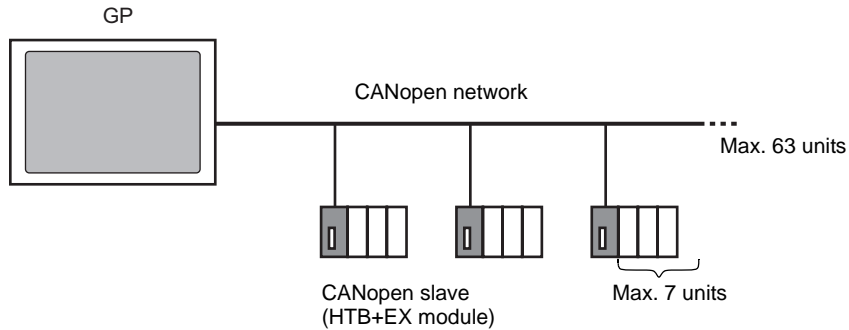


30.7 Controlling External I/O by Using HTB (CANopen Slave)

30.7.1 Summary

Model AGP-*****-CA1M contains the CANopen master driver. By connecting the CANopen slave unit "Hybrid Terminal Block" (model: HTB-1C0DM9LP; hereafter referred to as "HTB") to this GP, external I/O in a remote location can be controlled.

Moreover, by attaching an EX module to the HTB, I/O can be extended.



For the GP unit, an open network with maximum 63 units of connected HTB can be constructed. A maximum of 7 EX modules can be attached to each HTB unit.

■ CAN Standards

- CANopen specifications are defined by the CiA association and are available on the site www.can-cia.org.

NOTE

- To learn more about specifications and standard mechanisms CANopen, go to the homepage of the CiA (<http://www.can-cia.org>).

■ CANopen Master Driver

- Uses DS301V4.02, DSP302V3.2, and DS405V2.0 profiles. DS301 is a profile used for the application layer and for communication. DSP302 is a framework for the CANopen Manager and programmable CANopen devices. DS405V2.0 is a profile for IEC61131-3 programmable devices.
- Supports 11 bit COB-ID (CAN2.0A). Does not support 29 bit COB-ID (CAN2.0B).
- Communication is carried out with a PDO packet as the unit. For each HTB, PDO can support up to 16 units (Receive: number of RxPDO, 8 unit blocks [RPDO0~7], Send: number of TxPDF, 8 unit blocks [TPDO0~7]). The entire system can support up to 252 units of RxPDO and up to 252 units of TxPDO. When allocating to TPDO4~7 and RPDO4~7, with all the slave settings together, you can control 64 units.
- Does not support flying master^{*1}.
- Network configuration is saved in a concise DCF file^{*2}. This concise DCF file is transferred to the GP when transferring a project.

*1 Flying Master is a feature that allows the flying master to dynamically determine the master in a network when there are multiple devices that can be used as the CANopen master.

■ **HTB Features**

With Digital Electronics Corporation's CANopen slave unit, the following features can be used.

Feature	Points	Summary	Browse to
Standard I/O			
Standard Input	12	-	☞ "30.7.4 Standard I/O" (page 30-155)
Standard Output	8 pts	-	

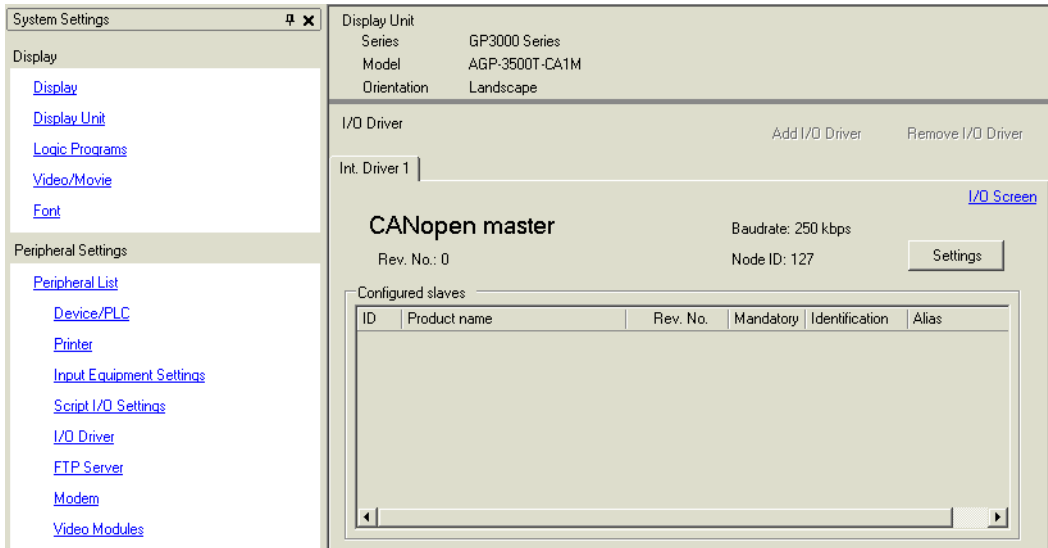
NOTE

- To extend I/O by using an EX module, refer to the following.
☞ "30.7.5 Expanding I/O Using EX Module" (page 30-161)

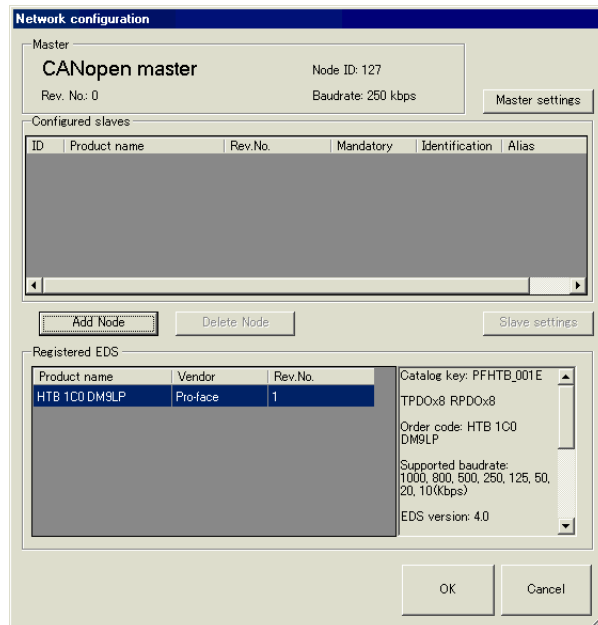
30.7.2 Setting Procedure

1 In Model Settings, select AGP-*****-CA1M.

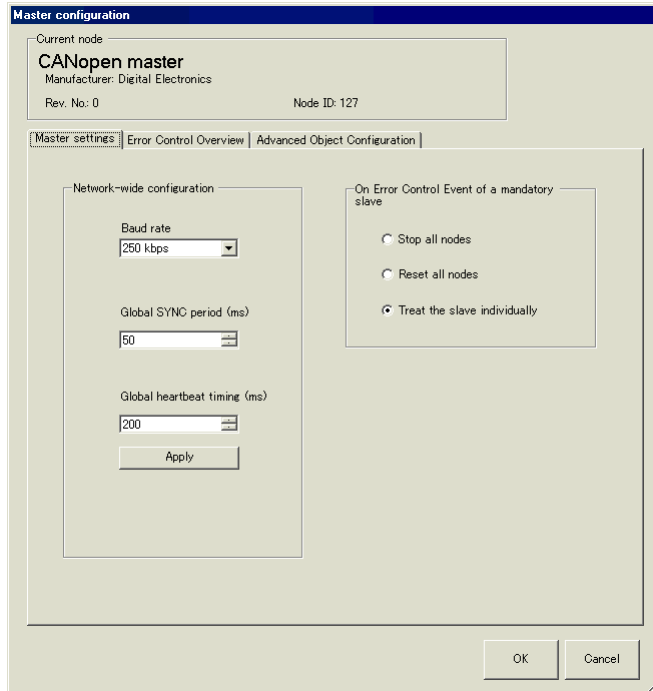
2 Open [I/O Driver] in System Settings.



3 Click [Settings] and the following dialog box appears.



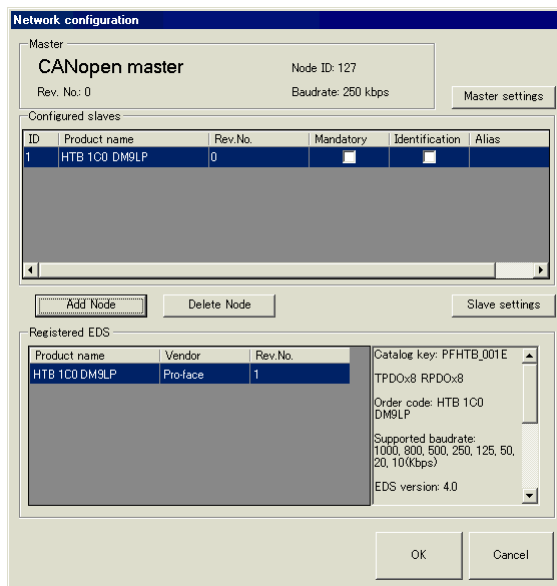
- Click [Master Settings] and the following dialog box appears. Configure all of the CANopen network settings, such as Baud rate, SYNC sending period, and master object settings. Click [OK] to enable the settings and the dialog box closes.



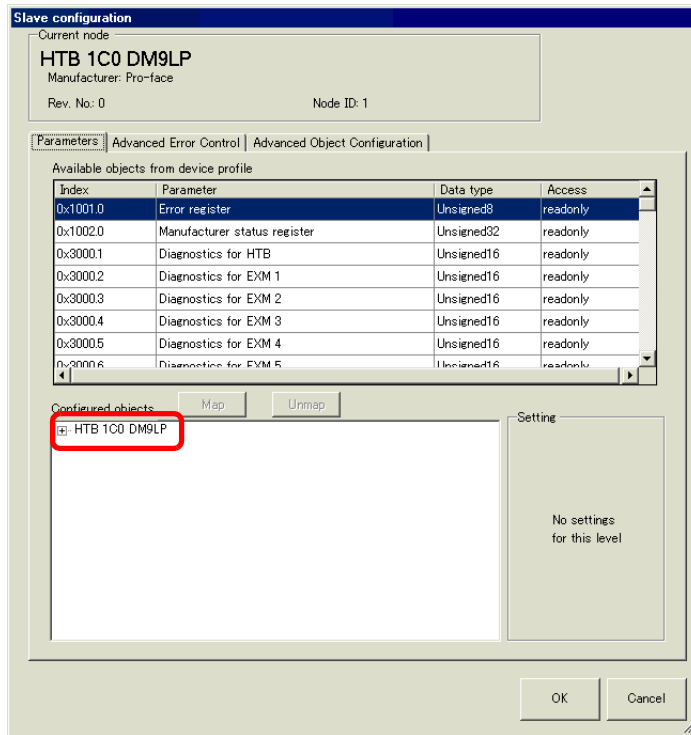
NOTE

- Set up the slave (HTB) baud rate on the HTB unit itself.

- Next, add a slave to the CANopen network. In [Registered EDS] list select the slave unit you want to add and click the [Add Node] button. The unit specified for [Configured Slave] is added.



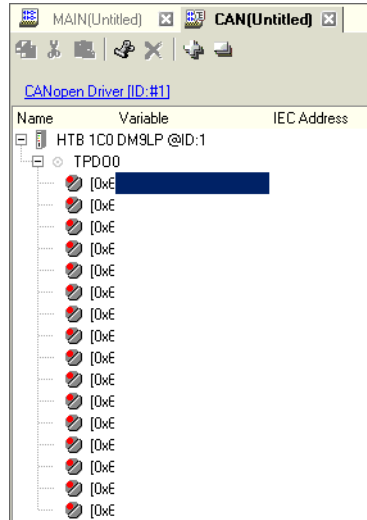
6 Click [Slave Settings] with the added slave unit selected and the following dialog box appears. According to the function you want to use, configure the communication parameter settings and set actions and values of objects to be used. Click [OK] to enable the settings and the dialog box closes.



NOTE

- When using an EX module, use a sub index object that corresponds to the attachment position of the EX module.
 ☞ "30.7.5 Expanding I/O Using EX Module" (page 30-161)

- 7 Click [I/O Screen] in the [I/O Driver Settings] screen or select [I/O Screen] in the [Screen List] window on Work Space to allocate a variable to each of the mapped objects. For instructions on how to allocate variables, refer to the following:
- ☞ "30.7.3 Allocating I/O (Common)" (page 30-152)



- 8 Create a Logic Screen and a Base Screen to access the allocated variables and transfer them to the GP.

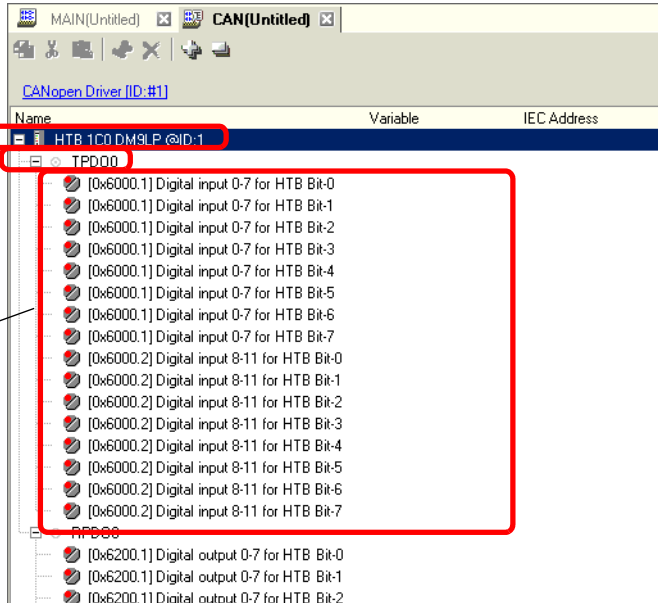
30.7.3 Allocating I/O (Common)

Objects set in [I/O Driver Settings] are reflected on the I/O Screen as terminals (I/O terminals). By allocating variables to terminals, I/O can be controlled.

Displays Slave Product Name, Node ID, or Comment.

PDO packet number.

Mapped objects display as terminals.



NOTE

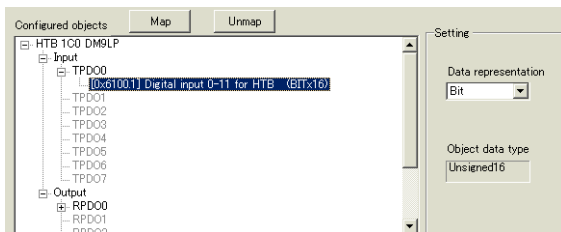
- When mapping TPDO4 to 7 and RPDO4 to 7, the total slave settings are limited to 64.

I/O Screen Display

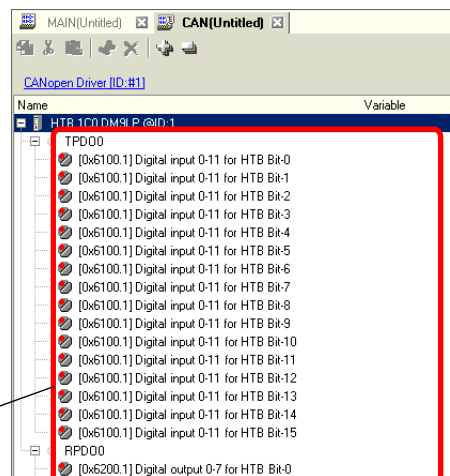
The terminal display on the I/O Screen differs depending on the type (such as, Bit, Byte, Word, Dword) of each object specified in the [Slave Settings] dialog box in [I/O Driver Settings].

Example 1: Set the Unsigned16 object "Digital input 0-11 for HTB" to "Bit" display

I/O Driver



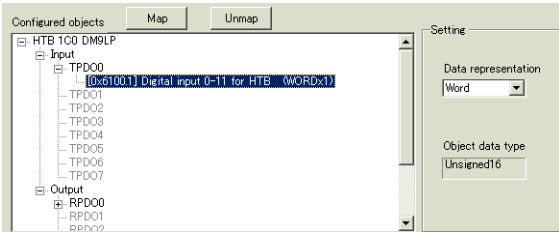
I/O Screen



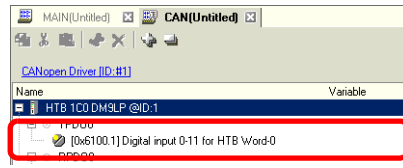
An Unsigned16 object displays as a 16 bit terminal.

Example 2: Set Unsigned16 object "Digital input 0-11 for HTB" as "Word" display settings.

I/O Driver



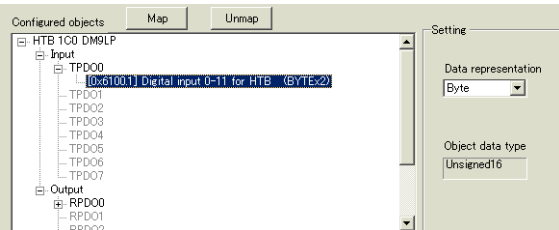
I/O Screen



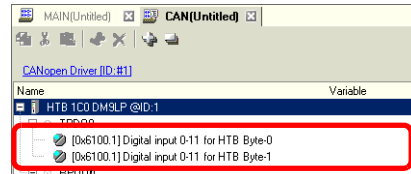
Unsigned16 displays as a 1 word terminal.

Example 3: Set the Unsigned16 object "Digital input 0-11 for HTB" to "Byte" display settings.

I/O Driver



I/O Screen



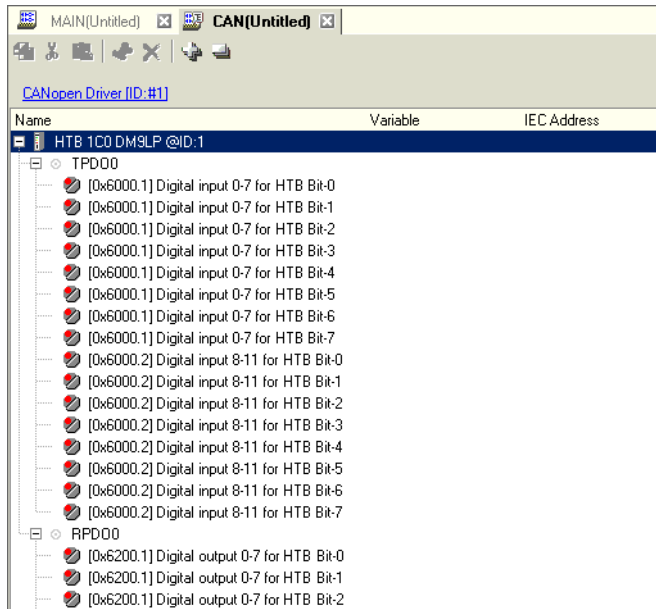
Unsigned16 displays as a 2 byte terminal.

NOTE

- In "Digital input 0-11 for HTB", Unsigned16 bit data lower bytes are allocated to the "Byte-0" terminal and higher bytes to the "Byte-1" terminal. The remaining bytes of the assigned variables cannot be used.

■ Mapping Variables

Mapping a variable per terminal of objects. To enter a variable, double-click the [Variable] column of the terminal you wish to allocate.



[IEC Address] appears automatically after entering a variable.

