FATEK®

# FBs-CBCANH

User Manual

CANopen Master Communication Board

# Index

# Table index

# Figure index

| Version | Date | Author | Description |
|---|---|---|---|
| V1.0 | 2016/12/26 | Edison Lin | Draft |
| V1.1 | 2017/01/04 | Edison Lin | Revised section 6.2.1.4 for Auto. Start Remote |
| V1.2 | 2017/3/24 | Edison Lin | - Revised manual design<br>- Remove single status indicator<br>- Revised ladder program example |
| V1.3 | 2017/5/24 | Curtis Li | Support importing EDS/DCF file |
| V1.4 | 2017/7/18 | Curtis Li | Revised function block |
| V1.5 | 2017/7/19 | Edison Lin | Fixed typo |
| V1.6 | 2017/8/4 | Edison Lin | Modified block ladder-related information |
| V1.7 | 2017/12/11 | Curtis Li | Add SDO task and NMT task, remove older info about SDO and NMT operation |

# 1. Overview

The CBCANH board is a CANopen master communication board for FBs-series PLC and can be mounted on the extended board slot of the CPU module thus saves space. Any FBs PLC can effectively control or exchange data with slave devices on the CANopen network with it.

CANopen is a CAN-based industrial protocol and is widely used in field applications, such as automatic controls for industrial machinery, automobile control systems, factory automation, medical equipment control, building automation, remote data collection and control, environmental monitoring and others.

Both RPDO number and TPDO number supported on CBCANH board are 60, and the range of corresponding PLC registers is from R3200 to R3679. For better user experience, board provides multiple convenient functions including AutoSDO, Auto Start Remote, SDO task and NMT task. By AutoSDO, a set of pre-defined SDO operations are executed automatically once the board is turned on. This is used to configure the target devices; By Auto Start Remote, all slaves in the network are turned into Operational state once the board is turned on; SDO and NMT task makes it possible that SDO and NMT operations are done by accessing PLC registers, and makes PLC programming easier.

# 2. Specification

| Item | Characteristics | |
|---|---|---|
| Compliance with | CAN 2.0A , DS301 V4.02 | |
| PDO number | RPDO | Max. 60 |
| | TPDO | Max. 60 |
| SDO number | Server | 1 |
| | Client | 1 |
| AutoSDO | Max. 30 groups, each of which supports up to 12 SDO operations | |
| Application parameter object | Max. 1000 | |
| Sync master | Configurable Sync time | |
| NMT master | Available through EasyCANHopener and NMT task | |
| Time stamp | Consumer | |
| Error control | Heartbeat | |
| Baud | 20K, 50K, 125K, 250K, 500K, 750K, and 1M | |

| | | |
|---|---|---|
| Configuration | Remote | Supported |
| | PC utility | EasyCANHopener |
| Remote PLC programming | Supported | |
| Vendor ID | 2EF(Hexadecimal) | |
| Signal terminal | 3 Pin spring terminal | |
| Electric isolation | Yes | |
| Voltage/current | 5V, 150mA | |
| Working temperature | 0 ~ 60 ℃ | |
| Storage temperature | -20 ~ 80 ℃ | |

Table 1 The specification of FBs-CBCANH

## 3. Installation and wiring

FBs PLC provides an extension slot on the left side of the CPU module for installing an additional communication board.


Figure 1 CBCANH top view

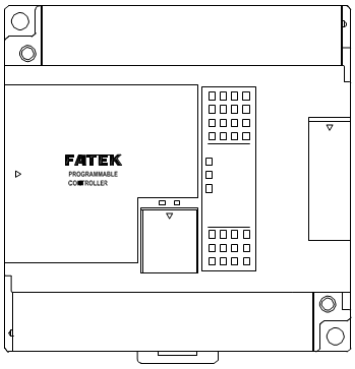



Figure 2 FBs PLC top view

The CBCANH connects to a CANopen network via a 3-pin spring style connector.

| Pin | Signal | Description |
|---|---|---|
| 1 | CAN_H | CAN_H bus line (dominant high) |
| 2 | CAN_L | CAN_L bus line (dominant low) |
| 3 | CAN_GND | Ground / 0V / V- |

Table 2 Definition of the pins

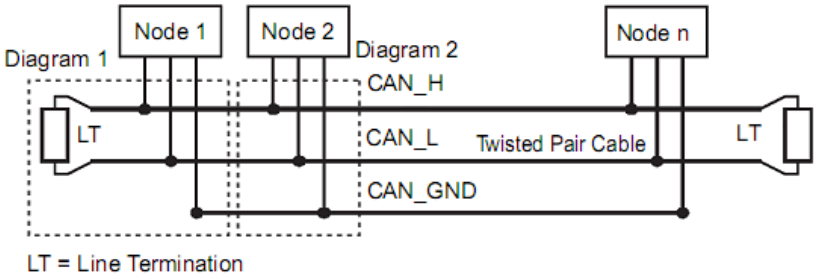



Figure 3 CANopen line termination

Normally for a node located at either left or right end of a network, a 120Ω 1/4W line termination resistor should be connected between CAN_H and CAN_L terminal pins as shown in above picture to ensure the signal quality, but for CBCANH, there's an easier way to implement the same function - short the jumper (JP4) labeled "Term" inside the CBCANH module. The following figure shows the location of the jumper.

Figure 4 Term jumper location

## 4. PLC application interface

The CPU module communicates with the CBCANH board via its internal registers. These registers basically fall into the following areas:

### 4.1 Communication interface area

In this area, there are 70 reserved registers, ranging from R3700 to R3769.

These registers shall be reserved for internal system operation and CAN NOT be used for user's applications since the configuration tool, EasyCANHopener, communicates with the CBCANH through this area.

## 4.2 Parameter data area

The size of this area is configurable up to 1000 registers. From the point of view of the network, these registers in this area act as objects for access. These registers can be addressed by their indexes and sub-indexes, thus they can be accessed by SDO services. With this function, users can change or adjust parameters of users' applications over network easily during system installation in the field.

## 4.3 Process data area

There are 80 registers in this area, ranging from R3200 to R3679. The real-time data(PDO) of the PLC is exchanged with other nodes on the network via this area.

| Sequence | Item | Function | Register |
|---|---|---|---|
| 1 | RPDO1 | Word #1 | R3200 |
| 2 | | Word #2 | R3201 |
| 3 | | Word #3 | R3202 |
| 4 | | Word #4 | R3203 |
| 5 ~ 240 | RPDO2 ~ RPDO60 | | R3204 ~ R3439 |
| 241 | TPDO1 | Word #1 | R3440 |
| 242 | | Word #2 | R3441 |
| 243 | | Word #3 | R3442 |
| 244 | | Word #4 | R3443 |
| 245 ~ 480 | TPDO2 ~ TPDO60 | | R3444 ~ R3679 |

Table 3 Process data area

The data length (0-4, with one register size per unit) of each PDO (Process Data Object) is configurable. When the data length is smaller than 4, preceding registers will be used first. For example, if the length of RPDO1 is 2, only the first two registers R3200 and R3201 will be used. The starting register number of each PDO is fixed and will not be changed even if a preceding PDO does not use up all of its usable registers. Any unoccupied registers in this range can be free to use by applications.

## 4.4 Module status area

| Sequence | Register | Function | |
|---|---|---|---|
| 1 | R3680 | Module Status | Low Byte:<br>Bit 0 : =0, Normal<br>=1, Stopped when excessive RX error occur while startup.<br>Bit 1: AutoSDO execution status<br>Bit 2: Reserved<br>Bit 3: =1, CAN Rx error<br>Bit 4: =1, CAN Tx error<br>High Byte:<br>Bit[15:8] CBCANH state.<br>=0, init. =5, OPERATIONAL. =4, STOPPED.<br>=127 PRE-OPERATIONAL |
| 2 | R3681 ~ R3684 | TPDO Status | Each bit represents the receiving status of each TPDO.<br>When a bit =1, it means data update is normal. |
| 3 | R3685 | Heart beat status | Nodes 1-15<br>When Bit[1]=1, it means the heartbeat in Node 1 is detected.<br>The detection cycle is determined by Heartbeat guard Time.<br>When Heartbeat guard time = 0, this register is not valid since CBCAN will not monitor heartbeat signals |
| 4 | R3686 | | Node 16-31 |
| 5 | R3687 | | Node 32-47 |
| 6 | R3688 | | Node 48-63 |
| 7 | R3689 | | Node 64-79 |
| 8 | R3690 | | Node 80-95 |
| 9 | R3691 | | Node 96-101 |
| 10 | R3692 | | Node 102-127 |
| 11 | R3693 | Time Stamp | Second (0-59) |
| 12 | R3694 | | Minute (0-59) |
| 13 | R3695 | | Hour (0-23) |
| 14 | R3696 | | Day (1-31) |
| 15 | R3697 | | Month (1-12) |
| 16 | R3698 | | Year (2000-2099) |

| | | | |
|---|---|---|---|
| 17 | R3699 | Time Stamp receiving indication | The value of this register increment by one when a new Time Stamp packet is received and rollover at 65535. |

Table 4 Module status area

## 4.5 Reserved PLC registers for block ladders

| Register | Description | |
|---|---|---|
| R3100 - R3199 | R3116 | CMR parameter: The number of register to send |
| | R3106 | CMR parameter: The base register to send |
| | others | Internal use |
| M1000, M1962 | Internal use | |

Table 5 Reserved PLC registers



# 5. LED Indicators

| Indicator State | Node Operation State |
|---|---|
| Single flash | PRE-OPERATIONAL |
| on and off every 2 secs | STOPPED |
| Blinking | OPERATIONAL |

Table 6 RUN LED(green)

| Indicator State | Error State |
|---|---|
| Off | No error |
| Single flash | Warning limit reach |
| Triple flash | Rcv. SYNC signal is time-out |
| Quadruple flash | Any one of expected TXPDOs is time-out |
| On | Bus off |

Table 7 ERR LED(red)

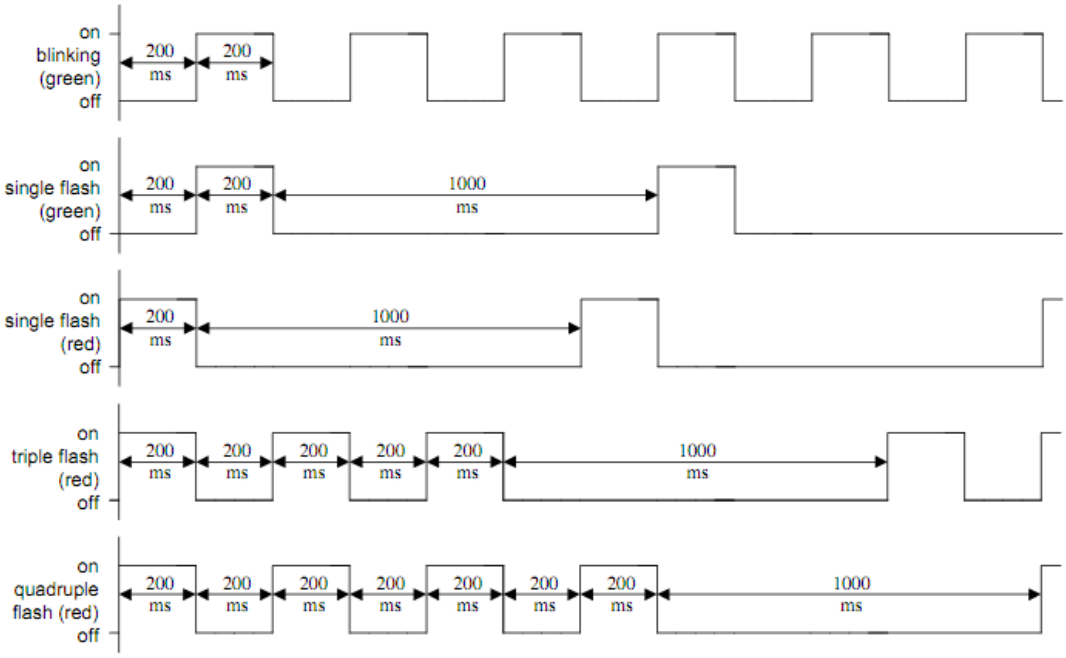



Figure 5 Indicator status and flash rate

## 6. EasyCANHopener

Major functions of the software:

