I-7565-CPM Intelligent USB/CANopen Master Converter

User's Manual

Tables of Content

General Information

1.1. CANopen Introduction

The CAN (Controller Area Network) is a serial communication protocol, which efficiently supports distributed real-time control with a very high level of security. It is an especially suited for networking "intelligent" devices as well as sensors and actuators within a system or sub-system. In CAN networks, there is no addressing of subscribers or stations in the conventional sense, but instead, prioritized messages are transmitted. CANopen is one kind of the network protocols based on the CAN bus and it is applied in a low level network that provides connections between simple industrial devices (sensors, actuators) and higher-level devices (controllers), as shown in Figure 1.1.

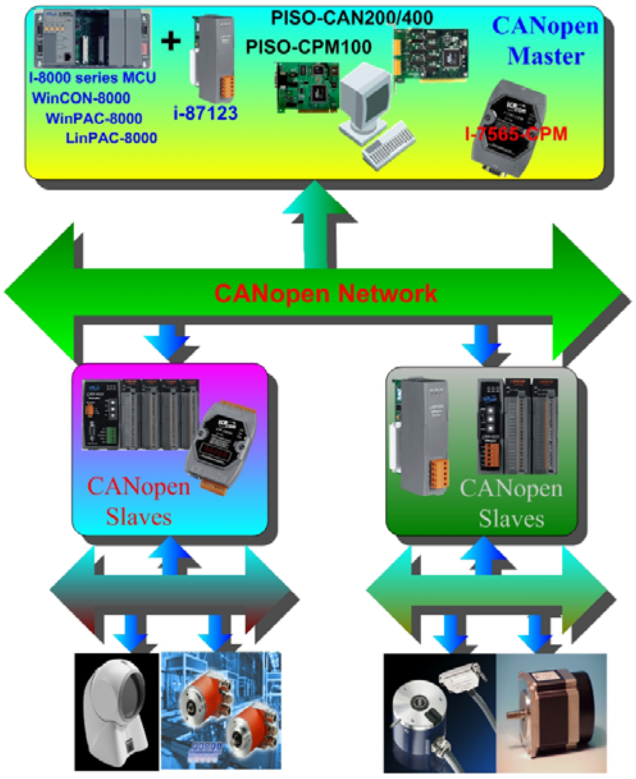


Figure 1.1 Example of the CANopen network

CANopen was developed as a standardized embedded network with highly flexible configuration capabilities. It provides standardized communication objects for real-time data (Process Data Objects, PDO), configuration data (Service Data Objects, SDO), network management data (NMT message, and Error Control), and special functions (Time Stamp, Sync message, and Emergency message). Nowadays, CANopen is used in many various application fields, such as medical equipment, off-road vehicles, maritime electronics, public transportation, building automation and so on.

1.2. CANopen Applications

CANopen is the standardized network application layer optimized for embedded networks. Its specifications cover the standardized application layer, frameworks for the various applications (e.g. general I/O, motion control system, maritime electronics and so forth) as well as device, interface, and application profiles.

The main CANopen protocol and products are generally applied in the low-volume and mid-volume embedded systems. The following examples show some parts of the CANopen application fields. (For more information, please refer to the web site, http://www.can-cia.org):

- Truck-based superstructure control systems
- Off-highway and off-road vehicles
- Passenger and cargo trains
- Maritime electronics
- Factory automation
- Industrial machine control
- Lifts and escalators
- Building automation
- Medical equipment and devices
- Non-industrial control
- Non-industrial equipment

1.3. I-7565-CPM Library Characteristics

In order to use the I-7565-CPM, we provide I-7565-CPM library. Users can use it to establish a CANopen communication network rapidly. Most of the CANopen communication objects, such as PDO, SDO and NMT, will be handled by the library function automatically. Therefore, it is helpful to reduce the complexity of developing a CANopen master interface, and let users ignore the detail CANopen protocol technology. This library mainly supports connection sets of master-slave architecture, which include some useful functions to control the CANopen slave devices in the CANopen network. The following figure describes the general application architecture of I-7565-CPM.

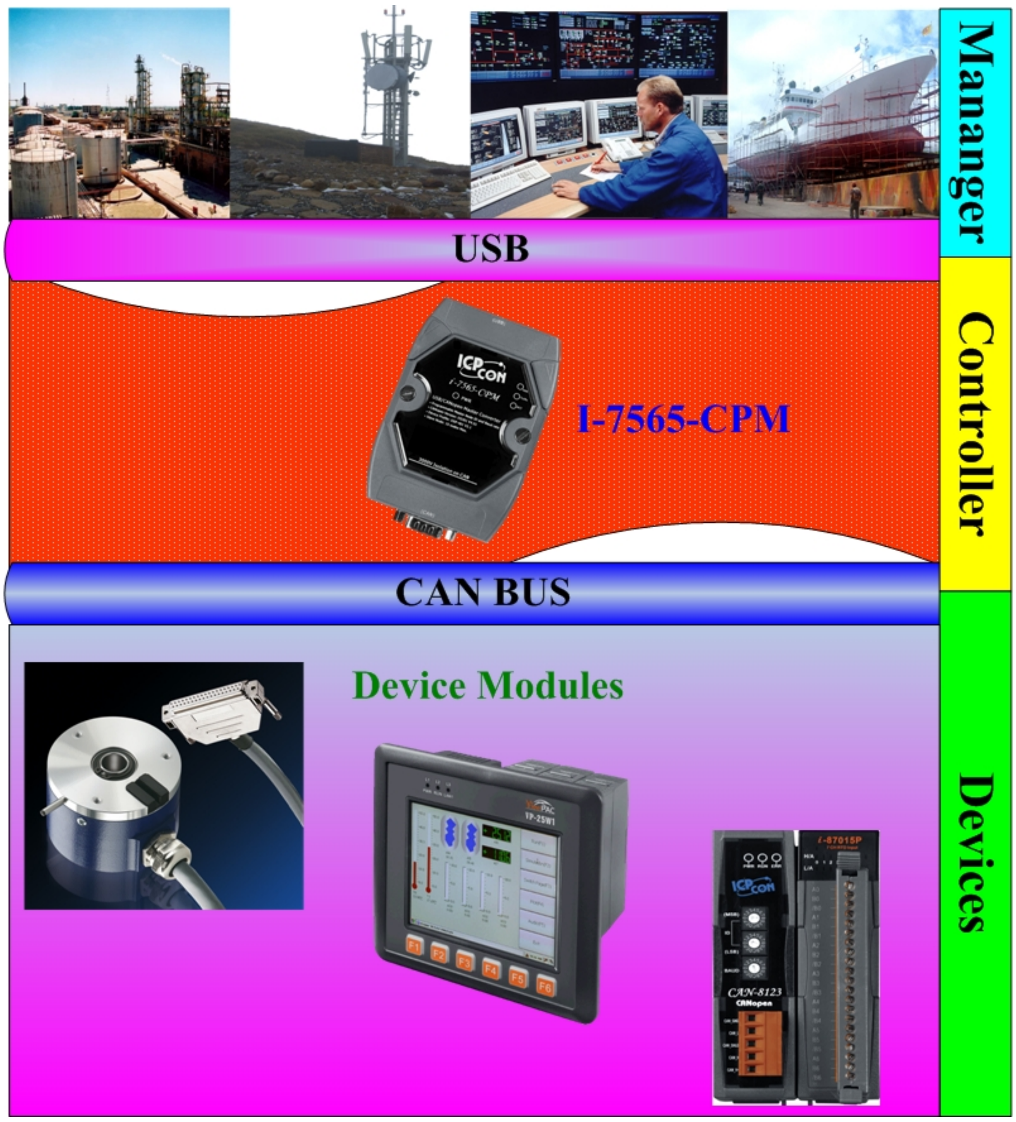


Figure 1.2 Application architecture

I-7565-CPM follows the CiA CANopen specification DS-301 V4.01, and supports the several CANopen features. The CANopen communication general concept is shown as Figure 1.3.

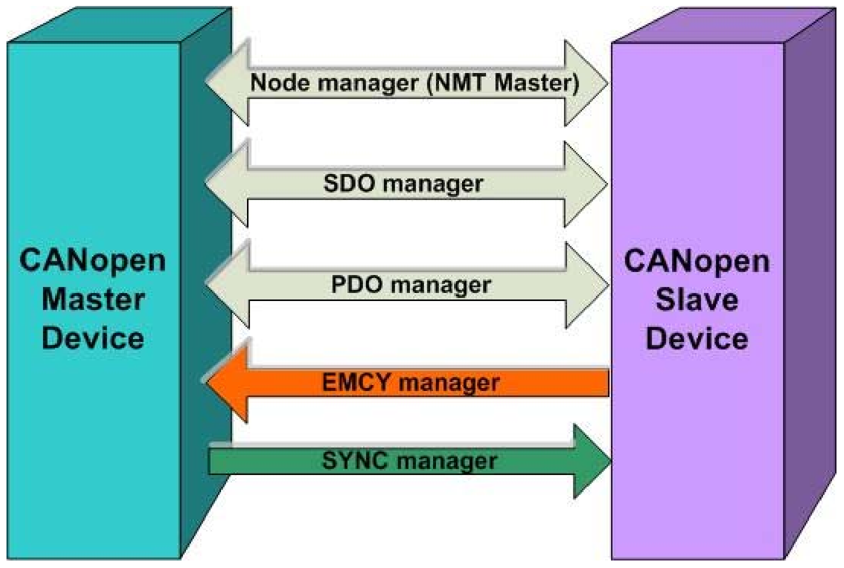


Figure 1.3 CANopen communication general concept

- **Node Manager (NMT Master)**
  - Functions for changing the slave device state
  - Node Guarding protocol for error control
  - Support Emergency (EMCY) messages
- **SDO Manager**
  - Expedited, segmented and block methods for SDO download and upload
- **PDO Manager**
  - Support all transmission types and event timer
- **SYNC Manager**
  - SYNC message production
  - SYNC cycles of 1ms resolution
- **EMCY Manager**
  - EMCY message consumer

For details about the CANopen functions described above, please refer to the function descriptions and demo programs shown in the chapter 3 and chapter 4.

Specifications

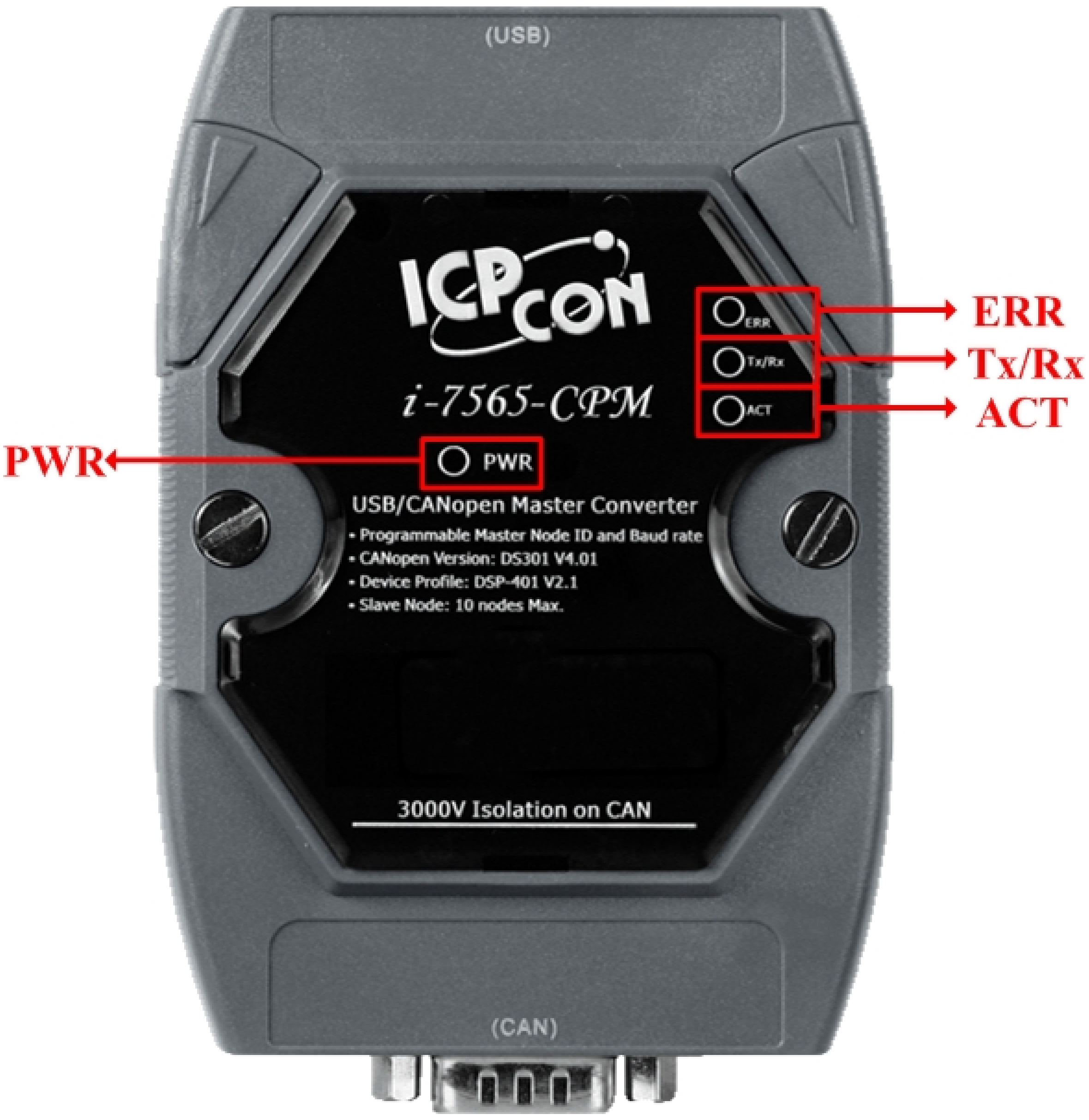
- CAN controller: Philip SJA1000T.
- CAN transceiver: Philip 82C250.
- Signal support: CAN_H, CAN_L.
- Microcontroller: 80186, 80M Hz CPU.
- Connector: 9-pin D-sub male connector.
- USB interface connector: USB Type B.
- USB Specification USB1.1 and USE 2.0.
- PWR LED, ACT LED, Tx/Rx LED, Error LED
- 120Ω terminal resister selected by jumper
- 2500 Vrms photo-isolation protection on CAN side
- 3000 Vrms galvanic DC/DC isolation on CAN side.
- 512 k bytes Flash memory.
- 512k bytes SARM.
- 16K EEPROM.
- Power Supply: By USB interface.
- Power Consumption: 3W
- Operating Temperature: -25℃ to +75℃
- Storage Temperature: -30℃ to +85℃
- Humidity: 5% ~ 95%
- Dimensions: 72mm x 101mm x 33mm (W x L x H).

Features

- Follow CiA DS-301 V4.01 and DSP-401 v2.1.
- Support maximum nodes up to 10
- Support 8 kinds baud: 10Kbps, 20Kbps, 50Kbps, 125Kbps, 250Kbps, 500Kbps, 800Kbps, and 1Mbps
- Support Node Guarding protocol
- Four indication LEDs (Pwr, Tx/Rx, Act and Err LEDs)
- Provide demos and utility
- Support all transmission types and SYNC message.
- Provide VC++ function libraries to process CANopen message

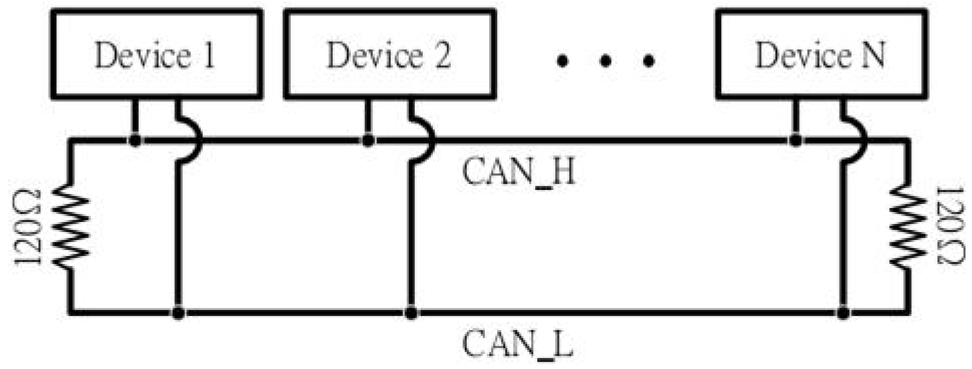
Hardware Specification

2.1. Hardware Structure


(USB)
ICPCON
i-7565-CPM
ERR
Tx/Rx
ACT
PWR
USB/CANopen Master Converter
• Programmable Master Node ID and Baud rate
• CANopen Version: DS301 V4.01
• Device Profile: DSP-401 V2.1
• Slave Node: 10 nodes Max.
3000V Isolation on CAN
(CAN)

2.2. Wire Connection

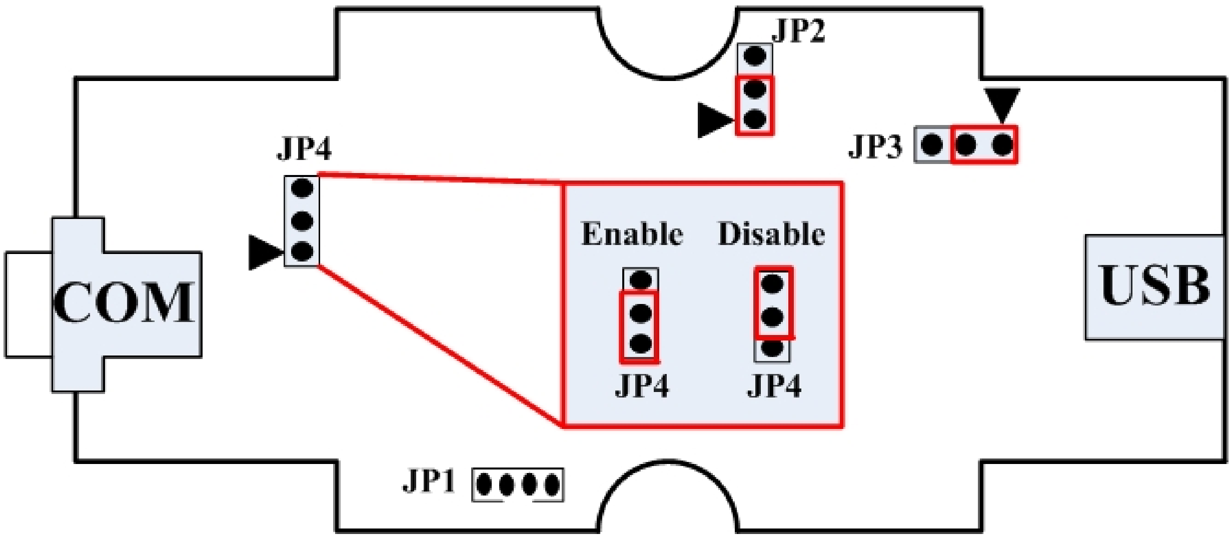
In order to minimize the reflection effects on the CAN bus line, the CAN bus line has to be terminated at both ends by two terminal resistances as in the following figure. According to the ISO 11898-2 spec, each terminal resistance is 120Ω(or between 108 Ω~132Ω). The length related resistance should have 70mΩ/m. Users should check the resistances of the CAN bus, before they install a new CAN network.



