and positive and negative torque limits (M_LIMIT_POS, M_LIMIT_NEG). 0x4000 is 100% of the reference torque. The relationship is shown as follows.

- **Calculation of torque reduction (MOMRED) and output torque**

$$Reduction\ value = \left(\frac{MOMRED}{4000h}\right) \times reference\ torque$$

$$Output\ torque = reference\ torque - reduction\ value = \left(1 - \frac{MOMRED}{4000h}\right) \times reference\ torque$$

- **Calculation of actual torque (M_ACT)**

$$M\_ACT = \left(\frac{motor\ torque}{reference\ torque}\right) \times 4000h$$

- **Calculation of other torque values**

$$Torque\ value = \left(\frac{Recieved\_data}{4000h}\right) \times reference\ torque.$$

It is recommended to process Recieved_data as a signed 16-bit integer.
If Recieved_data is an unsigned integer, when the value is larger than 32767 (7FFFh), it is required to minus 65536 before calculation.

To change the reference torque / force, set Pt42A for rotary motors; set Pt43A for linear motors.

Pt No.	Pt42A	PNU Number	0x242A		
Data Type	Uint32	Setting Range	0~2147483647	Default	0[*1]
Name	Motor reference torque	Unit	0.001 Nm	Applicable Motor	Rotary
Effective	Immediately[*2]	Attribute	Value	Applicable Telegram	102, 750

Pt No.	Pt43A	PNU Number	0x243A		
Data Type	Uint32	Setting Range	0~2147483647	Default	0[*1]
Name	Motor reference force	Unit	0.001 N	Applicable Motor	Linear
Effective	Immediately[*2]	Attribute	Value	Applicable Telegram	102, 750

Note:

*1 If the drive is set to factory default and the setup of Configuration Wizard is completed, it will automatically be set to the rated torque of motor based on the motor type.

*2 If the value is modified during the enabling and operating process, it will not be effective immediately. It will be effective after the motor is disabled.

6.11.1 Additional torque (M_ADD1)

When the additional torque is applied to the vertical axis, it can improve the downward sliding of the vertical axis during enabling process due to the weight of the load. The additional torque must correspond to the static force loaded on the axis, so that the required time for the motor output to reach the target torque after the brake is released is minimized, thereby the load can be held instantly.

- **Method of evaluating the load's static force**
  After the motor is enabled and the position of the motor is maintained at a fixed point, the motor torque at this time can be regarded as the load's static force on the axis. Different mechanical structures and the influence of friction may make the motor torque change. The average value obtained from multiple measurements or the minimum value is a more appropriate torque value.

- **Way to use**
  The controller must send the additional torque before enabling the motor. After the motor is enabled, make the additional torque return to zero in stages, and then perform motion control. By doing so, it can prevent the applied torque value from causing poor responses to the servo loop or generating vibrations.

Note:

During the enabling and operating process, if the applied torque value increases or decreases significantly for a split second, unpredictable vibrations and errors may occur.

6.11.2 Positive torque limit (M_LIMIT_POS), Negative torque limit (M_LIMIT_NEG)

Positive torque limit acts on the torque limit value for forward direction, while negative torque limit acts on the torque limit value for reverse direction. Between this function and the function in section 6.3 **Torque limit setting**, the smaller value will be viewed as the torque limit value.

- **Torque limit function**

  For detailed description of torque limit function, please refer to section 8.10 **Torque limit function** in "E1 Series Servo Drive User Manual" and "E2 Series Servo Drive User Manual."

Note:

(1) If positive torque limit and negative torque limit are set to 0 at the same time, this function is OFF.

(2) If bit 8 of Control word 2 in section 3.4.2 is set to "Activate Traverse to fixed stop", this function is OFF. Instead, torque reduction (MOMRED) will act on the torque limit value.

6.12 Traverse to fixed stop setting

Telegram 111 supports servo Pt parameters Pt42B ~ Pt42F for Traverse to fixed stop setting. This function is suited for the application that requires a constant torque output when traversing towards a specific position to clamp/press a workpiece, as shown in figure 6.12.1.

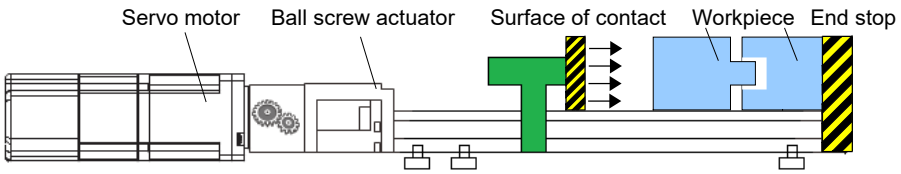


Figure 6.12.1

- **Signal chart for Traverse to fixed stop**

  To activate Traverse to fixed stop, set STW2.8 = 1 (“Activate Traverse to fixed stop”). The function starts operating after the status of POS_ZSW2.14 = 1 (“Traverse to fixed stopped activated”).