Data Type	Input	Output
Bit	IX	QX
Byte	IB	QB
Word	IW	QW
Dword	ID	QD

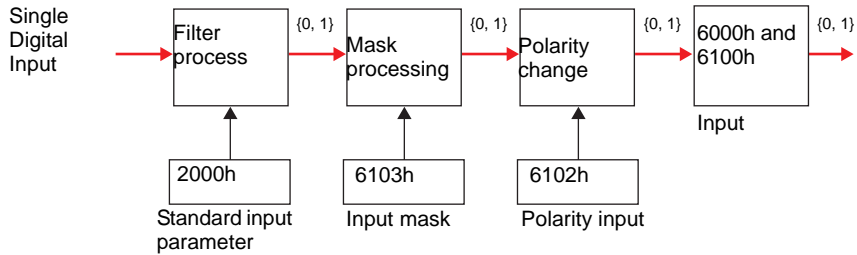
30.7.4 Standard I/O

■ Standard Input

Input bits are read into the sub indexes 1 and 2 of object 6000H (8 bit) and sub index 1 of object 6100h (16 bit) in 8 bit and 16 bit units. The following parameters can be changed:

- Standard input parameter (object 2000h)
- Polarity output (object 6102h)
- Input mask (object 6103h)

Object 2000h is used to set input I0 to I7 as digital input (or diagnostic input).

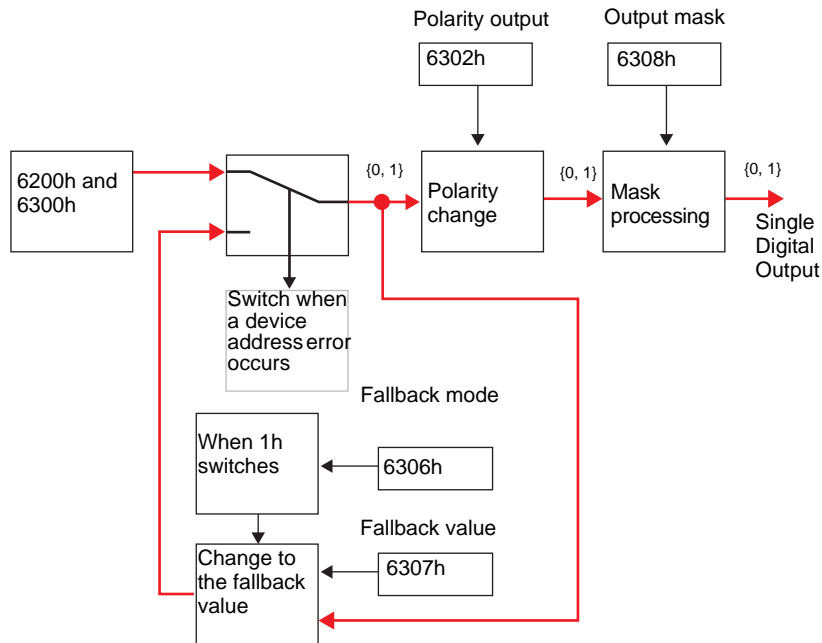


■ Standard Output

Output bits are written into sub index 1 of object 6200H (8 bit) and sub index 1 of object 6300h (16 bit) in 8 bit and 16 bit units. The following parameters can be changed.

- Polarity output (object 6302h)
- Output mask (object 6308h)

When an error such as failed communication with the GP occurs, it switches to the fallback mode.




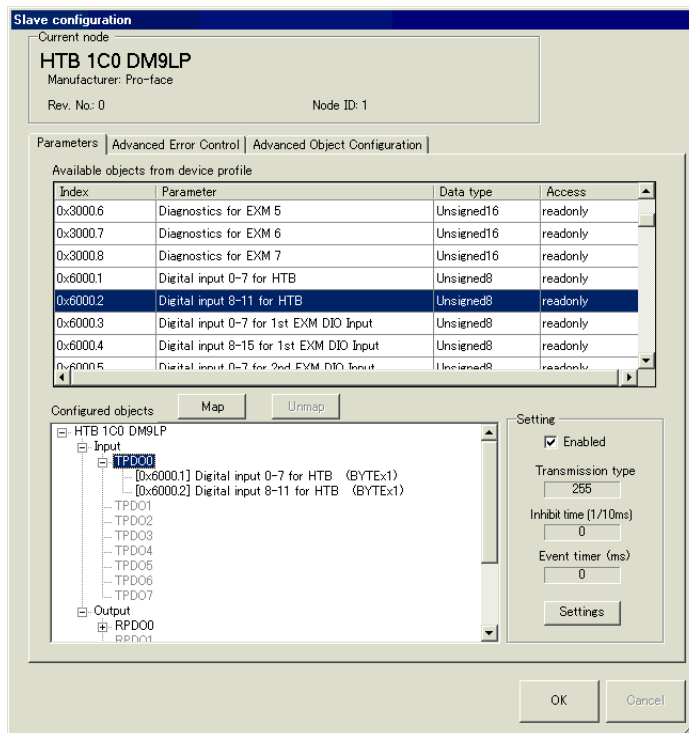
■ Setup Procedure

◆ Standard I/O

PDO mapping in slaves and how to configure objects are explained for cases in which 12-point input and 8-point output are used.

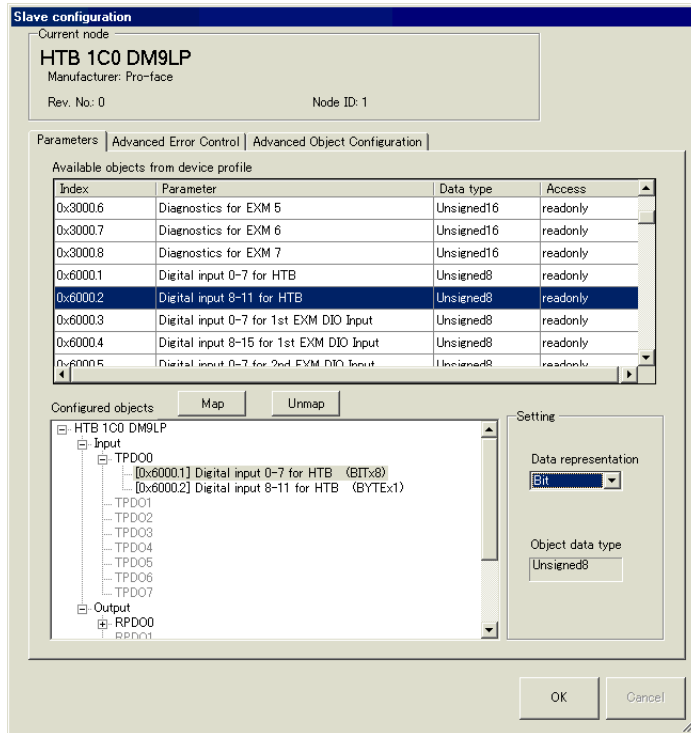
Settings in GP-Pro EX

- 1 Click [Settings] from System window [I/O Driver Settings] and add HTB to the network.
 "30.7.2 Setting Procedure" (page 30-148)
- 2 Click [Slave Settings] with the selected HTB and the following dialog box appears. In the [Parameters] tab, expand the [Configured objects] tree to open [TPDO0].

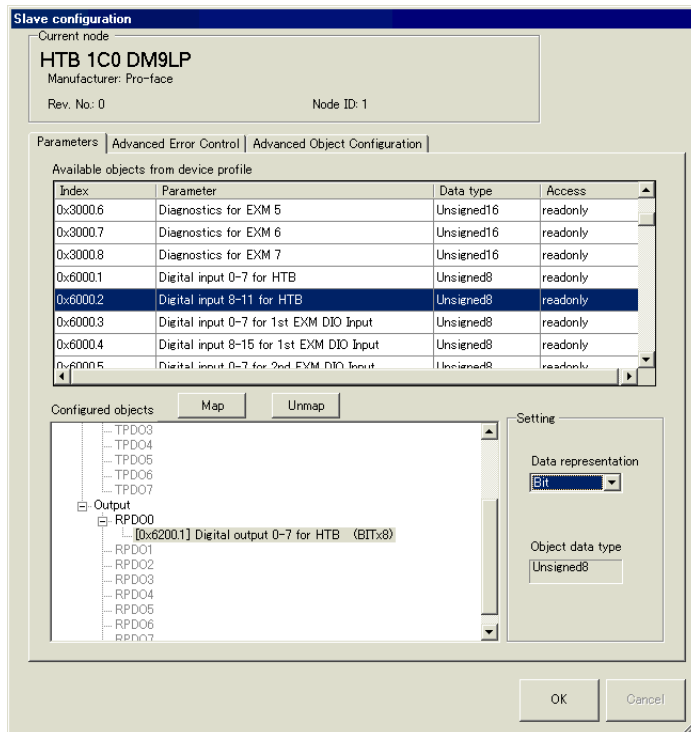


- 3 Check whether [0x6000.1] and [0x6000.2] have been allocated.

- 4 To change the data type of the object in TPDO, select the object and then select the [Data representation] in [Settings] (example, Bit).



- 5 Similarly, open RPDO0 and check whether [0x6200.1] and [0x6200.2] have been allocated. To change the data type of the object in RPDO, select the object and then and select [Data representation] in [Settings] (example, Bit).




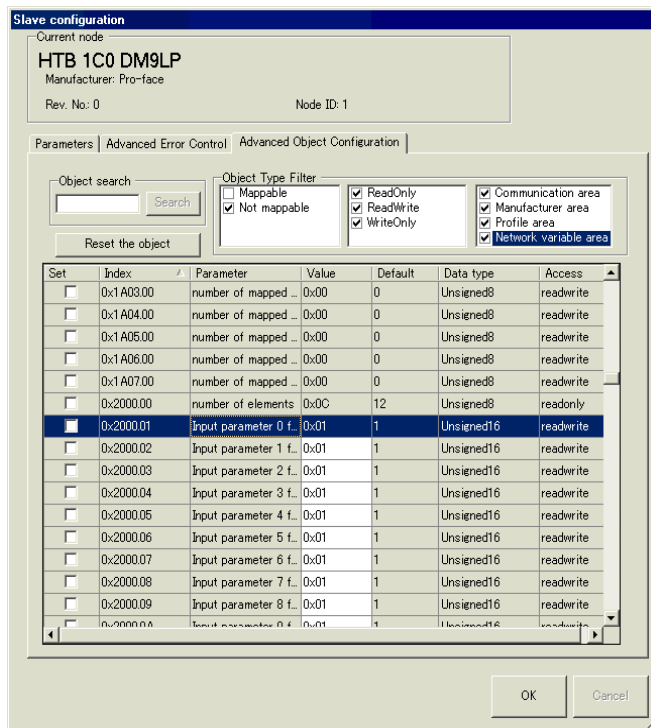
NOTE

- You can use "0x6100.1" instead of input value objects: "0x6000.1" and "0x6000.2"; and the object "0x6300.1" instead of output value objects: "0x6200.1" and "0x6200.2", respectively. Then select the mapped object and click the [Unmap] button to release cancel the assignment. After this, from [Available objects from device profile], select the object you want to map and then click the [Map] button to assign it to PDO.
- Map the input data from HTB (CANopen slave) to the GP (CANopen master) for TPDO, and map the object relating to the output data from the GP to HTB for RPDO.
- Map the error register "0x1001.1", the status register "0x1002.1" and the module diagnosis "0x3000.1" to TPDO as required.


6 Click the [Advanced Object Configuration] tab and configure the detailed settings of the object. Set the input filter (0x2000.1 to 0x2000.C), the fallback mode (0x6306.1) and the fallback value (0x6307.1) as required and click [OK] to close the dialog.

NOTE

- For details of objects, refer to the following.
 "30.7.10 List of HTB Objects" (page 30-211)



7 Click [I/O Screen] in [I/O Driver Settings] or select [I/O Screen] in the [Screen List] window on Work Space to assign a variable to each of the mapped objects. For how to assign variables, refer to the following.

-  "30.7.3 Allocating I/O (Common)" (page 30-152)

8 Create a Logic Screen and a Base Screen to access the allocated variables and transfer them to the GP.

NOTE

- To check set values, use SDOR instructions.

■ **Details of Objects Used in Standard I/O**

The following table shows I/O read/write actions of an HTB.

Object (Hex)	Sub index	Bit	Description	Parameter
6000	1	0	Input 0	8 bit word
		
	7	Input 7		
6000	2	0	Input 8	8 bit word
		
		3	Input 11	
		4 to 7	Unused	
6100	1	0	Input 0	16 bit word
		
		11	Input 11	
		12 to 15	Unused	
6200	1	0	Output 0	8 bit word
		
		7	Output 7	
6300	1	0	Output 0	16 bit word
		
		7	Output 7	
		8 to 15	Unused	

■ Details of I/O Setting Objects

Object (Hex)	Sub index	Bit	Description	Parameter
2000	1 ... 12	-	Input filter 0 ... Input filter 11	0: None 1: 3ms (default) 2: 12ms
6102	1	0 ... 11 12 to 15	Polarity input 0 ... Polarity input 11 Unused	0: Normal input 1: Reverse input
6103	1	0 ... 11 12 to 15	Input mask 0 ... Input mask 11 Unused	0: Enable mask (default) 1: Disable mask
6302	1	0 ... 11 8 to 15	Polarity output 0 ... Polarity output 7 Unused	0: Normally open output (NO) (default) 1: Normally closed output (NC)
6306	1	0 ... 7 8 to 15	Fallback mode output ^{*1} ... Fallback mode output 7 Unused	0: Retentive state 1: Fallback value enabled (default)
6307	1	0 ... 7 8 to 15	Fallback value output 0 ... Fallback value output 7 Unused	0: Fallback to 0 (default) 1: Fallback to 1
6308	1	0 ... 7 8 to 15	Output mask 0 ... Output mask 7 Unused	0: Retaining state (default) 1: Value of object (Hex) 6300

*1 Fallback mode is a feature that can define the state of output when a communication error occurs (example, turning ON whenever a communication error occurs).

NOTE

- When fallback mode (6306h) is 0, fallback value (6307h) is ignored.

30.7.5 Expanding I/O Using EX Module

Maximum of 7 units of EX modules can be attached to HTB. In addition to expansion of I/O for standard input and output, analog I/O can be used.

You can use the following EX module models and features:

◆ Models and Features of EX Modules


Feature	Type	Summary	Browse to
Standard Input	EXM-DDI8DT	DIO Module: 8-input sink/source	☞ " ◆ Standard Input (Model: EXM-DDI8DT)" (page 30-162)
	EXM-DDI16DT	DIO Module: 16-input sink./source	☞ " ◆ Standard Input (Model: EXM-DDI16DT)" (page 30-163)
Standard Output	EXM-DDO8UT	DIO module: 8-output sink	☞ " ◆ Standard Output (Model: EXM-DDO8UT, EXM-DDO8TT and EXM-DRA8RT)" (page 30-163)
	EXM-DDO8TT	DIO module: 8-output source	
	EXM-DRA8RT	DIO module: 8-output relay	
	EXM-DDO16UK	DIO Module: 16-output sink	☞ " ◆ Standard Output (Model: EXM-DDO16UK, EXM-DDO16TK and EXM-DRA16RT)" (page 30-164)
	EXM-DDO16TK	DIO Module: 16-output source	
	EXM-DRA16RT	DIO Module: 16-output relay	
Standard I/O	EXM-DMM8DRT	DIO Module: 4-input sink/source DIO Module: 4-output relay	☞ " ◆ Standard Input and Output (Model: EXM-DMM8DRT)" (page 30-165)
Analog Input	EXM-AMI2HT	Analog Module: 2-input voltage/current	☞ " ◆ Analog Input (Model: EXM-AMI2HT)" (page 30-170)
Analog Output	EXM-AMO1HT	Analog Module: 1-output voltage/current	☞ " ◆ Analog Output (Model: EXM-AMO1HT)" (page 30-171)
Analog IO	EXM-AMM3HT	Analog Module: 2-input voltage/current Analog Module: 1-output voltage/current	☞ " ◆ Analog I/O (Model: EXM-AMM3HT)" (page 30-172)
	EXM-ALM3LT	Analog Module: 2-input thermocouple/PT100 Analog Module: 1-output voltage/current	☞ " ◆ Analog I/O (Model: EXM-ALM3LT)" (page 30-174)

NOTE

- While HTB is powered ON, such as when it's running, do not install or remove HTB or the EX module.

■ **Standard I/O**

The following describes CANopen objects used in DIO for EX modules. Sub index number "n" (2 or more) is determined by the position of the module connected to HTB and the object that is used.

NOTE	<ul style="list-style-type: none"> For details on the relationship between sub index "n" and attached position of EX module, refer to the following:  " ■ Sub Index" (page 30-179)
-------------	---

◆ **Standard Input (Model: EXM-DDI8DT)**

DIO input module defines active state and input mask using configuration objects.

Object (Hex)	Sub index	Bit	Description	Parameter
6000	n	0	Input 0	8 bit word
		
		7	Input 7	
6100	n	0	Input 0	16 bit word
		
		7	Input 7	
		8 to 15	Unused	
6102	n	0	Polarity input 0	0: Normal input (default) 1: Reverse input
		
		7	Polarity input 7	
		8 to 15	Unused	
6103	n	0	Input mask 0	0: Enable mask (default) 1: Disable mask
		
		7	Input mask 7	
		8 to 15	Unused	

◆ Standard Input (Model: EXM-DDI16DT)

DIO input module defines active state and input mask using configuration objects.

Object (Hex)	Sub index	Bit	Description	Parameter
6000	n	0	Input 0	8 bit word
		
		7	Input 7	
	n+1	0	Input 8	8 bit word
		
		7	Input 15	
6100	n	0	Input 0	16 bit word
		
		15	Input 15	
6102	n	0	Polarity input 0	0: Normal input (default) 1: Reverse input
		
		15	Polarity input 15	
6103	n	0	Input mask 0	0: Enable mask (default) 1: Disable mask
		
		15	Input mask 15	

◆ Standard Output (Model: EXM-DDO8UT, EXM-DDO8TT and EXM-DRA8RT)

DIO output module defines fallback mode and values using configuration objects.

Object (Hex)	Sub index	Bit	Description	Parameter
6200	n	0	Output 0	8 bit word
		
		7	Output 7	
6300	n	0	Output 0	16 bit word
		
		7	Output 7	
		8 to 15	Unused	
6302	n	0	Polarity output 0	0: Normally open output (NO) (default) 1: Normally closed output (NC)
		
		7	Polarity output 7	
		8 to 15	Unused	
6306	n	0	Fallback mode output 0	0: Maintain the state 1: Fallback value enabled (default) (set to the set fallback value)
		
		7	Fallback mode output 7	
		8 to 15	Unused	
6307	n	0	Fallback value output 0	0: Fallback to 0 (default) 1: Fallback to 1
		
		7	Fallback value output 7	
		8 to 15	Unused	

Continued

Object (Hex)	Sub index	Bit	Description	Parameter
6308	n	0	Output mask 0	0: Maintain the state (default) 1: Value of the object (Hex) 6300
		
		7	Output mask 7	
		8 to 15	Unused	

◆ **Standard Output (Model: EXM-DDO16UK, EXM-DDO16TK and EXM-DRA16RT)**

DIO output module defines fallback mode and values using configuration objects.

Object (Hex)	Sub index	Bit	Description	Parameter
6200	n	0	Output 0	8 bit word
		
		7	Output 7	
	n+1	0	Output 8	8 bit word
...	...			
		7	Output 15	
6300	n	0	Output 0	16 bit word
		
		15	Output 15	
6302	n	0	Polarity output 0	0: Normally open output (NO) (default) 1: Normally closed output (NC)
		
		15	Polarity output 15	
6306	n	0	Fallback mode output 0	0: Maintain the state 1: Fallback value enabled (default)
		
		15	Fallback mode output 15	
6307	n	0	Fallback value output 0	0: Fallback to 0 (default) 1: Fallback to 1
		
		15	Fallback value output 15	
6308	n	0	Output mask 0	0: Maintain the state (default) 1: Value of the object (Hex) 6300
		
		15	Output mask 15	

◆ **Standard Input and Output (Model: EXM-DMM8DRT)**

DIO input and output module defines fallback mode and values using configuration objects.

