CAN Bus User Manual

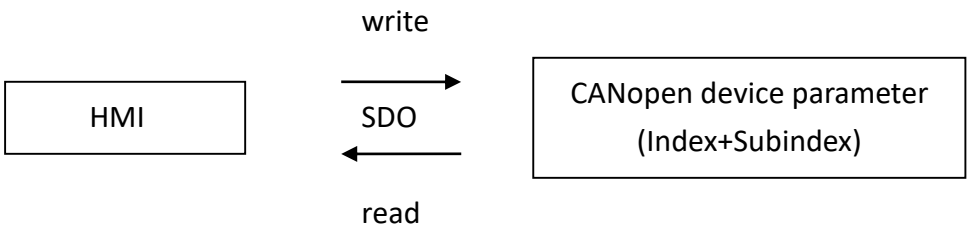
Using CANopen Slave Driver

V1.00

Table of Contents

1. SDO Communication

An important property of CANopen device, Object Dictionary (OD) is a table which stores the configuration and process data for all types of device. For direct access to CANopen devices, Service Data Objects (SDO) can be used. With SDO, Object Dictionary entries can be read and written with their Index + Subindex as defined in the OD. Generally speaking, SDO is used to access the parameters of the CANopen device.



Use SDO to read or write CANopen device parameters.

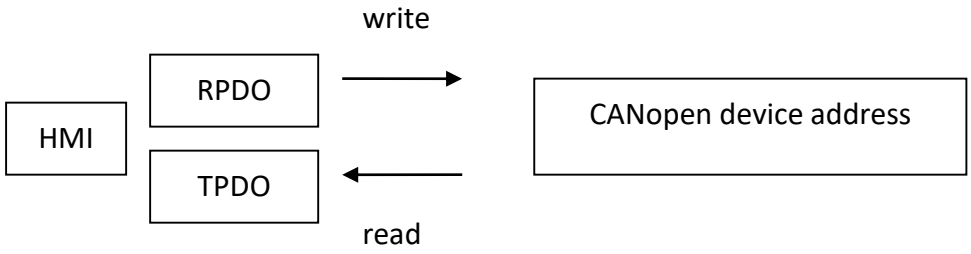
SDO can directly read or write a specific address. In EasyBuilder Pro, to access data by SDO, select the desired data type (8bit, 16bit, or 32 bit), and enter appropriate Index and Subindex values.
For example, to read Index=2200, Subindex=01, 16bit data, please enter 220001 in SDO_16bit device type.

Please Note that Subindex is not a required parameter. When subindex is not specified, use 00.

2. PDO Communication

In CANopen protocol, Process Data Object (PDO) is another way for data exchange. The data to be transmitted or received from its Object Dictionary can be copied into the data field of PDO. In this manner, HMI, by communicating with PDO, can read or write data in Object Dictionary. Generally speaking, PDO is better suited for data that changes dynamically.

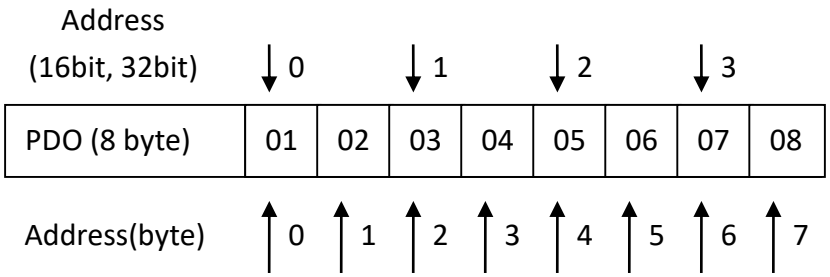
On PLC, PDO is divided into two types: Receive PDO (RPDO) and Transmit PDO (TPDO). However, for HMI, RPDO transfers data to PLC, and TPDO receives data from PLC.