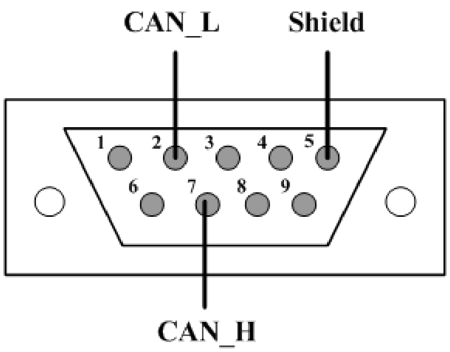
Moreover, to minimize the voltage drop over long distances, the terminal resistance should be higher than the value defined in the ISO 11898-2. The following table can be used as a good reference.

Bus Length (meter)	Bus Cable Parameters		Terminal Resistance (Ω)
	Length Related Resistance (mΩ/m)	Cross Section (Type)	
0~40	70	0.25(23AWG)~0.34 mm^2 (22AWG)	124 (0.1%)
40~300	<60	0.34(22AWG)~0.6 mm^2 (20AWG)	127 (0.1%)
300~600	<40	0.5~0.6mm^2 (20AWG)	150~300
600~1K	<20	0.75~0.8mm^2 (18AWG)	150~300

In the I-7565-CPM, there is a jumper for 120Ω terminal resistance. The JP4 of I-7565-CPM is for the terminal resistance. Its location is shown in the following figure



I-7565-CPM is equipped with one **9-pin D-sub male connector** for wire connection of the CAN bus. The connector's pin assignment is specified as follows.



9-pin D-sub male connector

Pin No.	Signal	Description
1	N/A	Non-available
2	CAN_L	CAN_L bus line (dominant low)
3	N/A	Non-available
4	N/A	Non-available
5	CAN_SHLD	Optional CAN Shield
6	N/A	Non-available
7	CAN_H	CAN_H bus line (dominant high)
8	N/A	Non-available
9	N/A	Non-available

Pin assignment of the 9-pin D-sub male connector

2.3. ACT LED

If the I-7565-CPM is running normally, the ACT LED will be turned on always. This LED is off, please check the power supply or contact to your distributor.

2.4. Tx/Rx LED

Each I-7565-CPM provides Tx/Rx LED to check the status of the transmission and reception of CAN messages. If I-7565-CPM is transmitting or receiving a CAN message, the Tx/Rx LED will blink. If I-7565-CPM's loading is heavy, the Tx/Rx LED will always turn on.

2.5. ERR LED

The ERR LED indicates the error status of the CAN physical layer. It also indicates the errors due to missing CAN messages.

2.6. PWR LED

The power consumption of I-7565-CPM is 3W. If the power is given normally, the ACT LED will be turned on always. This LED is off, please check the power supply or contact to your distributor.

I-7565-CPM Function Library

3.1. Function List

In order to use I-7565-CPM more easily, we provide some useful and easy-to-use functions in I-7565-CPM library. Users can control the I-7565-CPM by using these functions. The following table lists the all functions provided by the I-7565-CPM library.

Function Name	Description
I7565C_DLL_ver	Get the version of the I-7565-CPM library
I7565C_OpenCom	Open the PC Com port to connect the PC with I-7565-CPM
I7565C_Configure	Setting the CAN baud rate and activate the I-7565-CPM
I7565C_AddNode	Add a CANopen slave into I-7565-CPM management list. Afterwards, users can control this slave by using I-7565-CPM
I7565C_ShutdownMaster	Remove all nodes and stop I-7565-CPM
I7565C_CloseCom	Close the com port of PC which is opened by using function I7565C_OpenCom
I7565C_RemoveNode	Remove a CANopen slave from I-7565-CPM management list
I7565C_ChangeState	Change the NMT state of a CANopen node
I7565C_GetState	Get the NMT state of a CANopen node
I7565C_Guarding	Start the node guarding mechanism
I7565C_SendSYNC	Send SYNC message cyclically
I7565C_AbortSDO	Send SDO abort message
I7565C_ReadSDO	Read data by upload SDO protocol
I7565C_WriteSDO	Write data by download SDO protocol
I7565C_InstallPDO	Install and enable a specified PDO object. Afterwards, users can access the data of PDO
I7565C_SetPDOResponse	Set the PDO data response mechanisms
I7565C_RemovePDO	Remove a specified PDO object
I7565C_WritePDO	Use PDO to write data to CANopen node
I7565C_RemotePDO	Use PDO to get data from CANopen node
I7565C_PDOTxType	Set transmission type of a specified TxPDO
I7565C_ChangeSYNCID	Change SYNC COB-ID

I7565C_ChangeEMCYID	Change EMCY COB-ID
I7565C_WriteDO	Output 8 bits DO value
I7565C_WriteAO	Output one AO channel
I7565C_ReadDI	Read 8 bits DI value
I7565C_ReadAI	Read one AI channel
I7565C_ScanAllNode	Scan all CANopen slaves which live in the CANopen network
I7565C_AddPDOPolling	Setting a polling PDO. Afterwards, the PDO can be polled automatically by I-7565-CPM.
I7565C_AddSDOPolling	Setting a polling SDO. Afterwards, the SDO can be polled automatically by I-7565-CPM.
I7565C_DelPDOPolling	Cancel a specified PDO polling mechanism
I7565C_DelSDOPolling	Cancel a specified SDO polling mechanism
I7565C_GetPollingNodeSDO	Get index and sub-index of a polling SDO
I7565C_GetPollingPDOList	Get the ID list of all polling PDO
I7565C_GetAllUsefulCOBID	Get all of the useful COBID of a CANopen node
I7565C_SavePollingPara	Save all polling information into I-7565-CPM module’s EEPROM
I7565C_LoadPollingPara	Load all polling information from I-7565-CPM module’s EEPROM.
I7565C_ReceiveThread	Start the thread to receive all COM port’s data from I-7565-CPM into buffer
I7565C_GetPDOInfo	Get mapping information of a specified PDO
I7565C_GetPDOPollingTime	Get polling time information of a PDO
I7565C_GetSDOPollingTime	Get polling time information of a SDO
I7565C_GetEMCYInfo	Get EMCY message
I7565C_GetResponseValuOfPDOPolling	Get the PDO data if the PDO had be polled.
I7565C_GetResponseValuOfSDOPolling	Get the SDO data if the SDO had be polled.
I7565C_Heartbeat	Set Heartbeat Producer Time and Heartbeat Consumer Time

Table 3.1 Description of functions

3.2. Function Return Code

The following table interprets all the return code returned by the CANopen master library functions.

Return Code	Error ID	Description
0	I7565C_OK	OK
126	I7565C_Port_Open	PC COM port had be opened
127	I7565C_No_Port	No PC COM port is opened
128	I7565C_PortClose_ERR	Close PC COM port failure
129	I7565C_SetBaud_ERR	Setting the CAN baud rate of I-7565-CPM fails
130	I7565C_ AddNode_ERR	Adding node to I-7565-CPM fails
131	I7565C_RemoveNode_ERR	Removing node from I-7565-CPM module fails
132	I7565C_Parameters_ERR	The input parameters of the function is wrong
133	I7565C_ChangeState_ERR	Changing NMT status of a node fails
134	I7565C_NODENUMBER_ERR	The number of node is wrong
135	I7565C_GUARD_FAILED	Node guarding configuration is fails
136	I7565C_SetPDOResponse_ERR	PDO response mode configuration fails
137	I7565C_SendSYNC_ERR	SYNC time configuration fails
138	I7565C_GetState_ERR	Getting node NMT status fails
139	I7565C_ReadSDO_ERR	Sending a SDO to read a specified object fails
140	I7565C_WriteSDO_ERR	Sending a SDO to write a specified object fails
141	I7565C_AbortSDO_ERR	Aborting SDO message fails
142	I7565C_ChangeSYNCID_ERR	Change SYNC COBID failure
143	I7565C_InstallPDO _ERR	Install or modify PDO content error
144	I7565C_DATALEN_ERR	Data length is erroneous
145	I7565C_RemotePDO_ERR	Remove PDO error of node
146	I7565C_WritePDO_ERR	Writing PDO data error
147	I7565C_CHANNEL_ERR	This I/O channel isn't exist
148	I7565C_ScanAllNode_ERR	Scanning all CANopen slave nodes of CAN bus error
149	I7565C_AddPDOPolling_ERR	Set PDO Polling time error
150	I7565C_AddSDOPolling_ERR	Set SDO Polling time error
151	I7565C_DelPDOPolling_ERR	Delete PDO Polling time error
152	I7565C_GetPollingNodeSDO_ERR	Get the SDO polling time failure of node
153	I7565C_GetPollingPDOList _ERR	Fail in getting all polled PDO objects from

		I-7565-CPM module
154	I7565C_GetAllUsefulCOBID _ERR	Get all useful COBID failure of node
155	I7565C_SavePollingPara_ERR	Save all polling data to I-7565-CPM module’s error
156	I7565C_LoadPollingPara _ERR	Load polling parameters from I-7565-CPM failure
157	I7565C_GetPDOInfo _ERR	Get PDO information failure
158	I7565C_GetPDOPollingTime_ERR	Get PDO polling time failure
159	I7565C_GetSDOPollingTime_ERR	Get SDO polling time failure
160	I7565C_ShutdownMaster_ERR	Shutdown I-7565-CPM module error
161	I7565C_RemovePDO_ERR	Removing PDO failure
162	I7565C_PDOTxType_ERR	Set PDOTxType error
163	I7565C_DelSDOPolling_ERR	Delete SDO Polling time error
164	I7565C_NoEMCYInfo	No EMCY message in software buffer
165	I7565C_NoPDOPollingValue	No PDO message in software buffer during the mode of PDO is set to polling mode.
166	I7565C_NoSDOPollingValue	No SDO message in software buffer during the mode of the SDO is set to polling mode.
167	I7565C_Heartbeat_FAILED	Node Heartbeat configuration is fails.

Table 3.2 Description of return code

3.3. CANopen Master Library Application Flowchart

In this section, it describes the operation procedure about how to use the CANopen master library to build users application. This information is helpful for user to apply the CANopen master library easily. Besides, the CANopen operation principles must be obeyed when build a CANopen master application. For example, if the CANopen node is in the pre-operational status, the PDO communication object is not allowed to use. For more details, please refer to the demo programs in chapter 4.

When users programs apply the CANopen master library functions, the function I7565C_Configure must be called first. This function is used to initialize I-7565-CPM and to configure the CAN port.

After initializing the CAN interface successfully, users need to use the function I7565C_AddNode to install at least one CANopen slave into the I-7565-CPM management list. Then, the I-7565-CPM can control and access this CANopen slave.

If the function I7565C_Configure and I7565C_AddNode has been executed, the communication services (NMT, SYNC, EMCY, SDO, and PDO services) are applied automatically until calling the function I7565C_ShutdownMaster. The function I7565C_ShutdownMaster can stop all processes. It is usually used if users want to stop the I-7565-CPM.

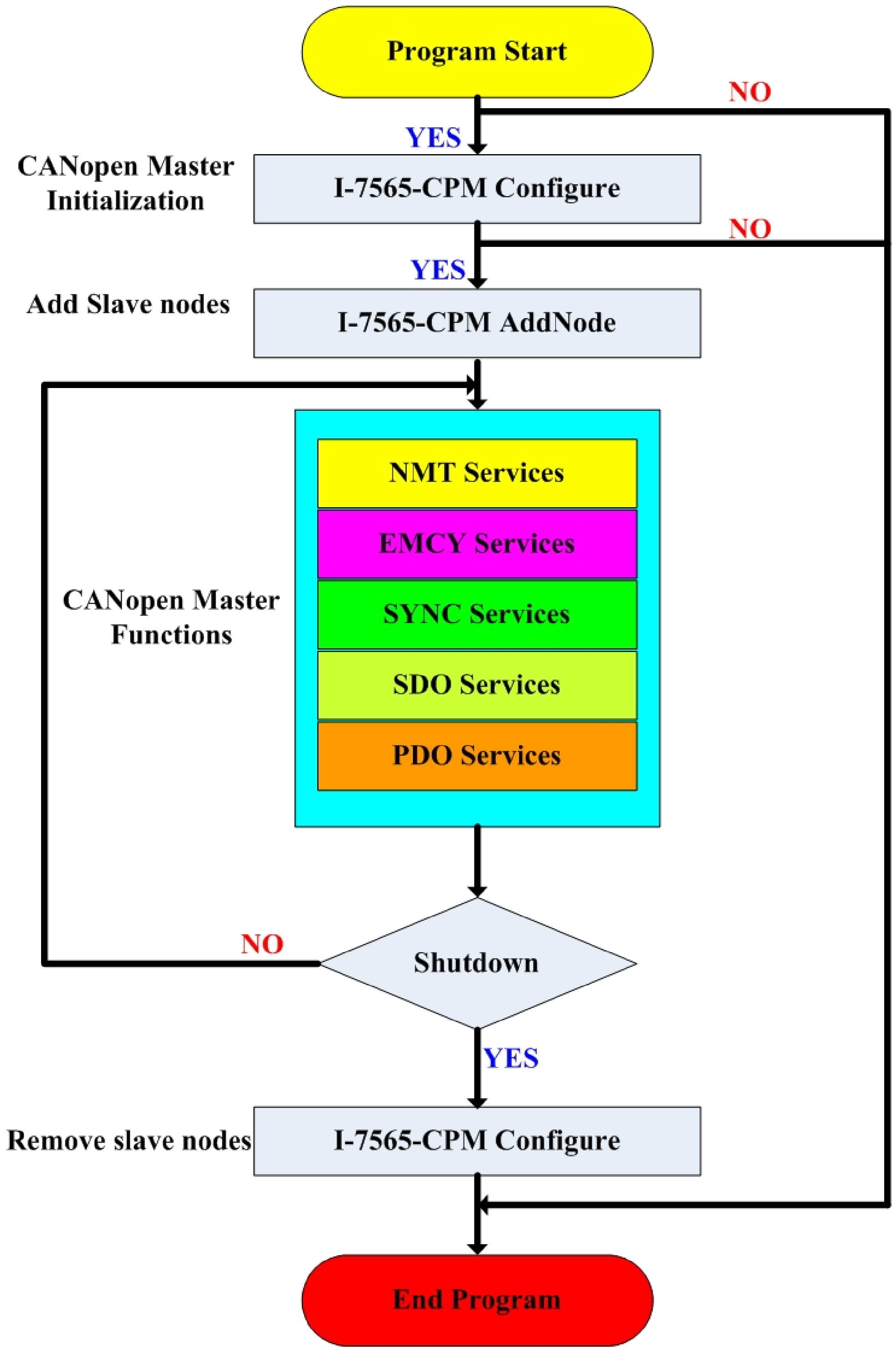


Figure 3.1 Main programming sequences

3.4. Communication Services Introduction

NMT Services

The CANopen master library provides several NMT services functions, such as the functions I7565C_AddNode, I7565C_RemoveNode, I7565C_ChangeState, I7565C_GetState, and I7565C_Guarding. As the prerequisite of the master, the slave nodes have to be registered by the function I7565C_AddNode with providing the slave Node-ID. The registered slave can be individually removed from the management list by the function I7565C_RemoveNode. Through NMT services, the NMT Master controls the state of the slave. Table 3.3 is the command value and corresponding NMT command for the input parameters of the function I7565C_ChangeState. When using the function I7565C_GetState, the slave states and their descriptions are shown in the table 3.4. The Node Guarding protocol is implemented via the function I7565C_Guarding. If the slave nodes are in the management list of I-7565-CPM, users can change the node guarding parameters defined in the slave by calling the function I7565C_Guarding.

Command Value	Description
1 (0x01)	Enter Operational
2 (0x02)	Stop
128 (0x80)	Enter Pre-Operational
129 (0x81)	Reset_Node
130 (0x82)	Reset_Communication

Table 3.3 NMT Command Specifier

State of Slave	Description
4 (0x04)	STOPPED
5 (0x05)	OPERATIONAL
127 (0x7F)	PRE-OPERATIONAL

Table 3.4 The State of The Slave

SDO Services

Initiate SDO download or Initiate SDO upload protocol is used when SDO data length is equal or less than 4 bytes. If the SDO message data length is over 4 bytes, segment SDO download or upload protocol will be used. After calling these two functions, I7565C_ReadSDO and I7565C_WriteSDO, the initiate protocol and segment protocol will be selected automatically according to the SDO data length.

I7565C_AbortSDO function can abort a pending SDO transfer at any time. Applying the abort service will have no confirmation from the salve device.

PDO Services

The function I7565C_InstallPDO is used for setting TPDOs or RPDOs mapping object. Each PDO object supports 0~8 application objects. These application objects defined in the CANopen specification DS401 are mapped to the DI/DO/AI/AO channels. After calling the function I7565C_InstallPDO, the PDO communication object will be mapped and activated. If the PDO communication object is not needed no more, use the function I7565C_RemovePDO to remove it.

The PDOs data are written to the PDO buffer by using the function I7565C_WritePDO. This function can write all PDO 8-bytes data or write only some bytes of PDO data. If users write some bytes of the PDO data, the other part of the PDO data will not be changed. Users can use the function I7565C_SetPDOResponse to change the response type of TPDO. In I7565C_SetPDOResponse function, there are three types, data event, timer event, and remote only to set, and, the data event is the default type of the DI channels, the remote only type is the default type of the AI channels.