<Standard input>

Object (Hex)	Sub index	Bit	Description	Parameter
6000	n	0 ... 3	Input 0 ... Input 3	8 bit word
		4 to 7	Unused	
6100	n	0 ... 3	Input 0 ... Input 3	16 bit word
		4 to 15	Unused	
6102	n	0 ... 3	Polarity input 0 ... Polarity input 3	0: Normal input (default) 1: Reverse input
		4 to 15	Unused	
6103	n	0 ... 3	Input mask 0 ... Input mask 3	0: Enable mask (default) 1: Disable mask
		4 to 15	Unused	

<Standard output>

Object (Hex)	Sub index	Bit	Description	Parameter
6200	n	0 ... 3	Output 0 ... Output 3	8 bit word
		4 to 7	Unused	
6300	n	0 ... 3	Output 0 ... Output 3	16 bit word
		4 to 15	Unused	
6302	n	0 ... 3	Polarity output 0 ... Polarity output 3	0: Normally open output (NO) (default) 1: Normally closed output (NC)
		4 to 15	Unused	
6306	n	0 ... 3	Fallback mode output 0 ... Fallback mode output 3	0: Maintain the state 1: Fallback value enabled (default)
		4 to 15	Unused	

Continued

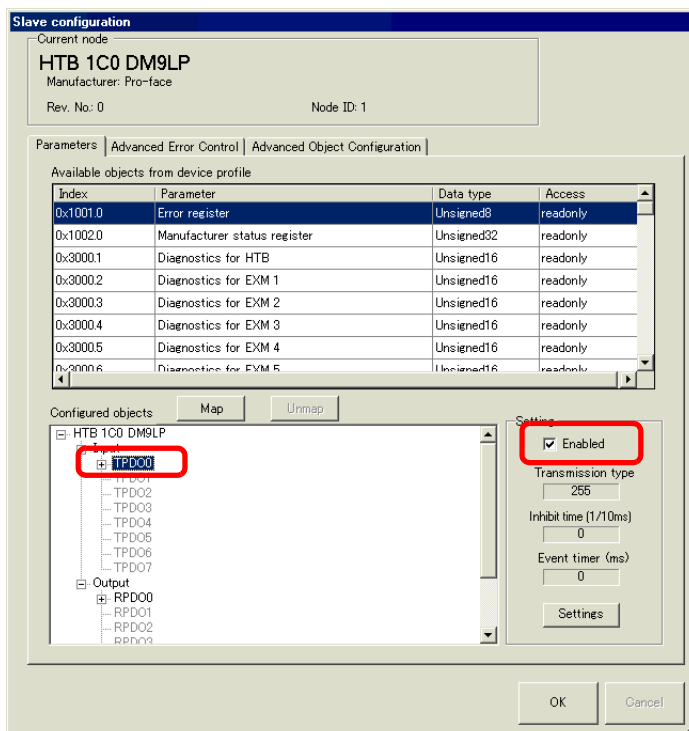
Object (Hex)	Sub index	Bit	Description	Parameter
6307	n	0	Fallback value output 0	0: Fallback to 0 (default) 1: Fallback to 1
		
		3	Fallback value output 3	
		4 to 15	Unused	
6308	n	0	Output mask 0	0: Maintain the state (default) 1: Value of the object (Hex) 6300
		
		3	Output mask 3	
		4 to 15	Unused	

◆ **Setup Procedure**

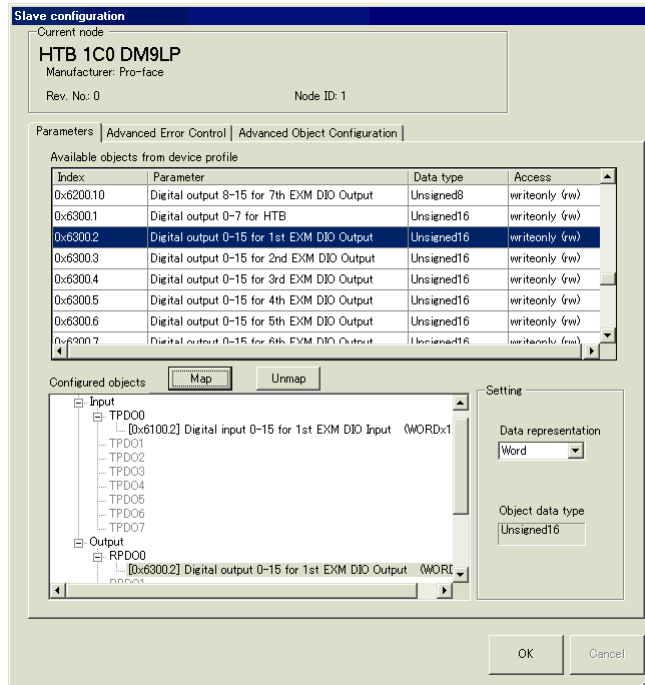
The following introduces how to map PDO on slave side and how to set objects when adding EX module "EXM-DMM8DRT" to HTB (using 12-input points and 8-output points).

<Settings in GP-Pro EX>

- 1 Click [Settings] from System window [I/O Driver Settings] and add HTB to the network.
 ☞ "30.7.2 Setting Procedure" (page 30-148)
- 2 Set standard input and output of HTB.
 ☞ "30.7.4 Standard I/O ■ Setup Procedure" (page 30-156)



- Set the standard I/O of EX module. Select the input value object "0x6100.2" to map to TPDO. In [Settings], select the data type. Then map "0x6300.2" to RPDO and select the data type.



NOTE

- Map the input data from HTB (CANopen slave) to the GP (CANopen master) for TPDO, and map the object relating to the output data from the GP to HTB for RPDO.
- Map error register "0x1001.1", status register "0x1002.1" and module diagnosis "0x3000.1" and "0x3000.2" to TPDO as required.

- Click [Advanced Object Configuration] tab to configure detailed settings of the object. Set fallback mode (0x6306.1 and 0x6306.2) and fallback value (0x6307.1 and 0x6307.2) as required and click [OK] to close the dialog box.

NOTE

- For details of objects, refer to the following.
[☞ "30.7.10 List of HTB Objects" \(page 30-211\)](#)

- Click [I/O Screen] in [I/O Driver Settings] or select [I/O Screen] in the [Screen List] window on Work Space to assign a variable to each of the mapped objects. For how to assign variables, refer to the following.
[☞ "30.7.3 Allocating I/O \(Common\)" \(page 30-152\)](#)

- Create a Logic Screen and a Base Screen to access the allocated variables and transfer them to the GP.

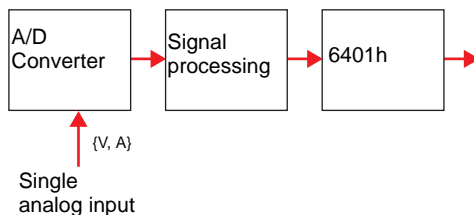
NOTE

- To check set values, use SDOR instructions.

■ Analog I/O

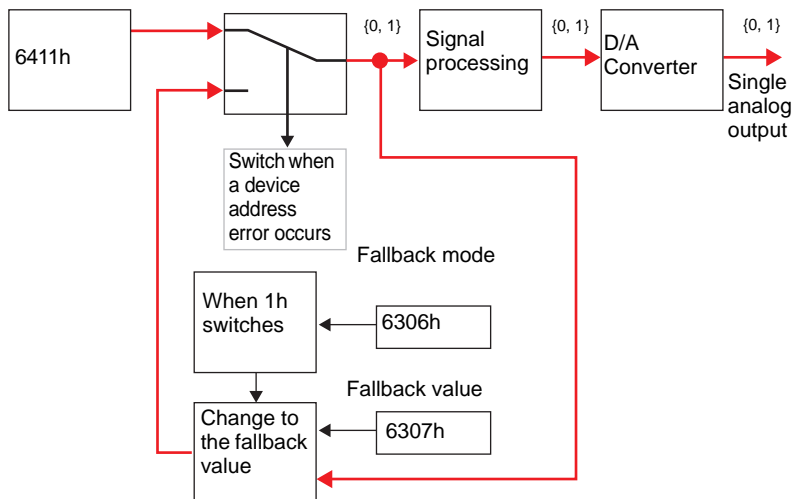
<Analog input>

It is read by the unit of 16 bit word. Each channel value is stored in sub index of the object 6401h.



<Analog output>

It is written by the unit of 16 bit word. Output bit is stored in sub index of the object 6411h. It is switched to fallback mode when an error occurs.



NOTE • Data is not updated unless "PDO transmission enabled/disabled" (6423h) is set to 1 (enabled).

<Analog I/O Settings - Common Objects (Model: EXM-AMI2HT, EXM-AMO1HT, EXM-AMM3HT and EXM-ALM3LT)>

The following describes CANopen objects used in analog I/O for EX modules. Sub index number "n" (2 or more) is determined by the position of the module connected to HTB and the object that is used.

NOTE • For details on the relationship between sub index "n" and attached position of EX module, refer to the following:
 ☞ " ■ Sub Index" (page 30-179)

EX modules that support analog have different setup objects between models. The following common values are used in all channel and registries.

<Analog input>

Object (Hex)	Description	Parameter
2100	Data Type	0: Unused 2: Current (4 to 20mA) 3: Voltage (0 to 10V) 5: Thermocouple (K type) 6: Thermocouple (J type) 7: Thermocouple (T type) 8: PT100
2101	Data range	0: Fixed 1: User setting 2: Celsius 3: Fahrenheit
2102	Lower limit (when 2101h = 1)	Lowest value
2103	Upper limit (when 2101h = 1)	Highest value
6401	Read value	
6422	PDO transmission channel number	Indicates the channel in which active bit caused an event.
6423	PDO transmission enabled/disabled	0: Disables PDO transmission 1: Enables PDO transmission Transmit the command to all the channels.

<Analog output>

Object (Hex)	Description	Parameter
2200	Data Type	0: Unused 2: Current (4 to 20mA) 3: Voltage (0 to 10V) 5: Thermocouple (K type) 6: Thermocouple (J type) 7: Thermocouple (T type) 8: PT100
2201	Data range	0: Fixed 1: User setting 2: Celsius 3: Fahrenheit
2202	Lower limit (when 2201h = 1)	Lowest value
2203	Upper limit (when 2201h = 1)	Highest value
6411	Written value	
6443	Fallback mode	0: Fallback 1: Maintain the state
6444	Fallback value	Fallback value

◆ Analog Input (Model: EXM-AMI2HT)

Analog input module defines input range and customizable value using configuration objects.

NOTE

- Analog channels can be mixed.

Channel	Object (Hex)	Sub index	Description	Parameter	Default of the parameter
CH1	2100	n	Data Type	0: Unused 2: Current (4 to 20mA) 3: Voltage (0 to 10V)	0
	2101	n	Data range	0: Fixed 1: User setting	1
	2102	n	Lower limit (when 2101h = 1)	Lowest value	0
	2103	n	Upper limit (when 2101h = 1)	Highest value	7FFFh
	6401	n	Read value	Input	0
	6422	1	PDO transmission channel number	Indicates the channel in which active bit caused an event.	0
	6423	0	PDO transmission enabled/disabled	0: Disable PDO transmission 1: Enable PDO transmission	0
CH2	Same as CH1. Sub index is n+1.				

◆ Analog Output (Model: EXM-AMO1HT)

Analog output module defines output range, customizable value, and fallback value using configuration object.

Channel	Object (Hex)	Sub index	Description	Parameter	Default of the parameter
CH1	2200	n	Data Type	0: Unused 2: Current (4 to 20mA) 3: Voltage (0 to 10V)	0
	2201	n	Data range	0: Fixed 1: User setting	1
	2202	n	Lower limit (when 2201h = 1)	Lowest value	0
	2203	n	Upper limit (when 2201h = 1)	Highest value	7FFFh
	6411	n	Written value	Output	0
	6443	n	Fallback mode	0: Maintain the state 1: Fallback	1
	6444	n	Fallback value (when 6443h = 1)	Fallback value	0

◆ **Analog I/O (Model: EXM-AMM3HT)**

Analog I/O module defines the range of I/O, customizable value, fallback mode, and fallback value using configuration objects.

NOTE

- Analog channels can be mixed.

Channel	Object (Hex)	Sub index	Description	Parameter	Default of the parameter
CH1 (Input)	2100	n	Data Type	0: Unused 2: Current (4 to 20mA) 3: Voltage (0 to 10V)	0
	2101	n	Data range	0: Fixed 1: User setting	1
	2102	n	Lower limit (when 2101h = 1)	Lowest value	0
	2103	n	Upper limit (when 2101h = 1)	Highest value	7FFFh
	6401	n	Read value	Input	0
	6422	1	PDO transmission channel number		0
	6423	0	PDO transmission enabled/disabled	0: Disable PDO transmission 1: Enable PDO transmission	0
CH2 (Input)	Same as CH1 (Input). Sub index is n+1.				

Continued

Channel	Object (Hex)	Sub index	Description	Parameter	Default of the parameter
CH1 (Output)	2200	n	Data Type	0: Unused 2: Current (4 to 20mA) 3: Voltage (0 to 10V)	0
	2201	n	Data range	0: Fixed 1: User setting	1
	2202	n	Lower limit (when 2201h = 1)	Lowest value	0
	2203	n	Upper limit (when 2201h = 1)	Highest value	7FFFh
	6411	n	Written value	Output	0
	6443	n	Fallback mode	0: Maintain the state 1: Fallback	1
	6444	n	Fallback value (when 6443h = 1)	Fallback value	0

◆ Analog I/O (Model: EXM-ALM3LT)

The temperature input module uses configuration objects to determine the thermocouple range and customizable values. The analog output module defines the output range, customizable values, fallback mode, and fallback value.

NOTE

- Analog channels can be mixed.

Channel	Object (Hex)	Sub index	Description	Parameter	Default of the parameter
CH1 (Input)	2100	n	Data Type	0: Unused 5: Thermocouple (K type) 6: Thermocouple (J type) 7: Thermocouple (T type) 8: PT100	0
	2101	n	Data range	0: Fixed 1: User setting 2: Celsius 3: Fahrenheit	1
	2102	n	Lower limit (when 2101h = 1)	Lowest value	0
	2103	n	Upper limit (when 2101h = 1)	Highest value	7FFFh
	6401	n	Read value	Input	0
	6422	1	PDO transmission channel number		0
	6423	0	PDO transmission enabled/disabled	0: Disable PDO transmission 1: Enable PDO transmission	0
CH2 (Input)	Same as CH1 (Input). Sub index is n+1.				

Continued

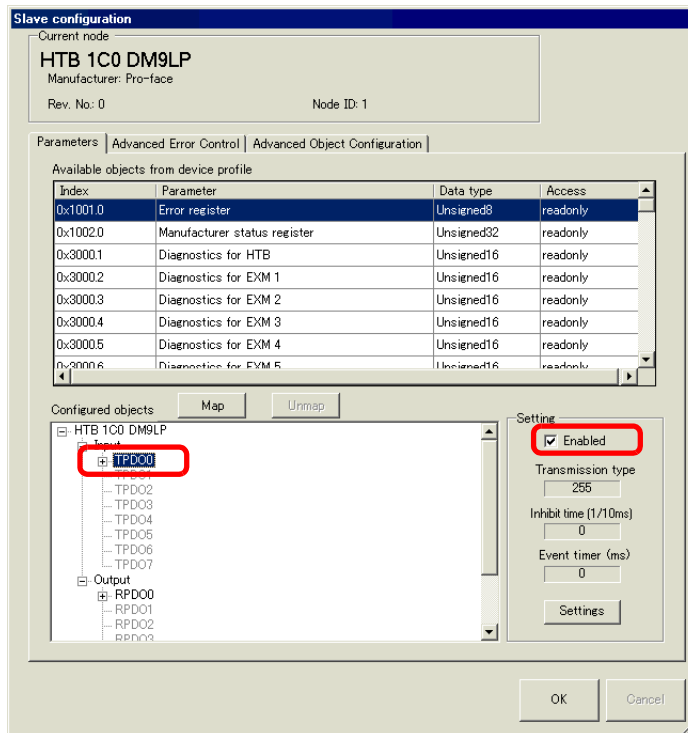
Channel	Object (Hex)	Sub index	Description	Parameter	Default of the parameter
CH1 (Output)	2200	n	Data Type	0: Unused 2: Current (4 to 20mA) 3: Voltage (0 to 10V)	0
	2201	n	Data range	0: Fixed 1: User setting	1
	2202	n	Lower limit (when 2201h = 1)	Lowest value	0
	2203	n	Upper limit (when 2201h = 1)	Highest value	7FFFh
	6411	n	Written value	Output	0
	6443	n	Fallback mode	0: Maintain the state 1: Fallback	1
	6444	n	Fallback value (when 6443h = 1)	Fallback value	0

◆ **Setup Procedure**

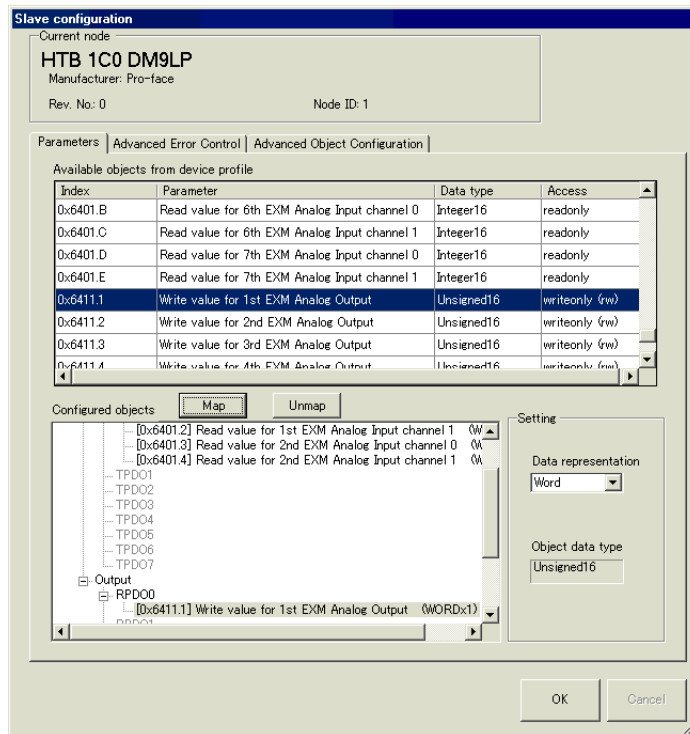
The following describes how to map PDO on slave side and how to set objects when adding EX module "EXM-AMI2HT" and "EXM-AMM3HT" to HTB (using 12-input points and 8-output points).

<Settings in GP-Pro EX>

- 1 Click [Settings] from System window [I/O Driver Settings] and add HTB to the network.
☞ "30.7.2 Setting Procedure" (page 30-148)
- 2 Set standard input and output of HTB.
☞ "30.7.4 Standard I/O ■ Setup Procedure" (page 30-156)



- 3 Set up analog input of EX module. Select the input value object "0x6401.1 to 0x6401.4" to map to TPDO. In [Settings], select the data type. In addition, map "0x6411.1" to RPDO and select the data type.