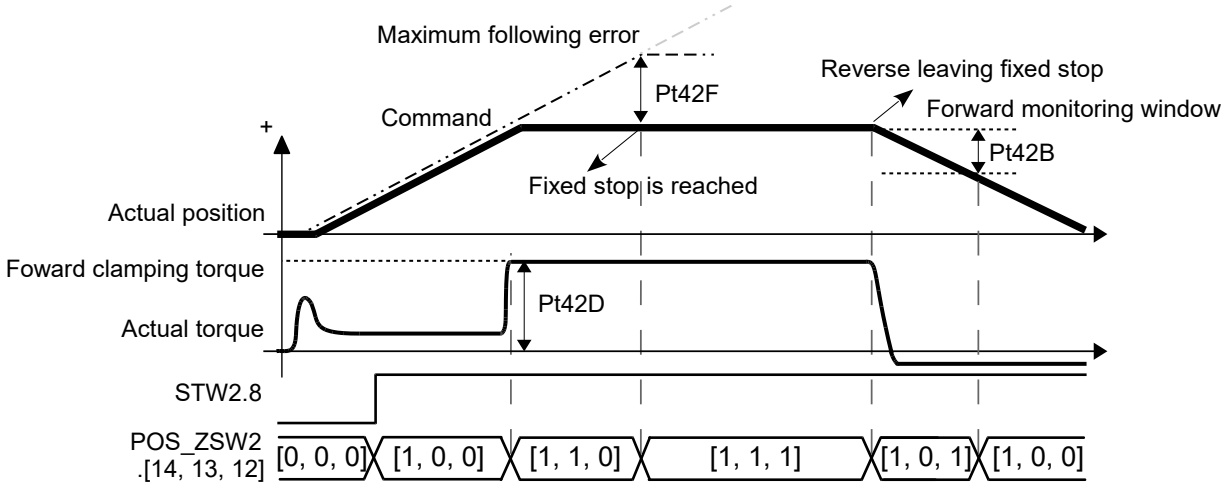


Figure 6.12.2

Note: If starting the homing procedure, the function will be temporarily deactivated until home position is set (ZSW1.11 = 1).

■ **Alarm detection: Fixed stop is not reached**

After the function starts operating, the alarm AL.EE0 will be output if the fixed stop is not reached after absolute positioning completed.

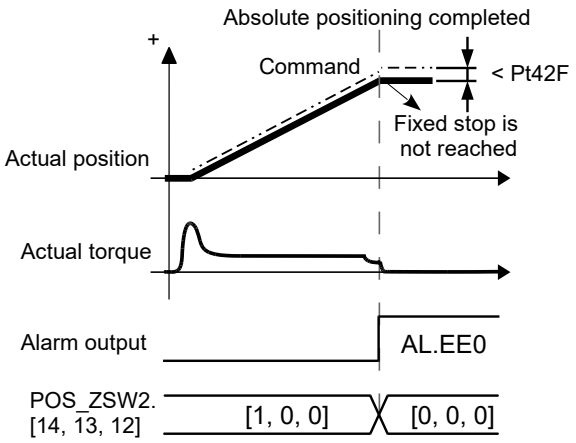


Figure 6.12.3

Alarm Number and Alarm Name	AL.EE0 Fixed stop application alarm
Cause	Fixed stop is not reached after absolute positioning completed.
Confirmation Method	Check if the workpiece is in place, and check if the given position command and the fixed stop maximum following error (Pt42F) fit with the application.
Corrective Action	Properly set the position command and the fixed stop maximum following error according to the specification of the workpiece.

■ **Alarm detection: The position goes outside the monitoring window after reaching the fixed stop**

After reaching the fixed stop, the alarm AL.EE0 will be output if the absolute positioning goes outside the monitoring window.