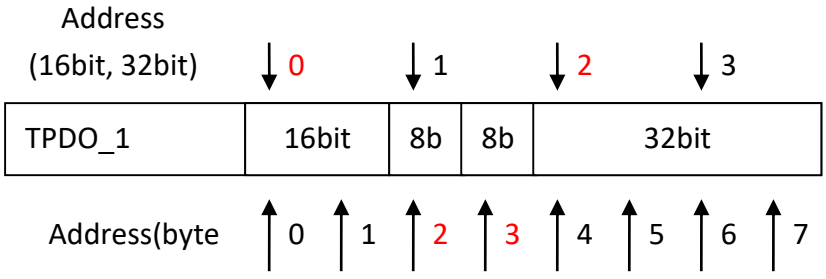
Use PDO to read or write CANopen device parameters.

3. TPDO Reading and RPDO Writing

Each PDO contains 8 bytes of data. To read or write a 16bit or 32bit data, use PDO device type; for byte data, use PDO(Byte) device type. The following figure illustrates the address mapping of PDO data field in EasyBuilder Pro.

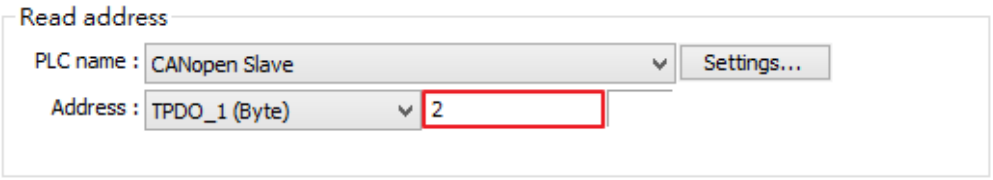


In accessing PDO data, one should select device types and addresses according to PDO data composition. Take TPDO_1, which contains data as follows: 16bit, 8bit, 8bit, and 32bit, as an example as shown in the following figure.



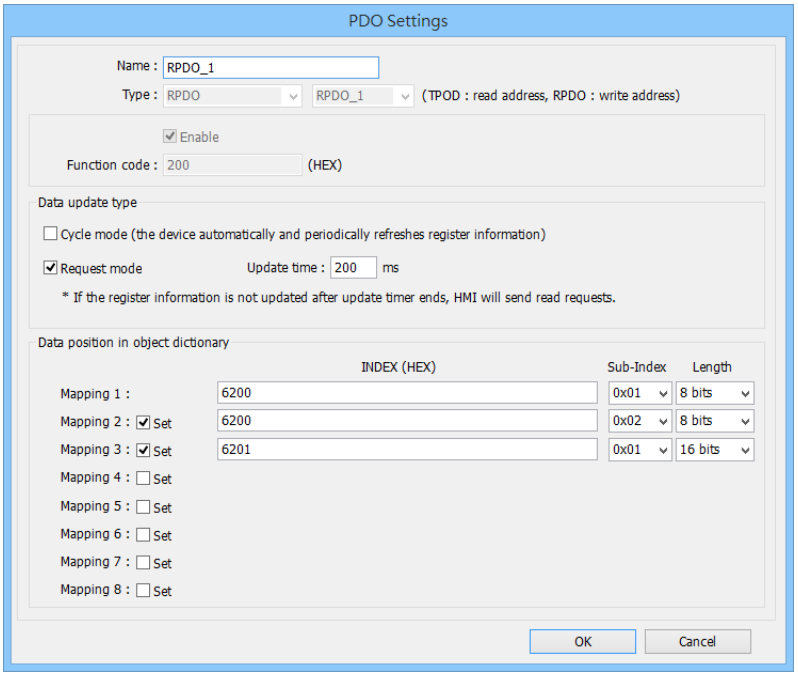
To display the data on HMI, the following device types are used in EasyBuilder Pro:

TPDO_1[0], TPDO_1(Byte)[2], TPDO_1(Byte)[3], TPDO_1[2]. The number enclosed in [ ] indicates the address.