NOTE

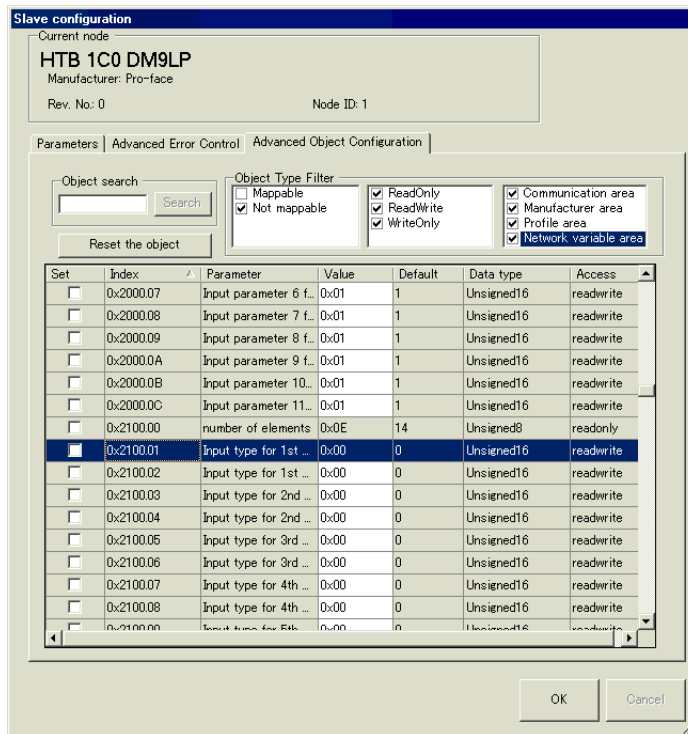
- Map the input data from HTB (CANopen slave) to the GP (CANopen master) for TPDO, and map the object relating to the output data from the GP to HTB for RPDO.
- Map error register "0x1001.1", status register "0x1002.1", and module diagnosis "0x3000.1", "0x3000.2" and "0x3000.2" to TPDO as required.

4 Open [Advanced Object Configuration] tab and configure detailed settings of the object. Set the data type of analog input (0x2100.1 to 0x2100.4) and the data type of analog output (0x2200.1).


Set the data range (Input: 0x2101.1 to 0x2101.4, output: 0x2201.1), PDO transmission enabled/disabled (0x6423), fallback mode (0x6443.1) and fallback value (0x6444.1), and then click [OK] to close the dialog box.

NOTE

- For details of objects, refer to the following.
 "30.7.10 List of HTB Objects" (page 30-211)



5 Click [I/O Screen] in [I/O Driver Settings] or select [I/O Screen] in the [Screen List] window on Work Space to assign a variable to each of the mapped objects. For how to assign variables, refer to the following.

 "30.7.3 Allocating I/O (Common)" (page 30-152)

6 Create a Logic Screen and a Base Screen to access the allocated variables and transfer them to the GP.

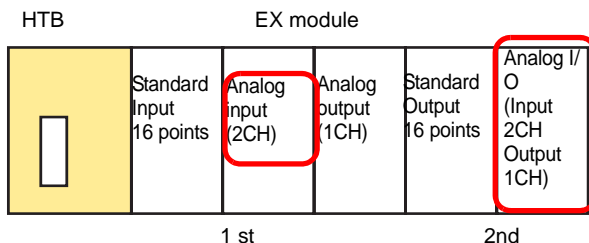
NOTE

- To check set values, use SDOR instructions.
- When changing the analog setting value used by SDOW, you need to use the expansion bus reset (0x3300) to stop the internal bus before making the change.

■ Sub Index

Sub index 0 stores the total number of channels used per feature, and from sub index 1 onward, each channel on the EX module is mapped, in order, beginning with the channel closest to HTB.

For example, for analog input feature



The analog input object's sub index contains the number of modules used in the analog input only, counting from those closest to HTB.

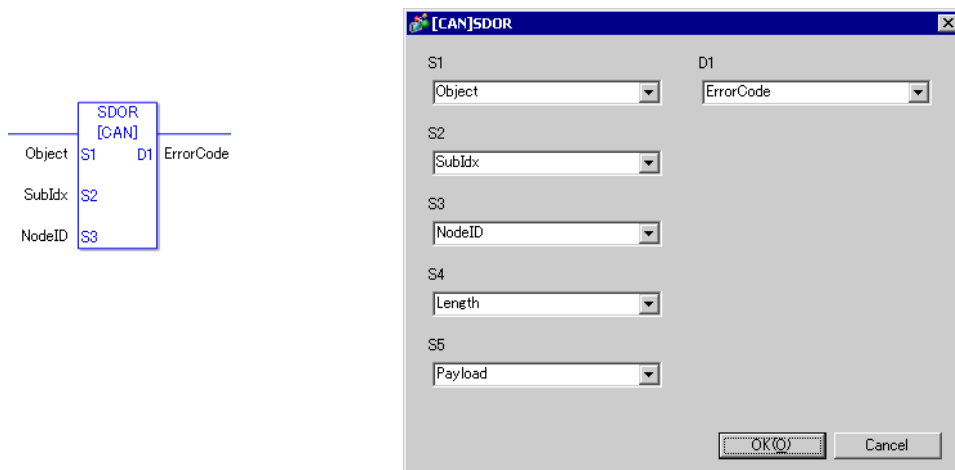
Sub index		
0	4	<- Total number of channels for analog input
1	Input data	<- CH1 data of the 1st EX module
2	Input data	<- CH2 data of the 1st EX module
3	Input data	<- CH1 data of the 2nd EX module
4	Input data	<- CH2 data for 2nd EX module

30.7.6 Using I/O Driver Instructions

You can change set values of objects and check the values while the program is running if you use I/O Driver Instructions on Logic Screen.

◆ SDOR

From service data objects (hereafter referred as "SDO"), read the value. Double-click the inserted instruction and the following dialog box appears.



NOTE

- Do not run two or more SDO instructions (SDOR, SDOW) at the same time. You can only run one SDO instruction at a time.

Operand Settings

S1: Specify index of the object.

S2: Specify sub index of the object.

S3: Specify node ID.

S4: Specify length (byte number) of SDO access.

S5: Specify where to store read data. If you specify an array element in the system variable #L_IOMasterDrv[0] to [255], that becomes the start address where the number of bytes specified in parameter S4 is written.

D1: Specify the variable where error codes are stored if an instruction does not act as expected.

< Error Code >

0: Normal

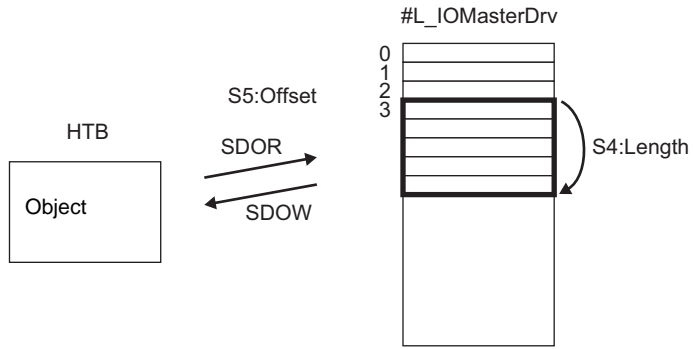
1: Error in operand number

111: Error in set up value for operand S4 or S5

120: Error in SDO communication or error in set up value for operand S1, S2, or S3

123: Timeout error

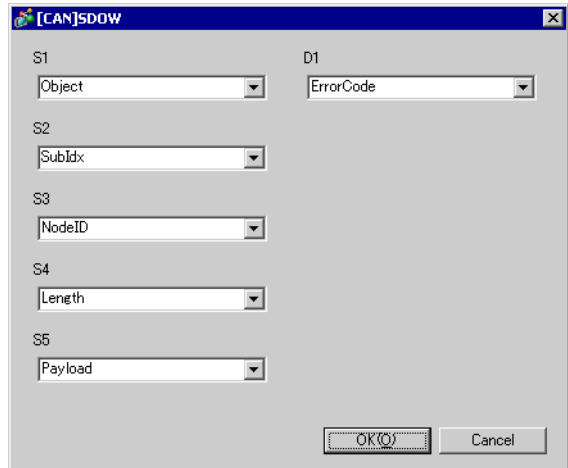
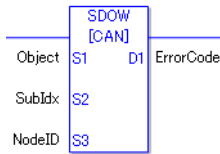
Example: S4=20, S5=3



◆ SDOW

Write in SDO.

Double-click the inserted instruction and the following dialog box appears.



NOTE

- Do not run two or more SDO instructions (SDOR, SDOW) at the same time. You can only run one SDO instruction at a time.

Operand Settings

S1: Specify index of the object.

S2: Specify sub index of the object.

S3: Specify node ID.

S4: Specify length (byte number) of SDO access.

S5: Specify where to write data. If you specify an array element in the system variable #L_IOMasterDrv[0] to [255], that becomes the start address in the specified object where the number of bytes specified in parameter S4 is written.

D1: Specify the variable where error codes are stored if an instruction does not act as expected.

< Error Code >

0: Normal

1: Error in operand number

111: Error in set up value for operand S4 or S5

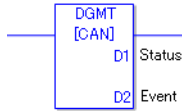
120: Error in SDO communication or error in set up value for operand S1, S2, or S3

123: Timeout error

◆ **DGMT**

Read the state on the master side.

Double-click the inserted instruction and the following dialog box appears.



Operand Settings

D1: Specify the variable to store status.

D2: Specify the variable to store information on an event.

- Status

31	16 15	8 7	0
Unused (0 fixed)		Communication status	Master status

Bit	Summary		Detail
0 to 7	0x00	Initializing	Initializing
	0x01	Reset request	Change to reset.
	0x40	Reset	Resetting
	0x60	Slave checking	Check slave mapping
	0x61	Resetting network	Reset all the nodes
	0x62	Network standby	Waits for a specified amount of time while the communication command can be reset.
	0x64	Initializing each slave	Initialize each slave in the network.
	0x70	Module defect	A deficiency exists for the module.
	0x80 to 83*1	Clearing	Scan the network. The state in which firmware can be started.
	0x90	Fatal error	Fatal error occurred in the network. Firmware will be reset.
	0xA0 to A3*1	Operating	The network is operating
	0xC0	Stopped	The network is stopped.
	0xE0 to E3*1	Preparing for operation	The network is in pre-operating state.
8	Queue overrun (RXLP)		Overrun of receiving queue with low priority*2 has occurred.

Continued

Bit	Summary	Detail
9	Controller overrun	CAN controller overrun has occurred
10	Controller bus off	CAN controller bus off has occurred
11	Controller error occurred	Error has occurred in CAN controller
12	Controller error recovered	CAN controller has recovered from error state
13	Queue overrun (TXLP)	Overrun of sending queue with low priority ^{*2} has occurred.
14	Queue overrun (RXHP)	Overrun of receiving queue of high priority ^{*3} has occurred.
15	Queue overrun (TXHP)	Overrun of sending queue with high priority ^{*3} has occurred.

*1 The following states are shown according to the state of lower 4 bits.

Bit 0: Error bit of the optional node or the node that is not set.

0: No error

1: Error in 1 or more optional nodes or the node that is not set.

Bit 1: Error bit of mandatory node

0: No error

1: Error in 1 or more mandatory nodes

Bit 2: Common action bit

0: No active nodes

1: At least one active node

Bit 3: Action bit of CANopen master module

0: Not active

1: Active

*2 Queue with low priority is used for heartbeat, node guard and SOD transfer.

*3 Queue with high priority is for messages of TPDO, NMT command, SYNC and EMCY.

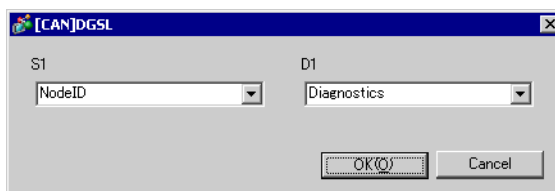
- Event

31	16 15	0
Unused (0 fixed)		Event information

Bit	Summary
0	Network communication error
1	Node ID error
2	Error controlling event of mandatory node
3	Identification error of mandatory node or incomplete concise DCF
4	Identification error of optional node
5 to 6	Reserved
7	Set if there are changes to bits in the bit list
8	Requested feature is unsupported
9	Data byte number of received RxPDO is too small
10	Incomplete concise DCF
11	Overrun of application specific SDO queue
12	Reserved
13	No slave
14	Change the state of the network with NMT command
15	Change the sate of the node with NMT command

◆ **DGSL**

Read the state on slave side.



Operand Settings


S1: Specify node ID.

D1: Specify the variable to store slave information.

- Slave information

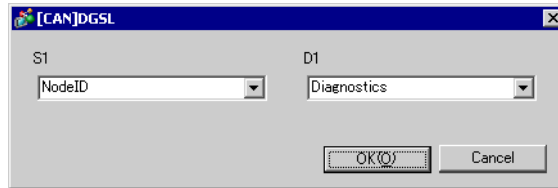
Bit	Summary	Detail
0	Mapping	0: Slave is not assigned to master. 1: Slave is assigned to master.
1	Configuration	0: Not set as slave. Startup is disabled. 1: Set as slave. Startup enabled.
2	Defect	0: Configured Slave 1: Unconfigured Slave
3	Emergency message (EMCY)	0: Slave has not sent the emergency message 1: Slave has sent the emergency message.
4	Operating	0: Slave is not operating. 1: Slave is operating.
5	Stopped	0: Slave is not stopped 1: Slave is stopped
6	Preparing for operation	0: Slave is not preparing for operation. 1: Slave is preparing for operation
7	Mismatch error	0: Normal concise DCF file 1: Incorrect concise DCF file
8	Slave configuration error	0: Objects of DCF and slave are matched. 1: Objects of DCF and slave are mismatched.
9	Identification error	0: Information of slave node is normal. 1: Information of slave node has error.
10 to 15	Reserved	-

◆ Setting Method

1 Open Logic Screen (MAIN or SUB) and right-click the rung to select [Insert Instructions (I)]. Click  to select [9. I/O Driver Instructions], point to [CAN], and click [DGSL] (instruction to read the state on slave side).

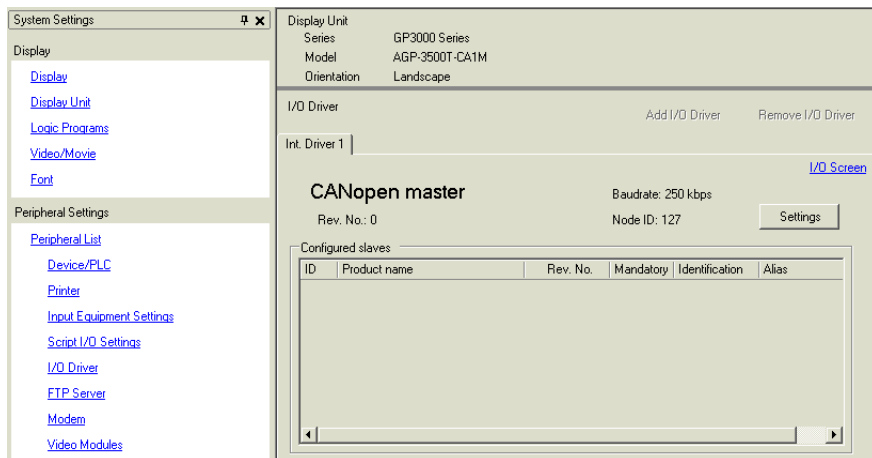
2 Mapping variables to operands.


Double-click the inserted instruction. The following dialog box appears. Map variables and click [OK].



30.7.7 [I/O Driver] Settings Guide

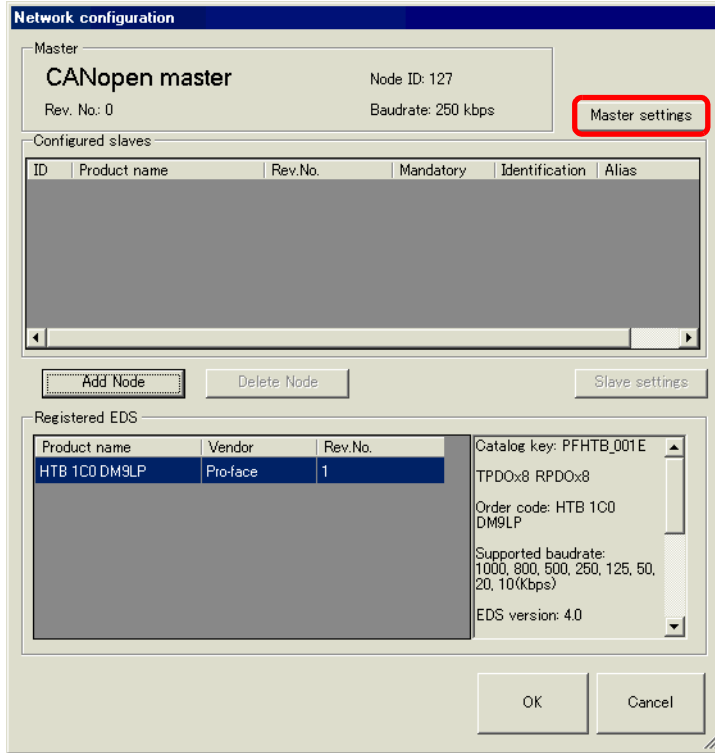
When AGP-*****-CA1M is selected in Display Settings, click [I/O Driver] in System Settings to open the following screen.




Setting	Description	
Revision Number	Displays revision number for EDS file of CANOpen master.	
Baud rate	Displays baud rate of CANOpen master.	
Node ID	Displays node ID of CANOpen master.	
Settings	Displays [Network Settings] dialog box.  " ■ Network Settings Dialog Box" (page 30-189)	
I/O Screen	Go to I/O Screen.	
Configured slaves	Slave devices added to CANopen network displays. The details of [Network Settings] and [Slave Settings] dialog boxes display in the list.	
	ID	Node ID of the slave in CANopen network.
	Product Name	Displays the product name of the slave.
	Revision Number	Displays the slave's EDS file revision number.*1
	Mandatory Node	Displays whether the slave is mandatory in the network.
	Identification	Displays whether or not to run the device type (object's 1000h) reference.
	Alias	Displays comments set for the node.

*1 EDS file is a text file in ASCII format that describes specification (example, useful features and objects) of the device. It is required to register and set the device to the network.

■ Network Settings Dialog Box



Setting	Description
Master Settings	Displays Master Settings dialog box.  " ◆ Master Settings Dialog Box" (page 30-191)
Configured slaves	Slave devices added to CANOpen network displays. The details of [Network Settings] and [Slave Settings] dialog boxes display in the list.
ID	Node ID of the slave in CANOpen network.
Product Name	Displays the product name of the slave.
Revision Number	Displays the slave's EDS file revision number.*1
Mandatory Node	Defines whether the slave is mandatory in the network. Select this option to turn ON bit 3 in object 1F81h
Identification	Defines whether or not to run the device type reference. Select this option to check if the structure matches the set up value in object 1000h. If 1000h is zero, reference does not take place.
Alias	Defines the comment set for the node.
Add Node	Add the slave selected in [EDS List] to [Configured slaves]. You can change Node ID (1 to 63) of the added node. You can also enter comments within 18 characters.
Delete Node	Deletes the slave selected in [Configured slaves] from the network.