In CANopen specification, users can get the TxPDOs data by applying the remote transmit request CAN frame. The function I7565C_RemotePDO is needed in this case. Of course, before using the function I7565C_RemotePDO, the response type needs to be remote only type firstly. Users can use the function I7565C_SetPDOResponse to do so.

SYNC Services

Calling the function I7565C_SendSYNC starts the SYNC object transmission. This function supports single SYNC message and cyclic SYNC message. The timer parameter of the function I7565C_SendSYNC can adjust the cyclic period of SYNC COB-ID sent by master if the cyclically parameter is 1. This timer parameter range is from 0 to 65535ms. If the timer parameter is set to 0, the SYNC object transmission will be stopped. When the cyclically parameter is 0, the function will send single SYNC message.

EMCY Services

Emergency objects are triggered by the occurrence of a device internal error situation. Users can call the function I7565C_RealDatas to receive EMCY message, please reference to the usage of I7565C_RealDatas.

3.5. Function Description

3.5.1. I7565C_DLL_ver

- **Description:**

  This function is used to obtain the version information of I7565C_DLL.lib library.

- **Syntax:**

  char* I7565C_DLL_ver (**void**)

- **Parameter:**

  None

- **Return:**

  Library version information.

3.5.2. I7565C_OpenCom

- **Description:**

The function must be applied when users want to open the PC COM port. If the PC COM port is opened successfully, user can obtain the PC COM port No. by checking the global variable **COM_Open.**

- **Syntax:**

int I7565C_OpenCom(**int** Port_number, char *ModuleName)

Parameter:

Port_number: [input]The PC COM port No. which users want to use. If Port_number is 0, the library will automatically scan all PC COM port and check which COM port is connected with I-7565-CPM. If the value is set from 1 to 255, it means that use the specified port No. (1~255) to open the PC COM port.

***ModuleName:** [output] When the PC has connected with GW-7433D successfully, the *ModuleName parameter will show "!007565CPM".

- **Return:**

I7565C_OK
I7565C_No_Port

3.5.3. I7565C_Configure

- **Description:**

  The function must be applied when configuring the CAN controller and initialize the I-7565-CPM. It must be called once before using other functions of I7565C_DLL.lib.

- **Syntax:**

  int I7565C_Configure(**unsigned char** Baudrate)

  Parameter:

  Baudrate: [input]The baud rate of the I-7565-CPM

Value	Baud rate
0	10 kbps
1	20 kbps
2	50 kbps
3	125 kbps
4	250 kbps
5	500 kbps
6	800 kbps
7	1 Mbps

- **Return:**

  I7565C_OK
  I7565C_SetBaud_ERR
  I7565C_No_Port

3.5.4. I7565C_AddNode

- **Description:**

  The function I7565C_AddNode can add a specified CANopen slave into the I-7565-CPM management list. After calling this function, the slave will be into the operational state directly and the default TxPDO COB-ID(0x180 + node ID, 0x280 + node ID, 0x380 + node ID, 0x480 + node ID)and RxPDO COB-ID(0x200 + node ID, 0x300 + node ID, 0x400 + node ID, 0x500 + node ID) will also be installed if the slave supports them. The added node can be removed from the management list by the function I7565C_RemoveNode*.*

- **Syntax:**

  int I7565C_AddNode(**unsigned char** node, **unsigned char** Timeout)

- **Parameter:**

  node: [input] Slave device Node-ID (1~127).

  Timeout: [input] When user uses this command to add a node into the I-7565-CPM firmware, set this parameter to define the time interval between each two commands sending in the adding node procedure. The unit is millisecond (ms). The range of this value is from 0 to 255 ms.

- **Return:**

  I7565C_OK
  I7565C_AddNode_ERR
  I7565C_TIMEOUT

3.5.5. I7565C_ShutdownMaster

- **Description:**

  The function I7565C_ShutdownMaster removes all the slaves which had added into the I-7565-CPM management list and stops all the functions of I-7565-CPM. The function must be called before exit the users' application programs.

- **Syntax:**

  int I7565C_ShutdownMaster (void)

- **Parameter:**

  None

- **Return:**

  I7565C_OK
  I7565C_ShutdownMaster_ERR
  I7565C_No_Port

3.5.6. I7565C_CloseCom

- **Description:**

  The function I7565C_CloseCom can close the specified PC COM port and release the resources of this PC COM port.

- **Syntax:**

  int I7565C_CloseCom(**int** COMPort)

- **Parameter:**

  COMPort: [input] The number of PC COM port which will be closed. (1~255).

- **Return:**

  I7565C_OK
  I7565C_PortClose_ERR
  I7565C_No_Port

3.5.7. I7565C_RemoveNode

- **Description:**

  The function I7565C_RemoveNode removes the slave with the specified Node-ID from the I-7565-CPM management list. It requires a valid Node-ID, which has installed by the function I7565C_AddNode before.

- **Syntax:**

  int I7565C_RemoveNode(**unsigned char** node)

- **Parameter:**

  node: [input] Slave device Node-ID (1~127).

- **Return:**

  I7565C_OK
  I7565C_RemoveNode_ERR
  I7565C_No_Port

3.5.8. I7565C_ChangeState

- **Description:**

  The function I7565C_ChangeState is used to change the NMT state of a slave. If the node parameter of this function is set to 0, the state of all nodes on the same CAN network will be changed.

- **Syntax:**

  int I7565C_ChangeState(**unsigned char** node,
  unsigned char state)

- **Parameter:**

  node: [input] Slave device Node-ID (1~127). Set this parameter to 0 to indicate all slave devices.
  state: [input] NMT command specifier.
  1: start remote node
  2: stop remote node
  128: enter PRE-OPERATIONAL
  129: Reset_Node
  130: Reset_Communication

- **Return:**

  I7565C_OK
  I7565C_ChangeState_ERR
  I7565C_No_Port

3.5.9. I7565C_GetState

- **Description:**

  The function I7565C_GetState can get the NMT state from the slave.

- **Syntax:**

  int I7565C_GetState(**unsigned char** node,
  unsigned char *state)

- **Parameter:**

  node: [input] Slave device Node-ID (1~127).
  ***state:** [output] The NMT state of node.

4:	STOPPED
5:	OPERATIONAL
127:	PRE-OPERATIONAL

- **Return:**

  I7565C_OK
  I7565C_GetState_ERR
  I7565C_No_Port

3.5.10. I7565C_Guarding

- **Description:**

  Use the function I7565C_Guarding to set Guard Time and Life Time Factor of the specified slave. Then, the master will automatically send the Guarding message to slave device according to the Guard Time parameters of this function. If the master doesn’t receive the confirmation of Guarding message form the salve, I-7565-CPM will send an error message to users. Users need to use I7565C_RealDatas to get this message. However, if the slave doesn’t receive the Guarding message during the node life period (Node life period = Guard Time * Life Time Factor), it will be triggered to send the EMCY message. It is recommended that Life Time Factor is set to more than 1.

- **Syntax:**

```
int I7565C_Guarding(unsigned char node,
                    unsigned short guardtime,
                    unsigned char lifetime)
```

- **Parameter:**

  node: [input] Slave device Node-ID (1~127).
  guardtime: [input] Guard Time (1 ~ 65535 ms).
  lifetime: [input] Life Time Factor (1 ~ 255).

- **Return:**

  I7565C_OK
  I7565C_GUARD_FAILED
  I7565C_No_Port

3.5.11. I7565C_SendSYNC

- **Description:**

  Use the function I7565C_SendSYNC to send a SYNC message with specified COB-ID cyclically. If the timer parameter of this function is 0, the I-7565-CPM will stop to send the SYNC message.

- **Syntax:**

  int I7565C_SendSYNC(**unsigned short** cobid,
  unsigned char cyclically,
  unsigned short timer)

- **Parameter:**

  cobid: [input] COB-ID used by the SYNC object.
  cyclically: [input] If the parameter is 0, the SYNC message will be send once. If the parameter is 1, the SYNC message will be send cyclically with the timer parameter.
  timer: [input] SYNC message transmission period. If the timer is 0, the SYNC transmission will be stopped. The parameter is useless if parameter cyclically is set to 0.

- **Return:**

  I7565C_OK
  I7565C_SendSYNC_ERR
  I7565C_No_Port

3.5.12. I7565C_AobrtSDO

- **Description:**

  Call function I7565C_AbortSDO to cancel the SDO transmission. The parameter node is used to specify which SDO communication will be terminated.

- **Syntax:**

  int I7565C_AbortSDO(**unsigned char** node,
  unsigned short index,
  unsigned char subindex)

- **Parameter:**

  node: [input] Slave device Node-ID (1~127).
  index: [input] The object index value of the object dictionary.
  subindex: [input] The object sub-index value of the object dictionary.

- **Return:**

  I7565C_OK
  I7565C_AbortSDO_ERR
  I7565C_No_Port

3.5.13. I7565C_ReadSDO

- **Description:**

  The function I7565C_ReadSDO is useful to the SDO upload from a specified slave. When users use this function, pass the slave device node ID, object index and sub-index into this function. This function supports both expedition mode and segment mode of SDO protocol.

- **Syntax:**

  int I7565C_ReadSDO(**unsigned char** node,
  unsigned short index
  unsigned char subindex
  unsigned char *len
  unsigned char *rdata)

- **Parameter:**

  node: [input] Slave device Node-ID (1~127).
  index: [input] Object index of object dictionary of slave devices.
  subindex: [input] Object sub-index of object dictionary of slave devices.
  ***len:** [output] Total data length.
  ***rdata:** [output] SDO data respond from the specified slave device.

- **Return:**

  I7565C_OK
  I7565C_ReadSDO_ERR
  I7565C_No_Port

3.5.14. I7565C_WriteSDO

- **Description:**

The function I7565C_WriteSDO can send out a SDO message to specified salve device. This procedure is also called download SDO protocol. The parameter “node” of the function I7565C_WriteSDO is used to point which slave device will receive this SDO message. Because the data length of each object stored in the object dictionary is different, users need to know the data length when writing the object of the object dictionary of the specified slave devices. This function supports both expedition mode and segment mode of SDO protocol.

- **Syntax:**

int I7565C_WriteSDO(**unsigned char** node,
unsigned short index, **unsigned char** subindex,
unsigned char len, **unsigned char** *tdata
unsigned char *rlen, **unsigned char** *rdata)

- **Parameter:**

node: [input] Slave device Node-ID (1~127).
index: [input] The index value of object of the object dictionary.
subIndex: [input] The sub-index value of object of the object dictionary.
len: [input] Total data size to be written.
***tdata:** [input] The SDO data written to slave device.
***rlen:** [input] Total data size of responded data.
***rdata:** [input] SDO data responded from the specified slave device.

- **Return:**

I7565C_OK
I7565C_No_Port
I7565C_NODENUMBER_ERR
I7565C_WriteSDO_ERR

3.5.15. I7565C_InstallPDO

- **Description:**

After calling the I7565C_InstallPDO function, a new PDO COBID will be installed or an old PDO COBID will be modified in the PDO object list of CANopen master library stack. If the slave device has defined the default PDO object, these default PDO will be installed automatically when the function I7565C_AddNode is called.

- **Syntax:**

int I7565C_InstallPDO(**unsigned char** node,
unsigned short cobid,
unsigned char txrxtype,
unsigned char channel,
unsigned char *tdata)

- **Parameter:**

node: [input] Slave device Node-ID (1~127).
cobId: [input] COB-ID used by the PDO object.
txrxtype: [input] PDO type (0: RxPDO, 1: TxPDO).
channel: [input] PDO mapping object sub-index value (1 ~ 8).
***tdata:** [input] 4-byte information of mapped application object. Users need to look up the user manual of CAN slave device to find the information of application object, and obey the following format to fill this parameter.

tdata[0] : The numbers of bit of specified application object.
tdata[1] : The sub-index of specified application object.
tdata[2] : The low byte of index of specified application object.
tdata[3] : The high byte of index of specified application object.

For example, there is an application object defined by some CAN slave device. This application object use index 0x6000 and sub-index 0x06. It is used to store a 16-bit data. When users add this specified application object in the PDO object list of I-7565-CPM, tdata[0] is set to 0x10 (for indicating the stored data to be 16-bit), tdata[1] is 0x06 (for indicating the sub-index to be 0x06), tdata[2] is 0x00 (for indicating the low byte of the index 0x6000), and tdata[3] is 0x60 (for indicating the high byte of the index 0x6000).

- **Return:**

I7565C_OK
I7565C_No_Port
I7565C_NODENUMBER_ERR
I7565C_InstallPDO_ERR

3.5.16. I7565C_SetPDOResponse

- **Description:**

When users set the slave device to some PDO transmission types, the slave will return the TxPDO message automatically if some event is triggered or the event timer is time up. Call the function I7565C_SetPDOResponse to set the PDO message modes. Afterwards, I-7565-CPM will follow the selected PDO message mode to feedback the received PDO message to users.

- **Syntax:**

int I7565C_SetPDOResponse(**unsigned char** node,
unsigned short cobid,
unsigned char mode)

- **Parameter:**

node: [input] Slave device Node-ID (1~127).
cobid: [input] COB-ID used by the PDO object.
mode: [input] PDO response mode.

Mode Value	Description
0	I-7565-CPM will not feedback any received PDO messages sent from slave expect remote-transmit-requist PDO
1	I-7565-CPM will feedback all of received PDO messages to users automatically.

- **Return:**

I7565C_OK
I7565C_NODENUMBER_ERR
I7565C_Parameters_ERR
I7565C_SetPDOResponse_ERR
I7565C_No_Port

3.5.17. I7565C_RemovePDO

- **Description:**

  The function I7565C_RemovePDO can remove a TxPDO or RxPDO which had been installed by the I7565C_InstallPDO. This function can also remove single channel data mapped in TxPDO or RxPDO.

- **Syntax:**

int I7565C_RemovePDO(**unsigned char** node,
unsigned short cobid,
unsigned char txrxtype,
unsigned char channel)

- **Parameter:**

node: [input] Slave device Node-ID (1~127).
cobid: [input] COB-ID used by the PDO object.
txrxtype: [input] PDO type (0: RxPDO, 1: TxPDO).
channel: [input] PDO mapping object sub-index value (0 ~ 8). If channel parameter is 0, the specified PDO object will be removed. If others (1 ~ 8), the specified sub-index content of the PDO will be removed.

Return:

I7565C_OK
I7565C_NODENUMBER_ERR
I7565C_Parameters_ERR
I7565C_RemovePDO_ERR
I7565C_No_Port

3.5.18. I7565C_WritePDO

- **Description:**

Call the function I7565C_WritePDO to send out a PDO message to the specified slave device. Before using this function, users need to use the function I7565C_InstallPDO to install the PDO object if users want to use non-default PDO. Then, change the NMT state of target slave device to operational mode by using the function I7565C_ChangeState if the slave is not in the operational mode. Use the parameter “offset” to set the start byte position of PDO data which need to be modified, and use the parameters “*rdata” and “len” to point the data and data length which users want to fill to the PDO data. The other data of the PDO which don’t be modified by this function will keep the original states.

- **Syntax:**

```
int I7565C_WritePDO(unsigned short cobid,
                    unsigned char offset,
                    unsigned char dlen,
                    unsigned char *rdata)
```

- **Parameter:**

cobid: [input] COB-ID used by the PDO object.
offset: [input] The start byte position of PDO data (0 ~ 7).
dlen: [input] data size (dlen + offset ≦ 8 (total length of the PDO)).
***rdata:** [output] The pointer for storing the PDO data.

- **Return:**

I7565C_OK
I7565C_DATALEN_ERR
I7565C_WritePDO_ERR
I7565C_No_Port

3.5.19. I7565C_RemotePDO

- **Description:**

  Use the function I7565C_RemotePDO to send a RTR (remote-transmit-request) PDO message to the slave device.

- **Syntax:**

```
int I7565C_RemotePDO(unsigned short cobid,
                     unsigned char *len,
                     unsigned char *rdata)
```

- **Parameter:**

  cobid: [input] COB-ID used by the PDO object.
  ***len:** [output] The data length of the RTR PDO message.
  ***rdata:** [output] The pointer for storing the PDO message received from the slave device.

- **Return:**

  I7565C_OK
  I7565C_RemotePDO_ERR
  I7565C_No_Port

3.5.20. I7565C_PDOTxType

- **Description:**

  Use this function to change transmission type of TxPDO. The default transmission type is 255.

- **Syntax:**

  int I7565C_PDOTxType(**unsigned char** node,
  unsigned short cobid
  unsigned char txtype)

- **Parameter:**

  node: [input] Slave device Node-ID (1~127).
  cobid: [input] COB-ID used by the PDO object.
  txtype: [input] Transmission type of TxPDO (0 ~ 255).

- **Return:**

  I7565C_OK
  I7565C_NODENUMBER_ERR
  I7565C_PDOTxType_ERR
  I7565C_No_Port

3.5.21. I7565C_ChangeSYNCID

- **Description:**

  Use the function I7565C_ChangeSYNCID to change the SYNC COB-ID of a specified slave device.

- **Syntax:**

  int I7565C_ ChangeSYNCID (**unsigned char** node,
  unsigned short cobid)

- **Parameter:**

  node: [input] Slave device Node-ID (1~127).
  cobid: [input] COB-ID used by the SYNC object.

- **Return:**

  I7565C_OK
  I7565C_ChangeSYNCID_ERR
  I7565C_No_Port
  I7565C_WriteSDO_ERR

3.5.22. I7565C_ChangeEMCYID

- **Description:**

  Use the function I7565C_ChangeEMCYID to change the EMCY COB-ID of a slave device.

- **Syntax:**

  int I7565C_ ChangeEMCYID (**unsigned char** node,
  unsigned short cobid)

- **Parameter:**

  node: [input] Slave device Node-ID (1~127).
  cobid: [input] COB-ID used by the EMCY object.

- **Return:**

  I7565C_OK
  I7565C_No_Port
  I7565C_WriteSDO_ERR
  I7565C_NODENUMBER_ERR
  I7565C_ReadSDO_ERR

3.5.23. I7565C_WriteDO

- **Description:**

  Use this function to output one byte (8 channels) DO data.