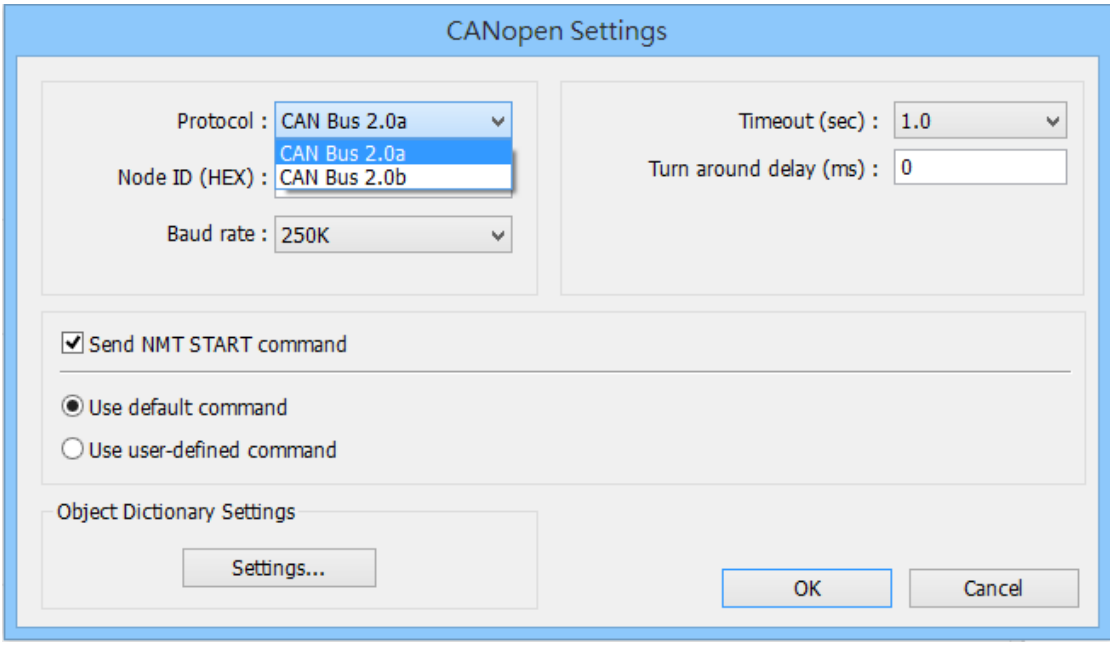


4. RPDO Reading

In order to monitor and write data to RPDO address on HMI more conveniently (for example, when using a Numeric object), in EasyBuilder Pro, RPDO is allowed to read data in Object Dictionary via SDO. Please define the data mapping in Object Dictionary in RPDO's setting page. The HMI will display the data in RPDO via SDO read process. In the RPDO's setting, the mapping should include: Index, Subindex and data length, as shown in the following figure.



5. CANopen Communication Parameter Settings



General Parameters:

Protocol & Node ID

- CAN Bus 2.0a:

  Node ID (range: 1 - 0xFF). The command sent from TPDO/RPDO is selected by Function Code + Node ID.

  For example, TPDO1:

  Function Code = 0x180

  Node ID = 0x01

  The command sent to read TPDO1 would be: 0x180 + 0x01 = 0x181

- CAN Bus 2.0b:

  The command sent from TPDO/RPDO is selected by Function Code + Node ID.

  For example, 0x10**00**1800 + 0x01 = 0x10**01**1800

Baud rate

Set to the same baud rate as the CANopen device.

Communication Parameters:

Send NMT START command

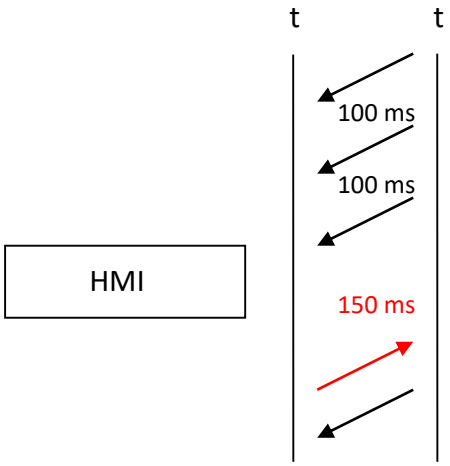
Initializes CANopen network. The device will enter operational state.

Update time and Cycle communication mode (TPDO)

- Update time:

When using a device that sends signals periodically, if no new data has been received

by the HMI within the Update Time, the HMI will send a read command.



The device sends a message every 100 ms. If the update time is set to 150 ms, and the message is not sent when reaching 150 ms, the HMI will send a read request.

- Cycle communication mode:

When using a device that sends out signals periodically and doesn't accept HMI command, please select this option. The HMI will not send read requests.