- Configuration of the CBCANH board, including RPDO and TPDO mapping, AutoSDO, SDO task, NMT task, Auto Start Remote, and other settings. Configuration can be exported as a chcfg file for duplication, storage, and modification offline.
- SDO Service: Users can either access a single controller object directly, or access multiple objects using batch operation. The batch contents can also be saved in files to be reused in the future.
- PLC remote access bridge: The FBs PLC is usually connected to the Winproladder via local serial port. When this function is enabled, the CBCANH can work as a remote programming gateway, and the Winproladder can connect to another FBs PLC equipped with CBCAN board at remote site through the CBCANH.
- NMT service: with a single operation, users can send “Start Remote”, “Enter PRE-OPERATIONAL”, “Reset Node”, “Reset Communication” or “Stop” commands to a specific node or all nodes on the network.
- CBCANH firmware update: Users can update the firmware of the CBCANH board on the PLC via the serial port.

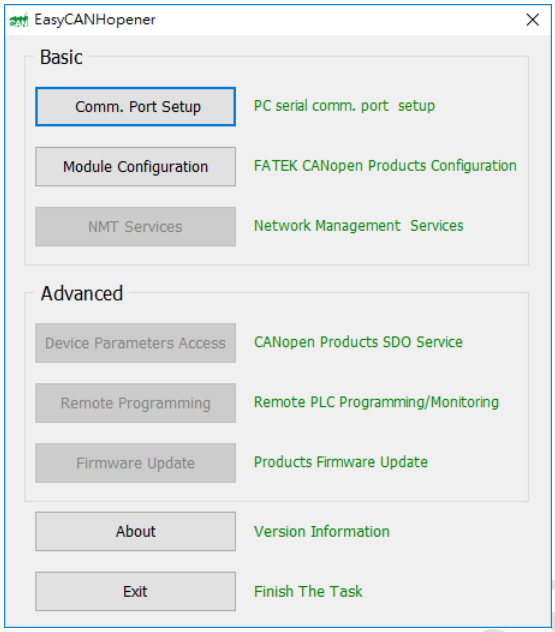



Figure 6 Main menu

## 6.1 PLC connection

All operations requiring a PLC connection will not be enabled before a successful serial communication with the PLC, so one has to use Comm. Port Setup to establish the connection beforehand, as shown in the Figure 7.

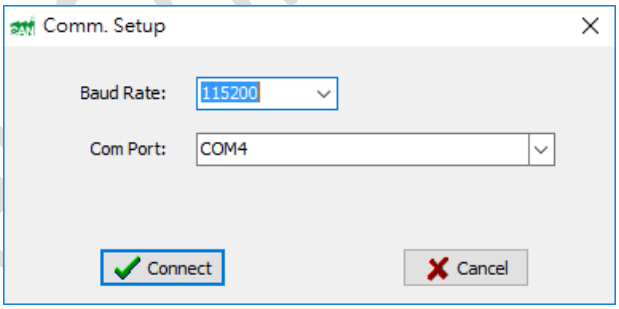



Figure 7 Baud setup

A full functioning main menu can be referred to in the Figure 8.

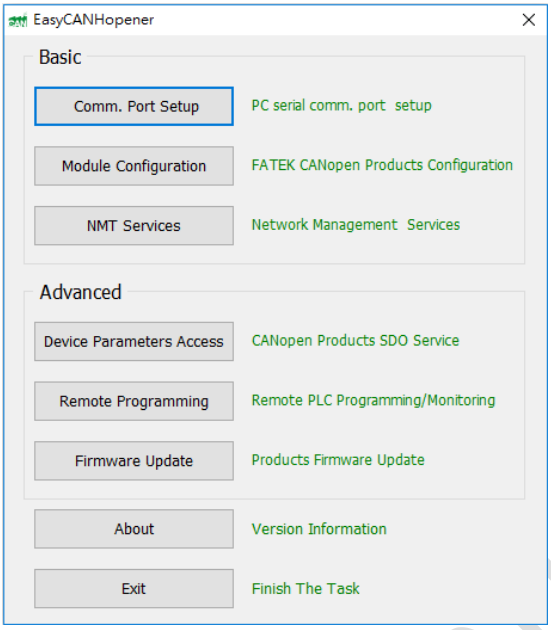



Figure 8 Main menu with full functions

## 6.2 Configuration setup

Click the “Module Configuration” button on the main page to enter the Configuration Setup screen as shown below.

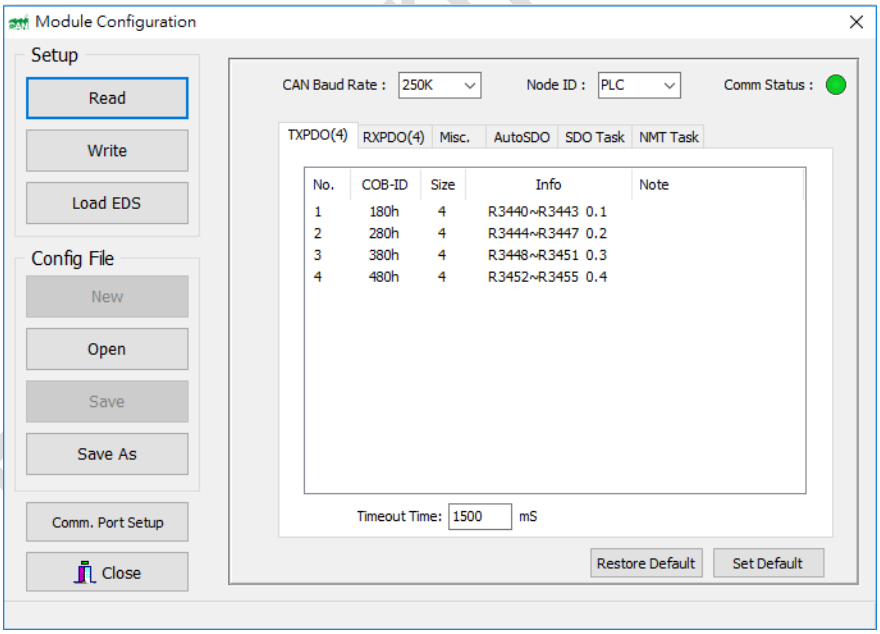



Figure 9 Configuration setup

### 6.2.1 Create new configuration

When entering the Configuration Setup screen, a new configuration file will be generated automatically and all parameters are in their

default values. Users should be aware that node ID in PDO default configuration is empty.

#### 6.2.1.1 Basic setup

**CAN Baud Rate:** This is the network operating speed. The default value is 250K bps. The range is 20K-1M bps. The same baud rate must be set for all node devices on the network; otherwise, they will not operate properly.

**Node ID:** This ID is set to indicate the node id of the CBCANH on the CANopen network and can be configured as 1-127 or “PLC”. When it is set to “PLC”, the node ID will be configured with the station number of the PLC CPU module for quick integration into the system without additional efforts. The default value for this setting is “PLC”.

**Comm Status:** An indicator shows the connection status with the PLC. Normally a green light should be observed.

#### 6.2.1.2 TPDO setup

Click the TPDO tab to enter the receive process data object (TPDO) setup page as shown in the figure below. It is used to receive real-time data from slaves.

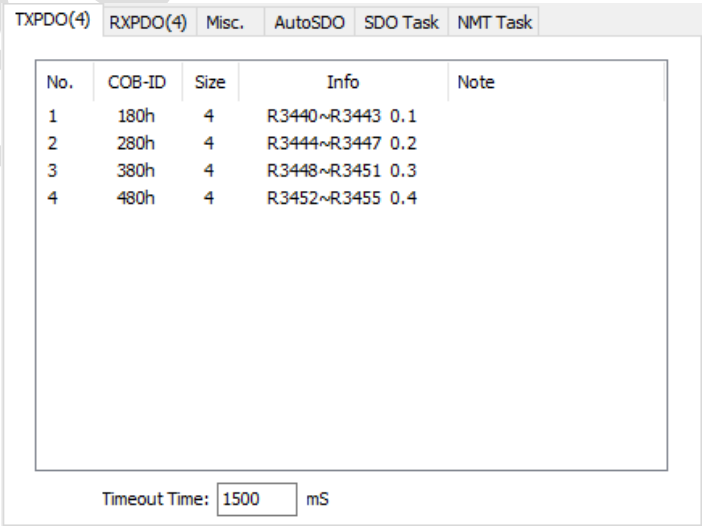



Figure 10 TPDO setup

The number in the parentheses of the tab name represents the

quantity of PDOs that has been defined.

As shown in the Figure 10, four TXPDOs have been defined. The TPDO can be configured by the pop-up menu as shown in the figure below.


Figure 11 TPDO pop-up menu

The functions in this popup menu are described as below:

- Edit: Right-click on the PDO you want to edit to start the pop menu, and select Edit… from the menu. Or, simply double-click the PDO you want to edit to start the following screen.

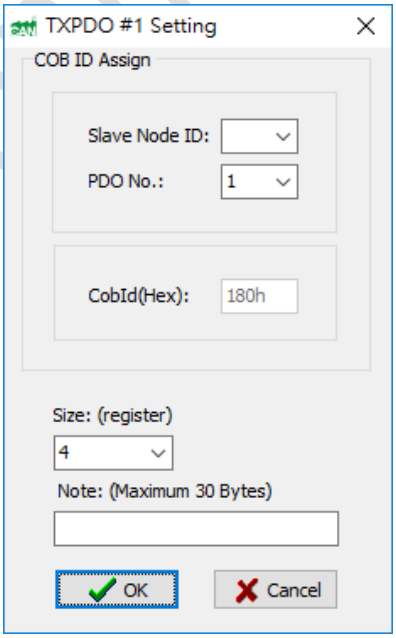



Figure 12 Edit TXPDO

From this screen, users can configure the communication object ID (COB-ID) and data length:

- COB-ID： Each node on the CANopen network has a unique set of COB-IDs for PDO mapping, and CBCANH

as a master needs to know what the mapping is like on the network in order to control or receive data as expected. Also each node can own up to 4 TXPDO COB-ID and 4 RXPDO COB-IDs for data exchange. Figure 12 shows a default TXPDO setting with empty node ID. Be sure to fill in a valid ID while configuring.