- **Syntax:**

```
int I7565C_WriteDO(unsigned char node,
                   unsigned char dochannel
                   unsigned char value)
```

- **Parameter:**

  node: [input] Slave device Node-ID (1~127).

  dochannel: [input] The sub-index of index 0x6200 of specified application object. Please refer to slave device user manual for more detail information.

  value: [input] The value for 8-channel digital output which is presented as 1 byte.

- **Return:**

  I7565C_OK
  I7565C_CHANNEL_ERR
  I7565C_WriteSDO_ERR
  I7565C_NODENUMBER_ERR
  I7565C_No_Port

3.5.24. I7565C_ReadDI

- **Description:**

  Use this function to read one byte (8 channels) DI data.

- **Syntax:**

```
int I7565C_ReadDI(unsigned char node,
                  unsigned char dichannel
                  unsigned char *value)
```

- **Parameter:**

  node: [input] Slave device Node-ID (1~127).
  dichannel: [input] The sub-index of index 0x6000 of specified application object. Please refer to slave device user manual for more detail information.
  ***value:** [output] The pointer for storing the value for 8-channel digital input which is presented as 1 byte.

- **Return:**

  I7565C_OK
  I7565C_CHANNEL_ERR
  I7565C_ReadSDO_ERR
  I7565C_NODENUMBER_ERR
  I7565C_No_Port

3.5.25. I7565C_WriteAO

- **Description:**

  Use this function to output one channel AO data.

- **Syntax:**

int I7565C_WriteAO(**unsigned char** node,
unsigned char aochannel
unsigned short value)

- **Parameter:**

node: [input] Slave device Node-ID (1~127).
aochannel: [input] The sub-index of index 0x6411 of specified application object. Please refer to slave device user manual for more detail.
value: [input] One 16-bit (or <16-bit) AO channel value which is presented as two bytes.

- **Return:**

I7565C_OK
I7565C_CHANNEL_ERR
I7565C_WriteSDO_ERR
I7565C_NODENUMBER_ERR
I7565C_No_Port

3.5.26. I7565C_ReadAI

- **Description:**

  Use this function to read one channel AI data.

- **Syntax:**

  int I7565C_ReadAI(**unsigned char** node,
  unsigned char aichannel
  unsigned short *value)

- **Parameter:**

  node: [input] Slave device Node-ID (1~127).
  aichannel: [input] The sub-index of index 0x6401 of specified application object. Please refer to slave device user manual for more detail.
  ***value:** [output] Read one 16-bit (or <16-bit) AI channel value which is presented as two bytes.

- **Return:**

  I7565C_OK
  I7565C_CHANNEL_ERR
  I7565C_ReadSDO_ERR
  I7565C_NODENUMBER_ERR
  I7565C_No_Port

3.5.27. I7565C_ScanAllNode

- **Description:**

  Use this function to get how many CANopen slave nodes are in the CANopen network. Users can get the ID number of those CANopen slave nodes.

- **Syntax:**

  int I7565C_ScanAllNode(**int ***NodeCount,
  unsigned char *value)

- **Parameter:**

  ***NodeCount:** [output] The numbers of slave devices
  ***value:** [output] The ID number (1~127) of the slave devices

- **Return:**

  I7565C_OK
  I7565C_ScanAllNode_ERR
  I7565C_No_Port

3.5.28. I7565C_AddPDOPolling

- **Description:**

  Use this function to set which PDO is polled and the PDO polling time. This function is useful for some devices whose PDO don't support event timer or event trigger. Afterwards, the I-7565-CPM will poll this PDO by using the period of the polling time.

- **Syntax:**

  int I7565C_ AddPDOPolling (**unsigned short** cobid,
  unsigned long time)

- **Parameter:**

  cobid: [input] The PDO COBID
  time: [input] The PDO Polling time(1~65535 ms).

- **Return:**

  I7565C_OK
  I7565C_AddPDOPolling_ERR
  I7565C_No_Port

3.5.29. I7565C_AddSDOPolling

- **Description:**

  Use this function to set which SDO be polled and the SDO polling time. Afterwards, the I-7565-CPM will poll this PDO by using the period of the polling time.

- **Syntax:**

```
int I7565C_ AddSDOPolling (unsigned char node,
                           unsigned short index,
                           unsigned char subindex,
                           unsigned long time)
```

- **Parameter:**

  node: [input] The node ID of the node whose SDO will be polled.
  index: [input] The SDO index number.
  subindex: [input] The SDO sub-index number.
  time: [input] The SDO Polling time(1~65535 ms).

- **Return:**

  I7565C_OK
  I7565C_AddSDOPolling_ERR
  I7565C_No_Port

3.5.30. I7565C_DelPDOPolling

- **Description:**

  Use this function to delete the polling mode of the specified PDO CODID.

- **Syntax:**

  int I7565C_ DelPDOPolling (**unsigned short** cobid**)**

- **Parameter:**

  cobid: [input] The PDO COB-ID.

- **Return:**

  I7565C_OK
  I7565C_DelPDOPolling_ERR
  I7565C_No_Port

3.5.31. I7565C_DelSDOPolling

- **Description:**

  Use this function to delete the polling mode of SDO.

- **Syntax:**

  int I7565C_DelSDOPolling (**unsigned char** node,
  unsigned short index**,**
  unsidned char subindex)

- **Parameter:**

  node: [input] The polled SDO of the node ID which want to be deleted.
  index: [input] The index of the polled SDO.
  subindex: [input] The sub-index of the polled SDO

- **Return:**

  I7565C_OK
  I7565C_DelSDOPolling_ERR
  I7565C_No_Port

3.5.32. I7565C_GetPollingNodeSDO

- **Description:**

  Use this function to read all of the polled SDO objects in the I-7565-CPM stack.

- **Syntax:**

  int I7565C_ GetPollingNodeSDO (**unsigned char** node,
  unsigned char *rdata)

- **Parameter:**

  node: [input] The number of the node ID of a specified CANopen slave node.
  *** rdata:** [output] The pointer for storing the list include the information of all the polled SDO objects in the I-7565-CPM stack.

- **Return:**

  I7565C_OK
  I7565C_GetPollingNodeSDO_ERR
  I7565C_No_Port

3.5.33. I7565C_GetPollingPDOList

- **Description:**

  Use this function to read all of the polled PDO objects in the I-7565-CPM stack.

- **Syntax:**

  int I7565C_ GetPollingPDOList (**unsigned char** *rdata)

- **Parameter:**

  *** rdata:** [output] The pointer for storing the list include the information of all the polled PDO objects in the I-7565-CPM stack.

- **Return:**

  I7565C_OK
  I7565C_GetPollingPDOList_ERR
  I7565C_No_Port

3.5.34. I7565C_GetAllUsefulCOBID

- **Description:**

  Use this function to get all valid COB-ID of NMT, SYCN, EMCY, PDO and SDO from a node.

- **Syntax:**

  int I7565C_ GetAllUsefulCOBID (**unsidned char** node,
  unsigned char *rdata)

- **Parameter:**

  node: [input] The number of the node ID of a specified CANopen slave node.
  *** rdata:** [output] The pointer for storing the list include the information of all the COB-IDs of a specified node.

- **Return:**

  I7565C_OK
  I7565C_GetAllUsefulCOBID_ERR
  I7565C_No_Port

3.5.35. I7565C_SavePollingPara

- **Description:**

  Use this function to save the numbers of the CANopen slaves in the I-7565-CPM management list, the numbers of the TxPDOs, the numbers of the polled SDO objects, the COB-ID and polling time of the polled TxPDO, the index and sub-index of the polled SDO objects, and the SDO polling time into EEPROM.

- **Syntax:**

  int I7565C_ SavePollingPara (**void**).

- **Parameter:**

- **Return:**

  I7565C_OK
  I7565C_SavePollingPara_ERR
  I7565C_No_Port

3.5.36. I7565C_LoadPollingPara

- **Description:**

  Use this function to load the parameters which are saved by function I7565C_SavePollingPara. When the I-7565-CMP reboots, users need to call this function to load the previous configuration. If users want to use default configuration, don’t call this function.

- **Syntax:**

  int I7565C_ LoadPollingPara (**void**)

- **Parameter:**

- **Return:**

  I7565C_OK
  I7565C_LoadPollingPara_ERR
  I7565C_No_Port

3.5.37. I7565C_ReceiveThread

- **Description:**

  Use this function to start the reception thread. Afterwards, the CANopen master library will receive the RS-232 data from I-7565-CPM automatically. This function must be called after using the I7565C_OpenCom function.

- **Syntax:**

  int I7565C_ReceiveThread(LPVOID pParam)

- **Parameter:**

  pParam: [output] The numbers of the CANopen slave devices which will be controlled by I-7565-CPM.

- **Return:**

3.5.38. I7565C_GetPDOInfo

- **Description:**

  Use this function to get the transmit PDO mapping configuration of the specified PDO of the I-7565-CPM.

- **Syntax:**

  int I7565C_GetPDOInfo (**unsigned short** cobid,
  unsigned char *rdata)

- **Parameter:**

  cobid: [input] The COB-ID of a PDO.
  *** rdata:** [output] The pointer for storing the transmit PDO mapping configuration information.

- **Return:**

  I7565C_OK
  I7565C_GetPDOInfo_ERR
  I7565C_No_Port

3.5.39. I7565C_GetPDOPollingTime

- **Description:**

  Use this function to receive data of the PDO's polling time.

- **Syntax:**

  int I7565C_GetPDOPollingTime (**unsigned short** cobid,
  unsigned char *rdata)

- **Parameter:**

  cobid: [intput] The COBID of PDO polling time object.
  ***rdata:** [output] Receiving data of the PDO's polling time.

  Return:

  I7565C_OK
  I7565C_GetPDOPollingTime_ERR
  I7565C_No_Port

3.5.40. I7565C_GetSDOPollingTime

- **Description:**

  Use this function to receive data of the SDO's polling time of node.

- **Syntax:**

  int I7565C_ GetSDOPollingTime (**unsigned char** node,
  unsigned short index,
  unsigned char subindex**,**
  unsigned char * rdata)

- **Parameter:**

  node: [input] The number of CANopen slave node.
  index: [input] The object's index of SDO.
  subindex: [input] The object's subindex of SDO.
  ***rdata:** [output] Receiving data of the SDO's polling time (1~65535 ms) of node.

- **Return:**

  I7565C_OK
  I7565C_GetSDOPollingTime_ERR
  I7565C_No_Port

3.5.41. I7565C_ GetEMCYInfo

- **Description:**

  Use this function to get the EMCY message. If there are some EMCY messages which is sent from CANopen slaves and is stored in the software buffer, users can use this function to get it.

- **Syntax:**

  int I7565C_GetEMCYInfo(**unsigned char** *rdata)

- **Parameter:**

  ***rdata:** [output] The data pointer to get the EMCY message.

- **Expression:**

  ***rdata =** "<EM04084830811107000000000"
  Definition word = "<EM"
  Node id = "04"
  EMCY COBID = "084"
  Length of bytes = "8"
  byte 0 = "30"
  byte 1 = "81"
  byte 2 = "11"
  byte 3 = "10"
  byte 4 = "07"
  byte 5 = "00"
  byte 6 = "00"
  byte 7 = "00"

  (Guard fail and Heartbeat fail)
  ***rdata =** "<R08_02"
  Definition word = "<R08_"
  Node id = "02"

3.5.42. I7565C_GetResponseValuOfPDOPolling

- **Description:**

  Use this function to get the PDO response data. If the mode of the PDO is set to polling mode, user can use this function to get the PDO response data stored in the software buffer.

- **Syntax:**

  int I7565C_GetResponseValuOfPDOPolling (**unsigned char** *rdata)

- **Parameter:**

  ***rdata:** [output]. The data pointer to get the PDO data.

- **Expression:**

  ***rdata =** ”< P1842D03F”
  Definition word = “<P“
  PDO COBID = “184”
  Length of bytes = ”2”
  byte 0 = “D0”
  byte 1 = “3F”
  byte 7 = “00”

3.5.43. I7565C_GetResponseValuOfSDOPolling

- **Description:**

  Use this function to get the SDO response data. If the mode of the SDO is set to polling mode, user can use this function to get the SDO response data stored in the software buffer.

- **Syntax:**

  int I7565C_GetResponseValuOfSDOPolling (**unsigned char** *rdata)

- **Parameter:**

  ***rdata:** [output]. The data pointer to get the SDO data.

- **Expression:**

  ***rdata =** ”<i0454F006001D0”

  Definition word = “<I”

  Node id = “04”

  Length of bytes = “5”

  byte 0 = “4F”

  byte 1 = “00”

  byte 2 = “60”

  byte 3 = “01”

  byte 4 = “D0”

3.5.44. I7565C_Heartbeat

- **Description:**

  Use the function I7565C_Heartbeat to set Heartbeat Producer Time and Heartbeat Consumer Time. Then, the slave will automatically send the Heartbeat message to master device according to the Heartbeat Time parameters of this function, and the master can count the time according to the Consumer Time parameters of this function,. If the master doesn’t receive the confirmation of Heartbeat message form the salve, I-7565-CPM will send an error message to users. Users need to use I7565C_GetEMCYInfo to get this message. However, if the master doesn’t receive the Heartbeat message during the Consumer Time it will be triggered to send the EMCY message.

- **Syntax:**

  int I7565C_Heartbeat (**unsigned char** node, **unsigned short** C_HeartTime**,**
  DWORD P_HeartTime)

- **Parameter:**

  node: [input] The number of CANopen slave node.
  C_HeartTime: [input] The time parameter of Heartbeat Consumer.
  P_HeartTime: [input] The time parameter of Heartbeat Producer.

- **Return:**

  7565C_OK
  I7565C_Heartbeat_FAILED
  I7565C_No_Port;

Chapter 4 I-7565-CPM Utility

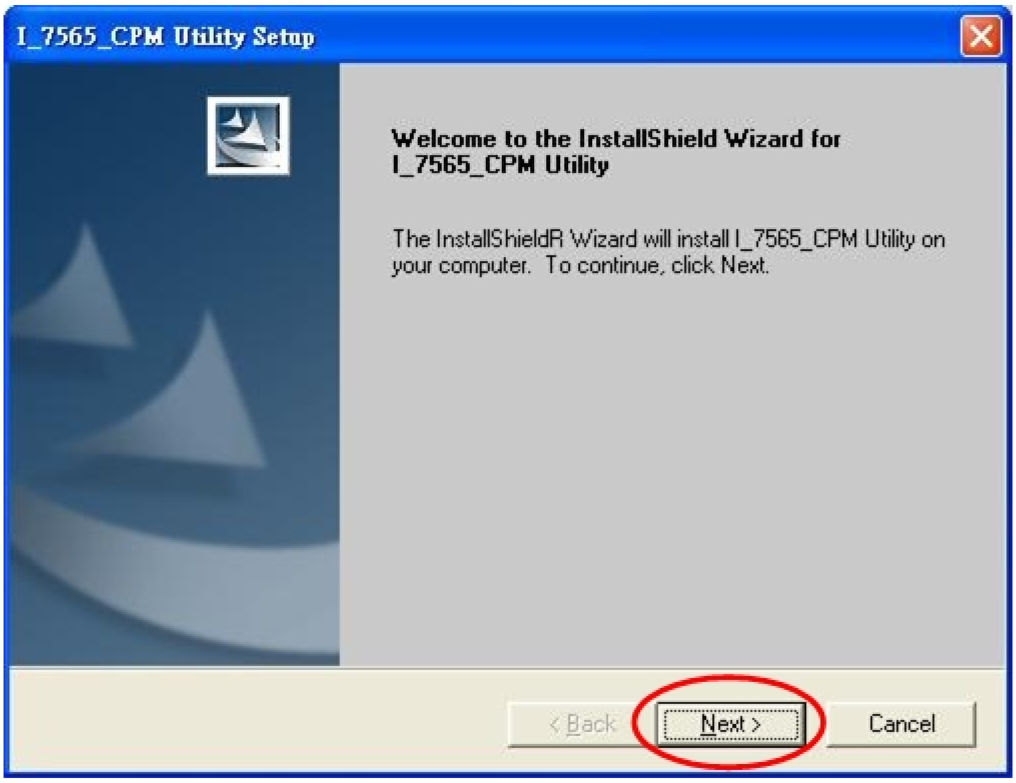
4.1. Install I-7565-CPM Utility

We provide the I-7565-CPM Utility. There are several functions supported by the utility. They are shown as follows.

1. Configure NMT mode, SYNC COBID, SYNC transmission time, Guard time, Life time, EMCY COBID, PDO polling time and SDO polling time.
2. Install the PDO or SDO object so that users can use it by using the utility.
3. Remove CANopen node, PDO object or SDO object.
4. Support TxPDO, RxPDO, TxSDO, and RxSDO.
5. Record EMCY messages.

Step 1: Download the I_7565_CPM Utility setup file from the web site http://www.icpdas.com/download/can/index.htm or the CD in the product box. The path is “/Napdos/iCAN/CAN_MASTER/I-7565-CPM”.

Step 2: Execute the I_7565_CPM SETUP.exe file to install the CANopen master utility.