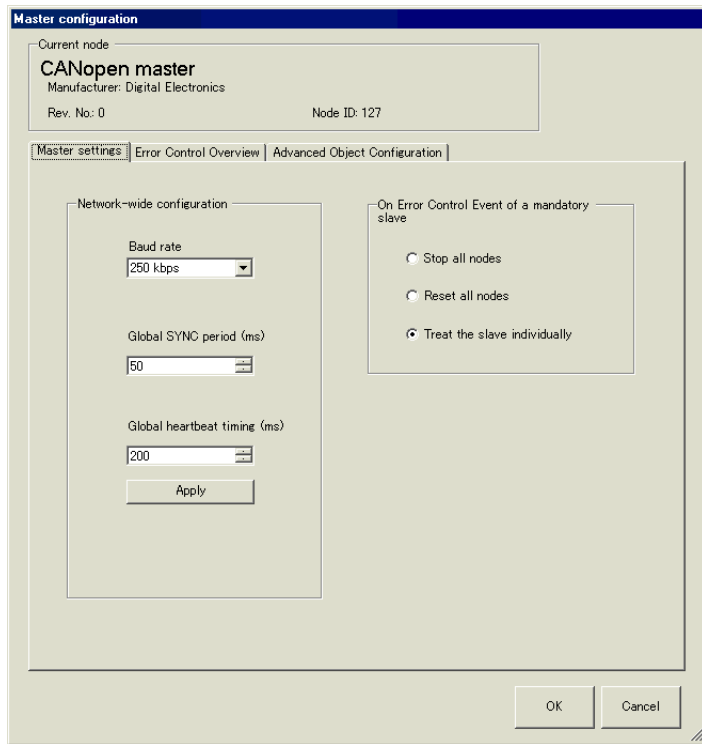
Continued

Setting	Description
Slave Settings	Displays Slave Settings dialog box. ☞ " ◆ Slave Settings Dialog Box" (page 30-197)
Registered EDS	Displays available slaves in the list.
Product Name	Displays the product name of the slave.
Vendor	Displays a vender name of slave.
Revision Number	Displays the revision number for EDS file of the slave.
Detail Settings	Information on the currently selected EDS displays. The contents that will display include catalog key, number of TPDO and RPDO, order code, corresponding baud rate, EDS version, EDS information, creator, created date, editor, and edited date.

*1 EDS file is a text file in ASCII format that describes specification (example, useful features and objects) of the device. It is required to register and set the device to the network.

◆ **Master Settings Dialog Box**

- Master Settings



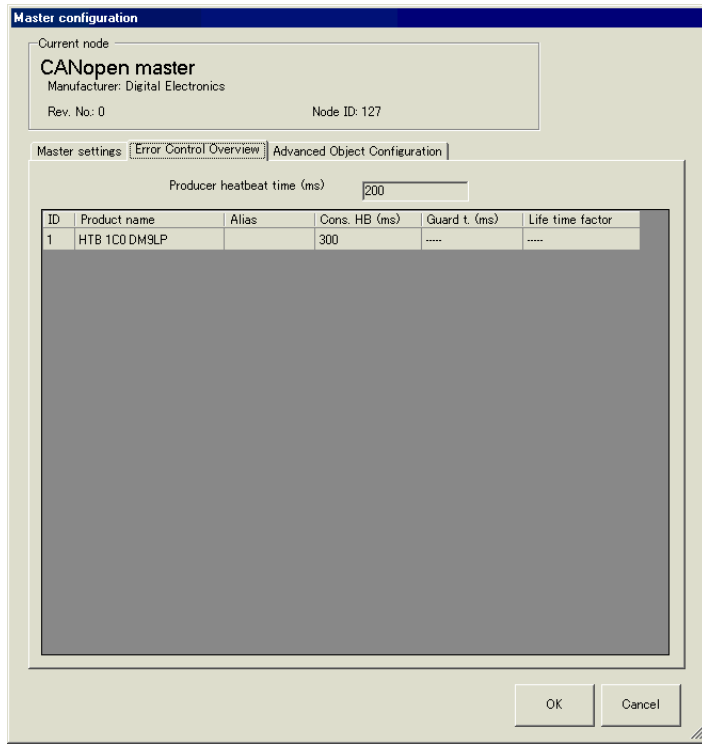
Setting	Description														
Network-wide configuration	Configures CANopen network settings.														
Baud rate	<p>Selects corresponding baud rate from [50kbps], [125kbps], [250kbps], [500kbps], [800kbps] and [1000kbps].</p> <table border="1"> <thead> <tr> <th>Baud rate (bps)</th> <th>Wire length (m)</th> </tr> </thead> <tbody> <tr> <td>50k</td> <td>1000</td> </tr> <tr> <td>125k</td> <td>500</td> </tr> <tr> <td>250k</td> <td>250</td> </tr> <tr> <td>500k</td> <td>100</td> </tr> <tr> <td>800k</td> <td>25</td> </tr> <tr> <td>1000k</td> <td>4</td> </tr> </tbody> </table> <p>NOTE • Set the slave (HTB) baud rate on the HTB unit itself.</p>	Baud rate (bps)	Wire length (m)	50k	1000	125k	500	250k	250	500k	100	800k	25	1000k	4
Baud rate (bps)	Wire length (m)														
50k	1000														
125k	500														
250k	250														
500k	100														
800k	25														
1000k	4														
SYNC Period	Defines the frequency of the SYNC signal sent by the slave unit. The set up range is 0, or 3 to 32767, and stored in object 1006h.														

Continued

Setting		Description												
	Heartbeat Time	<p>Set the heartbeat time sent from slave to master (or from master to slave). The unit is in milliseconds (ms) and setting range is from 0, or 50 to 21844. Press [Apply to All Nodes] and the value of object 1016h (consumer heartbeat time) for CANopen master changes to the automatically optimized value. The value of the object 1017h (producer heartbeat time) for all slaves is overwritten by the value set here; the value is automatically optimized.</p> <p>NOTE</p> <ul style="list-style-type: none"> To individually specify heartbeat time on slave side, it can be set in the [Slave Settings] dialog box. 												
Boot-up configuration		Set the action when starting CANopen master.												
	On Error Control Event of a mandatory slave	<p>Select the action when error occurs in the slave set in [Mandatory] Node] from [Stop All Nodes], [Reset All Nodes] and [Treat the slave individually]. When this item is set, bit 4 to 5 of the object 1F80h for CANopen master is as follows:</p> <table border="1"> <thead> <tr> <th></th> <th>Bit 4</th> <th>Bit 5</th> </tr> </thead> <tbody> <tr> <td>Stop all nodes</td> <td>ON</td> <td>ON</td> </tr> <tr> <td>Reset all nodes</td> <td>ON</td> <td>OFF</td> </tr> <tr> <td>Treat the slave individually</td> <td>OFF</td> <td>OFF</td> </tr> </tbody> </table>		Bit 4	Bit 5	Stop all nodes	ON	ON	Reset all nodes	ON	OFF	Treat the slave individually	OFF	OFF
	Bit 4	Bit 5												
Stop all nodes	ON	ON												
Reset all nodes	ON	OFF												
Treat the slave individually	OFF	OFF												

- Error Control Overview

Displays the state of each node. No settings are allowed.



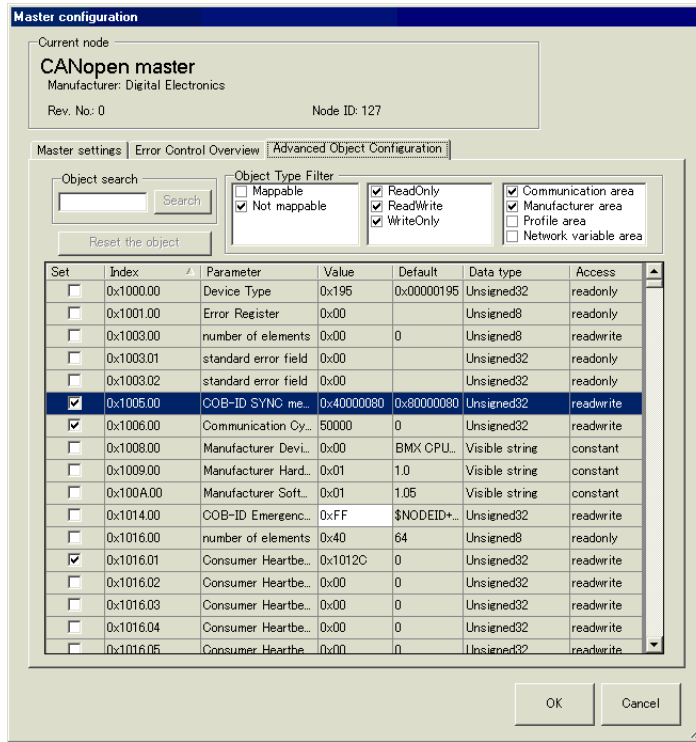
Setting	Description
Producer Heartbeat Time	Displays the sending period of the heartbeat set in [Heartbeat Time] in [Master Settings] tab. Unit is in milliseconds (ms). It is stored in the object 1017h for CANopen master.
Node ID	Node ID in CANopen network.
Product Name	Display EDS product name.
Alias	Displays comments set for the node.
Consumer Heartbeat	Display consumer heartbeat set in each slave. It is stored in the object 1016h for slave.
Guard Time	When [Use Node Guard] is selected in [Advanced Error Control] tab in [Slave Settings], displays [Guard Time] that is set. It is stored in the object 100Ch on slave side.
Life time factor	When [Use Node Guard] is selected in [Advanced Error Control] tab in [Slave Settings], displays [Life time factor] that is set. It is stored in the object 100Dh on slave side.

- Advanced Object Configuration

You can change the values of read-write and write-only (not grayed out) objects. The values set here are written in the objects during startup.

NOTE

- To use Advanced Object Configuration, you must have sufficient knowledge about CANopen.



Setting	Description
Object Search	Enter the object you want to search, or enter the parameter text string, then click [Search] to find the corresponding object in the object list.
Reset the object	Returns the set up value of the object selected in the [Object List] to its initial value.

Continued

Setting	Description
Object type filter	<p>Select the extracting conditions you want to display in [Available objects from device profile] from the following three categories:</p> <ul style="list-style-type: none"> • PDO mapping <ul style="list-style-type: none"> PDO mapping enabled PDO mapping disabled • Access method <ul style="list-style-type: none"> Read only Write only Read/Write enabled • Data Area <ul style="list-style-type: none"> Commutation Area (1000h to 1FFFh) Manufacturer Area (2000h to 5FFFh) Profile Area (6000h to 9FFFh) Network variable area (A000h to AFFFh)

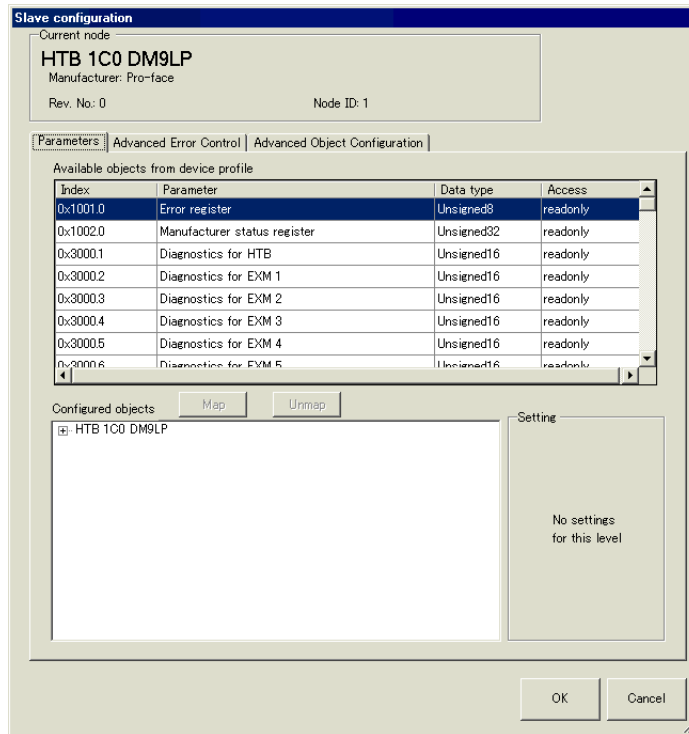
Continued

Setting	Description
Available objects from device profile	Display the list of objects according to the conditions selected in [Object type filter].
Set	When the "Set Value" changes, this check box is automatically selected; you can confirm that the object has changed. When the check box is cleared, the object returns to default.
Index	Display index and sub index of the object. The sub index appears after the comma. For example, for "0x1003.2", "0x1003" is index and "2" is sub index.
Parameter	Display parameter name of the object.
Setting Value	The value of the object can be changed. The number with "0x" indicates a hexadecimal, otherwise without, it indicates a decimal number.)
Default	Display default of the object.
Data Type	<p>Display data type of the object. The following data types are included.</p> <ul style="list-style-type: none"> • Boolean (Single bit value) • Integer8 (Integer with 8 bit code) • Integer16 (Integer with 16 bit code) • Integer24 (Integer with 24 bit code) • Integer32 (Integer with 32 bit code) • Integer40 (Integer with 40 bit code) • Integer48 (Integer with 48 bit code) • Integer56 (Integer with 56 bit code) • Integer64 (Integer with 64 bit code) • Unsigned8 (Integer without 8 bit code) • Unsigned16 (Integer without 16 bit code) • Unsigned24 (Integer without 24 bit code) • Unsigned32 (Integer without 32 bit code) • Unsigned40 (Integer without 40 bit code) • Unsigned48 (Integer without 48 bit code) • Unsigned56 (Integer without 56 bit code) • Unsigned64 (Integer without 64 bit code) • Float (32 bit single accuracy fixed decimal point) • Float64 (64 bit single accuracy fixed decimal point) • Visible String (Text string including ASCII text) • Octet string (Array of integer without 8 bit code) • Unicode string (Array of integer without 16 bit code) • Bit string (Array of single bit) • Time of day (48 bit value indicating time and date) • Time difference (48 bit value indicating time) • Domain (Application specific data block) • Reserved (Reservation)
Access method	<p>Displays access method of the objects. The following types are included.</p> <ul style="list-style-type: none"> • readonly (Read only) • writeonly (Write only) • readwrite (Read/Write) • constant (Constant)

◆ **Slave Settings Dialog Box**

Configure detailed settings of the slave selected in [Slave Settings].

- Communication parameters

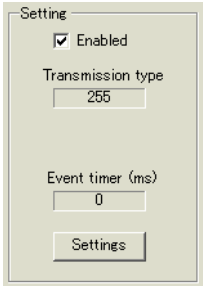
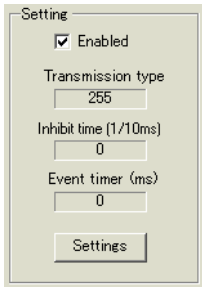


Setting	Description
Available objects from device profile	Display the list of objects that can be mapped to PDO.
Index	Display index and sub index of the object. The sub index appears after the comma. For example, for "0x1003.2", "0x1003" is index and "2" is sub index.
Parameter	Display parameter name of the object.

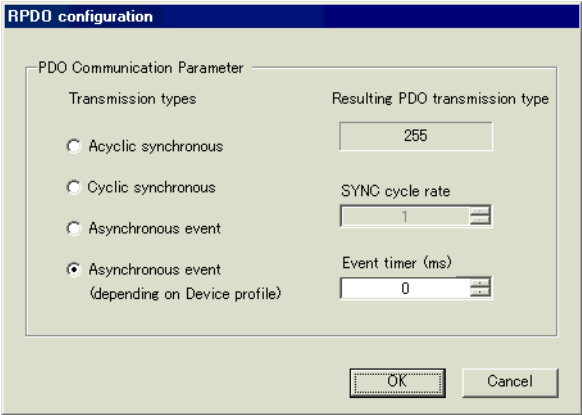
Continued

Setting	Description
<p style="writing-mode: vertical-rl; transform: rotate(180deg);">Available objects from device profile</p> <p>Data Type</p>	<p>Display data type of the object. The following data types are included.</p> <ul style="list-style-type: none"> • Boolean (Single bit value) • Integer8 (Integer with 8 bit code) • Integer16 (Integer with 16 bit code) • Integer24 (Integer with 24 bit code) • Integer32 (Integer with 32 bit code) • Integer40 (Integer with 40 bit code) • Integer48 (Integer with 48 bit code) • Integer56 (Integer with 56 bit code) • Integer64 (Integer with 64 bit code) • Unsigned8 (Integer without 8 bit code) • Unsigned16 (Integer without 16 bit code) • Unsigned24 (Integer without 24 bit code) • Unsigned32 (Integer without 32 bit code) • Unsigned40 (Integer without 40 bit code) • Unsigned48 (Integer without 48 bit code) • Unsigned56 (Integer without 56 bit code) • Unsigned64 (Integer without 64 bit code) • Float (32 bit single accuracy fixed decimal point) • Float64 (64 bit single accuracy fixed decimal point) • Visible String (Text string including ASCII text) • Octet string (Array of integer without 8 bit code) • Unicode string (Array of integer without 16 bit code) • Bit string (Array of single bit) • Time of day (48 bit value indicating time and date) • Time difference (48 bit value indicating time) • Domain (Application specific data block) • Reserved (Reservation)
<p>Access method</p>	<p>Displays access method of the objects. The following types are included.</p> <ul style="list-style-type: none"> • readonly (Read only) • writeonly (Write only) • readwrite (Read/Write) • constant (Constant)
<p>Mapping</p>	<p>Map the object selected in [Available objects from device profile] to [Configured objects] tree.</p>
<p>Remove forces</p>	<p>Remove the object mapped to [Configured objects] tree.</p>
<p>Configured objects</p>	<p>Mapped objects per slave displays in tree configuration. Map the object. TPDO PDO sent from slave to master. When data is input from the external I/O that's connected to the selected slave, map the object here. RPDO PDO sent from master to slave. When data is output from the external I/O that's connected to the selected slave, map the object here.</p>

Continued

Setting	Description
Settings	Configure detailed settings of the item selected in [Configured objects]. Setting contents vary from when: the slave is selected on the tree, TPDO or RPDO is selected, or the mapped object is selected.
When TPDO/ RPDO Is Selected	Set PDO enabled/disabled and set [Transmission Type], [Inhibit Time] and [Event Timer]. <div style="display: flex; justify-content: space-around;">   </div> <hr/> <div style="border: 1px solid black; padding: 2px; display: inline-block;">NOTE</div> <ul style="list-style-type: none"> When PDO is disabled, when you re-open the project, all the information set up in the PDO is initialized to its defaults.
Enabled	Select the check box to enable PDO.
Transmission Type	Display the transmission type for PDO in [Change Settings].
Inhibit Time	Display time when continuously transmitting PDO is inhibited only for TPDO. Set in [Change Settings], unit is 100 microseconds.
Event Timer	Displays transmission interval in which PDO is continuously sent. Set in [Change Settings]. Unit is ms.