- Size Setup: The valid size of the TXPDO can be from one to four 16-bit words.
- Note: Comment of a TPDO

- Delete: Right-click the PDO you want to delete and select “Delete” from the pop menu to delete the PDO.

- Insert: Right-click any location where a PDO to be inserted and select “Insert” from the pop menu to start the PDO Edit screen. After the configuration on the popup menu is done, a new PDO will be inserted at the location addressed.

- Add: Right-click the screen to start the popup menu. Select “Add” to start the PDO Edit screen. After the configuration on the popup menu is done, a new PDO will be added at the end of the list.

- Up: Right-click the PDO to be moved up. Select “Up” on the popup menu to move the PDO to one position upward.

- Down: Right-click the PDO to be moved down. Select “Down” on the popup menu to move the PDO to one position downward.

- Delete All: Right-click on the screen to start the pop menu. Select “Delete All” to delete all PDOs in the list.

There is an additional item “Timeout Time” at the bottom of the TPDO setup page as shown in Figure 10:

***Timeout Time:***

This setting is used to indicate the deadline after the reception of a TPDO message. If TPDO messages are received properly within

this interval, the corresponding bit of PLC internal register R3681 “TPDO status” will be set to 1. If not, the bit will be set to 0 to indicate the TPDO timeout event. The default value for this setting is 1.5 seconds.

#### 6.2.1.3 RPDO setup

The RPDO setup is similar to the TPDO’s; this section only described the difference between TPDO and RPDO setup. Please refer to the section 6.2.1.2 for those similar settings.

There is one additional item “Transmission Type” added to the RXPDO Edit page:

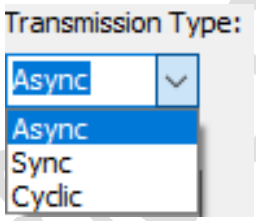



Figure 13 Transmission type

Transmission Type

- Async (asynchronous transmission): A RPDO is transmitted when the RPDO data is changed or event time is elapsed. After RPDO messages are transmitted, the end of the Inhibit Time must be reached before sending next RPDO message. With this function, users can prevent the communication bandwidth from being consumed by unexpected high frequency of input signals.

- Sync. (synchronous transmission): Whenever the status changes, the RPDO message will be transmitted immediately after the next SYNC message is received.

- Cyclic: RPDO messages are transmitted periodically in time with the cyclic time setting.

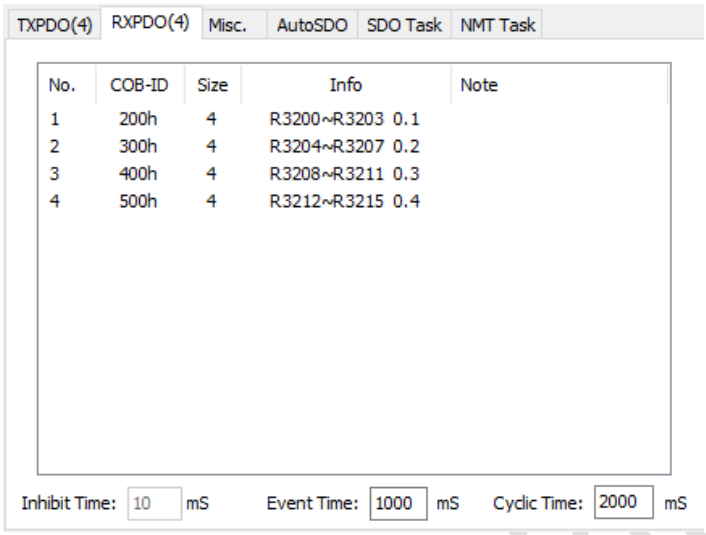



Figure 14 RXPDO setup

***Inhibit Time*** The minimum transmission time interval when RXPDO are transmitted in Async. mode. After the transmitting of RXPDO messages, the next RXPDO message will not be sent until the end of inhibit time even if the status is changed during the inhibit time period. The default value is 10ms and cannot be changed.

***Event Time*** The maximum transmission time interval when RXPDO are transmitted in Async. mode. After the transmitting of RXPDO messages, the next RXPDO message will be sent immediately after the end of event time even if the status is not changed during the event time period. The default value is 1 second.

***Cycle Time*** The transmission time interval used in the Cyclic mode. The default value is 2 seconds.

#### 6.2.1.4 Miscellaneous

Click the “Misc.” tab to edit miscellaneous setup as shown in the Figure 15 below.

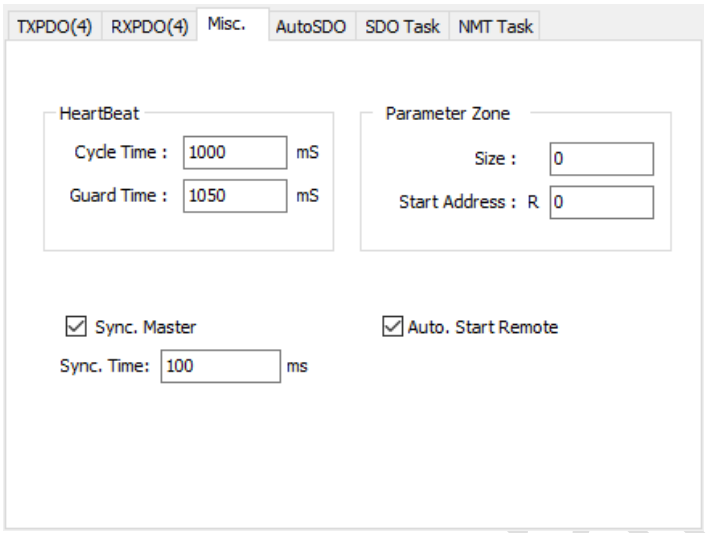



Figure 15 Misc. setup

*Heartbeat* This informs the existence of a node within a specific time interval. In normal circumstances, a node will send a heartbeat signal at a regular interval. Other nodes determine if this node is operating normally based on this signal.

- *Cycle Time:* This represents the regular interval the node sends the heartbeat signal. If the value is "0", this means the heartbeat function is disabled.

- *Guard Time:* This represents the time criteria for a node to detect if the heartbeat of other nodes is normal. In general, individual nodes have individual cycle times, thus users should set different cycle times for different nodes. However, this system assumes that the cyclic time of all nodes are the same in order to simplify the configuration process. In this case, the value must be greater than the default value.

***Parameter data*** In general, there are usually some parameters that do not change frequently. These parameters can be mapped to the CANOpen objects through the SDO (Service Data Object) service for remote access over the network.

- *Size:* This setting is to indicate the quantity of the mapped registers. The maximum value is 1000.

- *Start Address:* This represents the start address of the mapped register. Only R-registers can be mapped as parameter registers.

***Sync. Master*** When this option is enabled (checked), the CBCANH will operate as a sync. master node which transmits synchronous messages.

***Sync. Time*** When Sync master option is enabled, this value represents the interval of synchronous messages transmission.

***Auto Start Remote*** When enabled, the CBCANH will put all the slaves on the network in OPERATIONAL mode using NMT after boot up and the execution of AutoSDO if any.

#### 6.2.1.5 AutoSDO

Maximum 30 groups are supported and each of which can be configured up to 12 SDO operations. Each group can have unique node ID or share the same node ID if more operations are needed. AutoSDO will be executed sequentially during power on or through a ladder program and the result can be checked through the configuration tool. It could be used to configure slaves automatically before the execution of a ladder program.

***WR*** A SDO operation which writes data into a slave device, e.g. PDO mapping and any preset value.

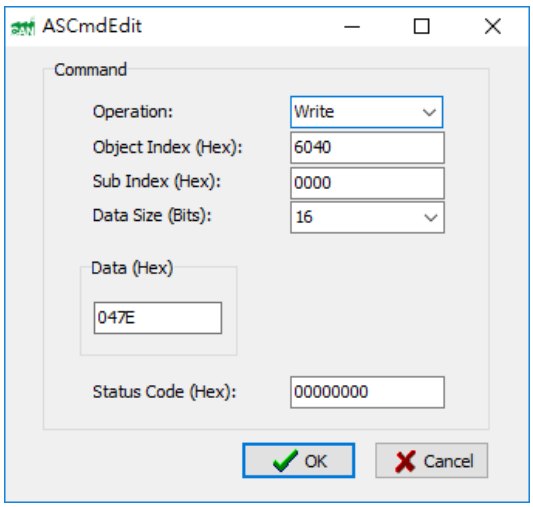


Figure 16 AutoSDO write operation

***MR*** A SDO operation which reads data from a slave device and compares it with preset data. A mask will be used to logical AND with data read, which makes a BIT comparison possible. Status Code shows the result of monitoring.

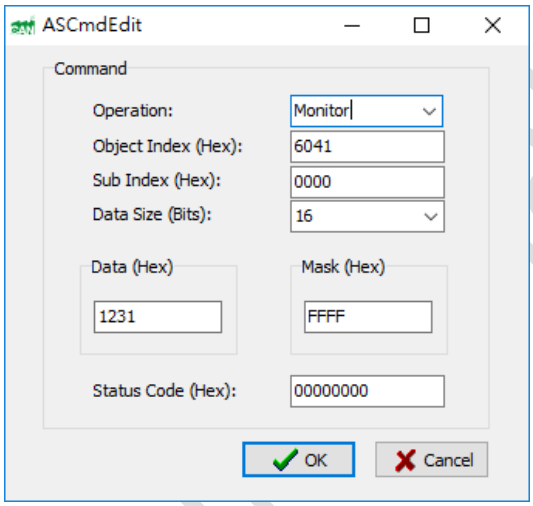



Figure 17 AutoSDO monitor

### 6.2.2 SDO Task

SDO task makes it possible that SDO operations can be done by accessing registers of PLC. Adding a new SDO task is as simple as configuring it with index and sub-index of a specific node, operation mode, type and start address of corresponding PLC registers in SDO task page. After the setup is finished, accessing the corresponding PLC registers is the same as accessing SDO data. Maximum 32 operations are supported.

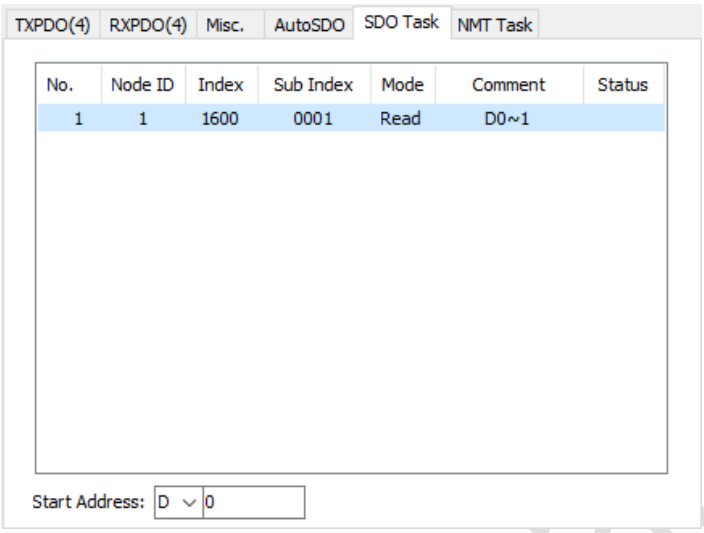



Figure 18 SDO task setup

As shown in Figure 18, SDO task supports two modes, including read and write. Both of them support data access in three variant data length (8/16/32 bits). Status code shows the result of execution, either success or error code is returned.