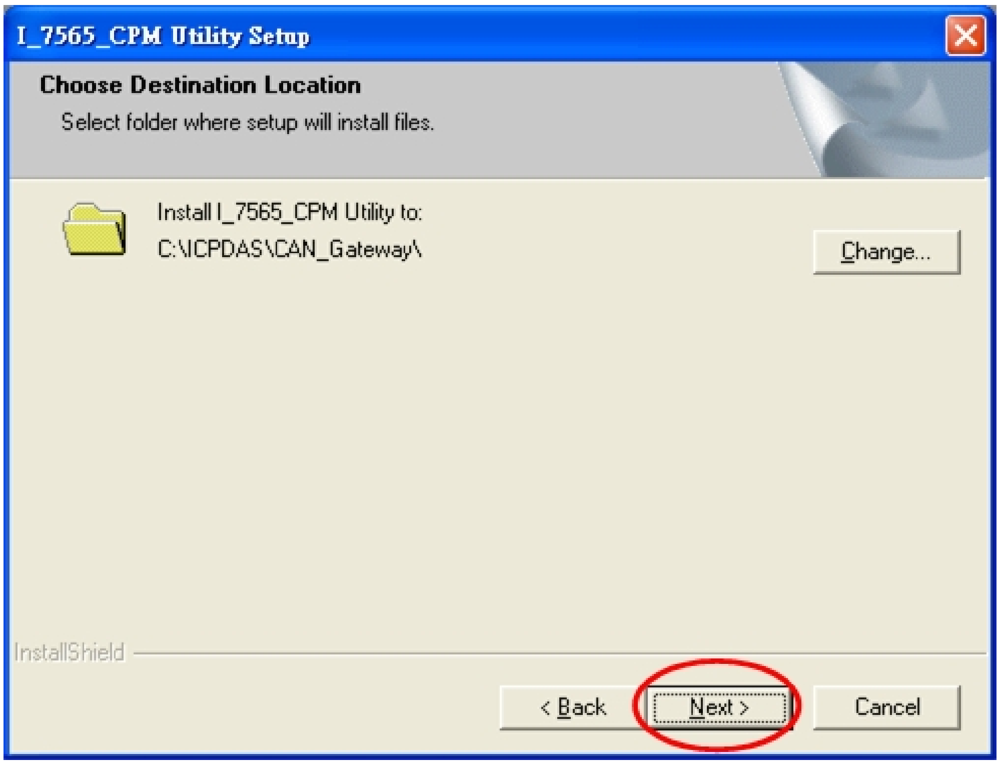


I_7565_CPM Utility Setup

Welcome to the InstallShield Wizard for I_7565_CPM Utility

The InstallShieldR Wizard will install I_7565_CPM Utility on your computer. To continue, click Next.

Step 3: Click the “Next” button to continue, then users will see the default path, if users want to change the install path, click the “Change” button to change the install path.

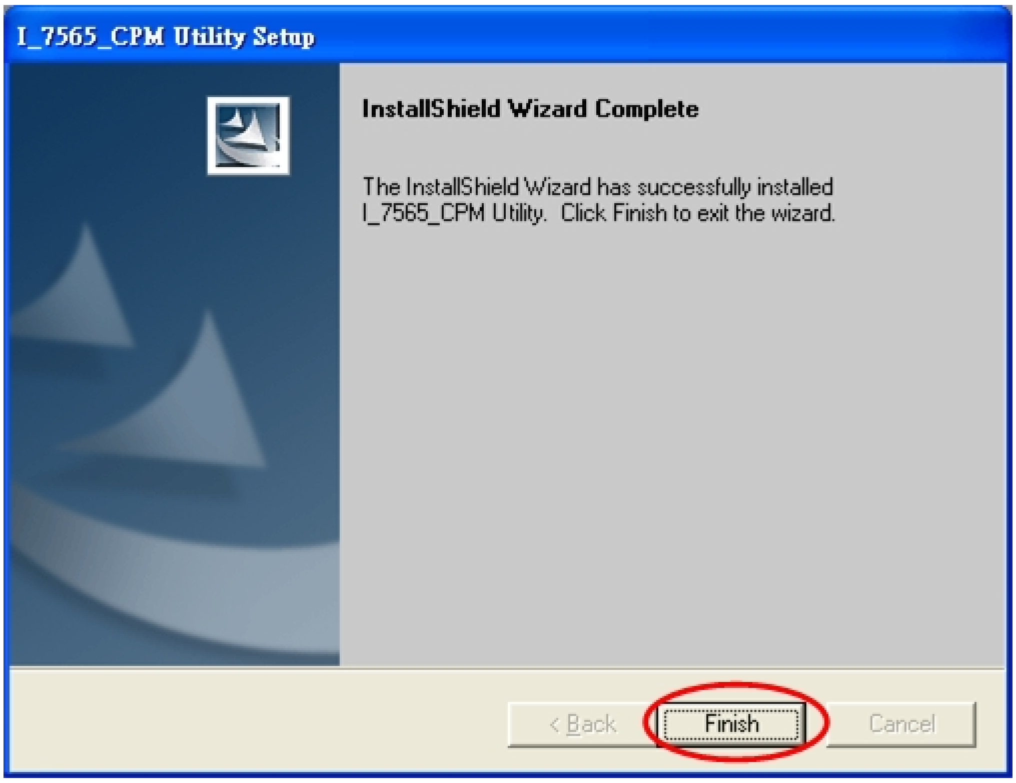

I_7565_CPM Utility Setup

Choose Destination Location

Select folder where setup will install files.

Install I_7565_CPM Utility to:
C:\ICPDAS\CAN_Gateway\

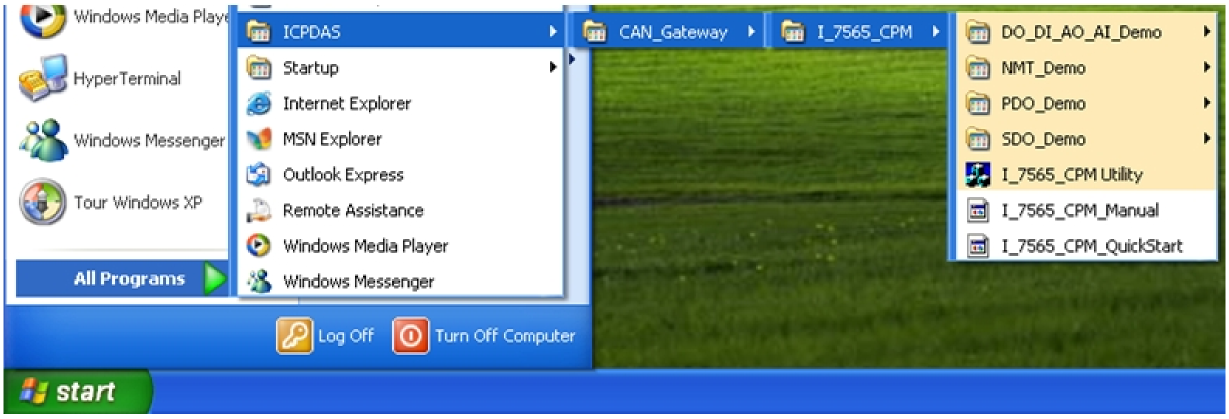
Step 4: Click the “Finish” button to finish the installation.


I_7565_CPM Utility Setup

InstallShield Wizard Complete

The InstallShield Wizard has successfully installed I_7565_CPM Utility. Click Finish to exit the wizard.

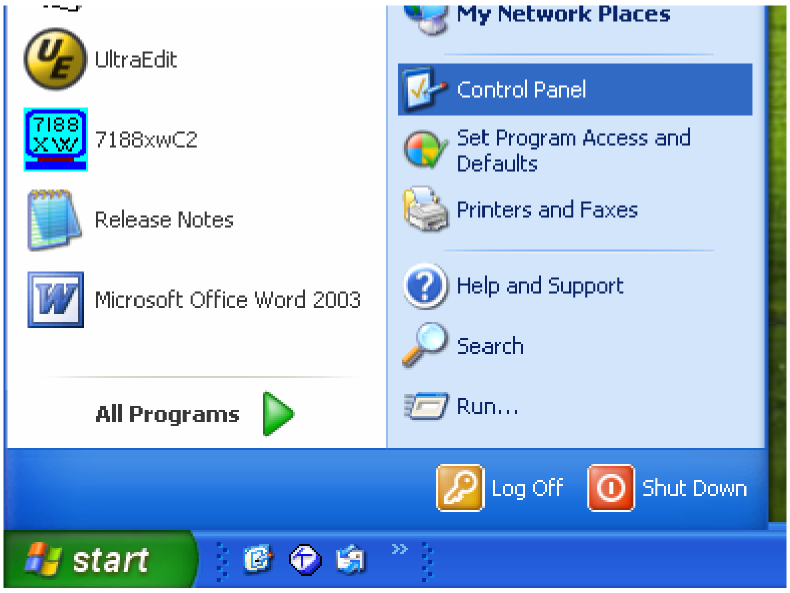
Step 5: After finishing the installation of the I-7565-CPM Utility, users can find the I-7565-CPM Utility, the NMT_ Demo, the PDO_Demo, the SDO_Demo, and the DO_DI_AO_AI_Demo.



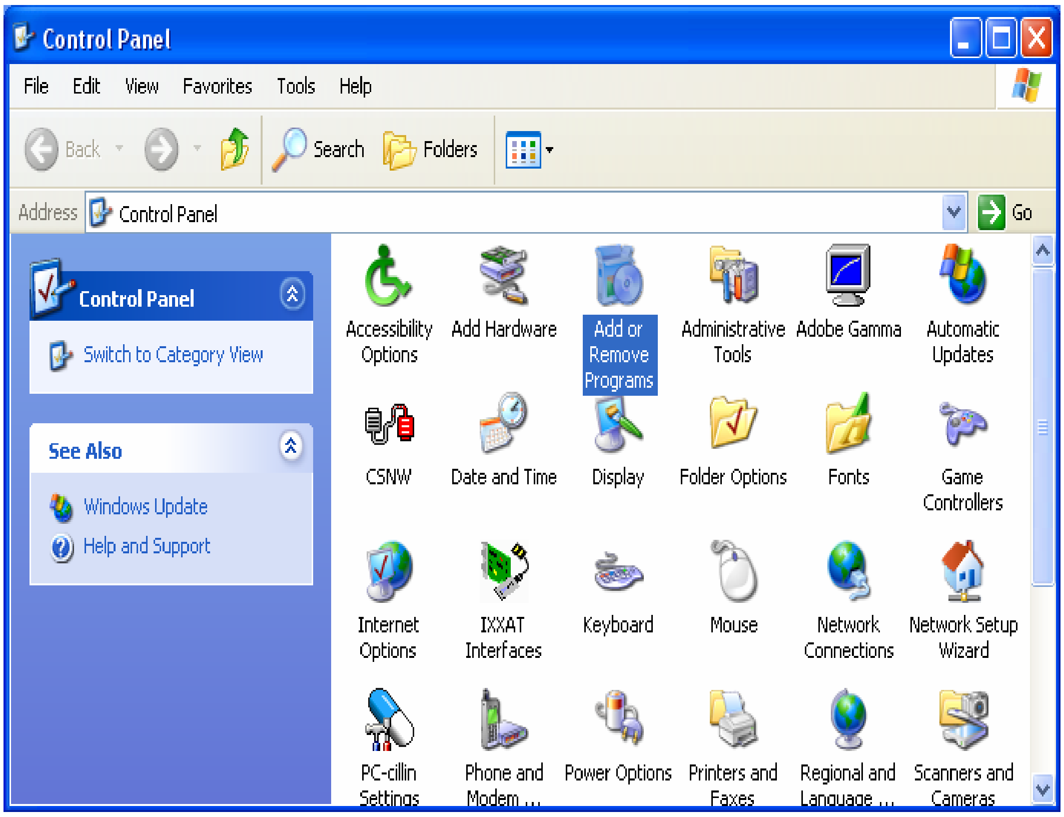
4.2. Uninstall I-7565-CPM Utility

You can uninstall the I-7565-CPM Utility software from the control panel by using the following steps.

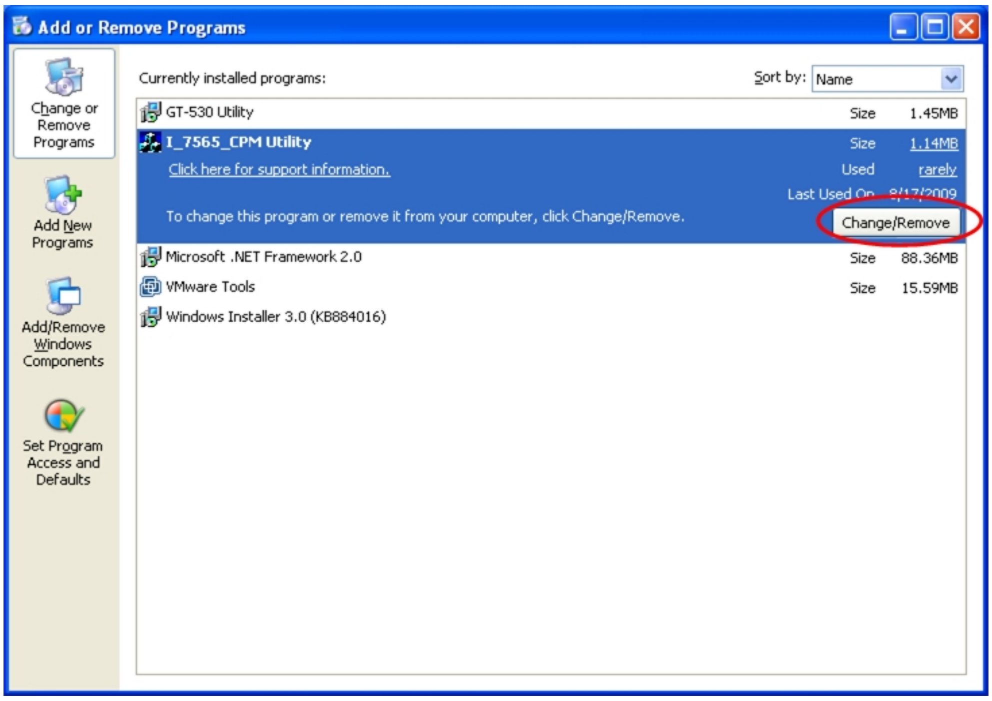
Step 1: Click “Start” in the task bar, then clicks the control panel as the following figure.



Step 2: Click the “Add or Remove Programs” icon to open the dialog.



Step 3: Find out the I-7565-CPM utility, and click “Change/Remove” button.


Add or Remove Programs
Change or Remove Programs
Add New Programs
Add/Remove Windows Components
Set Program Access and Defaults
Sort by: Name
Change/Remove

Currently installed programs:

GT-530 Utility Size 1.45MB

I_7565_CPM Utility Size 1.14MB

Click here for support information. Used rarely

Last Used On 8/17/2009

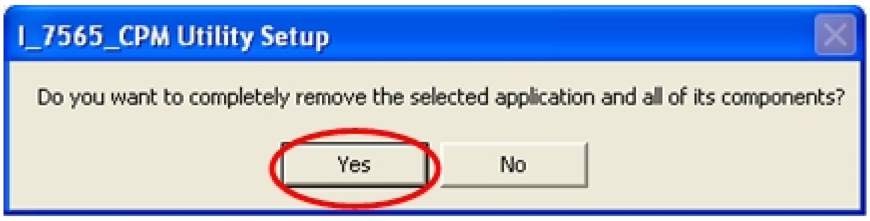
To change this program or remove it from your computer, click Change/Remove.

Microsoft .NET Framework 2.0 Size 88.36MB

VMware Tools Size 15.59MB

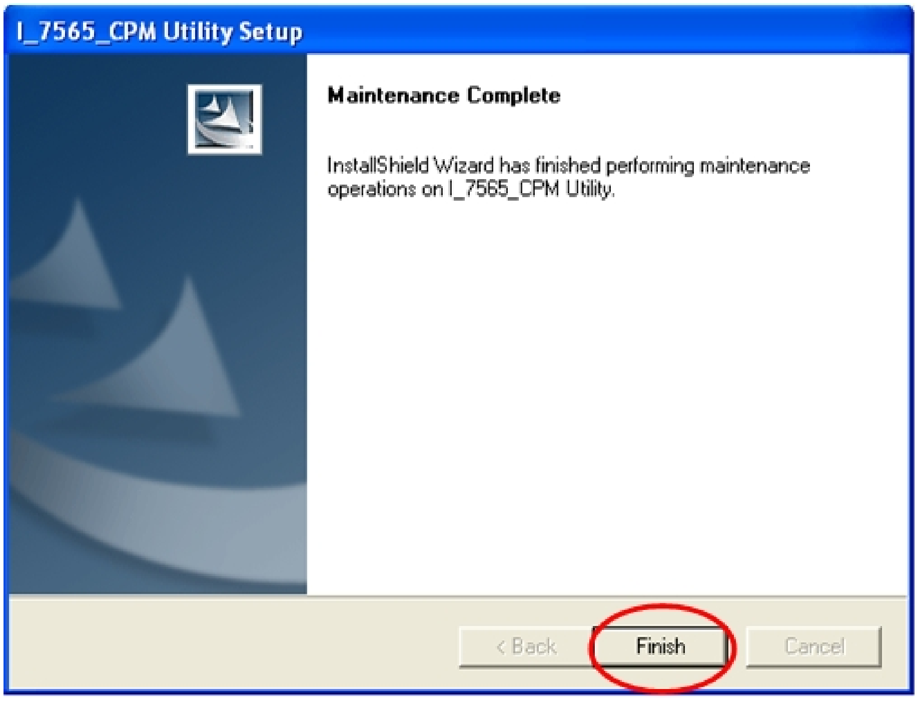
Windows Installer 3.0 (KB884016)

Step 4: Click the “Yes” button to remove it.



Do you want to completely remove the selected application and all of its components?

Step 5:Click the “Finish” button to finish it.



Maintenance Complete

InstallShield Wizard has finished performing maintenance operations on I_7565_CPM Utility.