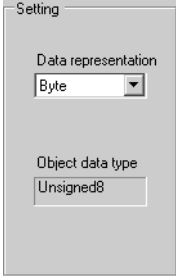
Continued

Setting		Description												
Settings	When TPDP/RPDO is selected	<p>Click to display the following screen. Different screens are displayed for RPDO and TPDO.</p> <p>When RPDO is selected</p> 												
		<p>Change Settings</p> <ul style="list-style-type: none"> • Acyclic synchronous (non cyclic) Continuously receive PDO according to SYNC signal. [Transmission Type] is fixed to 0. • Cyclic synchronous (cyclic) Receive PDO according to SYNC signal when it's matched to [SYNC Cycle Rate] that is set. [SYNC Cycle Rate] is set between 1 and 240. The [SYNC Cycle Rate] number is the same as the [Transmission Type] number. • Asynchronous event (manufacturer event) Receive PDO including slave equipment manufacturer specific objects according to [Event Timer] that has been set. [Transmission Type] is fixed to 254. • Asynchronous event (standard event) Receive PDO including objects except manufacturer specific objects according to [Event Timer] that has been set. [Transmission Type] is fixed to 255. Set the [Event Timer] from 0 to 32767, in millisecond units. <p>The information of PDO that has been set is reflected for the following objects:</p> <table border="1" data-bbox="430 1460 1186 1667"> <thead> <tr> <th>Object number</th> <th>Sub index</th> <th>ItemName</th> </tr> </thead> <tbody> <tr> <td rowspan="5">0x1400 + Node ID</td> <td>0</td> <td>Number of RPDO</td> </tr> <tr> <td>1</td> <td>COB ID</td> </tr> <tr> <td>2</td> <td>Transmission Type</td> </tr> <tr> <td>3</td> <td>Inhibit Time</td> </tr> <tr> <td>5</td> <td>Event Timer</td> </tr> </tbody> </table>	Object number	Sub index	ItemName	0x1400 + Node ID	0	Number of RPDO	1	COB ID	2	Transmission Type	3	Inhibit Time
Object number	Sub index	ItemName												
0x1400 + Node ID	0	Number of RPDO												
	1	COB ID												
	2	Transmission Type												
	3	Inhibit Time												
	5	Event Timer												

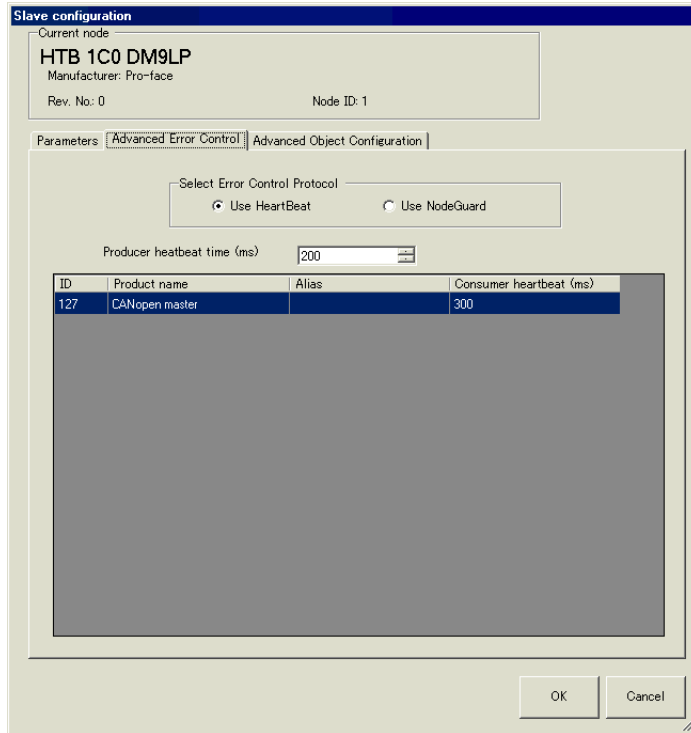
Continued

Setting		Description														
Settings	When TPDP/RPDO is selected	<p>TPDO</p> <div style="text-align: center;"> </div> <ul style="list-style-type: none"> • Acyclic synchronous (non cyclic) Continuously send PDO constantly according to SYNC signal. [Transmission Type] is fixed to 0. • Cyclic synchronous (cyclic) Send PDO according to SYNC signal when it's matched to [SYNC Cycle Rate] that is set. [SYNC Cycle Rate] is set between 1 and 240. The [SYNC Cycle Rate] number is the same as the [Transmission Type] number. • Asynchronous event (manufacturer event) Send PDO including slave equipment manufacturer specific objects according to [Event Timer] that has been set. [Transmission Type] is fixed to 254. [Event Timer]: Set between 0 and 32767. The unit is in milliseconds (ms). [Inhibit Time]: Set between 0 and 32767. The unit is 100 microseconds. This must be less than [Event Timer]. • Asynchronous event (standard event) Send PDO including objects except manufacturer specific objects according to [Event Timer] that has been set. [Transmission Type] is fixed to 255. <p>The information of PDO that has been set is reflected for the following objects:</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 30%;">Object number</th> <th style="width: 20%;">Sub index</th> <th style="width: 50%;">ItemName</th> </tr> </thead> <tbody> <tr> <td rowspan="5" style="vertical-align: middle;">0x1800 + Node ID</td> <td style="text-align: center;">0</td> <td>Number of TPDO</td> </tr> <tr> <td style="text-align: center;">1</td> <td>COB ID</td> </tr> <tr> <td style="text-align: center;">2</td> <td>Transmission Type</td> </tr> <tr> <td style="text-align: center;">3</td> <td>Inhibit Time</td> </tr> <tr> <td style="text-align: center;">5</td> <td>Event Timer</td> </tr> </tbody> </table>	Object number	Sub index	ItemName	0x1800 + Node ID	0	Number of TPDO	1	COB ID	2	Transmission Type	3	Inhibit Time	5	Event Timer
		Object number	Sub index	ItemName												
0x1800 + Node ID	0	Number of TPDO														
	1	COB ID														
	2	Transmission Type														
	3	Inhibit Time														
	5	Event Timer														
Change Settings																

Continued

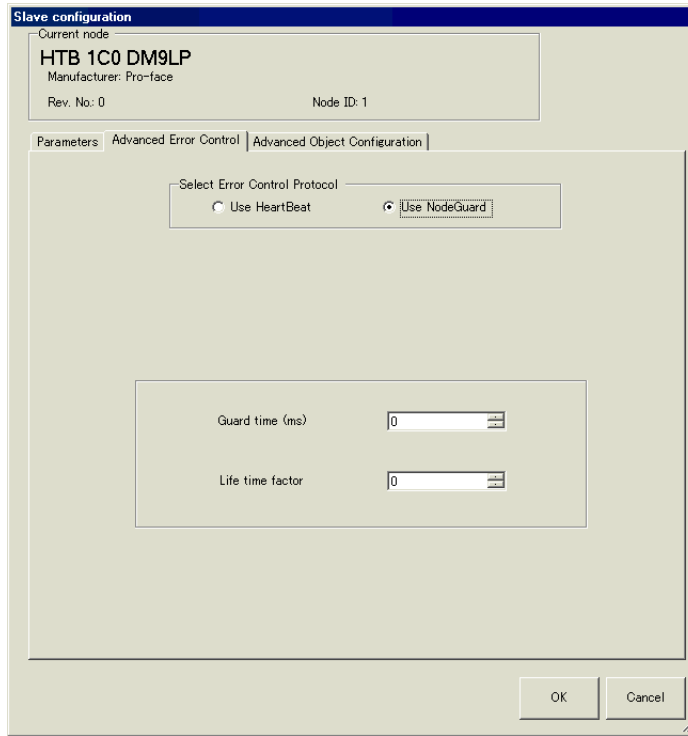
Setting		Description
Settings	When the object is selected	<p>Configure how to work with the mapped object. Display on I/O Screen varies depending on the type set here.</p> <p>☞ "30.7.3 Allocating I/O (Common) ■ I/O Screen Display" (page 30-152)</p> <ul style="list-style-type: none"> • Data representation Select from "Bit", "Byte", "Word" and "Dword" (Double Word). • Object data type Data type of the selected object displays. <div style="text-align: center;">  </div>

- Advanced Error Control (when heartbeat is selected)



Setting	Description
Producer Heartbeat Time	Set the transmission cycle of heartbeat. The unit is milliseconds (ms) and setting range is 0, or 50 to 21844, stored in object 1017h on the slave.
ID	Node ID of CANopen master displays.
Product Name	Product name of CANopen masters displays.
Alias	Comments of CANopen masters displays.
Consumer Heartbeat	Consumer heartbeat of CANopen master displays. Stored in the object 1016h on master side.

- Advanced Error Control (when no guard is selected)



Setting	Description
Guard Time	When NMT master polls the slaves, sets up the frequency for when slaves receive polling requests. This field's units are in milliseconds, and the set up range is 0, or 50 to 32767, stored in object 100Ch. Heartbeat is used when set to zero.
Life time factor	Set the time to monitor errors when NMT master polls the slaves. An error occurs if it is not polled when the time which is the value set in [Guard Time] multiplied by the value set here has passed. Setting range is 0, or 2 to 255. Stored in the object 100Dh.

- Advanced Object Configuration

You can change the values of read-write and write-only (not grayed out) objects can be changed. The values set here are written in the objects on slave side on startup.

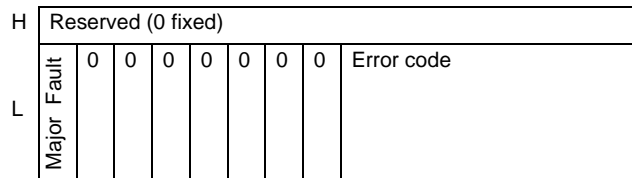
Detail settings are the same as those of [Advanced Object Configuration] tab in [Master Settings] dialog box.

☞ 30.7.7 [I/O Driver] Settings Guide ◆ Master Settings Dialog Box •Advanced Object Configuration 30-194

30.7.8 Error information

The system variable #L_IOStatus stores error information in the bottom 8 bits.

#L_IOStatus



Major Fault

"1" is set when it detects a malfunction that needs to stop the logic.

■ Error code

	Error code	Error message	Summary		Solution
Project date related error	000	No error	-	Initializing error	-
	001	Illegal instruction(s)	I/O Driver Instruction Error		Check the set value of operand for I/O Driver Instructions and re-transfer the project.
	002	Too many instructions	Number of used I/O Driver Instructions is greater than 16		Check the number of used I/O Driver Instructions and re-transfer the project.
	003	Invalid driver ID	The driver/unit results in an error while registering, and they have not been registered.		The project file may not have been sent correctly. Re-transfer the project file.
	004	Repeat driver setting	The driver is registered twice.		
	005	Invalid level	The driver is not correct.		
	006	Invalid address	The driver information is in correct. The controller information is not correct.		
		007	Driver not registered		Driver is not registered

Continued

	Error code	Error message	Summary		Solution	
Project data related error	008	Invalid PDO configuration	PDO Settings Error	Initialization error	Confirm PDO Settings and re-transfer the project.	
	009	Invalid terminal type	Invalid terminal type		The project file may not have been sent correctly. Re-transfer the project file.	
	010	Concise DCF not found	There is no concise DCF			
	011	Invalid Concise DCF	Invalid concise DCF			
	012	Invalid firmware	Inappropriate firmware			
	013	Configuration could not download to CANopen Module	Failure in downloading set value			Reset AGP. If the problem is unresolved, please contact your support center.
	014	Invalid value	Error code is 0 when error flag is set			Reset AGP. If the problem is unresolved, please contact your support center.
H/W related error	050	I/O board ID mismatch	The connected I/O board is not correct.	Hardware error	Display type may be different. Confirm the display type and transfer the project file again.	
	051	CANopen unit initialization error	The I/O board initialization fails.		The project file might not have been sent properly. Transfer the project file again. If the problem is still not solved, there may be a problem with the hardware. Contact your support center.	

Continued

	Error code	Error message	Summary		Solution
Application related error	100	Fatal error: Bus off	Bus failure (such as noise failure), hardware failure, abnormal baud rate settings, etc.	Fatal master error	Confirm the connection state of the communication cable and check the baud rate settings of all the nodes to re-transfer the project. If the problem is still not solved, there may be a problem with the hardware. Contact your support center.
	101	Fatal error: high priority receive queue overrun	Excessive amount of received data such as PDO, NMT, and SYNC.		Review the network settings by decreasing PDO.
	102	Fatal error: high priority transmit queue overrun	Excessive amount of sending data such as PDO, NMT and SYNC.		Review the network settings by increasing node guard intervals.
	103	Fatal error: low priority receive queue overrun	Excessive amount of receiving data such as heartbeat, node guard, and SDO.		
	104	Fatal error: low priority transmit queue overrun	Excessive amount of transmitting data such as heartbeat, node guard, and SDO.		Review the network settings by decreasing PDO and SDO.
	105	Fatal error: CAN controller overrun	Data loss due to excessive amount of data such as PDO and SDO.		
	106	Fatal error: duplicate node ID detected	More than two identical node IDs are used.		Confirm whether node ID is duplicated and re-transfer the project.
	107	Fatal error: module has unsupported features	Unsupported node exists		Review the node configuration.

Continued

	Error code	Error message	Summary		Solution
Application related error	108	Fatal error: invalid Concise DCF	Network configuration is wrong	Fatal master error	Please download concise DCF again. If the problem is unresolved, check all the EDS files and review the network settings.
	109	Fatal error: mandatory module configuration error	Concise DCF file of the mandatory node is invalid		Check the EDS file and confirm whether it matches the hardware configuration.
	110	Master fatal error	Fatal error has occurred in CANopen master.		Reset the display.
	111	Invalid value in instruction	An I/O driver instruction has run with an operand value that is out of range.	Runtime error	Please run each I/O driver with values inside specified ranges.
	112	Mandatory module is missing	There is a problem with the defined mandatory node.		Please check if the mandatory node is connected properly and if it is powered up.
	120	SDO communication error	Node object information specified in SDO communication is invalid, or communication partner is unavailable for communication.		Confirm whether the specified information of the SDO communication is correct. If correct, confirm the communication state of the network or reset the network.
	121	CANopen module timeout (Input)	I/O firmware is not updating input data.		I/O firmware is not operating properly. Reset the display.
	122	CANopen module timeout (Output)	I/O firmware does not update output data.		
	123	CANopen module interface is busy	I/O firmware is not running SDO communication.	I/O firmware is not running properly. Please reset the display unit.	

Continued

	Error code	Error message	Summary		Solution
Application related error	130	Incorrect PDO received	Invalid size for the received PDO	Nonfatal error	Confirm the EDS file.
	131	SDO queue overrun	Excessive SDO communication data size		Make SDO communication data size smaller.
	132	Master alone	Slave does not exist		Confirm whether slave is correctly connected and that the power is on.
	150	Identity error of an optional slave	The set node is different from the connected node.	Slave error	Check the EDS file and network settings to ensure that the node and EDS file are correct. You can find the node ID of the slave using an offline function.
	151	Unexpected state for one or more optional slaves	Incorrect configuration of the optional node		Confirm slave configuration in the network settings and re-transfer the project.
	152	Unexpected state for one or more mandatory slaves	Mismatched state of the Mandatory node and that of the network		Project may not have been transferred correctly to the display unit. Transfer the project again.
	153	Faulty slave	Abnormal Slave		Check whether correct EDS file is used. Using an offline function, you can find node ID of slave.
	154	Inconsistent Concise DCF for one or more slaves	CDCF is mismatched with the object dictionary of slave.		Check the EDS file and network settings to ensure that the node and EDS file are correct. You can find the node ID of the slave using an offline function.
	155	Concise DCF mismatch for one or more slaves	Slave configuration and the object directory are mismatched.		
	156	Identity error for one or more slaves	One or multiple set slaves do not match the connected slave.		

30.7.9 Restrictions

- If a project with a different I/O Driver version is transferred, it takes time to start the GP.
- If there are many PDO data, the number of processes executing the logic increases, so scanning may take longer than the set scan time.
- If Stop All Nodes is specified in Mode when a critical node error occurs, all the nodes must be restarted.
- For HTB and EX module standard outputs, do not perform write operations from multiple objects.
- When the CANopen communication cable is disconnected, upon restoration the CANopen network is reset. (The slave HTB unit is also reset.)
- The number of PDO the HTB unit can use is 16 total. As all the objects cannot be mapped, please map only required objects to PDO.
- The number of I/O bit points you can set up is 512 (input bits: 256 points, output bits: 256 points). The number of I/O integer points you can set up is 128 (input integer: 64 points, output integer: 64 points).
- Please do not set values outside the valid range. Even if you set up a value outside the range, an error may not display.
- If you open the network setup dialog box, despite not making any changes, a save project message may display when you close the project.
- After mapping I/O, deleting objects, disabling PDO, or deleting the slave will not cancel the I/O mapping.
- When using the CANopen driver, the undo operation is not available in the associated I/O screen or I/O driver settings.

30.7.10 List of HTB Objects

HTB objects are described in the following three categories.

- ☞ " ■ Communication Setting Objects (1000h to 1FFFh)" (page 30-211)
- ☞ " ■ Manufacturer Objects (2000h to 5FFFh)" (page 30-230)
- ☞ " ■ Standard Device Object (6000h to 9FFFh)" (page 30-241)

■ Communication Setting Objects (1000h to 1FFFh)

Object (Hex)	Parameter	Summary	Reference page
1000	Device Type	Device type	☞ 30-212
1001	Error Register	Error register	☞ 30-213
1002	Manufacturer Status Register	Status register	☞ 30-214
1003	Pre-defined Error Field (PEF)	PEF (Pre-defined Error Field)	☞ 30-215
1005	COB-ID SYNC message	SYNC COB-ID	☞ 30-216
1006	Communication Cycle Period	SYNC sending period	☞ 30-217
1008	Manufacturer Device Name	Manufacturer device name	☞ 30-217
100A	Manufacturer Software Version (MSV)	MSV (Manufacturer Software Version)	☞ 30-218
100C	Guard Time	Guard Time	☞ 30-218
100D	Lifetime Factor	Life time factor	☞ 30-218
1010	Store parameters	Storing parameters	☞ 30-219
1011	Restore Default parameters	Restoring default parameters	☞ 30-220
1014	COB-ID Emergency (EMCY) message	EMCY COB-ID	☞ 30-221
1016	Consumer heartbeat time	Consumer heartbeat time	☞ 30-221
1017	Producer heartbeat time	Producer Heartbeat Time	☞ 30-222
1018	Identity Object	Identity object	☞ 30-222
1027	Module list	Module list	☞ 30-223
1200	Server SDO Parameter	Server SDO parameter	☞ 30-223
1400 to 1407	Receive PDO communication Parameter	RPDO communication parameter	☞ 30-224
1600 to 1607	Receive PDO Mapping Parameter	RPDO Mapping Parameter	☞ 30-225
1800 to 1807	Transmit PDO Communication Parameter	TPDO communication parameter	☞ 30-226
1A00 to 1A07	Transmit PDO Mapping Parameter	TPDO Mapping Parameter	☞ 30-229

◆ **1000h: Device Type**

The last byte indicates the profile number (401 (191h) for CANopen specification I/O).

The last byte is called "additional information", which indicates feature details of the device.

31	16 15	4 3 2 1 0
Additional information	Unused	d c b a

- a: The device has standard input
- b: The device has standard output
- c: The device has analog input
- d: The device has analog output

Properties of the object

Object properties:

Sub index	Description	Data Type	Default	Access	PDO mapping	Backup
0	-	UNSIGNED32	-	ro (read only)	X	X

◆ Object 1001h: Error Register

This object is used by the device to display an internal failure. The corresponding bit becomes active when failure is detected.

The following failures can display:

Bit	Description	Remarks
0	General error	Set to 1 when failure including invalid parameter is detected in HTB and its EX module.
1	Current	From analog input of EX module (when set for current)
2	Voltage	From analog input of EX module (when set for voltage)
3	Temp.	-
4	Communication error	CANopen error indicator. Can be supported by stack.
5	Reserved	-
6	Reserved	-
7	Manufacturer specific	Set to 1 when internal bus failure is detected between HTB and EX module.

These bits indicate boolean "OR" in case of failure in HTB and EX module.

Properties of the object

Object properties:

Sub index	Description	Data Type	Default	Access	PDO mapping	Backup
0	-	UNSIGNED8	-	ro (read only)	0	X

◆ **Object 1002h: Status Register**

This object is stored in EMCY message.