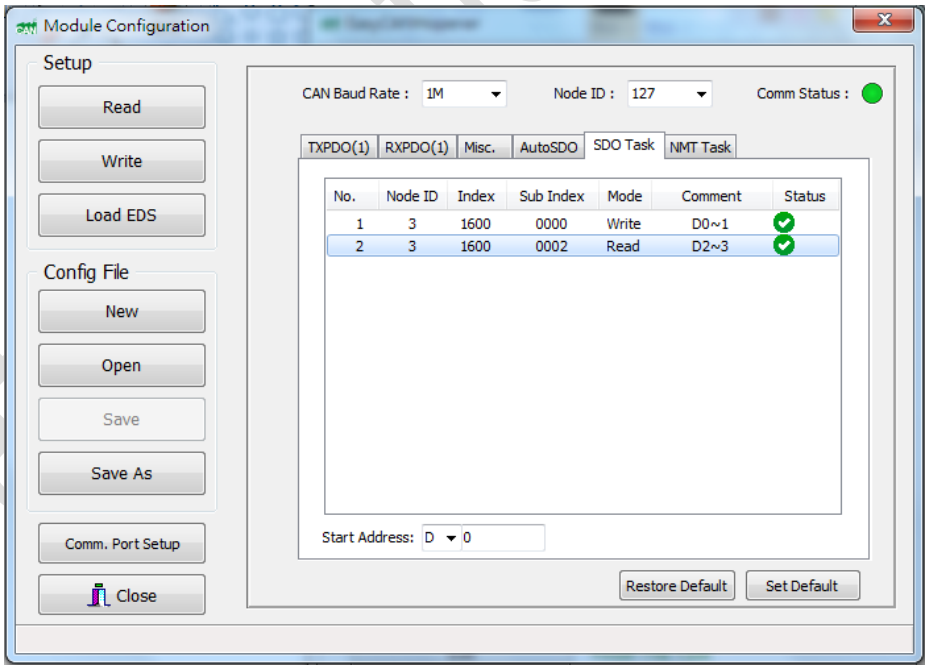



Figure 19 SDO task page

If the operation mode is read, the data accessed from slaves is put into the corresponding PLC registers. If the operation mode is write, the data which is about to be transmitted is put into the

corresponding PLC registers. As shown in Figure 19, each SDO task occupies two PLC registers regardless of the data size.

### 6.2.3 NMT Task

NMT task makes it possible that NMT commands can be done by accessing registers of PLC. Adding a new NMT task is as simple as configuring it with target node, NMT command, type and start address of corresponding PLC registers in NMT task page. After the setup is finished, accessing the corresponding PLC registers is the same as executing NMT command. Maximum 32 operations are supported.

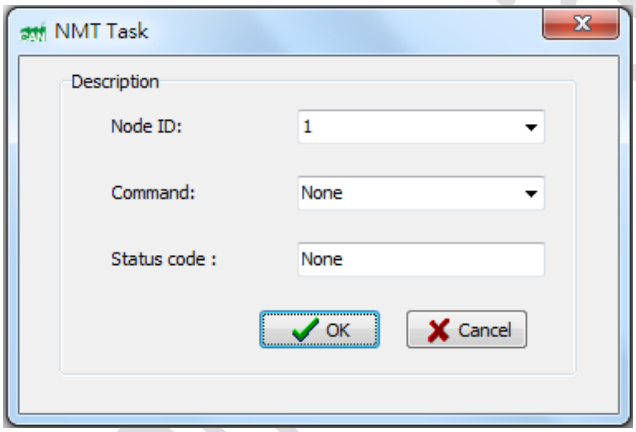



Figure 20 NMT task setup

The NMT task setup page is shown in Figure 20. Besides the same NMT commands that NMT Services has, NMT task has one additional command named “none”. NMT command “none” has no default command and used in the situation when the operation is decided sometime later. Status code shows the result of execution, either success or error code is returned.

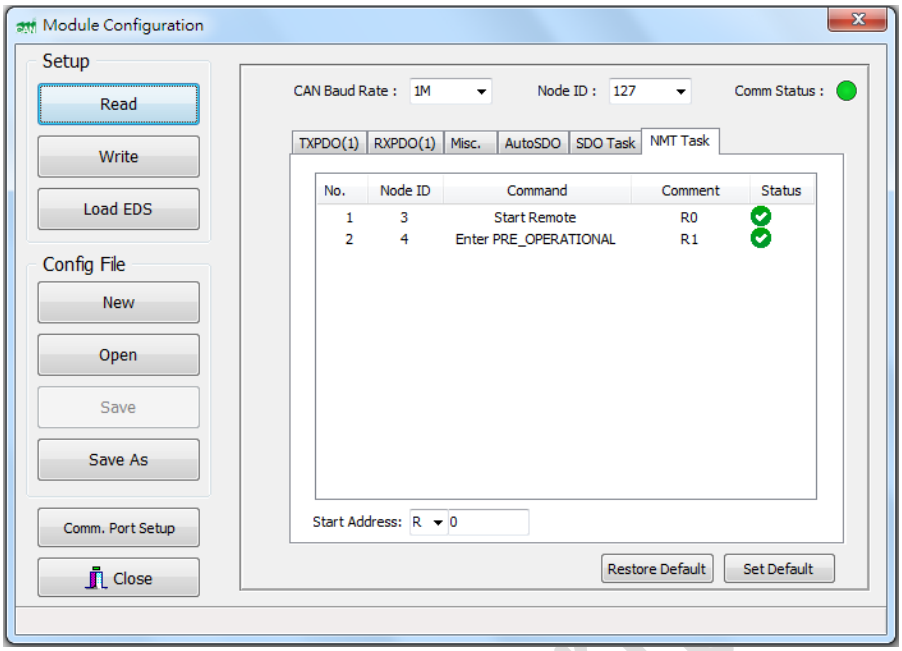



Figure 21 NMT task page

As shown in Figure 10, each NMT task occupies one PLC register. The Most Significant Byte(MSB) of corresponding PLC register stores NMT command, while the Least Significant Byte(LSB) of corresponding PLC register is used to trigger command and store the result of execution. The corresponding value of NMT command, as shown in Table 1, is filled into MSB. The value filled into LSB to trigger command is shown in Table 2.

| NMT Command | Value |
|---|---|
| **None** | 0 |
| **Start Remote** | 1 |
| **Enter Pre-Operational** | 2 |
| **Reset Node** | 3 |
| **Reset Communication** | 4 |
| **Stop** | 5 |

Table 8 Corresponding values of NMT commands

| Status Code | Description |
|---|---|
| **0001h** | Successful |
| **0002h** | Failed |
| **Other Values** | Trigger Command |

Table 9 Corresponding values of status codes

For example, to execute NMT command “Start Remote” through NMT task, one should set the content of corresponding PLC register as value shown below:

**0103H**

After execution, status code returns the result. If it works successfully, the content of corresponding PLC register should be the same as value shown below:

**0101H**

### 6.2.4 Save configurations to files

Click the “Save…” or “Save As” button. The save configuration window will pop up. Input the name of the file to be saved and click the “Save” button to save the configurations.

### 6.2.5 Open old configuration files

Click the “Open...” button. The open configuration file window will pop up. Click the  button and select the configuration files that have been saved. Click the OK button to open the selected configuration file.

### 6.2.6 Read CBCANH configurations

Click the “Read” button to retrieve current configuration in the module.

### 6.2.7 Write configurations to CBCANH Board

Click the “Write” button to write configuration into the module.

### 6.2.8 Load EDS

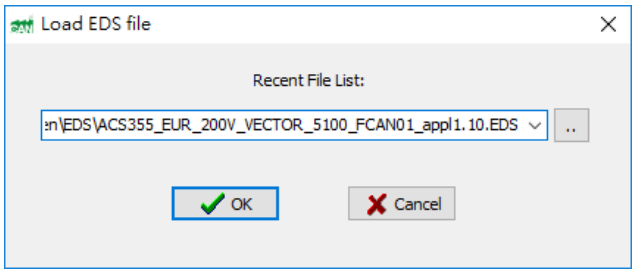



Figure 22 Load EDS file

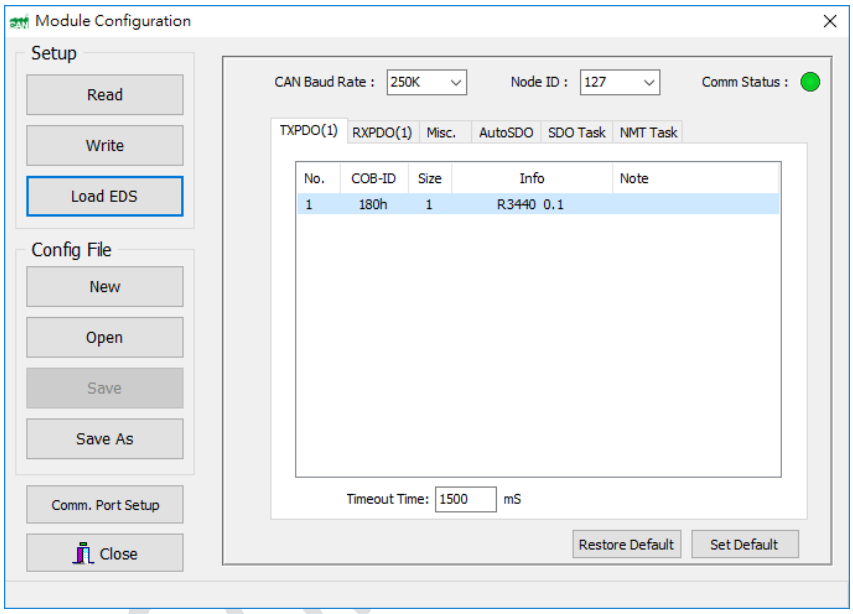



Figure 23 After loading a EDS file

### 6.2.9 End configuration setup

Click the “Close” button to exit the relevant configuration setup functions.

## 6.3 SDO Service

Click the “Device Parameter Access” button on the main page to enter the SDO Service screen as shown below. The setup items are described below.

**Node ID:** ID of one of the nodes on the network.

**Index:** Object index

**Sub-Index:** Object sub-index

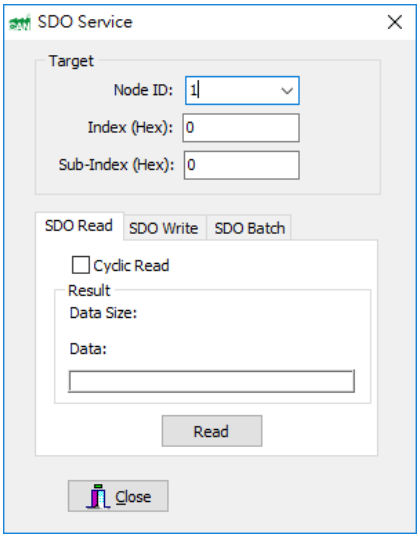



Figure 24 SDO service page

### 6.3.1 SDO Read

Click the SDO Read tab to enter the SDO Service screen as shown in Figure 25.