4.3. Driver install of I-7565-CPM

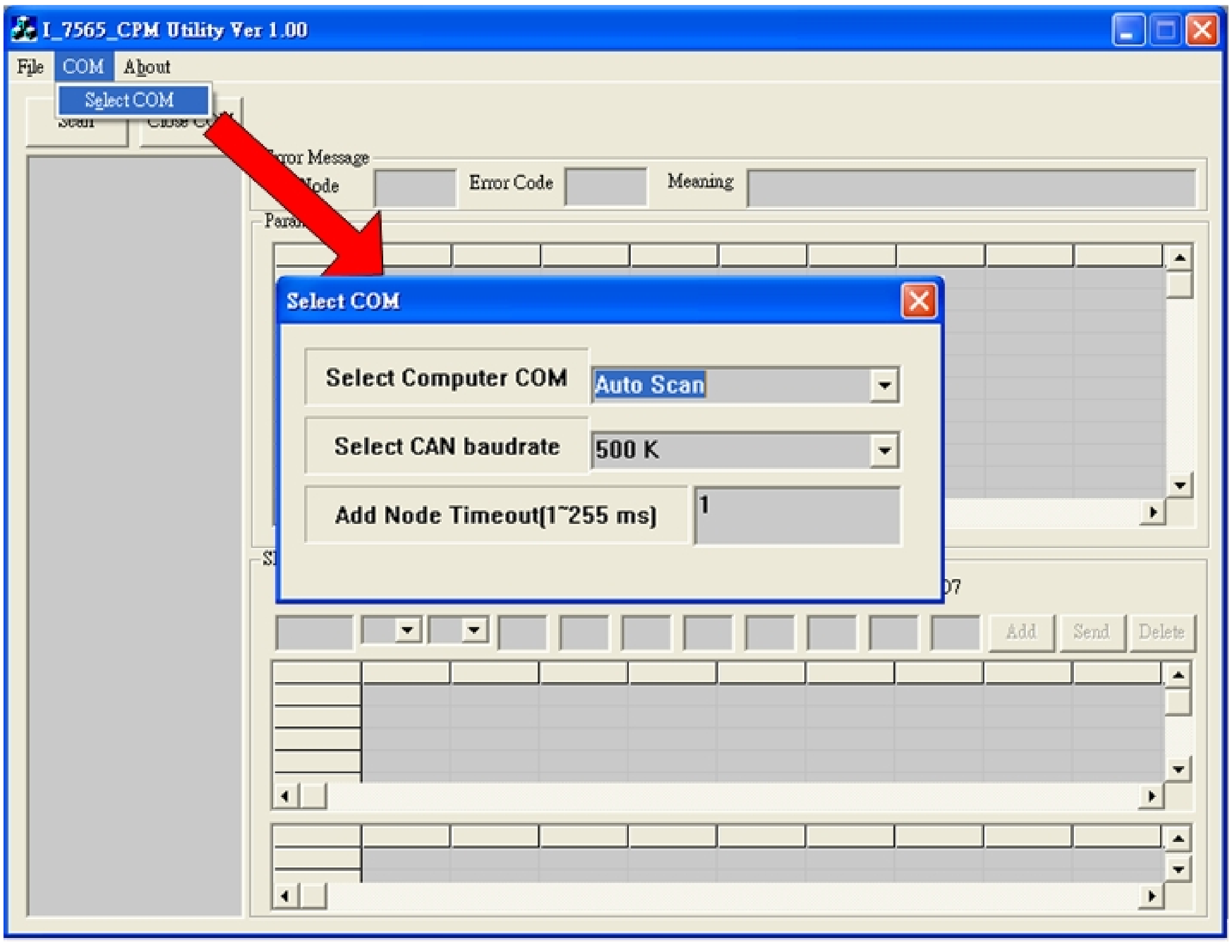
Before using the I-7565-CPM utility, or demos, user have to install the USB driver of I-7565-CPM which is at the path C:\ICPDAS\CAN_Gateway\I_7565_CPM\USB Driver\ i-756x driverinstaller.exe.

Please run the I-756x driverinstaller.exe to install the USB driver in your computer,

4.4. Using I-7565-CPM Utility

4.4.1. Configure COM Port and CAN baud rate

The I-7565-CPM utility provides some useful and easy-to-use functions. Before using it, users need to initiate the I-7565-CPM. Here are the steps. Run the I-7565-CPM.exe. The application interface will be shown as follows. Then Click the “Select COM” item to select the computer COM port, CAN bus baud rate and “Add Node Timeout”. The “Add Node Timeout” is the time interval between each two commands in adding node procedure. Afterwards, click the “Scan” button to scan the I-7565-CPM. If there is an I-7565-CPM connected with users’ computer, the CANopen slaves which are in the same CANopen network with I-7565-CPM will be scanned,



Select COM

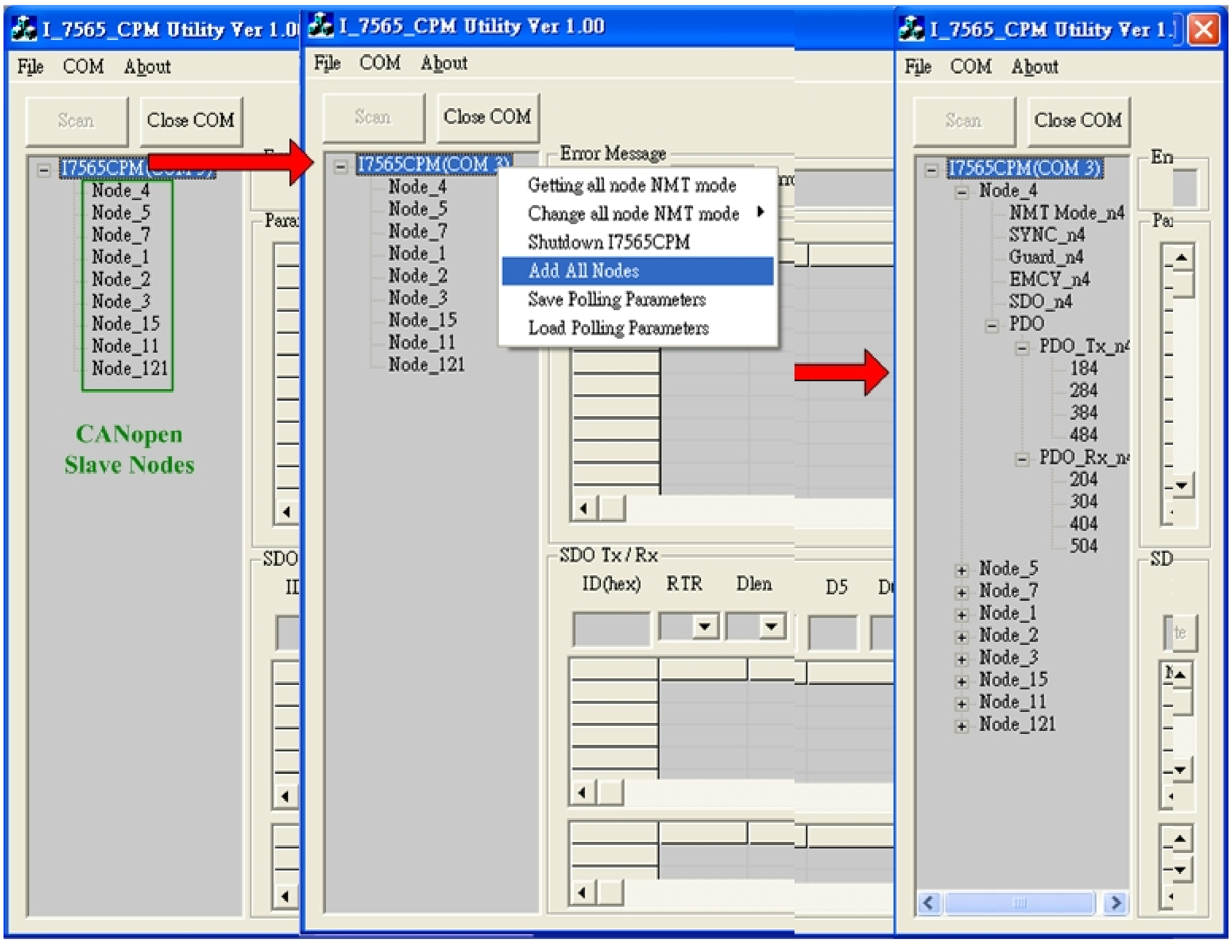
Select Computer COM: Auto Scan

Select CAN baudrate: 500 K

Add Node Timeout(1~255 ms): 1

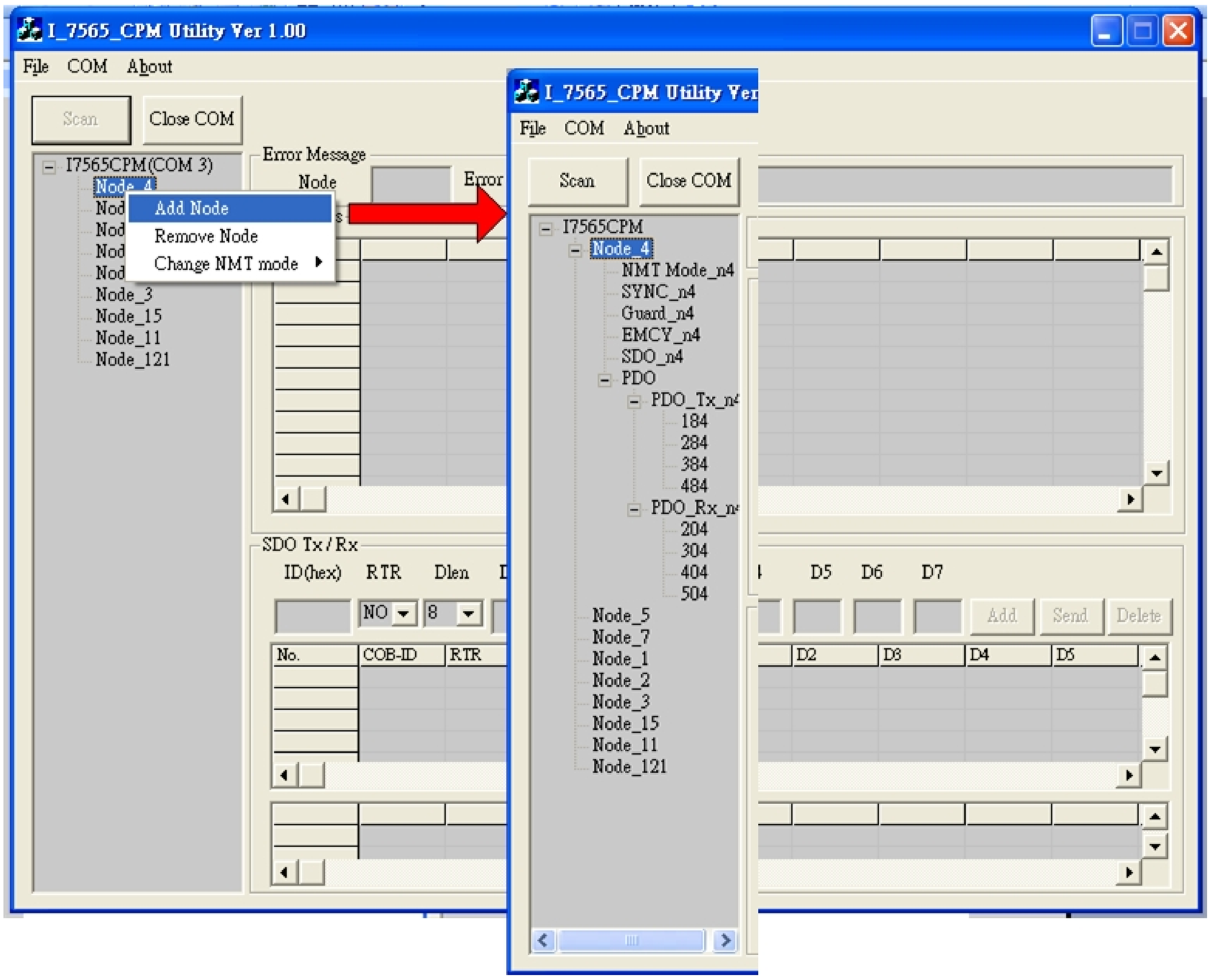
Select the “I7565CPM” item in the list, and right click it add all scanned nodes by using item “Add All Nodes”. Then, users can configure the parameters of the CANopen nodes on the pop up dialog.

1. **Getting all node NMT mode** → show the NMT states of all nodes in the filed “parameters” immediately.
2. **Change all node NMT mode** → Change the NMT states of all nodes, such as Stop mode, Pre-Operation mode, Reset Node mode and Reset Communication mode.
3. **Shutdown I7565CPM** → Shutdown the I-7565-CPM and change the NMT states of all CANopen nodes to be Pre-Operation mode.
4. **Add All Nodes** → Add all scanned CANopen nodes into the management list of the I-7565-CPM. Afterwards, users can use these nodes in the utility.
5. **Save Polling Parameters** → Save the PDO and SDO polling configuration into the EEPROM of the I-7565-CPM.
5. **Load Polling Parameters** → Load the PDO and SDO configuration from the EEPROM of the I-7565-CPM.



If users just want to add some node, select the CANopen nodes in the list, right click it, and select the item “Add Node” in the pop up dialog.
In order to configure this node after finishing the adding procedure, users can use the following method in the pop-up dialog.

1. **Add Node** → Add the specified CANopen nodes into the management list of the I-7565-CPM. Afterwards, users can use it in the utility.
2. **Remove Node** → Remove the CANopen slave node from the management list and EEPROM of the I-7565-CPM.
3. **Change NMT mode** → Change the node state of the specified node. The NMT state may be Stop mode, Pre-Operation mode, Reset Node mode or Reset Communication mode.



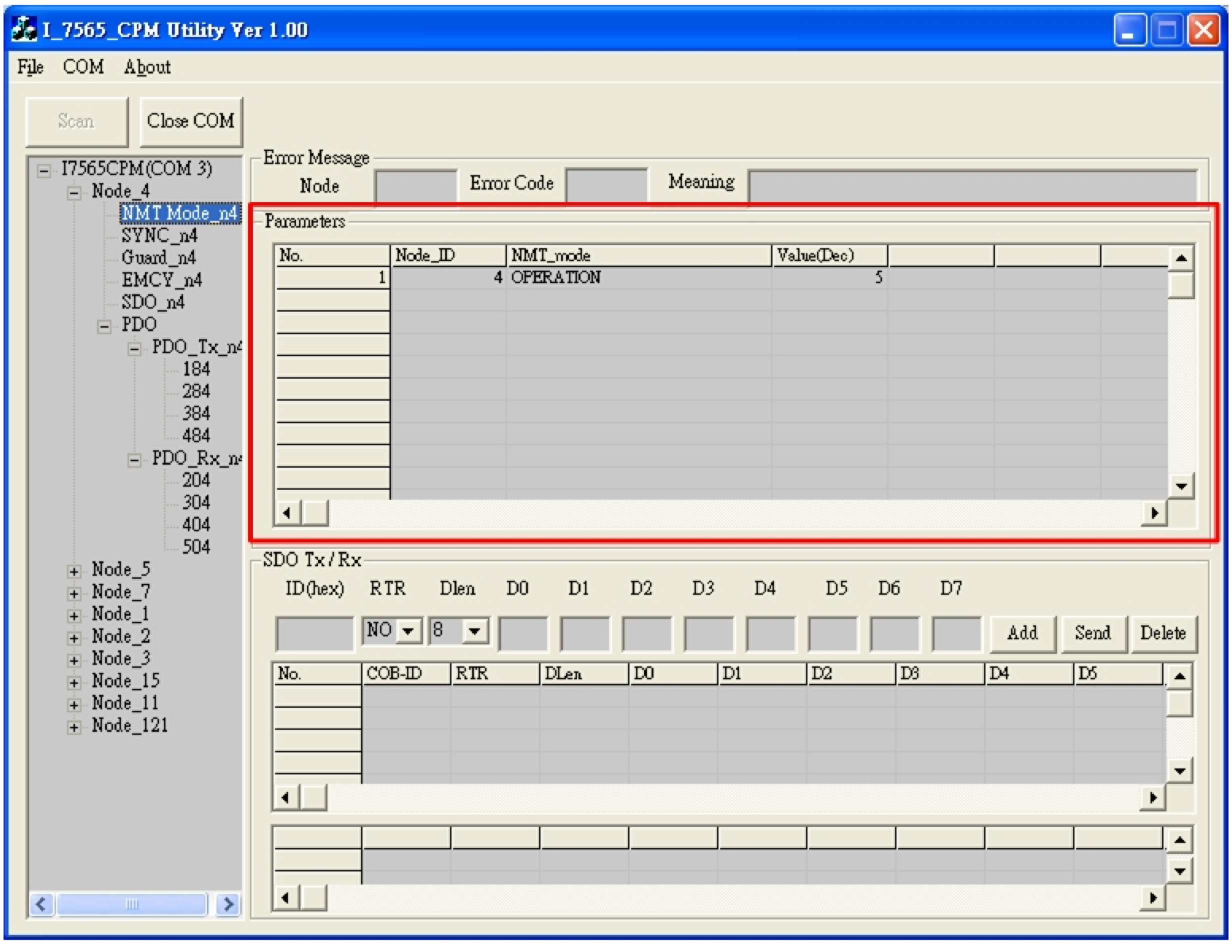
4.4.2. NMT Function

When user clicks the item “NMT_Mode_nx”, the parameter will be shown in the filed “Parameters” as follows.

NMT_Mode_nx

n → node.

x → ID Number(1, 2, … or 127).



No.	Node_ID	NMT_mode	Value(Dec)
1	4	OPERATION	5

4.4.3. SYNC Function

When users click the item "SYNC_nx", the parameter will be shown in the filed "Parameters". Right click to select the item "Setting SYNC", the "setting SYNC" dialog will be displayed, as follows.

Users can set the SYNC COB-ID and SYNC Send Mode.