Error code stored in the last word.

Additional information stored in the top word.

31	16 15	0
Additional information	Error code	

Properties

Object properties:

Sub index	Description	Data Type	Default	Access	PDO mapping	Backup
0	-	UNSIGNED32	-	ro (read only)	0	X

When each of the following bits is 0, it indicates normal and when 1, it indicates failure.

Register	Feature	Description
Bit 0 to 15	State of HTB + EX module	Bit 0 to 8: Unused Bit 9: Communication or external failure Bit 10 to 12: Unused Bit 13: Invalid configuration (There is no EX module or the configuration is invalid) Bit 14, 15: Unused
Bit 16 to 31	State of EX module	Bit 16: HTB state bit Bit 17: 1st EX module state bit Bit 18: 2nd EX module status bit Bit 19: 3rd EX module status bit Bit 21: 4th EX module status bit Bit 22: 5th EX module status bit Bit 23: 6th EX module status bit Bit 24: 7th EX module status bit Bit 25 to 31: Unused

NOTE

- EX is counted 1st, 2nd, etc. from the side closest to HTB.

◆ Object 1003h: PEF (Pre-defined Error Field)

This object is used to latch the latest failure and its properties.

- Error code is latched by the bottom two bytes. For details, refer to the error code list.
- "Additional information" is latched by the top two bytes. The top bytes of additional information (1002h, bits 16 to 23) is set up with the same information as the bottom bytes (1001h).
- The latch error number is stored in sub index 0.

Properties of the object

Object properties:

Sub index	Description	Data Type	Default	Access	PDO mapping	Backup
0	Sub index count = Total number of latched errors	UNSIGNED8	0	rw (read-write)	X	X
1	Latest error	UNSIGNED32	-	ro (read only)	X	X
2	2nd last error	UNSIGNED32	-	ro (read only)	X	X
...						
10						

When a new error occurs, the displayed code moves to the sub index at the upper level. That is, error of sub index 1 goes to sub index 2, and error of sub index 2 goes to sub index 3. It is processed similarly. A maximum of 254 pieces of error information are stored.

History of error codes can be deleted by writing value 0 to sub index 0 of the object 1003h.

The error codes are not deleted from PEF even if the error is fixed.

All failures are notified by sending EMCY message ("emergency" message). When the error conditions are removed, a no-error EMCY message is sent (error code 0x0000).

Error code list

Error code (hexadecimal display)	Diagnostic message	Cause
0000	ERROR_RESET_OR_NO_ERROR	Error has disappeared
1000	GENERIC_ERROR	Internal communication error
6101	SOFTWARE_RX_QUEUE_OVERRUN	Overflow of receiving memory
6102	SOFTWARE_TX_QUEUE_OVERRUN	Overflow of sending memory
8100	COMMUNICATION	Error in sending and receiving synchronous counter. EMCY is sent when the counter is more than 96.
8120	CAN_IN_ERROR_PASSIVE_MODE	CAN controller interrupt
8130	LIFE_GUARD_ERROR	Node-Guarding error
8140	BUS_OFF	Overflow of sending buffer counter

◆ **Object 1005h: SYNC COB-ID**

This object stores the synchronous message ID.

Properties of the object

Object properties:

Sub index	Description	Data Type	Default	Access	PDO mapping	Backup
0	-	UNSIGNED32	80h	rw (read-write)	X	O

◆ **Object 1006h: Global SYNC period**

This object indicates time interval between two SYNC signals. This interval is at least 10ms and the minimum increment needs to be 1ms. It must be entered with double word.

Field is set to 0 when this object is not used.

When the value 10,000 to 10,000,000 is entered, HTB needs to receive SYNC signals within this time interval. If it is not received, it is shifted to pre-operational state. Maximum tolerance is 1% of the set value. Monitoring elapsed time starts when the first SYNC signal is received.

Properties of the object

Object properties:

Sub index	Description	Data Type	Default	Access	PDO mapping	Backup
0	-	UNSIGNED32	0	rw (read-write)	0	0

Coding switching cycle period

Threshold is shown in the following table.

Value type	Decimal value	Hexadecimal value	Synchronization interval (ms)
Standard value	0	0000 0000	-
Min	10000	0000 2710	10
-	25000	0000 61A8	25
-	250000	0003 D090	250
-	1000000	000F 4240	1000
-	5000000	004C 4B40	5000
Max	10000000	0098 9680	10000

◆ **Object 1008h: Manufacturer Device Name**

Properties of the object

Object properties:

Sub index	Description	Data Type	Default	Access	PDO mapping	Backup
0	-	STRING	HTB1C0DM9LP	ro (read only)	X	X

◆ **Object 100Ah: MSV (Manufacturer Software Version)**

This object stores details of HTB firmware version in 'Vxx.yy' format.

Properties of the object

Object properties:

Sub index	Description	Data Type	Default	Access	PDO mapping	Backup
0	-	STRING	-	ro (read only)	X	X

◆ **Object 100Ch: Guard Time**

Properties of the object

Object properties:

Sub index	Description	Data Type	Default	Access	PDO mapping	Backup
0	-	UNSIGNED16	0	rw (read-write)	X	0

Guard time is stored in millisecond (ms).

Guard time x Lifetime factor (retry count) = Monitoring time

◆ **Object 100Dh: Lifetime factor**

Properties of the object

Object properties:

Sub index	Description	Data Type	Default	Access	PDO mapping	Backup
0	-	UNSIGNED8	0	rw (read-write)	X	0

◆ Object 1010h: Storing parameters

This object is used to store parameters of HTB and EX module in backup memory.

Properties of the object

Object properties:

Sub index	Description	Data Type	Default	Access	PDO mapping	Backup
0	Sub index number	UNSIGNED8	4	ro (read only)	X	X
1	Store all the parameters	UNSIGNED32	-	rw (read-write)	X	X
2	Store communication parameter (1000h to 1FFFh)	UNSIGNED32	-	rw (read-write)	X	X
3	Unused	-	-	-	-	-
4	Application parameter (2000h to 9FFFh)	UNSIGNED32	-	rw (read-write)	X	X

Set Editor Language

To store parameters, it is necessary to write "save" text (6576 6173h) to the corresponding index.

	Top byte		Bottom byte	
ISO 8859 (ASCII) signature	e	v	a	s
Hexadecimal value	65h	76h	61h	73h

Information on storage feature is read from sub index. Obtained result 0000 0001h indicates parameters are restored by the module only when applicable command is received.

◆ Object 1011h: Restoring Default Parameters

This object is used to restore parameters of HTB and EX module. Restoring parameters is considered only after the power is turned on.

Properties of the object

Object properties:

Sub index	Description	Data Type	Default	Access	PDO mapping	Backup
0	Sub index number	UNSIGNED8	4	ro (read only)	X	X
1	Restore all the default parameters	UNSIGNED32	-	rw (read-write)	X	X
2	Restore default communication parameters (1000h to 1FFFh)	UNSIGNED32	-	rw (read-write)	X	X
3	Unused	-	-	-	-	-
4	Restore default application parameters (2000h to 9FFFh)	UNSIGNED32	-	rw (read-write)	X	X

Set Editor Language

To restore parameters, it is necessary to write "load" text (6461 6F6Ch) to the corresponding index.

	Top byte		Bottom byte	
ISO 8859 (ASCII) signature	d	a	o	L
Hexadecimal value	64h	61h	6Fh	6Ch

Information on whether default parameters of the module can be restored is read from sub index. Obtained result 0000 0001h indicates that parameters are restored only when an applicable command is received by the module.

◆ Object 1014h: COB-ID EMCY (Emergency) Message

This object stores the EMCY emergency message ID.

Properties of the object

Object properties:

Sub index	Description	Data Type	Default	Access	PDO mapping	Backup
0	-	UNSIGNED32	80h + node ID	rw (read-write)	X	O

◆ Object 1016h: Consumer Heartbeat Time

This object is used when time interval set in ms for which HTB must receive heartbeat message from the monitored CANopen master. HTB is designed to monitor only one CANopen master.

The value of this objects must be greater than that of the object 1017h.

Time needs to be a multiple of 1ms.

Properties of the object

Object properties:

Sub index	Description	Data Type	Default	Access	PDO mapping	Backup
0	Sub index number	UNSIGNED8	1	ro (read only)	X	O
1	Consumer heartbeat time	UNSIGNED32	0	rw (read-write)	X	O

Contents of variables

Contents of sub index 1 are as follows.

Bit	31 (MSB) to 24	23 to 16	15 to 0 (LSB)
Value	0h (Reserved)	Address of monitored CANopen master	Monitoring time (ms)

CANopen master is not monitored if the value of the object is 0.

◆ Object 1017h: Producer Heartbeat Time

This object is used when the time interval is set in ms for the module to generate heartbeat message.

Default monitoring method is node guard. Heartbeat is used when any value except 0 is written in this object.

Properties of the object

Object properties:

Sub index	Description	Data Type	Default	Access	PDO mapping	Backup
0	-	UNSIGNED16	0	rw (read-write)	X	O

◆ Object 1018h: Identity Object

This object stores the device information. It includes manufacturer's CiA identifier (vendor ID), product code, and revision number of HTB.

Revision information is divided into the following two parts.

- Major revised part (top word) indicates modification of CANopen feature.
- Minor revised part (bottom word) indicates modification of HTB feature only.

Properties of the object

Object properties:

Sub index	Description	Data Type	Default	Access	PDO mapping	Backup
0	Input number	UNSIGNED8	3h	ro (read only)	X	X
1	Vendor ID	UNSIGNED32	12Bh	ro (read only)	X	X
2	Product code	UNSIGNED32	FEFBh	ro (read only)	X	X
3	Revision number	UNSIGNED32	-	ro (read only)	X	X

◆ **Object 1027h: Module List**

This object stores the list of EX modules connected to the bus.

Properties of the object

Object properties:

Sub index	Description	Data Type	Default	Access	PDO mapping	Backup
0	Number of sub index - Number of EX modules connected	UNSIGNED16	7	ro (read only)	X	X
1	Product code of the first module	UNSIGNED16	-	ro (read only)	X	X
...						
7	Product code of the last module	UNSIGNED16	-	ro (read only)	X	X

◆ **Object 1200h: Server SDO Parameter**

Description

This object stores the message ID for SDO communication.

Properties of the object

Object properties:

Sub index	Description	Data Type	Default	Access	PDO mapping	Backup
0	Sub index number	UNSIGNED8	0012h	ro (read only)	X	X
1	COB-ID of receive SDO (client -> server)	UNSIGNED32	600h + node ID	ro (read only)	X	X
2	COB-ID of transmit SDO (server -> client)	UNSIGNED32	580h + node ID	ro (read only)	X	X

◆ **Object 1400h to 1407h: RPDO communication parameter**

This object stores information of received PDO.

Properties of the object

Object properties:

Sub index	Description	Data Type	Default	Access	PDO mapping	Backup
0	Sub index number	UNSIGNED8	2	ro (read only)	X	O
1	COB-ID	UNSIGNED32	200h + Node ID (object 1400h) 80000200h + (object's bottom digit * 100) + Node ID (object 1401~1403h) 80000000h + Node ID (1404~1407h)	rw (read-write)	X	O
2	Transmission mode	UNSIGNED8 (1 to 5 for RPDO) UNSIGNED32 (6 to 8 for RPDO)	255	rw (read-write)	X	O

Transmission mode

PDO transmission mode can be set as in the following table.

Transfer code	Transmission mode					Remarks
	Cyclic	Noncyclic	Synchronous	Asynchronous	RTR only	
0		×	×			Send PDO in the first Sync message after the event
1 to 240	×		×			Send PDO per x pieces of Sync message
241 to 251	Reserved					-
252 to 253	Reserved					-
254				×		Send PDO when an event occurs
255				×		Send PDO when an event occurs

NOTE

- In Modes 254 and 255, the event to trigger transmission is defined by the message producer.

◆ **Object 1600h to 1607h: RPDO Mapping Parameter**

This object describes the objects transferred by PDO.

Properties

Object properties:

Sub index	Description	Data Type	Default	Access	PDO mapping	Backup
0	Sub index number	UNSIGNED8	-	rw (read-write)	X	O
1	First object in PDO	UNSIGNED32	-	rw (read-write)	X	O
2	Second object in PDO	UNSIGNED32	-	rw (read-write)	X	O
...						
8	Last object in PDO	UNSIGNED32	-	rw (read-write)	X	O

Configuration of data fields

Each data object that is transferred is indicated in the following format.

Bit	31 (MSB) to 16	15 to 8	7 to 0 (LSB)
Data	Index number of the transferred object	Sub index number of the transferred object	Length of the transferred object
Example	6200h	01h	08h

NOTE

- Maximum total length of the data transferred by PDO is 8 bytes.

◆ **Object 1800h to 1807h: TPDO Communication Parameter**

This object stores information of sent PDO.

Properties

Object properties:

Sub index	Description	Data Type	Default	Access	PDO mapping	Backup
0	Sub index number	UNSIGNED8	5	ro (read only)	X	O
1	COB-ID	UNSIGNED32	180h + Node ID (object 1800h) 80000180h + (object's bottom digit * 100) + Node ID (object 1801~1803h) 80000000h (1804~1807h)	rw (read-write)	X	O
2	Transmission mode	UNSIGNED8	255	rw (read-write)	X	O
3	Inhibit Time	UNSIGNED32	0	rw (read-write)	X	O
4	Unavailable					
5	Event Timer	UNSIGNED32	0	rw (read-write)	X	O

Transmission mode

PDO transmission mode can be set as in the following table.

Transfer code	Transmission mode					Remarks
	Cyclic	Noncyclic	Synchronous	Asynchronous	RTR only	
0		X	X			Send PDO in the first Sync message after the event
1 to 240	X		X			Send PDO per x pieces of Sync message
241 to 251	Reserved					-

Continued

Transfer code	Transmission mode					Remarks
	Cyclic	Noncyclic	Synchronous	Asynchronous	RTR only	
252	Unused					Receive SYNC message and send PDO with remote request
253	Unused					Update the data and send PDO with remote request
254				X		Send PDO when an event occurs
255				X		Send PDO when an event occurs

NOTE

- For digital I/O and analog I/O, the event changes the value.

COB-ID configuration

Configuration of COB-ID that supports CAN2.0 is shown in the following table:

Bit number	Value	Description
31 (MSB)	0	PDO objects exist
	1	No PDO objects exist
30	-	Reserved
29	0	11-Bit ID (CAN 2.0A)
28 to 11	0	For bit 29 = 0
10 to 0 (LSB)	X	Bit of ID 10 to 0

Inhibit Time

For "Send PDO", you can enter Inhibited Time value in this 16 bit field. When the data loads, the PDO sender checks whether "Inhibited Time" has passed after the previous send. You can send a new PDO only when "Inhibited Time" passes. "Inhibited Time" helps avoid overload of the CAN bus in asynchronous sending (sending mode 255). "Inhibited Time" is in multiple of 100s of the object 1800,03 to 1807,03.

Examples of the value are shown in the following table:

Value	Event timer (ms)
0000h	0
64h	10
3E8h	100
1388h	500
2710h	1000
FFFFh	6553

Event Timer

"Event Timer" acts only in the asynchronous transmission mode (255). Temporary transmission is sent when the data changes before "Event Timer" is ended. When a value greater than 0 is written in this 16 bit field, TPDO is continuously sent after "Event Timer" ends. The value written in 1800.05 to 1805.05 corresponds to "Event Timer" in milliseconds (ms). Data is transferred without data changes.

Examples of the value are shown in the following table:

Value	Event timer (ms)
0000h	0
64h	10
3E8h	100
1388h	500
2710h	1000
FFFFh	6553

◆ Object 1A00h to 1A07h: TPDO Mapping Parameter

This object describes the objects transferred by PDO.

Properties

Object properties:

Sub index	Description	Data Type	Default	Access	PDO mapping	Backup
0	Sub index number	UNSIGNED8	-	rw (read-write)	X	O
1	First object in PDO	UNSIGNED32	-	rw (read-write)	X	O
2	Second object in PDO	UNSIGNED32	-	rw (read-write)	X	O
...						
8	Last object in PDO	UNSIGNED32	-	rw (read-write)	X	O

Configuration of data fields















Each data object that is transferred is indicated in the following format.

Bit	31 (MSB) to 16	15 to 8	7 to 0 (LSB)
Data	Index number of the transferred object	Sub index number of the transferred object	Length of the transferred object
Example	6200h	01h	08h

NOTE

- Maximum total length of the data transferred by PDO is 8 bytes.

■ Manufacturer Objects (2000h to 5FFFh)

Object (Hex)	Parameter	Summary	Reference page
2000	Local digital parameter	Standard input parameter	 30-231
2100	Analog input type	Analog input data type	 30-232
2101	Analog input range	Analog input data range	 30-233
2102	Analog input minimum	Analog input minimum	 30-234
2103	Analog input maximum	Analog input maximum	 30-234
2200	Analog output type	Analog output data type	 30-235
2201	Analog output range	Analog output data range	 30-235
2202	Analog output minimum	Analog output lower limit	 30-236
2203	Analog output maximum	Analog output upper limit	 30-236
3000	Module diagnostics	Module diagnosis	 30-237
3200	Parameter status	Parameters status	 30-238
3201	Configuration stack number	Configuration stack number	 30-238
3202	Restore saved parameters	Restoring last save parameters	 30-239
3300	Extension bus reset	Reset Extension Bus	 30-240

◆ Object 2000h: Standard Input Parameter

Used only for standard input of communication blocks.

Properties

Object properties:

Sub index	Description	Data Type	Default	Access	PDO mapping	Backup
0	Standard Input	UNSIGNED16	12	ro (read only)	X	O
1	Standard input parameter 0	UNSIGNED16	1	rw (read-write)	X	O
...	-	-	-	-	-	-
12	Standard input parameter 11	UNSIGNED16	1	rw (read-write)	X	O

Configurable sub index value

0: No input filter

1: Input filter 3ms

2: Input filter 12ms

◆ Object 2100h: Analog Input Data Type

This object defines the analog input data type of the EX module.

Properties

Object properties:

Sub index	Description	Data Type	Default	Access	PDO mapping	Backup
0	Sub index number (analog input number)	UNSIGNED8	n	ro (read only)	X	O
1	CH1 input type	UNSIGNED16	0	rw (read-write)	X	O
...						
n	Input type of the last channel	UNSIGNED16	0	rw (read-write)	X	O

Configurable sub index value

0: Unused

2: Current (4 to 20mA)

3: Voltage (0 to 10V)

5: Thermocouple (K type)

6: Thermocouple (J type)

7: Thermocouple (T type)

8: Pt100

NOTE

- If you try to set an incompatible value (example, enter PT100 in normal analog input), the "Abort Code" is generated.

◆ Object 2101h: Analog Input Data Range

This object defines the data range of each analog input in EX module.

Properties

Object properties:

Sub index	Description	Data Type	Default	Access	PDO mapping	Backup
0	Sub index number (analog input number)	UNSIGNED8	n	ro (read only)	X	O
1	Measurement unit of CH1	UNSIGNED16	0011h	rw (read-write)	X	O
...						
n	Measurement unit of the last channel	UNSIGNED16	0011h	rw (read-write)	X	O

Configurable sub index value

0: Fixed

1: User setting

2: Celsius

3: Fahrenheit

NOTE

- If you try to set an incompatible value (example, Celsius in normal analog input), the "Abort Code" is generated.

◆ Object 2102h: Analog Input Lower Limit

This object stores the minimum value of the user-defined measurement range (object 2101h sub index = 1).