***Cyclic Read*** Select (check) this option to read SDO data repeatedly. This function is useful for observing the dynamic change of objects.

***Read*** Click the button to start reading SDO data. The results will be displayed in the window as shown below.

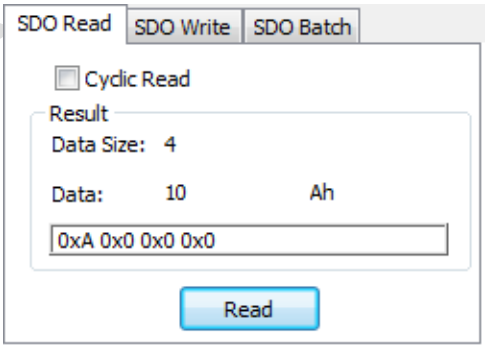



Figure 25 SDO read page

The value “4” in the Data Size item means that the size of the data is 4 bytes. The data content are given in decimal (10), hexadecimal (AH) and string type (0xa 0x0 0x0 0x0).

### 6.3.2 SDO write

Click the SDO Write tab to enter the SDO Write service screen as shown in Figure 26.

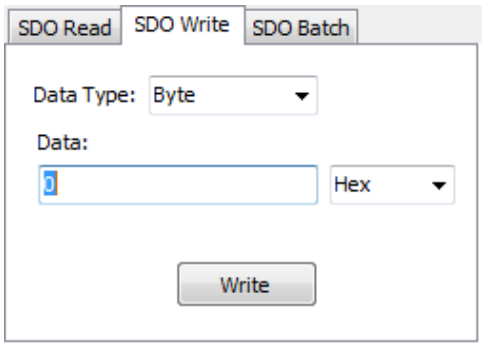



Figure 26 SDO write page

***Data Type*** Type of object data. Only Byte, Word and Dword data are supported.

***Data*** The value to set. Users can select data format from the pull-down list next to the box: Hex (hexadecimal) or Decimal.

***Write*** Click the button and start to write. An error window will pop up if there is any error.

### 6.3.3 SDO batch

By using SDO batch, a list of pre-defined multiple SDO commands can be sent out sequentially. It is very useful when multiple CBCANH modules need to be set up with similar configuration, thus users don't need to configure all functions one by one.

First, click the SDO Batch tab to enter the SDO batch screen as shown in Figure 27.

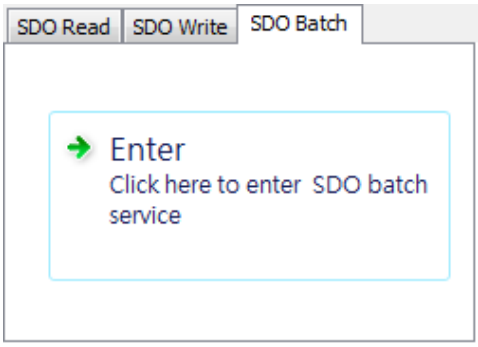



Figure 27 SDO batch

Click on the SDO batch window in Figure 27 to start SDO batch editing as shown in Figure 28.

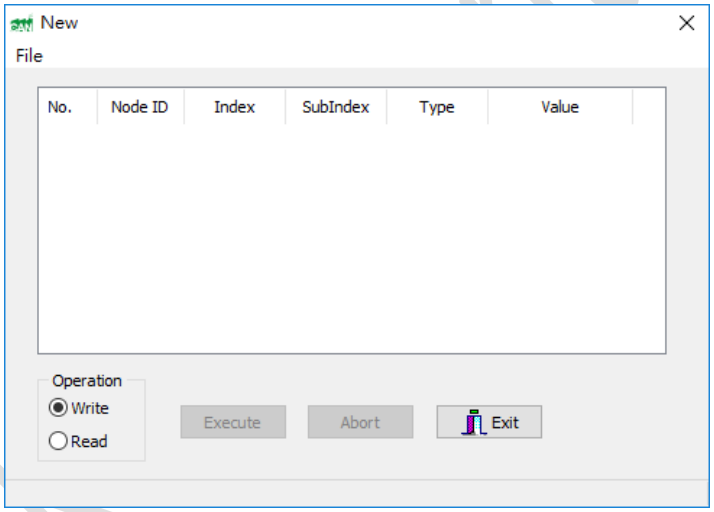



Figure 28 SDO batch setup page

### 6.3.3.1 Create batch contents

After entering the SDO batch service screen, there should be no batch contents. Then, users can create the batch content. The menu pops up as shown below.

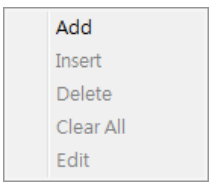



Figure 29 pop-up menu for creating batch

**Add:** Right-click the batch content window and select “Add” from the pop menu to start the following window.

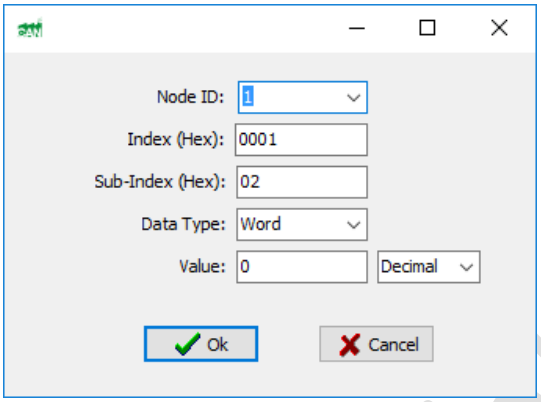



Figure 30 Add a SDO command

***Node ID*** Select the ID of the node you want to write SDO data to.

***Index*** Input the index value.

***Sub-Index*** Input the sub-index value.

***Data Type*** Select SDO data type: Byte, Word or Dword.

***Value*** Input a value.

Click the OK button to complete the creation of the new batch content.

**Insert:** Right-click the location where the batch command to be inserted and select “Insert” from the pop-up menu. Click the “OK” button to insert a new batch command in the specified location.

**Delete:** Right-click the batch item you want to delete and select “Delete” from the pop-up menu to delete the SDO command.

**Delete All:** Right-click the batch content window and select “Delete All” from the pop-up menu to delete all batch commands.

**Edit:** Right-click the batch item you want to edit and select “Edit” from the pop-up menu or double-click the command item to edit its content. Then, click the “OK” button to complete modifying the content.

#### 6.3.3.2 Save batch contents

Click “File” from the main menu at the top of the window and click “Save” or “Save As” from the pull-down to save the current batch contents in a file.

#### 6.3.3.3 Read this batch file

Click “File” from the main menu at the top of the window and click “Open” from the pull-down list to read the contents of the batch file.

#### 6.3.3.4 Execute batch write

First, select the “Write” option in the operation group. Then, click the “Execute” button to start batch write. SDO write commands will be sent out according to the contents in the batch window.

#### 6.3.3.5 Execute batch read

First, select the “Read” option in the operation group. Then, click the “Execute” button to start batch read. SDO read commands will be sent out according to the contents in the batch window. It differs from “batch write” in the read-back values will overwrite the original ones.

### 6.4 PLC remote monitoring service

With this service, the ladder programming software, Winproladder, connects to the remote PLC equipped with the CBCANH board via the PC serial port. The ladder programs on the remote PLC therefore can be modified using CANopen network.

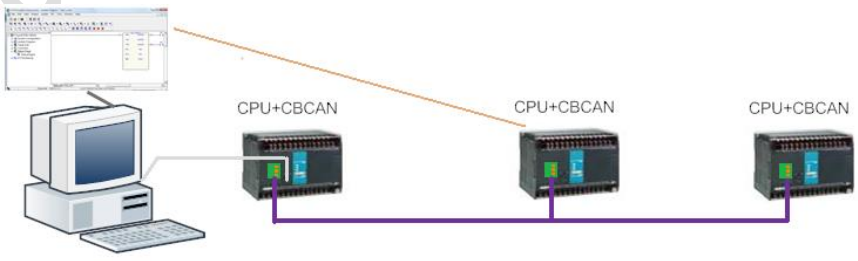



Figure 31 PLC remote monitoring service

#### 6.4.1 Open service

Click the “Remote Programming” button on the main page to enter the remote monitoring service screen as shown in Figure 32.

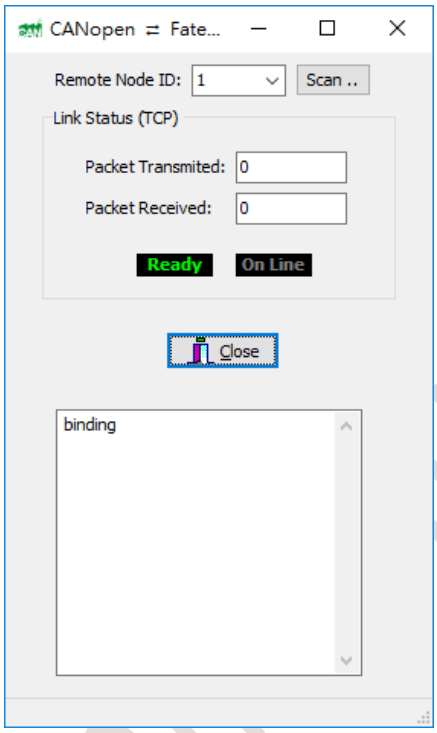



Figure 32 Gateway status window

Items in the screen are described as follows:

**Remote Node ID:** Input the ID of the node which you want to connect.

**Link Status:** Auxiliary status.

**Packet Transmitted:** The quantity of data packets transmitted to the remote PLC.

**Packet Received:** The quantity of data packets received from the remote PLC.

This service is completed through the local CBCAN module which operates as a gateway between the remote PLC and local Winproladder software. It is based on the TCP communication at port 500.

**Ready:** When the system is ready for the service, the button is green.

**On Line:** When the TCP network and application are linked, the button is green.

### 6.4.2 Run service

Users must connect to local machine IP 127.0.0.1 at port #500 via TCP connection to subscribe the service. Therefore, an example of Winproladder settings is shown in Figure 33:

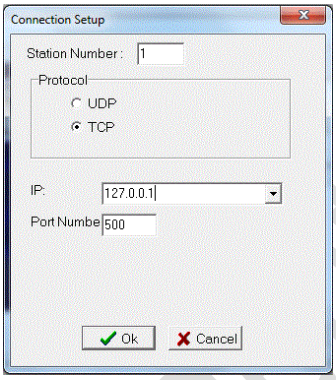



Figure 33 WinProladder connection setup page

As it is connected via the TCP communication, users can link up with the CANopen network with Ethernet from a remote site.

## 6.5 NMT Services

Click the “NMT Service” button on the main page to enter the NMT Services screen as shown in Figure 34.