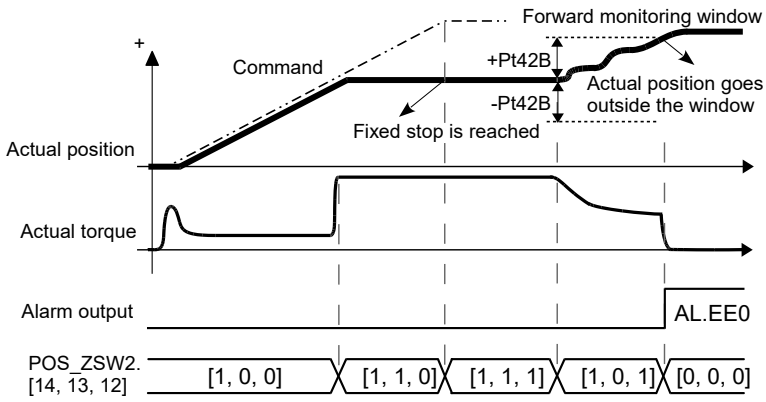


Figure 6.12.4

Alarm Number and Alarm Name	AL.EE0 Fixed stop application alarm
Cause	Actual position goes outside the fixed stop monitoring window after the fixed stop is reached.
Confirmation Method	Check if the mechanism and the workpiece are installed properly and stably.
Corrective Action	Adjust the mechanism. Widen the fixed stop monitoring window (Pt42B and Pt42C) or lower the fixed stop clamping torque (Pt42D and Pt42E) if necessary.

Pt No.	Pt42B	PNU Number	0x242B		
Data Type	Uint32	Setting Range	0~2147483647	Default	100
Name	Forward fixed stop monitoring window	Unit	Control unit	Applicable Motor	All
Effective	Immediately	Attribute	Value	Applicable Telegram	111

Pt No.	Pt42C*1	PNU Number	0x242C		
Data Type	Uint32	Setting Range	0~2147483647	Default	0
Name	Reverse fixed stop monitoring window	Unit	Control unit	Applicable Motor	All
Effective	Immediately	Attribute	Value	Applicable Telegram	111

Pt No.	Pt42D	PNU Number	0x242D		
Data Type	Uint32	Setting Range	0~2147483647	Default	1000000*2
Name	Forward fixed stop clamping torque (force)	Unit	0.001 Nm(N)	Applicable Motor	All
Effective	Immediately	Attribute	Value	Applicable Telegram	111

Pt No.	Pt42E*1	PNU Number	0x242E		
Data Type	Uint32	Setting Range	0~2147483647	Default	1000000
Name	Reverse fixed stop clamping torque (force)	Unit	0.001 Nm(N)	Applicable Motor	All
Effective	Immediately	Attribute	Value	Applicable Telegram	111

Pt No.	Pt42F*3	PNU Number	0x242F		
Data Type	Uint32	Setting Range	0~2147483647	Default	1000
Name	Fixed stop maximum following error	Unit	Control unit	Applicable Motor	All
Effective	Immediately	Attribute	Value	Applicable Telegram	111

Note:

*1 If setting to 0, the parameter setting corresponds to forward usage will take effect.

*2 If the drive is set as default and the setup of Configuration Wizard is completed, it will automatically be set to 10% of the motor rated torque.

*3 If Pt42F is smaller than Pt42B or Pt42C, fixed stop monitoring window will not work properly.

7. Appendix

7.1 Configure PROFINET communication by Thunder

Thunder offers users an interface to set the parameters related to PROFINET communication. Select **Tools** in the menu bar and click **PROFINET setup** to open “PROFINET setup” window.