Properties

Object properties:

Sub index	Description	Data Type	Default	Access	PDO mapping	Backup
0	Sub index number (analog input number)	UNSIGNED8	n	ro (read only)	X	O
1	Minimum measured value of CH1	INTEGER16	0	rw (read-write)	X	O
...						
n	Minimum measured value of the last channel	INTEGER16	0	rw (read-write)	X	O

◆ Object 2103h: Analog Input Upper Limit

This object stores the maximum of the user-defined measurement range (object 2101h sub index = 1) is stored here.

Properties

Object properties:

Sub index	Description	Data Type	Default	Access	PDO mapping	Backup
0	Sub index number (analog input number)	UNSIGNED8	n	ro (read only)	X	O
1	Maximum measured value of CH1	INTEGER16	7FFFh	rw (read-write)	X	O
...						
n	Maximum measured value of the last channel	INTEGER16	7FFFh	rw (read-write)	X	O

◆ Object 2200h: Analog Output Data Type

This object defines data type of the analog output in the EX module.

Properties

Object properties:

Sub index	Description	Data Type	Default	Access	PDO mapping	Backup
0	Sub index number (analog output number)	UNSIGNED8	n	ro (read only)	X	O
1	Output type of CH1	UNSIGNED16	0	rw (read-write)	X	O
...						
n	Output type of the last channel	UNSIGNED16	0	rw (read-write)	X	O

Configurable sub index value

0: Unused

2: Current (4 to 20mA)

3: Voltage (0 to 10V)

NOTE

- If you try to set an incompatible value (example, voltage (0 to 10V) in normal analog input), the "Abort Code" may be generated.

◆ Object 2201h: Analog Output Data Range

This object defines data range of each analog output in the EX module.

Properties

Object properties:

Sub index	Description	Data Type	Default	Access	PDO mapping	Backup
0	Sub index number (analog output number)	UNSIGNED8	n	ro (read only)	X	O
1	Measurement unit of CH1	UNSIGNED16	0011h	rw (read-write)	X	O
...						
n	Measurement unit of the last channel	UNSIGNED16	0011h	rw (read-write)	X	O

Configurable sub index value

0: Fixed

1: User setting

◆ **Object 2202h: Analog Output Lower Limit**

This object stores the minimum value of the user set writing range (object 2201h sub index = 1).

Properties

Object properties:

Sub index	Description	Data Type	Default	Access	PDO mapping	Backup
0	Sub index number (analog output number)	UNSIGNED8	n	ro (read only)	X	O
1	Minimum value of CH1	INTEGER16	0	rw (read-write)	X	O
...						
n	Minimum value of the last channel	INTEGER16	0	rw (read-write)	X	O

◆ **Object 2203h: Analog output Upper Limit**

This object stores the maximum value of the user set writing range (object 2201h sub index = 1).

Properties

Object properties:

Sub index	Description	Data Type	Default	Access	PDO mapping	Backup
0	Sub index number (analog output number)	UNSIGNED8	n	ro (read only)	X	O
1	Output type of CH1	INTEGER16	7FFFh	rw (read-write)	X	O
...						
n	Output type of the last channel	INTEGER16	7FFFh	rw (read-write)	X	O

◆ **Object 3000h: Module Diagnosis**

The object stores the HTB and EX module specific diagnosis.

Properties

Object properties:

Sub index	Description	Data Type	Default	Access	PDO mapping	Backup
0	Sub index number (module number)	UNSIGNED8	n	ro (read only)	X	X
1	HTB diagnosis	UNSIGNED16	0	ro (read only)	O	X
2	First EX module diagnosis	UNSIGNED16	0	ro (read only)	O	X
...						
n	Last EX module diagnosis	UNSIGNED16	0	ro (read only)	O	X

MappingHTB bit

- Bit 2: Hardware failure (external power failure, common in all the channels)
- Bit 3: Invalid module configuration
- Bit 8: Value error occurred in the last command
- Bit 9: Value consistency error occurred in the last command

DIO module bit mapping

- Bit 2: Hardware failure (external power failure, common in all the channels)
- Bit 3: Invalid module configuration
- Bit 8: Value error occurred in the last command
- Bit 9: Value consistency error occurred in the last command

Analog module bit mapping

- Bit 0: All channels are operating normally
- Bit 1: Module is initialized for all channels
- Bit 2: Hardware failure (external power failure, common in all the channels)
- Bit 3: Invalid module configuration
- Bit 4: Converting data of input CH1 (data is unavailable)
- Bit 5: Converting data of input CH2 (data is unavailable)
- Bit 6: Thermocouple input CH1 is not configured
- Bit 7: Thermocouple input CH2 is not configured
- Bit 8: Value error occurred in the last command
- Bit 9: Value consistency error occurred in the last command
- Bit 10: Analog input data CH1 has exceeded the range
- Bit 11: Analog input dataCH2 has exceeded the range
- Bit 12: Incorrect wiring (analog input data CH1 is less than the rage)
- Bit 13: Incorrect wiring (analog input data CH2 is less than the rage)
- Bit 14: Unused
- Bit 15: Output channels are unavailable

◆ **Object 3200h: Parameters Status**

This object stores the parameters status.

Stored object values:

0: Using default parameter

1: Using stored parameter

2: Using current parameter (unsaved)

Properties

Object properties:

Sub index	Description	Data Type	Default	Access	PDO mapping	Backup
0	-	UNSIGNED16	0	ro (read only)	X	X

◆ **Object 3201h: Configuration Stack Number**

This object stores the number of backups executed after the previous restoration of default parameter.

Properties

Object properties:

Sub index	Description	Data Type	Default	Access	PDO mapping	Backup
0	-	UNSIGNED16	-	ro (read only)	X	X

◆ **Object 3202h: Restore Last Saved Parameters**

Restore last saved parameters.



Unexpected action may occur in the equipment

When parameters are restored in "action" state, unexpected actions may occur in the equipment resulting in damage and physical injury. If it is necessary to operate, set HTB in the "pre-action" state, or ensure physical and equipment safety before attempting other operations.

Failure to follow these instructions may result in serious injury or death.

To restore parameters, you need to write the text string "load" to the corresponding index. You may need to reverse the string to "daol" or 64616F6Ch.

NOTE

- When the restoration proceeds (object 3202h is state 0), this command is ignored.
- Parameters are restored immediately.

Properties

Object properties:

Sub index	Description	Data Type	Default	Access	PDO mapping	Backup
0	Restore specific application parameters	UNSIGNED32	1	rw (read-write)	X	X

Restored objects are as follows.

1005H, 1006H, 100CH, 100DH, 1014H, 1016H, 1017H, 1400H to 1407H, 1600H to 1607H, 1800H to 1807H, 1A00H to 1A07H, 2000H, 2100H, 2101H, 2102H, 2103H, 2104H, 2105H, 2106H, 2200H, 2201H, 2202H, 2203H, 2303H, 2304H, 2305H, 2306H, 2307H, 2354H, 2355H, 2403H, 2404H, 2405H, 2406H, 2407H, 2408H, 2409H, 240AH, 2454H, 2455H, 2457H, 2458H, 2502H, 2503H, 2504H, 2505H, 2506H, 2507H, 2508H, 2557H, 6102H, 6103H, 6302H, 6306H, 6307H, 6308H, 6421H, 6424H, 6425H, 6426H, 6443H, 6444H

◆ **Object 3300h: Reset Extension Bus**

This object is used to update settings of EX module parameters.

- When this value is set to 1, the internal bus stops.
- When this value is set to 0, the internal bus is enabled if there is consistency with EX module parameters.

 **WARNING**

Unexpected action may occur in the equipment

When the internal bus is stopped, all of the output for EX module is set to 0 and HTB output shifts to fallback state.

















When extension bus stops in "action" state, unexpected actions may occur in the equipment resulting in damage and physical injury. If it is necessary to operate, set HTB in the "pre-action" state, or ensure physical and equipment safety before attempting other operations. Failure to follow these instructions may result in serious injury or death.

Properties

Object properties:

Sub index	Description	Data Type	Default	Access	PDO mapping	Backup
0	-	UNSIGNED16	0	rw (read-write)	X	X

■ Standard Device Object (6000h to 9FFFh)

Object (Hex)	Parameter	Summary	Reference page
6000	Read Input 8 bit	Input (8 bit)	 30-242
6100	Read Input 16 bit	Input (16 bit)	 30-242
6102	Polarity inputs 16 bits	Polarity input	 30-243
6103	Filter Input 16-bit	Input mask	 30-244
6200	Write outputs 8 bits	Output (8 bit)	 30-244
6300	Write outputs 16 bits	Output (16 bit)	 30-245
6302	Polarity Outputs 16 bit	Polarity output	 30-246
6306	Fallback mode Outputs 16 bit	Fallback mode output	 30-247
6307	Fallback Value Output 16 bit	Fallback value output	 30-247
6308	Filter Mask output 16 bit	Output mask	 30-248
6401	Analog Read Input 16 bit	Analog input	 30-249
6411	Analog Write Output 16 bits	Analog output	 30-249
6422	Analog Input interrupt Source	Analog input PDO transmission channel number	 30-250
6423	Analog Input global Interrupt enable	Analog input enabled/disabled	 30-250
6443	Analog output fallback mode	Analog output fallback mode	 30-251
6444	Analog output fallback value	Analog output fallback value	 30-251

◆ **Object 6000h: Input (8 Bit)**

This object reports 8 bit digital input value.

Sub index management:

- Each digital EX module uses an even-numbered sub index (allocated with 16 bit word).
- EX module and its sub index are given a number in the order closer to HTB.

Properties

Object properties:

Sub index	Description	Data Type	Default	Access	PDO mapping	Backup
0	Sub index number (number of 8 bit input)	UNSIGNED8	n	ro (read only)	X	X
1	HTB input 0 to 7	UNSIGNED8	-	ro (read only)	O	X
2	HTB input 8 to 15	UNSIGNED8	-	ro (read only)	O	X
3	First EX module input 0 to 7	UNSIGNED8	-	ro (read only)	O	X
...						
n	Last EX module input 8 to 15	UNSIGNED8	-	ro (read only)	O	X

◆ **Object 6100h: Input (16 Bit)**

This object reports 16 bit digital input value.

Properties

Object properties:

Sub index	Description	Data Type	Default	Access	PDO mapping	Backup
0	Sub index number (number of 16 bit input)	UNSIGNED8	n	ro (read only)	X	X
1	Read HTB input 0 to 15	UNSIGNED16	-	ro (read only)	O	X
2	Read first EX module input 0 to 15	UNSIGNED16	-	ro (read only)	O	X
...						
n	Read last EX module input 0 to 15	UNSIGNED16	-	ro (read only)	O	X

◆ Object 6102h: Polarity Input

This object defines the input polarity.

- 0 = Input is not reversed
- 1 = Input is reversed

NOTE

- The I/O state indicator LED continues to display the actual power status of the connected HTB. It is not affected by this object.

Properties

Object properties:

Sub index	Description	Data Type	Default	Access	PDO mapping	Backup
0	Sub index number (number of 16 bit input)	UNSIGNED8	n	ro (read only)	X	O
1	Input polarity of HTB	UNSIGNED16	0	rw (read-write)	X	O
2	Polarity input of the first EX module	UNSIGNED16	0	rw (read-write)	X	O
...						
n	Polarity input of the last EX module	UNSIGNED16	0	rw (read-write)	X	O

◆ **Object 6103h: Input Mask**

This object is used to set the mask for input.

- 0 = Input is read
- 1 = Input is ignored

NOTE	• The I/O state indicator LED continues to display the actual power status of the connected HTB. It is not affected by this object.
-------------	---

Properties

Object properties:

Sub index	Description	Data Type	Default	Access	PDO mapping	Backup
0	Sub index number (number of 16 bit input)	UNSIGNED8	n	ro (read only)	X	O
1	Input mask of HTB	UNSIGNED16	0	rw (read-write)	X	O
2	Input mask of the first EX module	UNSIGNED16	0	rw (read-write)	X	O
...						
n	Input mask of the last EX module	UNSIGNED16	0	rw (read-write)	X	O

◆ **Object 6200h: Output (8 Bit)**

This object manages the state of digital output.

Sub index management:

- Each digital EX module uses an even-numbered sub index (allocated with 16 bit word).
- EX module and its sub index are given a number in the order closer to HTB.

Properties

Object properties:

Sub index	Description	Data Type	Default	Access	PDO mapping	Backup
0	Sub index number (number of 8 bit output)	UNSIGNED8	n	ro (read only)	X	X
1	HTB output 0 to 7	UNSIGNED8	0	rw (read-write)	O	X

Continued

Sub index	Description	Data Type	Default	Access	PDO mapping	Backup
2	HTB output 8 to 15 (unused in 8 bit output)	UNSIGNED8	0	rw (read-write)	O	X
3	Last EX module output 0 to 7	UNSIGNED8	0	rw (read-write)	O	X
...						
n	Last EX module output 8 to 15	UNSIGNED8	0	rw (read-write)	O	X

◆ **Object 6300h: Output (16 Bit)**

This object manages the state of digital output.

Properties

Object properties:

Sub index	Description	Data Type	Default	Access	PDO mapping	Backup
0	Sub index number (number of 16 bit output)	UNSIGNED8	-	ro (read only)	X	X
1	HTB output 0 to 15	UNSIGNED16	0	rw (read-write)	O	X
2	Last EX module output 0 to 15	UNSIGNED16	0	rw (read-write)	O	X
...						
n	Last EX module output 0 to 15	UNSIGNED16	0	rw (read-write)	O	X

◆ **Object 6302h: Polarity Output**

This object defines output polarity.

NOTE

- The I/O state indicator LED continues to display the actual power status of the connected HTB. It is not affected by this object.

Properties

Object properties:

Sub index	Description	Data Type	Default	Access	PDO mapping	Backup
0	Sub index number (number of 16 bit output)	UNSIGNED8	2	ro (read only)	X	O
1	Polarity of HTB digital output	UNSIGNED16	0	rw (read-write)	X	O
2	Polarity of digital output in the last EX module digital output	UNSIGNED16	0	rw (read-write)	X	O
...						
n	Polarity of digital output in the last EX module	UNSIGNED16	0	rw (read-write)	X	O

Polarity

Summary of properties of this output are shown in the following table:

State	Description
1 (Output is reversed)	0 V = 1 24 V = 0
0 (Output is not reversed)	0 V = 0 24 V = 1

◆ Object 6306h: Fallback Mode Output

This object sets actions when wires are disconnected or control stops.

State	Description
0	Retain value
1	Fallback value (defined by the object 6307h)

Properties

Object properties:

Sub index	Description	Data Type	Default	Access	PDO mapping	Backup
0	Sub index number (number of 16 bit output)	UNSIGNED8	n	ro (read only)	X	O
1	HTB fallback mode	UNSIGNED16	FFFFh	rw (read-write)	X	O
2	Fallback mode of the last EX module	UNSIGNED16	FFFFh	rw (read-write)	X	O
...						
n	Fallback mode of the last EX module	UNSIGNED16	FFFFh	rw (read-write)	X	O

◆ Object 6307h: Fallback Value Output

This object indicates fallback value applied by the output if an internal or communication failure occurs and when the object 6306h is set to 1 for the corresponding bit.

If the object 6308h is 0, fallback does not run.

Properties

Object properties:

Sub index	Description	Data Type	Default	Access	PDO mapping	Backup
0	Sub index number (number of 16 bit output)	UNSIGNED8	n	ro (read only)	X	O
1	HTB fallback value	UNSIGNED16	0	rw (read-write)	X	O
2	Fallback value of the first EX module	UNSIGNED16	0	rw (read-write)	X	O
...						
n	Fallback value of the last EX module	UNSIGNED16	0	rw (read-write)	X	O

◆ **Object 6308h: Output Mask**

This object sets the mask for output.

State	Description
0	Holds the current output value
1	Approve writing to output (corresponds to the value of the object 6200h or 6300h)

NOTE

- The I/O status indicator LED continues to display the actual power status of a connected HTB. It is not affected by this object.

Properties

Object properties:

Sub index	Description	Data Type	Default	Access	PDO mapping	Backup
0	Sub index number (number of 16 bit output)	UNSIGNED8	n	ro (read only)	X	O
1	HTB output mask	UNSIGNED16	FFFFh	rw (read-write)	X	O
2	Output mask of the first EX module	UNSIGNED16	FFFFh	rw (read-write)	X	O
...						
n	Output mask of the last EX module	UNSIGNED16	FFFFh	rw (read-write)	X	O

◆ **Object 6401h: Analog Input**

This object reports the analog input value.

Sub index management:

- One sub index is used in each channel.
- EX module and its sub index are given a number in the order closer to HTB.

NOTE • These rules are applied to all the objects related to analog input.

Properties

Object properties:

Sub index	Description	Data Type	Default	Access	PDO mapping	Backup
0	Analog channel number n	UNSIGNED8	n	ro (read only)	X	X
1	Analog input value of CH1 in the first analog module	UNSIGNED16	0	ro (read only)	O	X
...						
n	Analog input value of the last analog module	UNSIGNED16	0	ro (read only)	O	X

◆ **Object 6411h: Analog Output**

This object writes the value of analog output.

Sub index management:

- One sub index is used in each channel.
- EX module and its sub index are given a number in the order closer to HTB.

NOTE • These rules are applied to all the objects related to analog output.

Properties

Object properties:

Sub index	Description	Data Type	Default	Access	PDO mapping	Backup
0	Analog channel number	UNSIGNED8	n	ro (read only)	X	X
1	Write analog output value of CH1 in the first analog module CH1.	UNSIGNED16	0	rw (read-write)	O	X

Continued

Sub index	Description	Data Type	Default	Access	PDO mapping	Backup
...						
n	Write the last analog output value of the last analog module	UNSIGNED16	0	rw (read-write)	O	X

◆ **Object 6422h: Analog Input PDO Transmission Channel Number**

This object stores the PDO transmission channel number.

Properties

Object properties:

Sub index	Description	Data Type	Default	Access	PDO mapping	Backup
0	Sub index number	UNSIGNED8	1	ro (read only)	X	X
1	Channel number that generates PDO transmission	UNSIGNED32	-	ro (read only)	O	X

◆ **Object 6423h: Analog Input Enabled/Disabled**

This object enables trigger events of analog input. If the value is FALSE, PDO is not transmitted.

Properties

Object properties:

Sub index	Description	Data Type	Default	Access	PDO mapping	Backup
-	-	BOOLEAN	FALSE	rw (read-write)	X	X

◆ Object 6443h: Analog Output Fallback Mode

This object instructs the fallback mode applied by the output when an internal or communication failure occurs.

0: Retain the value

1: Fallback Value

Properties

Object properties:

Sub index	Description	Data Type	Default	Access	PDO mapping	Backup
0	Sub index number (analog input number)	UNSIGNED8	n	ro (read only)	X	O
1	Fallback mode of CH1	UNSIGNED8	1	rw (read-write)	X	O
...						
n	Fallback mode of the last channel	UNSIGNED8	1	rw (read-write)	X	O

◆ Object 6444h: Analog Output Fallback Value

This object indicates fallback value applied by the output if an internal or communication failure occurs and when the object 6443h is set to 1 for the corresponding sub index 1.

Properties

Object properties:

Sub index	Description	Data Type	Default	Access	PDO mapping	Backup
0	Sub index number	UNSIGNED8	n	ro (read only)	X	O
1	Fallback value of CH1	INTEGER32	0	rw (read-write)	X	O
...						
n	Fallback value of the last channel	INTEGER32	0	rw (read-write)	X	O