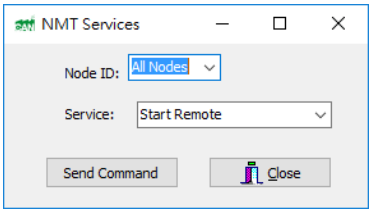



Figure 34 NMT service

**Node ID:** Input the ID of the NMT slave node to be managed. Users can select either a single node (1-127) or all nodes.

**Service:** The service includes “Start Remote”, “Enter PRE_OPERATIONAL”, “Reset Node”, “Reset Communication”, and “Stop”

After the configuration is done, click the “Send Command” button to start NMT service.

## 6.6 CBCANH Firmware Update

Click the “Firmware Update” button on the main page to enter the Firmware Update screen as shown in Figure 35.

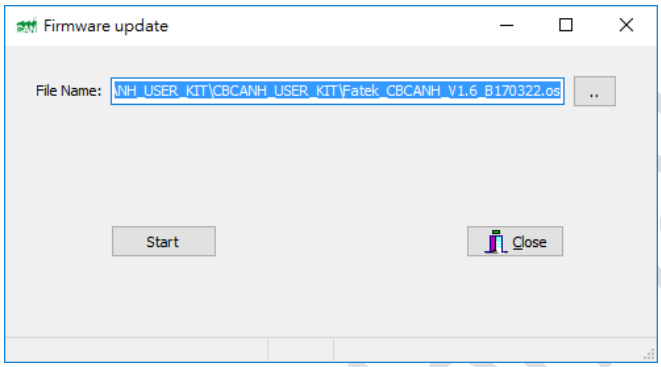



Figure 35 Firmware update

### 6.6.1 Select firmware files

Click the  button to open the file selection window. After selecting the firmware file, press the “OK” button, the following confirmation dialog box will pop up with the corresponding firmware information.

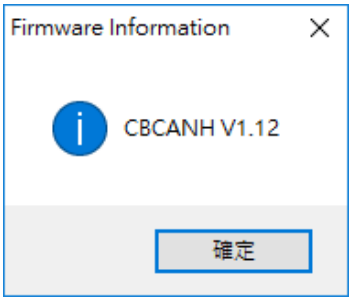



Figure 36 Firmware information

### 6.6.2 Start firmware update

Click the “Start” button to start the firmware update.

## 6.7 End operations

Click the “Exit” button on the main page to close EasyCANHOpener utility.

# 7. Block ladders for CBCANH control

- CMR

As shown in Figure 37, it is a calling block used to update a certain set of PLC registers which maps to configured RPDOs to the CBCANH. Place only one of it in the bottom of the main ladder program and make sure the corresponding reserved registers have been set as intended. Refer to section 4.5 and the appendix A for more information.

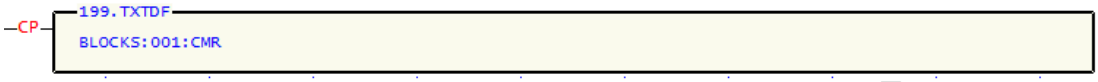



Figure 37 CBCANH-specific block ladder - CMR

- AUTOSDO_CTRL

As shown in Figure 38, it is a calling block which provides an alternative way to execute AutoSDO groups in the ladder program. Users should call the block ladder from a sub-function in case of triggering, as shown in Figure 39.

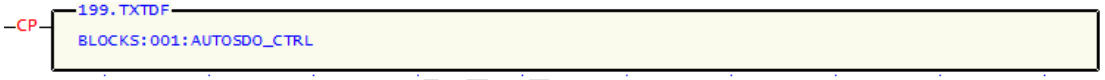



Figure 38 CBCANH-specific block ladder - AUTOSDO_CTRL

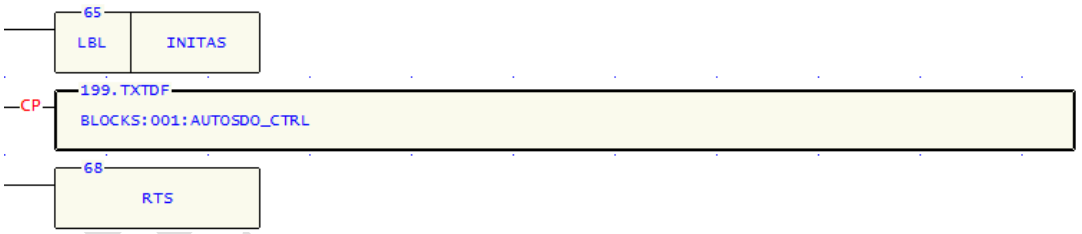



Figure 39 Calling AUTOSDO_CTRL from a sub-function

# Appendix A

## 1.1 Application Notes

CBCANH is a CANopen master communication module which provides the FBs series PLC with the ability to exchange data with or control slave devices. Most of the CANopen objects in the CBCANH are configurable via EasyCANHopener utility and shall be configured properly according to field network requirements. The purpose of this section is to provide a quick configuration reference to assist users to embrace the world of CANopen control.

### 1.1.1 Example: Homing control of a Estun servo drive

The environment setup is a CANopen network with CBCANH as a master device and Estun servo drive as a slave device. The ladder program example uses a timer to activate or deactivate the homing operation.
The CBCANH as a master uses the configuration in the following table to connect to the CANopen network.

| items | Values |
|---|---|
| Node ID | 8 |
| Slave Node ID | 1 |
| Baud | 1M bps |
| Sync. master | Disabled |
| Error control | Heartbeat producer (1000 ms)<br>Heartbeat consumer (1050 ms) |
| Number of AutoSDO group | 2 |
| Number of RPDO | 1 (1 Word)<br>Transmission type: Async. Mode |
| Number of TPDO | N/A |

Table 10 network configuration

All settings can be configured via EasyCANHopener to properly prepare the CBCANH for the example. The node ID and baud are usually first things to configure in terms of master settings. To start configuration, click the “Module Configuration” button on the main page of EasyCANHopener.

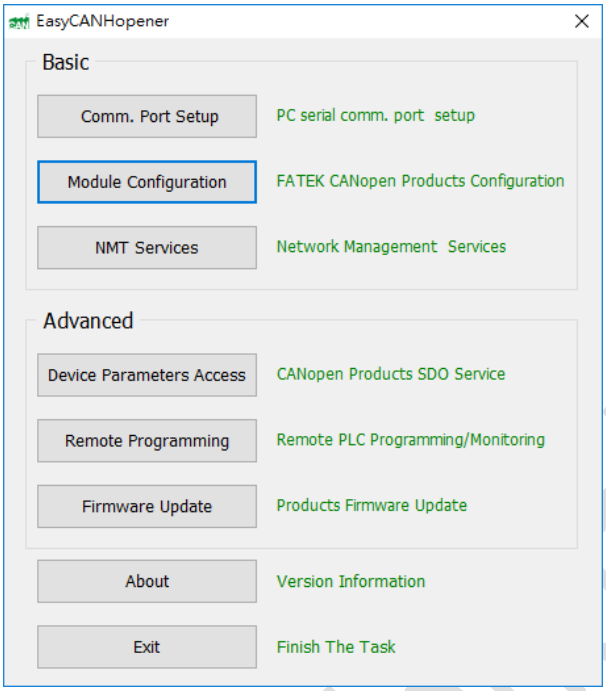



Figure 40 Main menu

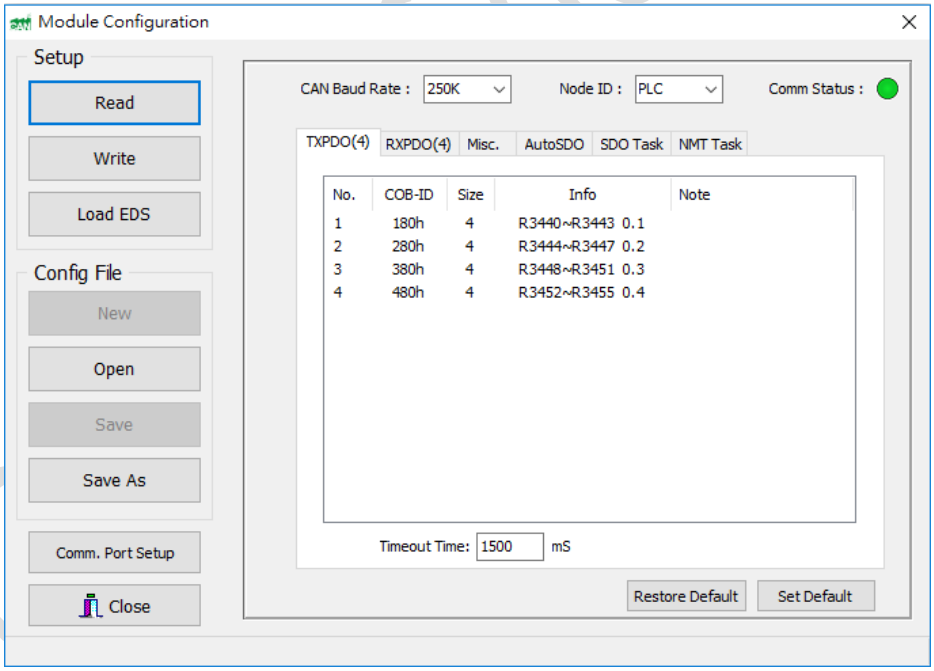


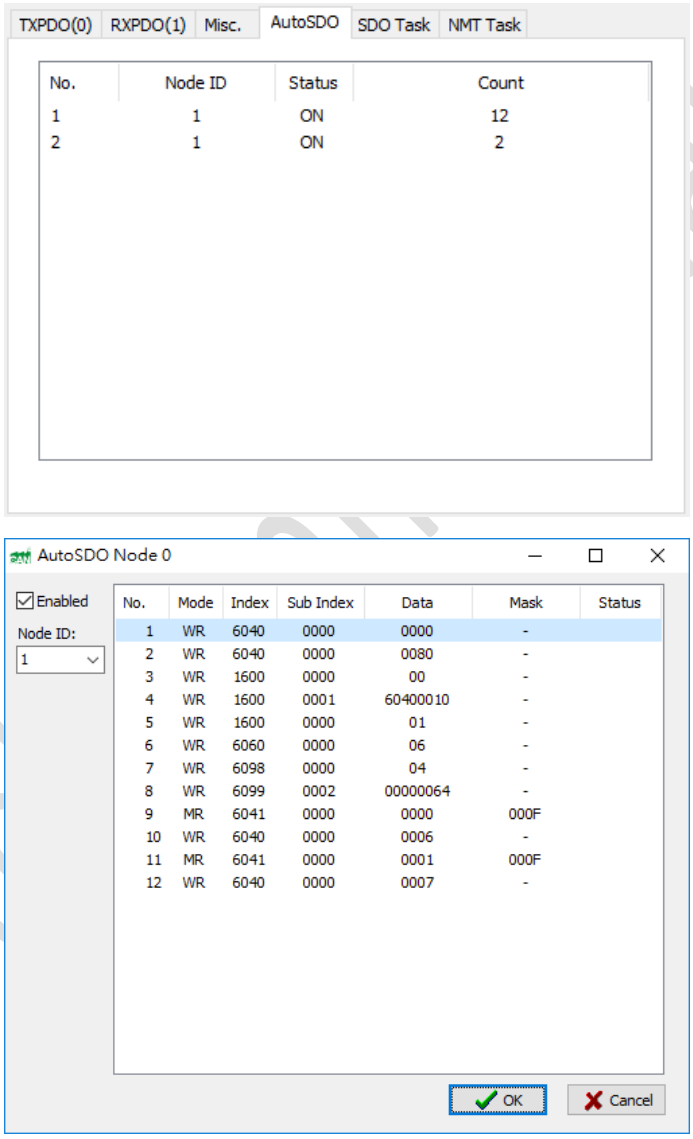

Figure 41 Module configuration

The node ID of the Estun servo drive is 1, thus any other ID except 1 in the range of 1 and 127 can be assigned to CBCANH. Assuming that the FBs PLC uses 8 as the station number, the node ID for the CBCANH can be configured as “PLC” to use the same.