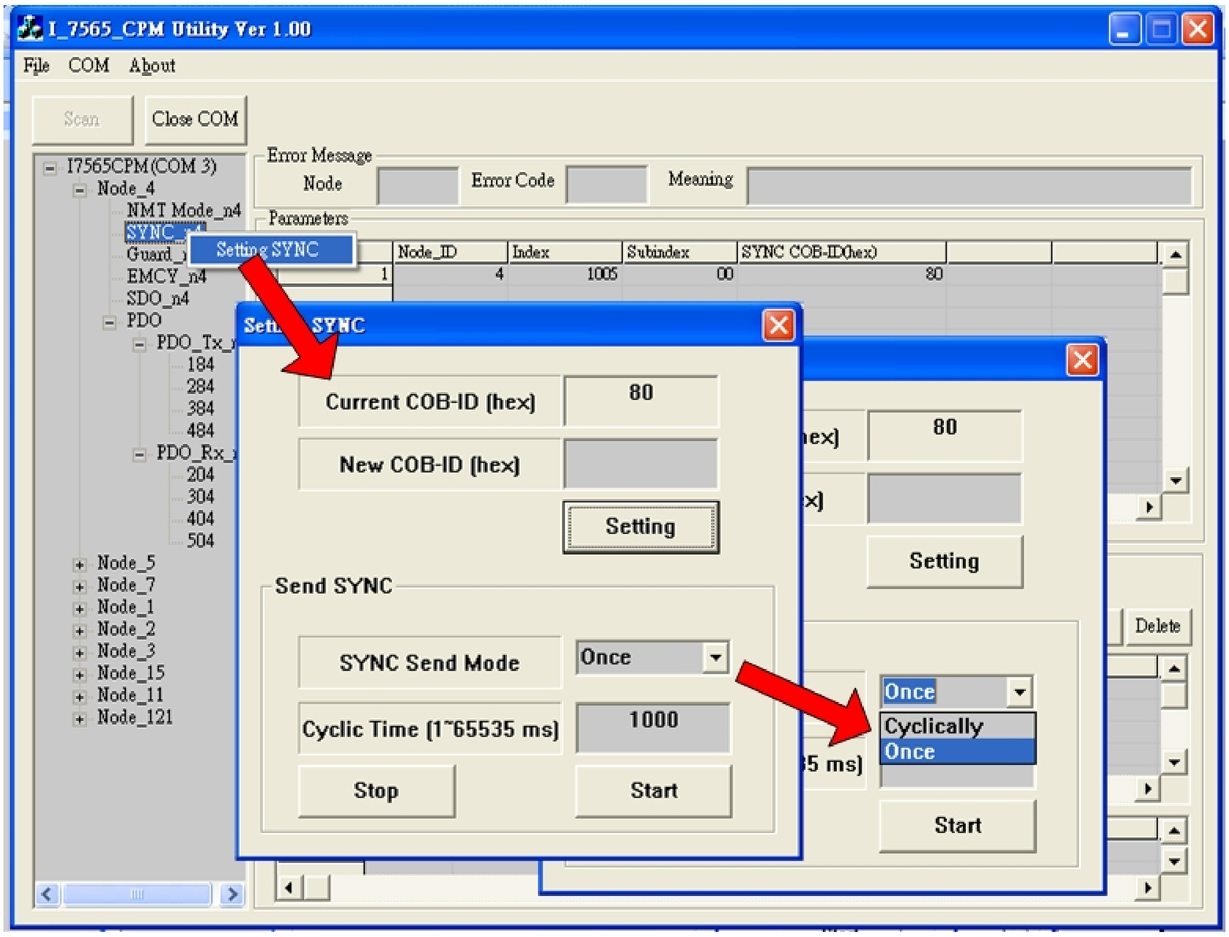
Cyclically → the I-7565-CPM will send the SYNC message according to the interval "Cyclic Time (1~65535 ms)".

Once → the I-7565-CPM will send the SYNC message once after "Cyclic Time (1~65535 ms)".

SYNC_nx

n → node.

x → ID Number(1, 2, … or 127).



Node_ID	Index	Subindex	SYNC COB-ID(hex)
1	4	1005	00	80

Current COB-ID (hex): 80
New COB-ID (hex):
SYNC Send Mode: Once
Cyclic Time (1~65535 ms): 1000

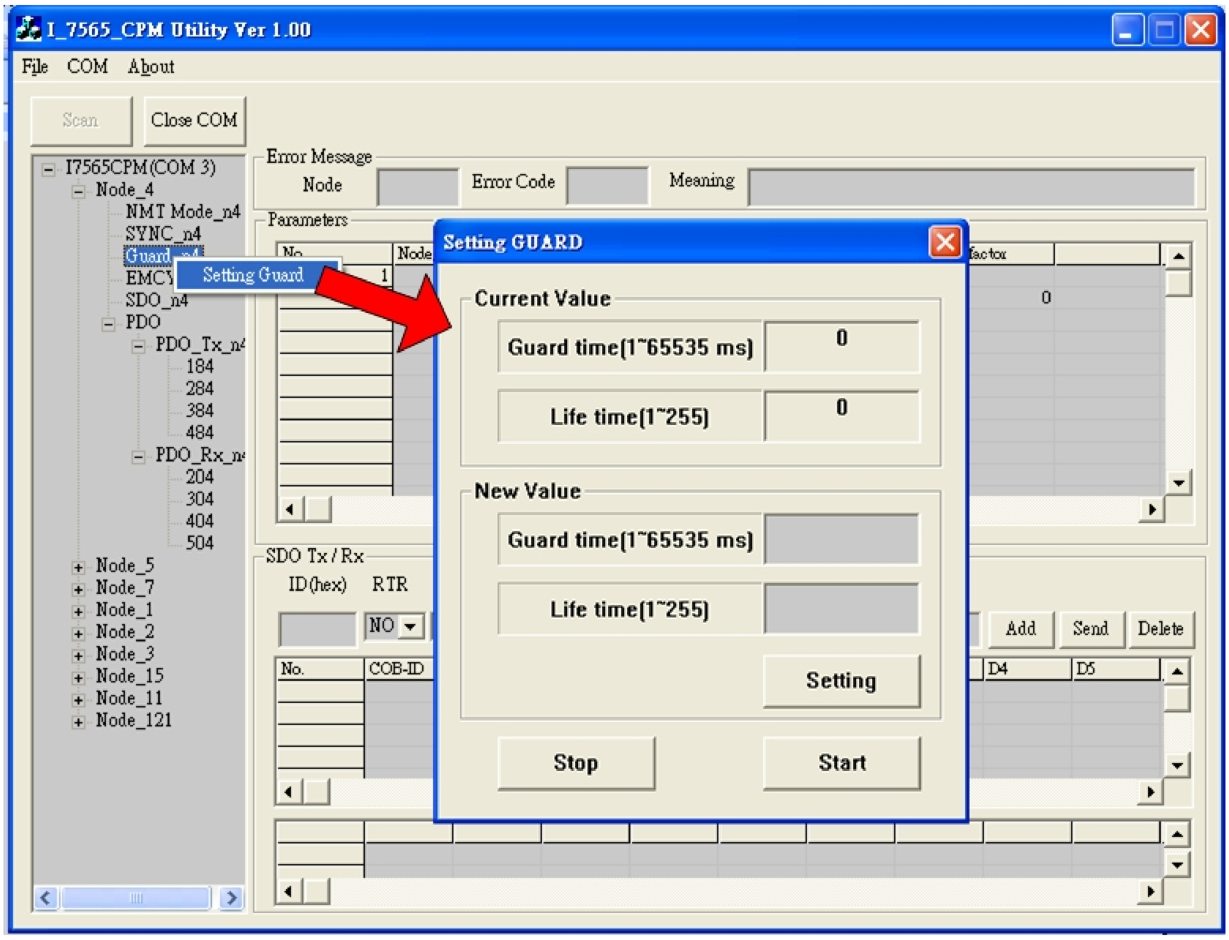
4.4.4. Guard Function

When users click the item “Guard_nx” in the list, the parameters will be shown in the “Parameters” field. Right click to select “Setting Guard” item. The “setting GUARD” dialog will pop up. Users can configure the “Guard time (1~65535 ms)” and “Life time(1~255)” to CANopen node, as follows.

Guard_nx

n → node.

x → ID Number(1, 2, … or 127).



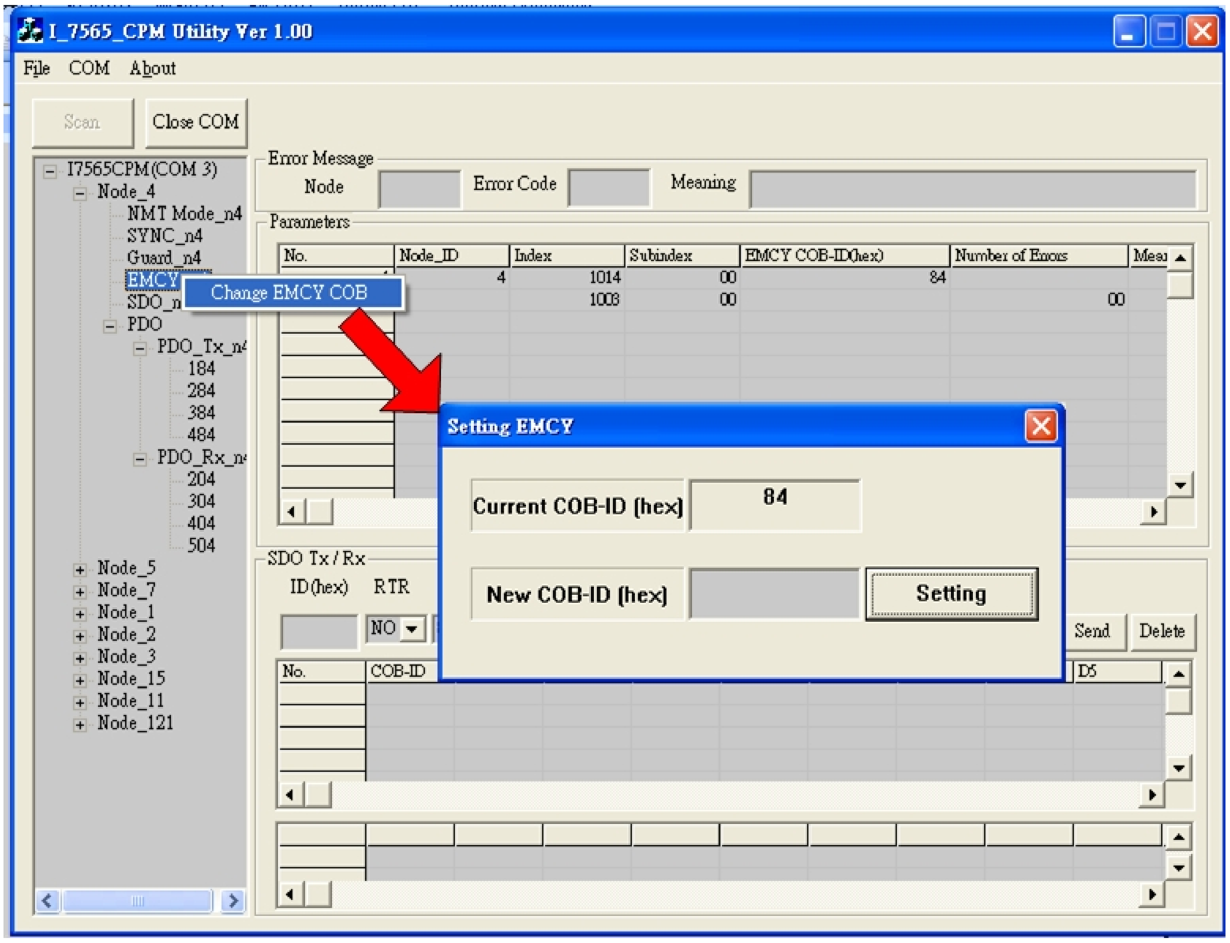
4.4.5. EMCY Function

When users click the item "EMCY_nx" in the list, the parameter will be shown in the "Parameters" field, right click to select "Change EMCY COB" item. The "setting EMCY" dialog will pop up. User can set the "New COB-ID(hex)" to CANopen node, as follows.

EMCY_nx

n → node.

x → ID Number(1.2.3.4.5.6……….127).



No.	Node_ID	Index	Subindex	EMCY COB-ID(hex)	Number of Errors	Mea
	4	1014	00	84		
		1003	00			00

Current COB-ID (hex)	84
New COB-ID (hex)	

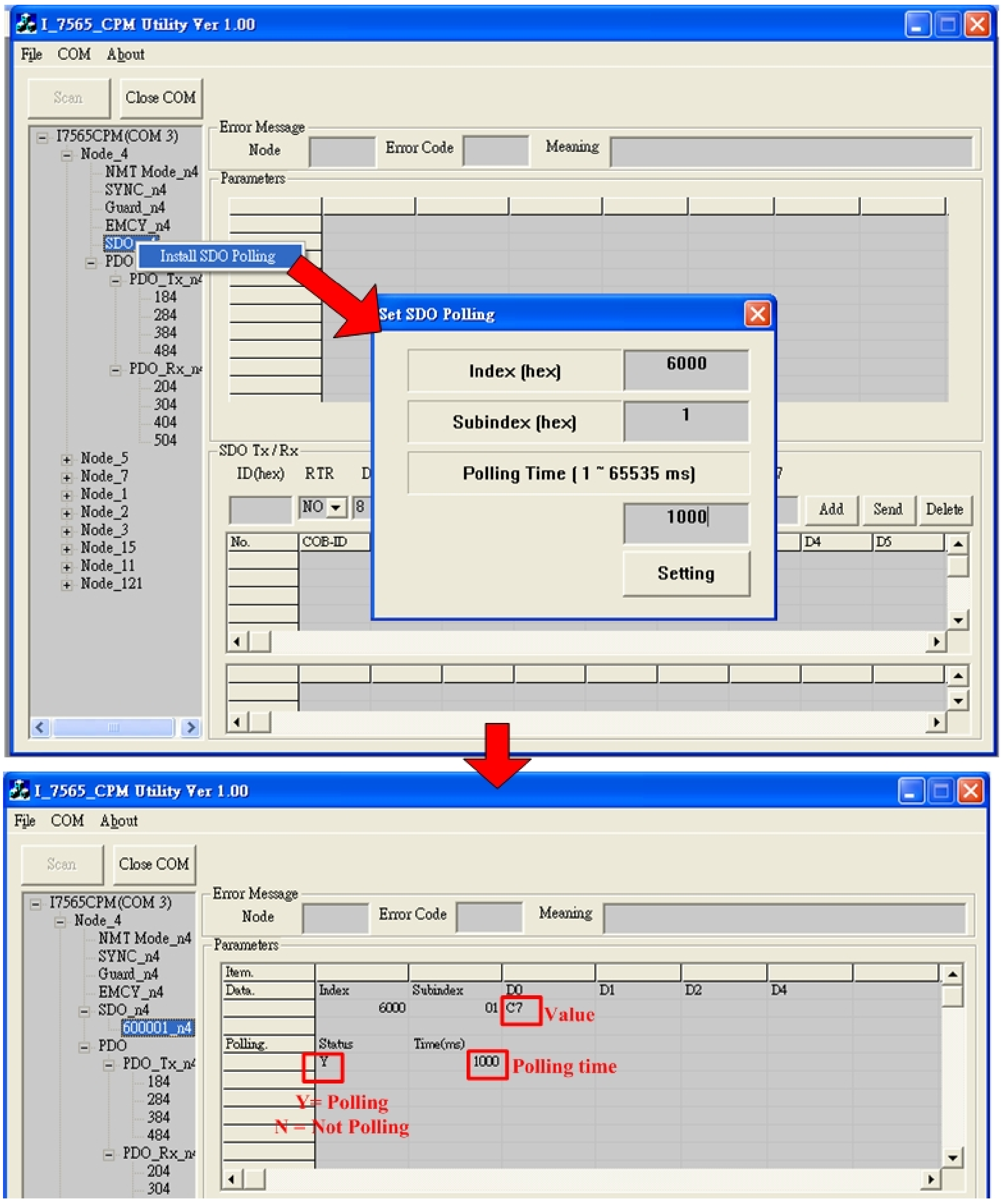
4.4.6. SDO Function

Right click “SDO_nx” item in the list to select “Install SDO Polling”. The “Set SDO Polling” dialog will pop up. Users can set the index, sub-index and polling time of the SDO of the specified CANopen node.

SDO_nx

n → node.

x → ID Number(1, 2, … or 127).



Set SDO Polling	
Index (hex)	6000
Subindex (hex)	1
Polling Time (1 ~ 65535 ms)	1000

Item.							
Data.	Index	Subindex	D0	D1	D2	D4	
	6000	01	C7				
Polling.	Status	Time(ms)					
	Y	1000					

If users want to remove the polling mode of a SDO, click the “600001_nx” item under the SDO_nx title in the list. Then right click to select “Remove SDO Polling” item to remove it from I-7565-CPM.

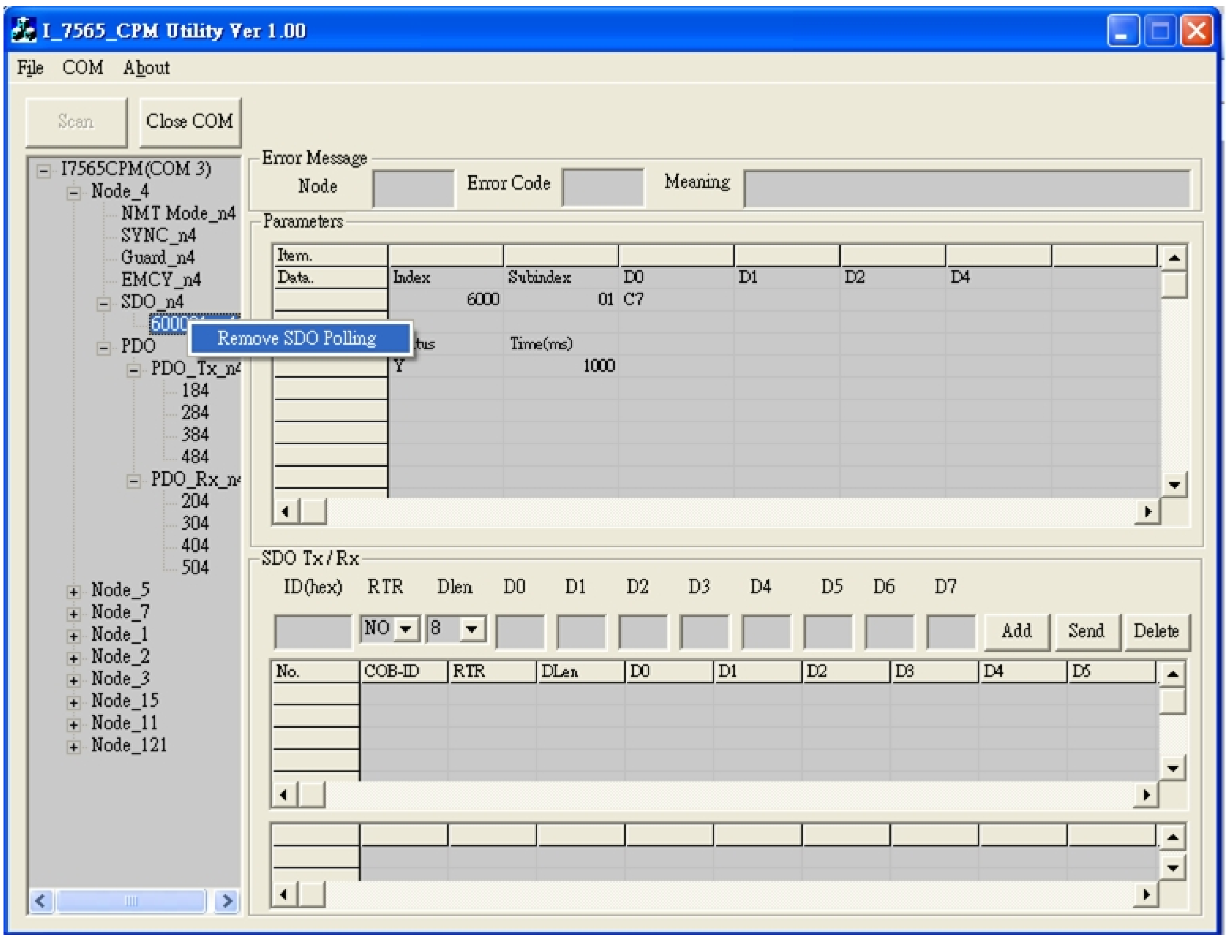
600001_nx

6000 → index.