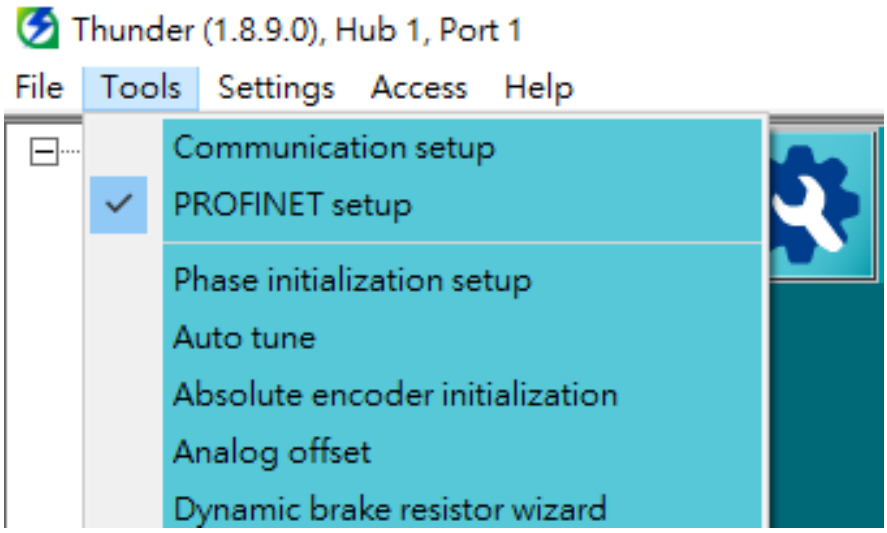


Figure 7.1.1

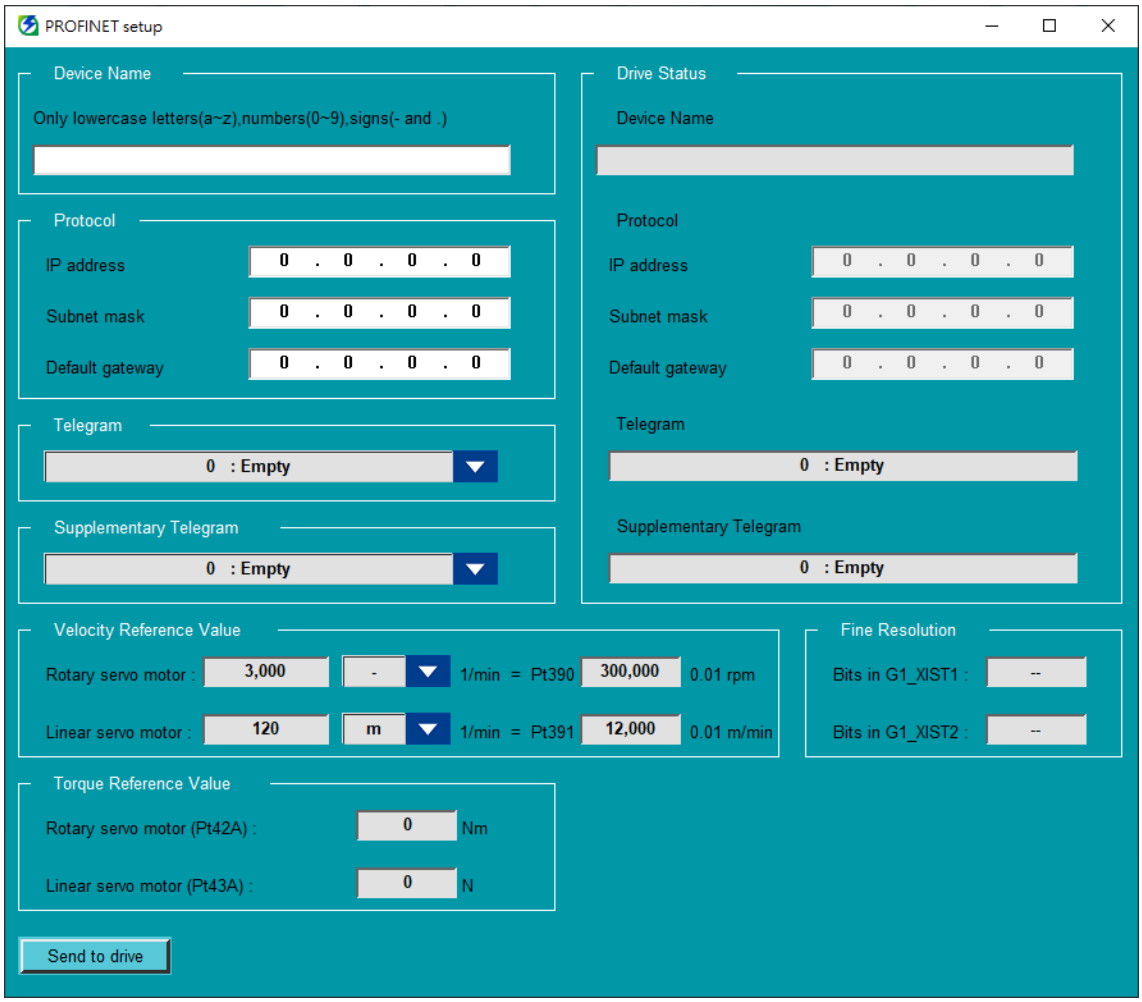


Figure 7.1.2

Drive Status displays the current activated setting of servo drive. To change the current setting, set **Device name**, **Protocol**, **Telegram**, **Supplementary Telegram**, **Velocity Reference Value**, **Fine Resolution** and **Torque Reference Value**. Then, click **Send to drive** to send the new setting to servo drive. The new setting will be activated after the process is completed.

Note:

(1) The device name must be unique within a PROFINET network.

(2) **Velocity Reference Value**, **Fine Resolution** and **Torque Reference Value** should always correspond with the reference speed, the encoder resolution and the reference torque configured in controller if application requires. Otherwise, these can be ignored. Based on the required configuration, inactive parameters do not need to be set, and the background color of these value columns is gray.

(3) For the setting of **Fine Resolution**, users can refer to “Application Note E1 PROFINET Drive Complete Setup with Siemens TIA Portal.”