Figure 42 Baud and node ID configuration

**AutoSDO setting**

Configure as shown in the Figure 43



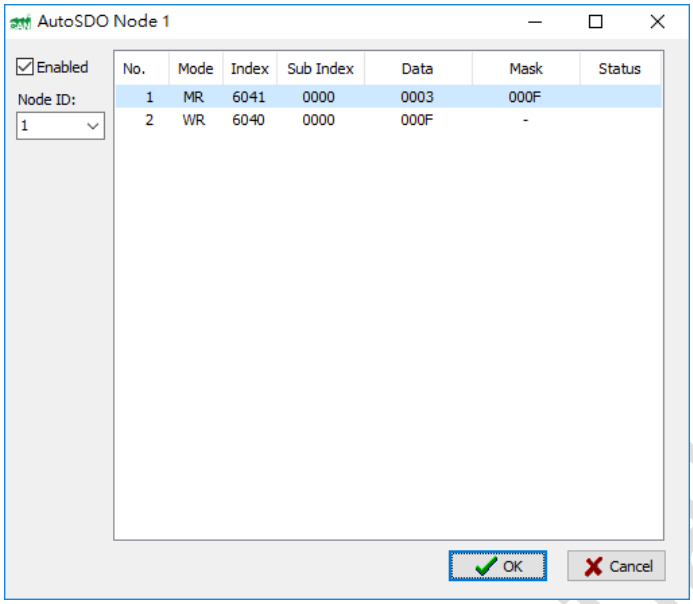



Figure 43 AutoSDO setup

Two AutoSDO groups with 14 SDO operations are created for the Estun servo drive following the homing example given in the CANopen operation manual of Estun servo drive. The node IDs of the groups should be 1.

AutoSDO write operation is as shown in the Figure 44

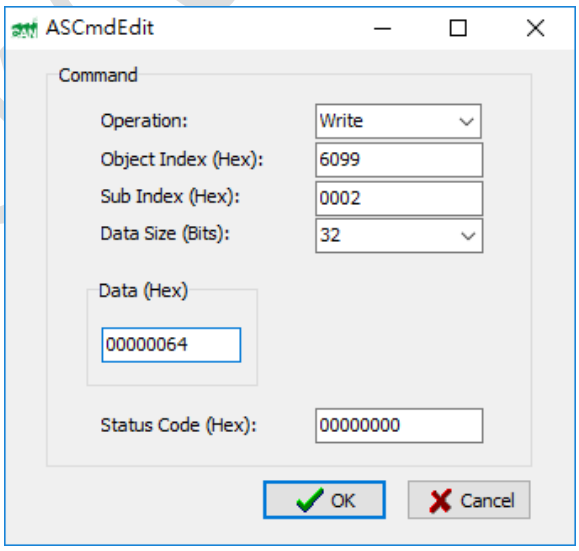



Figure 44 AutoSDO write

WR is a SDO download operation，which writes a given length of data(8/16/32 bits) into a target object dictionary index.

AutoSDO monitor operation is as shown in the Figure 45

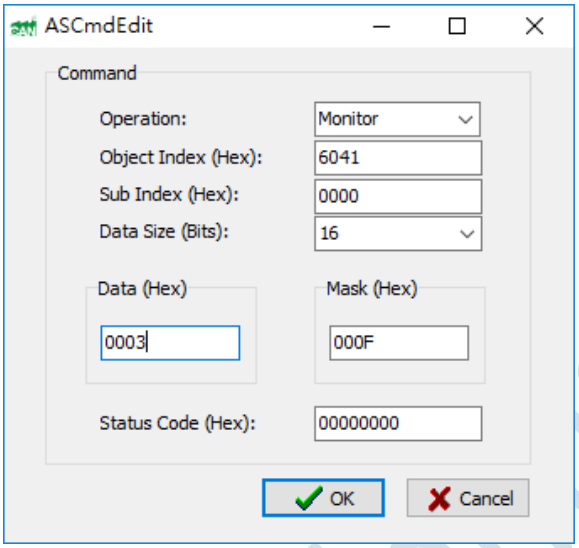



Figure 45 AutoSDO MR

MR is a SDO operation which monitors the value of a specific object dictionary index. MR shows success only if the following equation for a monitoring operation satisfies：

***Data == AND(SDO Upload Data, Mask)***

**RPDO setting**

As shown in the Figure 46, only one RXPDO is required for configuration in the example.

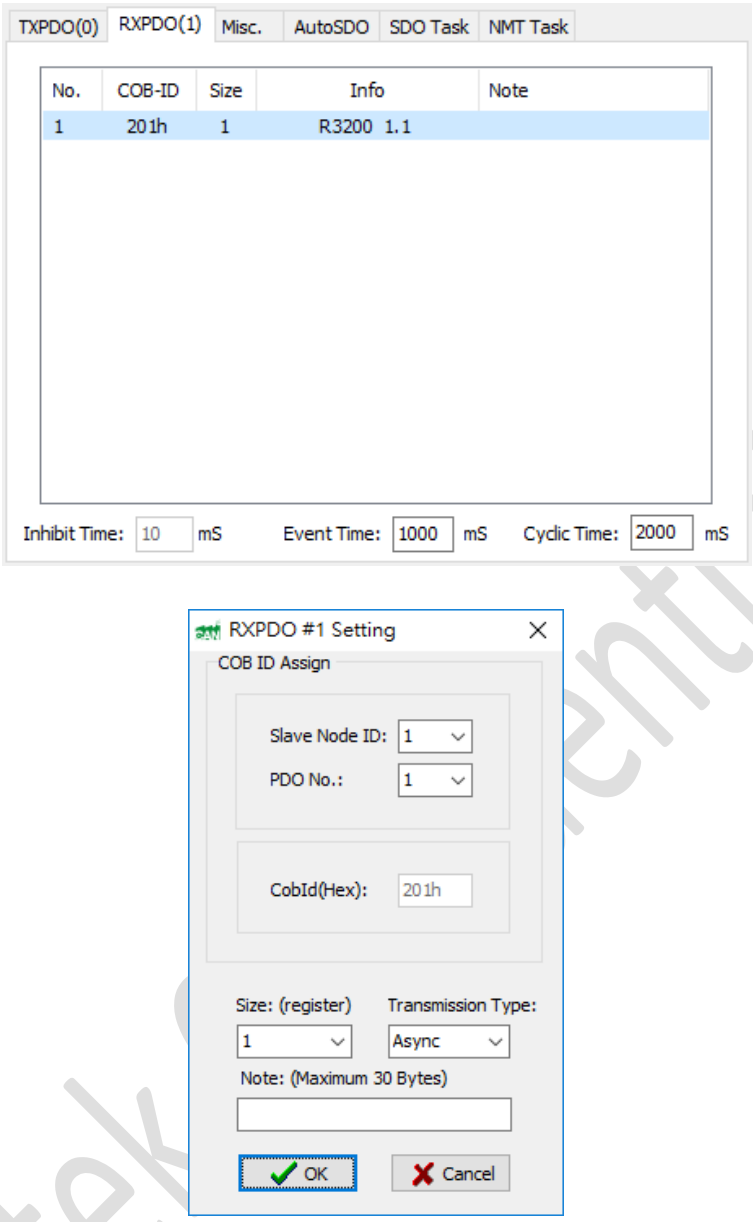



Figure 46 RXPDO configuration

Three additional settings are related to the transmission type and are located at the bottom of RXPDOs list page. “Cyclic Time” is only valid when “Cyclic” transmission type is used. “Inhibit Time” and “Event Time” are two transmission parameters for “Async” type. In this application, there is no requirement specified for these two parameters. “Event Time” may use the default settings 1000ms, while “Inhibit Time” is fixed to 10ms and it cannot be changed. The “Event Time” 1000ms which means the consecutive RXPDO messages shall be transmitted within 1000ms even if no event occurs. The setting 10ms for “Inhibit Time” means the consecutive RXPDO messages will

not be transmitted within 10 ms even if events are triggered during that period.

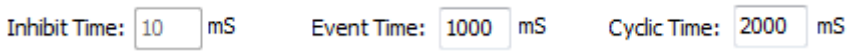



Figure 47 Transmission parameter setting

**Misc. Setting**

On Misc. setting page, there are three main functions “HeartBeat”, “Parameter Zone”, and “Sync. Master”. The “Parameter Zone” configuration is not related to the CANopen network communication and can be disabled (set to 0 to disable) for this application. “Sync Master” can be disabled for it is not used in the example. The “Guard Time” should use a higher value than the cycle time defined in the slave devices. In this example, “Guard Time” is set to the value 1050 ms for monitoring the heartbeat messages on the network.

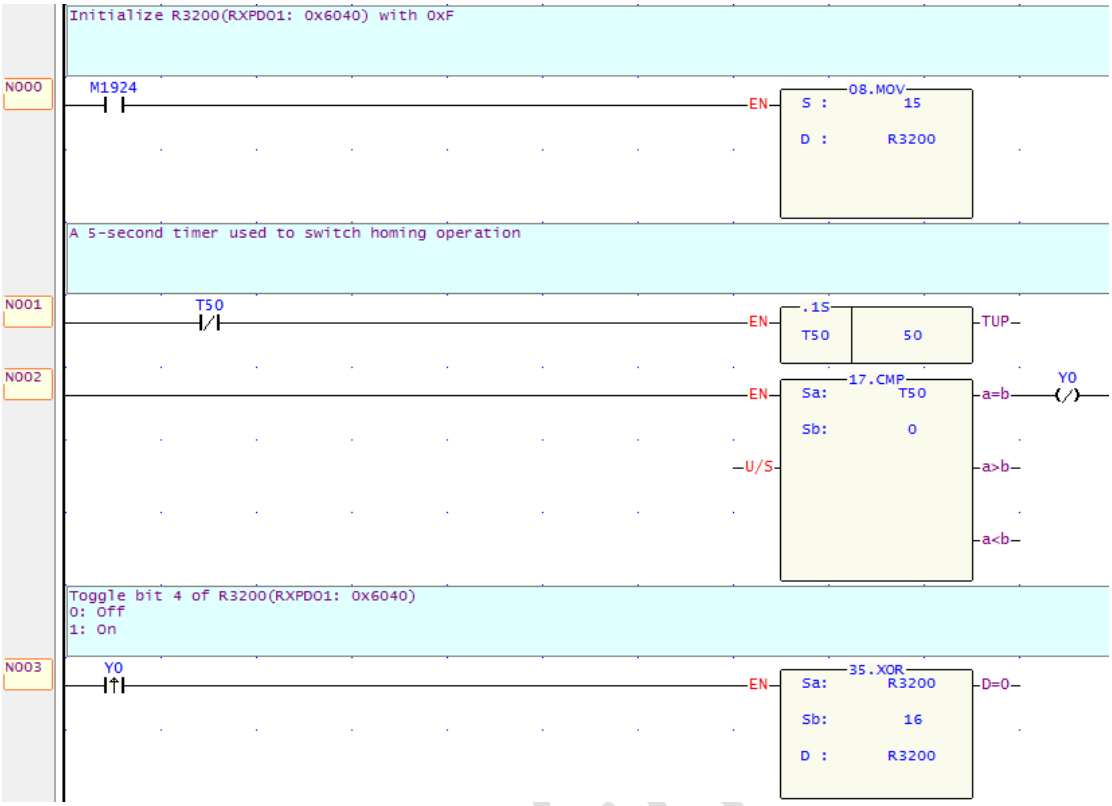
**Write Configuration to CBCANH**

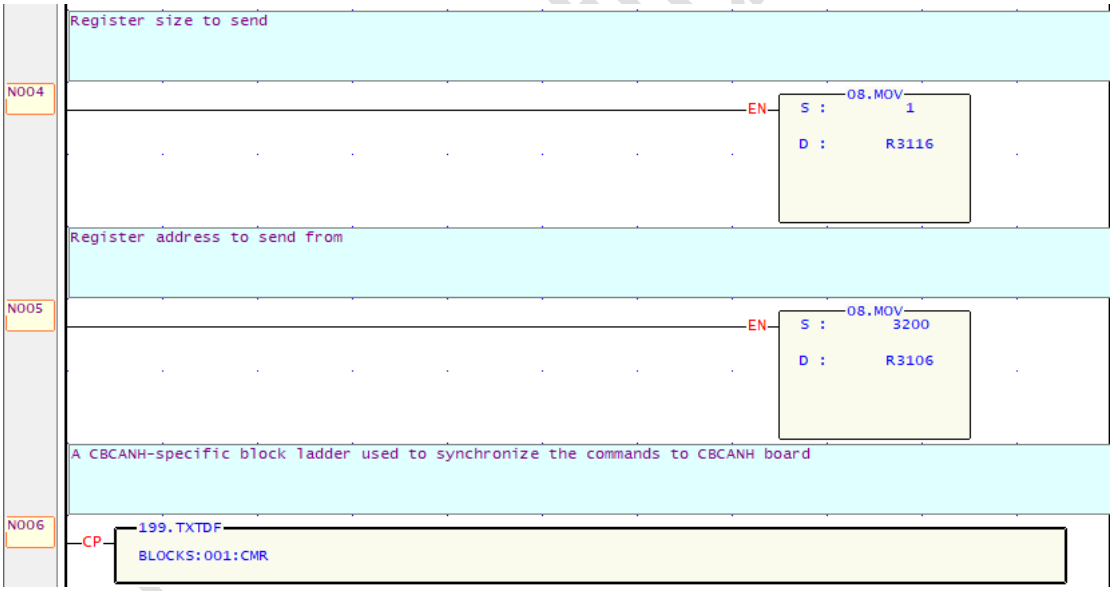
After above settings are done, the configuration then can be downloaded to CBCANH. Click “Write” button to start the procedure.

**PDO <-> PLC register mapping**

After downloading the new configuration to the CBCANH, the CBCANH should be able to communicate with the slaves. The CBCANH supports maximum 60 RXPDOs and 60 TXPDOs, or 15 drive devices. Each PDO supports maximum 8 bytes of data, and mapped to the FBs PLC internal registers. The CBCANH maps the received data of TXPDOs to the internal registers R3440~R3679, and the internal registers R3200~R3439 are mapped to the RXPDOs by using the block ladder CMR. In the example, RXPDO1 is read from R3200.

**Example program**


Initialize R3200(RXPDO1: 0x6040) with 0xF
N000
M1924
EN
08.MOV
S : 15
D : R3200
A 5-second timer used to switch homing operation
N001
T50
EN
.1S
T50
50
TUP
N002
EN
17.CMP
Sa: T50
Sb: 0
U/S
a=b
a>b
a<b
Y0
Toggle bit 4 of R3200(RXPDO1: 0x6040)
0: Off
1: On
N003
Y0
EN
35.XOR
Sa: R3200
Sb: 16
D : R3200
D=0


Register size to send
N004
EN
08.MOV
S : 1
D : R3116
Register address to send from
N005
EN
08.MOV
S : 3200
D : R3106
A CBCANH-specific block ladder used to synchronize the commands to CBCANH board
N006
CP
199.TXTDF
BLOCKS:001:CMR

## Appendix B

### The object dictionary of CBCANH

. Communication Object

| Index | SubIndex | Name | Data Type | Access | Default |
|---|---|---|---|---|---|
| 1000H | 0 | Equipment Type | I32U | R | 0 |
| 1001H | 0 | Error Register | I8U | R | 0 |
| 1005H | 0 | COB-ID of SYNC | I32U | R | 80H |
| 100CH | 0 | Guard Time | I16U | R | 0 |
| 100DH | 0 | Life Time Factor | I8U | R | 0 |
| 1016H | Consumer Heartbeat Time | | | | |
| | 0 | Item Count | I8U | R | 1 |
| | 1 | Consumer Heartbeat Time | I32U | R | 1020 |
| 1017H | 0 | Producer Heartbeat Time | I32U | R | 1000 |
| 1018H | Identity Object | | | | |
| | 0 | Item Count | I8U | R | 4 |
| | 1 | Vendor code | I32U | R | 2EFH |
| | 2 | Product Code | I32U | R | 0 |
| | 3 | Revision No. | I32U | R | 0 |
| | 4 | Serial No. | I32U | R | 0 |
| 1400H | RXPDO1 Communication Parameters | | | | |
| | 0 | Item Count | I8U | R | 2 |
| | 1 | RXPDO1 COB-ID | I32U | R | CFG |
| | 2 | Transmission type | I8U | R | 0xff |
| 1401H~ 143BH | RXPDO2~RXPDO60 Communication Parameters | | | | |
| 1600H | RXPDO1 Mapped Objects | | | | |
| | 0 | Item Count | I8U | R | 4 |
| | 1 | RXPDO1 Mapped Object #1 | I32U | R | CFG |
| | 2 | RXPDO1 Mapped Object #2 | I32U | R | CFG |
| | 3 | RXPDO1 Mapped Object #3 | I32U | R | CFG |
| | 4 | RXPDO1 Mapped Object #4 | I32U | R | CFG |
| 1601H~ 163BH | RXPDO2 ~ RXPDO60 Mapped Objects | | | | |

Communication Object (Cont.)

| Index | SubIndex | Name | Data type | Access | Default |
|---|---|---|---|---|---|
| 1800H | TXPDO1 Communication Parameters | | | | |
| | 0 | Item Count | I8U | R | 5 |
| | 1 | TXPDO1 COB-ID | I32U | R | CFG |
| | 2 | TXPDO1 Transmission Type | I32U | R | 0xff |
| | 3 | TXPDO1 Inhibit Time | I32U | R | CFG |
| | 4 | Reserved | - | - | - |
| | 5 | Event Time | I16U | R | CFG |
| 1801H ~ 183BH | TXPDO2 ~ TXPDO60 Communication Parameters | | | | |
| 1A00H | TXPDO1 Mapped Objects | | | | |
| | 0 | Item Count | I8U | R | 4 |
| | 1 | TXPDO1 Mapped Object #1 | I32U | R | CFG |
| | 2 | TXPDO1 Mapped Object #2 | I32U | R | CFG |
| | 3 | TXPDO1 Mapped Object #3 | I32U | R | CFG |
| | 4 | TXPDO1 Mapped Object #4 | I32U | R | CFG |
| 1A01H ~ 1A3BH | TXPDO2 ~ TXPDO60 Mapped Objects | | | | |

. PLC Parameter Zone Objects (Max. 1000 R register)

| Index | SubIndex | Name | Date type | Access | Default |
|---|---|---|---|---|---|
| 2000H | 1 | P Zone+0 | WORD | RW | |
| 2000H | 2 | P Zone+1 | WORD | RW | |
| 2000H | n | P Zone+n-1 | . | | |
| 2000H | 100 | P Zone+99 | WORD | RW | |
| 2001H | 1 | P Zone+100 | WORD | RW | |
| 2001H | 2 | P Zone+101 | WORD | RW | |
| ~ | ~ | ~ | WORD | RW | |
| 2009H | 100 | P Zone+999 | WORD | RW | |

. PLC Process Data Objects

| Index | SubIndex | Name | Data type | Access | Map |
|---|---|---|---|---|---|
| TXPDO process data | | | | | |
| 2010H | 1 | R3200 | *I16U* | *R* | Yes |
| | 2 | R3201 | | R | |
| | 3 | R3202 | | R | |
| | 4 | R3203 | | R | |

| | | | | | |
|---|---|---|---|---|---|
| 2011H ~ 204BH | R3204~R3439 | | | | |
| RXPDO process data | | | | | |
| 204CH | 1 | R3440 | *I16U* | *R* | Yes |
| | 2 | R3441 | | R | |
| | 3 | R3442 | | R | |
| | 4 | R3443 | | R | |
| 204DH ~ 2087H | R3444~R3679 | | | | |

.Program Version Objects

| Index | SubIndex | Name | Data type | Access | Default |
|---|---|---|---|---|---|
| 4000H | 0 | Item Count | I8U | R | 2 |
| | 1 | CBCANH Firmware version | I32U | R | |
| | 2 | Ladder software version | I32U | R | |

Ladder software version 由 PLC 的 R3697 决定

. Electron board Version Objects

| Index | SubIndex | Name | Data type | Access | Default |
|---|---|---|---|---|---|
| 4001H | 0 | Item Count | I8U | R | 1 |
| | 1 | Electron board version | I32U | R | |

**SDO ERROR CODE**

| Error Code Name | Error Value | Description |
|---|---|---|
| ABORT_TIME_OUT | 0x05040000L | SDO service Time out |
| ABORT_NO_OBJ | 0x06020000L | No such object |
| ABORT_RO | 0x06010002L | Attempt to write a read-only object |
| ABORT_SYS_LENGTH | 0x06040047L | Data length exceed system allow |
| ABORT_NO_SEGEMNT | 0x06010000L | Not support segment transfer |
| ABORT_OBJ_LENGTH | 0x06070010L | Not match object length |
| ABORT_SYNC | 0x05040001L | Command specifier not valid |
| ABORT_TOGGLE_BIT | 0x05030000L | Toggle bit not alternated |
| ABORT_PARM_LENGTH | 0x06070012L | Length of service parameter too high |
| ABORT_WO | 0x06010001L | Attempt to read a write-only object |
| ABORT_READ_LENGTH | 0x05040005L | Object length too big to read |