01 → subindex.

n → node.

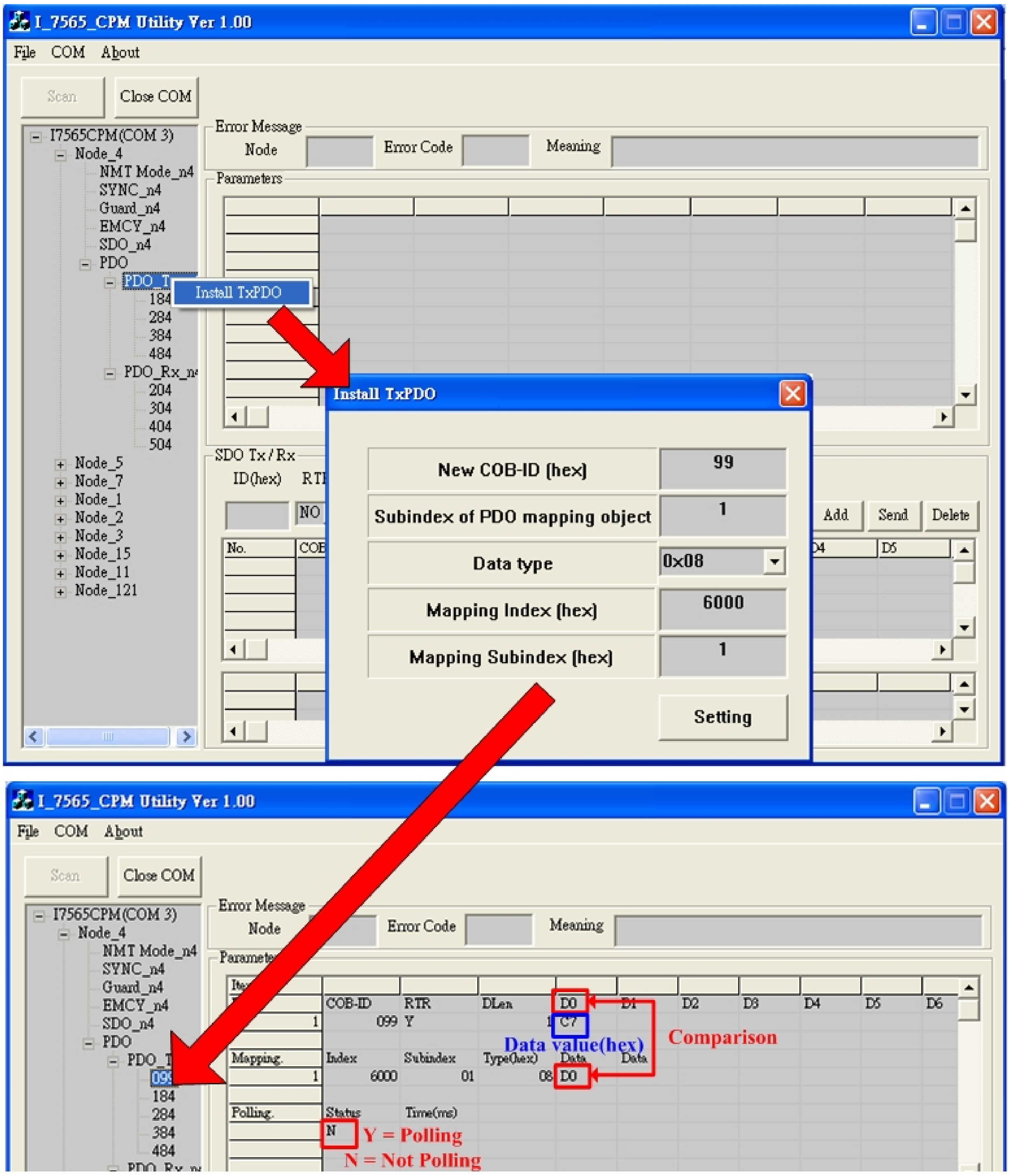
x → ID Number(1.2.3.4.5.6……….127).



Item.						
Data.	Index	Subindex	D0	D1	D2	D4
	6000	01	C7			
	tus	Time(ms)				
	Y	1000				

4.4.7. TxPDO Function

Click the “PDO_Tx_nx” item in the list, then right click to select “Install TxPDO” item. The “Install TxPDO” dialog will be shown. Users have to set the “New COB-ID”, the “Subindex of PDO mapping object”, the “Data type”, the “Mapping index” and “Mapping sub-index” of the “Install TxPDO” dialog.

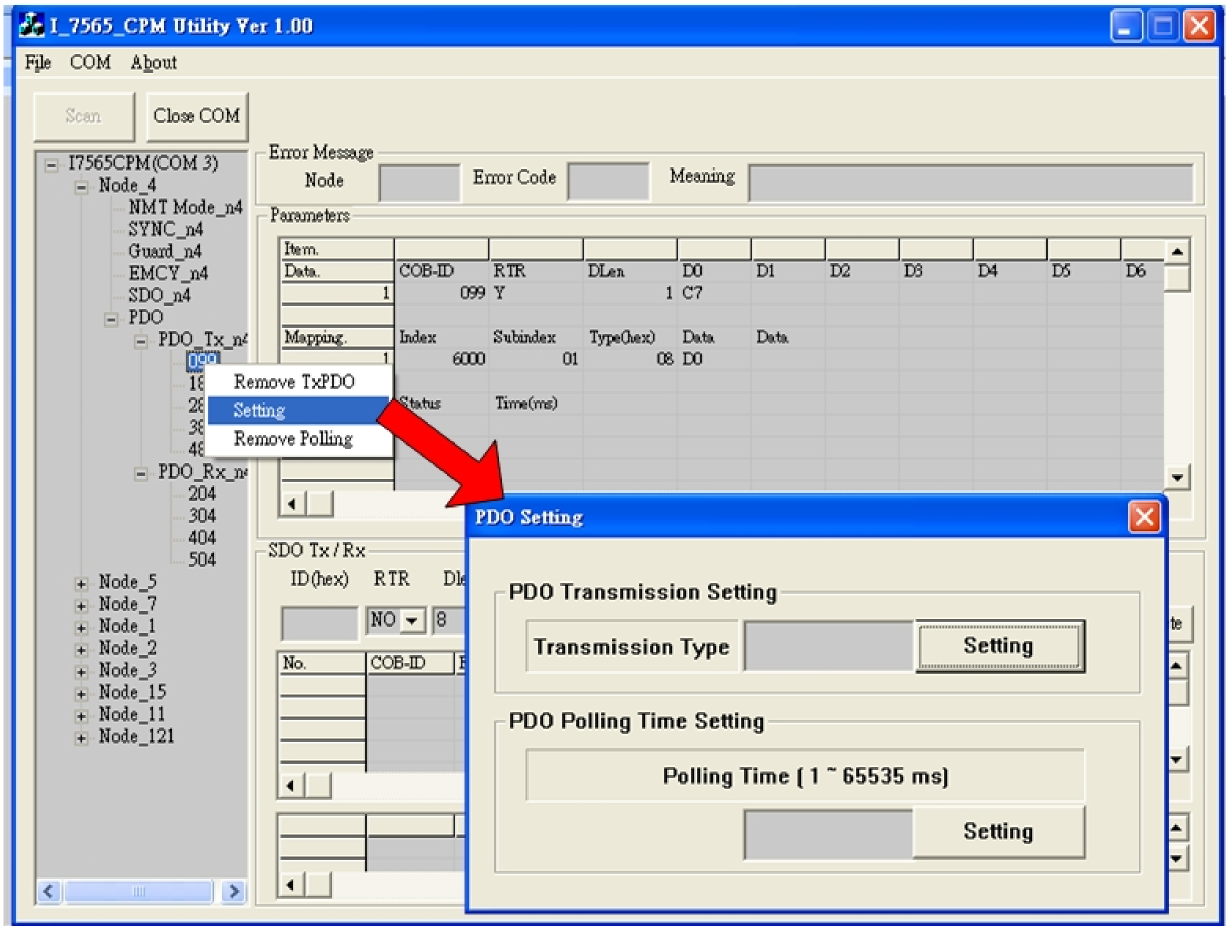


New COB-ID (hex)	99
Subindex of PDO mapping object	1
Data type	0x08
Mapping Index (hex)	6000
Mapping Subindex (hex)	1

Item										
	COB-ID	RTR	DLen	D0	D1	D2	D3	D4	D5	D6
1	099	Y	1	C7						
Mapping	Index	Subindex	Type(hex)	Data	Data					
1	6000	01	08	D0						
Polling	Status	Time(ms)								
	N									

Click the PDO COB-ID under the PDO_Tx_nx title in the list, then right click to select "Setting" item, "PDO Setting" dialog will pop up. Users can set TxPDO transmission type and polling time in the dialog.

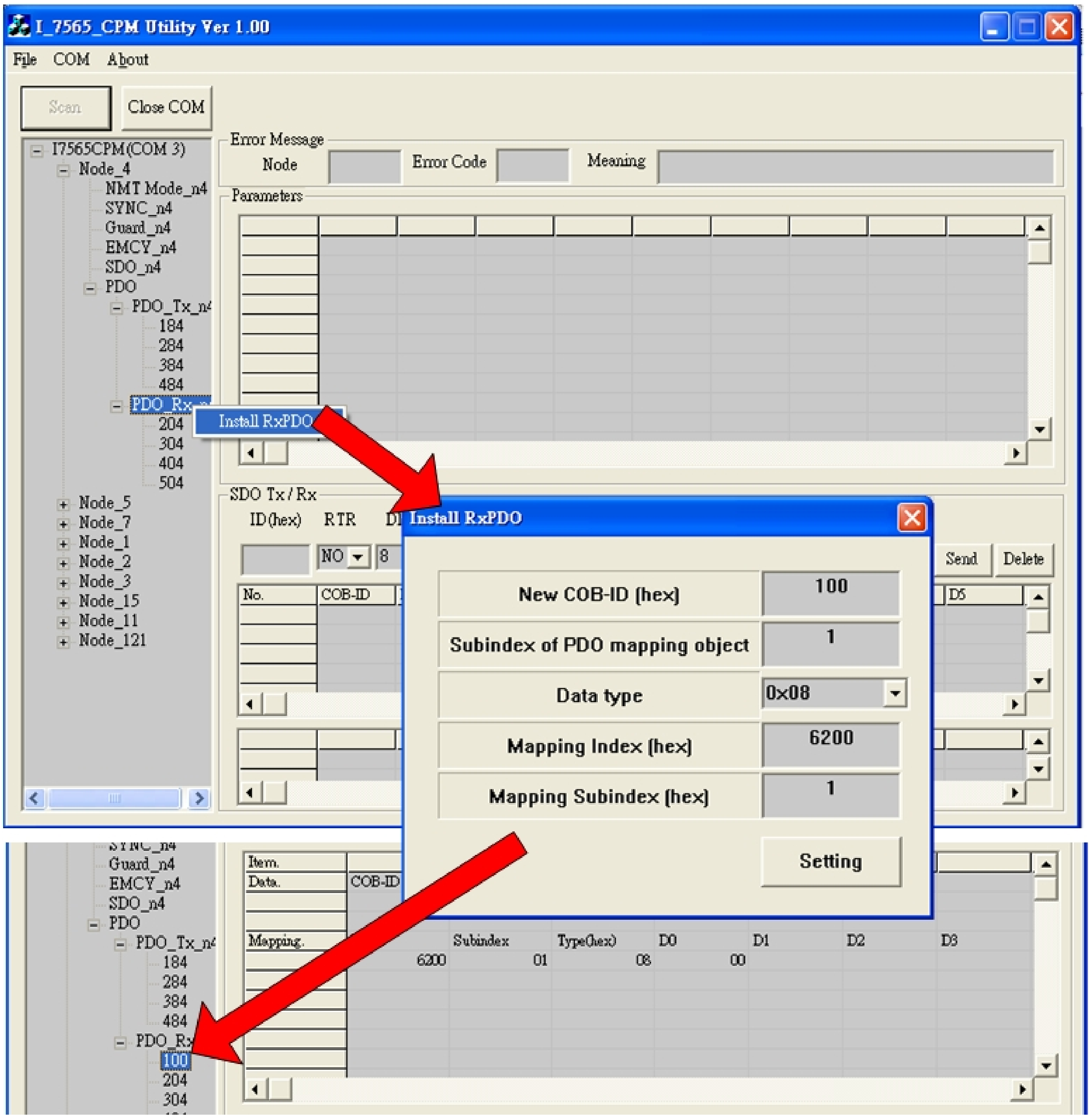
1. Remove TxPDO → Remove selected PDO form a CANopen node.
2. Setting → Set PDO transmission type or polling time of a CANopen node.
3. Remove Polling → Remove selected PDO polling time.



Item.										
Data.	COB-ID	RTR	DLen	D0	D1	D2	D3	D4	D5	D6
1	099	Y	1	C7						
Mapping.	Index	Subindex	Type(hex)	Data	Data					
1	6000	01	08	D0						
	Status	Time(ms)								

4.4.8. RxPDO Function

Click the “PDO_Rx_nx” item, and right click to select “Install RxPDO” item. The “Install RxPDO” dialog will pop up. Users have to set the “New COB-ID”, the “Subindex of PDO mapping object”, the “Data Type”, the “Mapping index” and “Mapping subindex” of the “Install RxPDO” dialog.

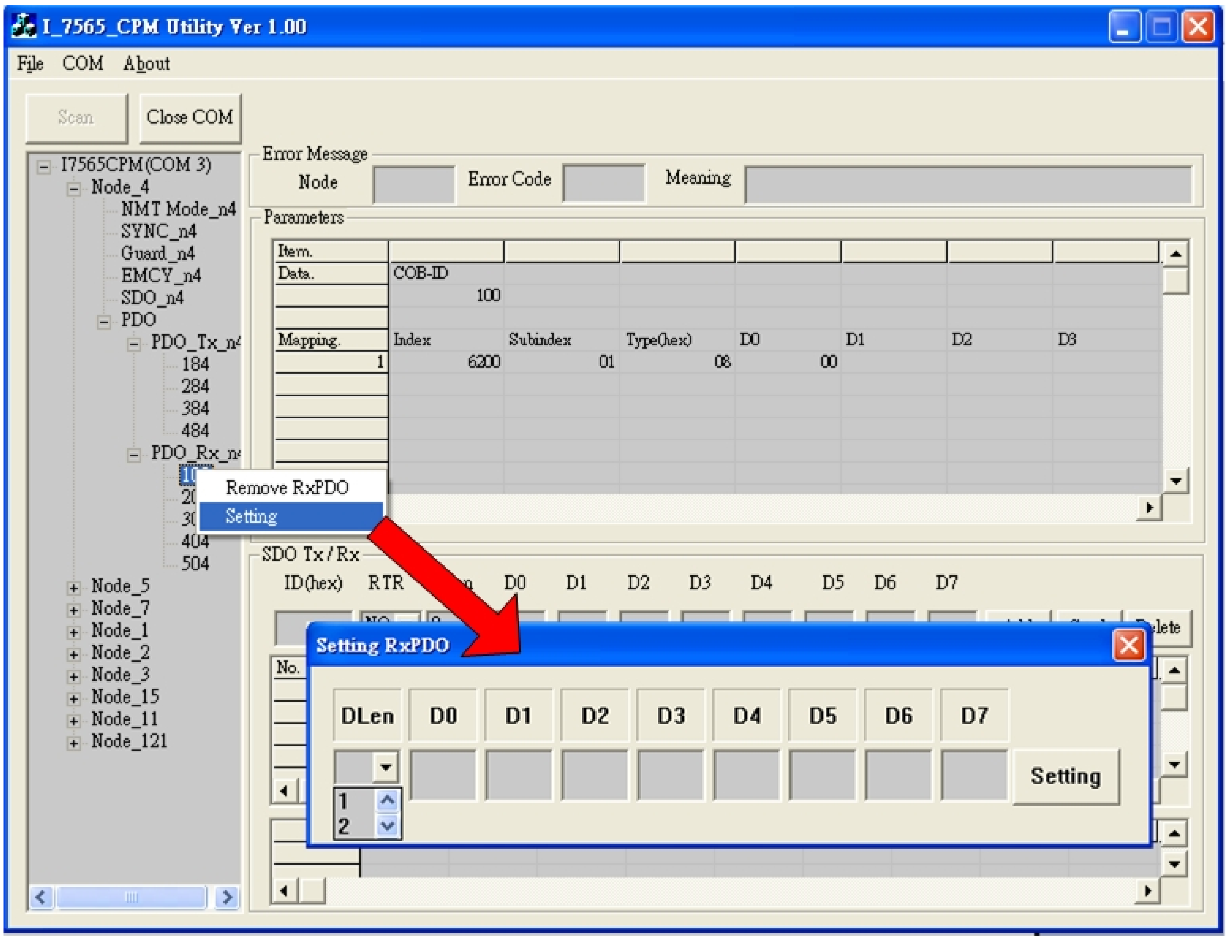


New COB-ID (hex)	100
Subindex of PDO mapping object	1
Data type	0x08
Mapping Index (hex)	6200
Mapping Subindex (hex)	1

User can click the PDO COB-ID under the title PDO_Rx_nx in the list, and right click to select “Setting” item, The “Setting RxPDO” dialog will pop up. Users can fill the PDO data which are sent to CANopen slave node in this dialog.

1. Remove RxPDO → Removing selected PDO form CANopen node.

2. Setting → Send a RxPDO to CANopen node.



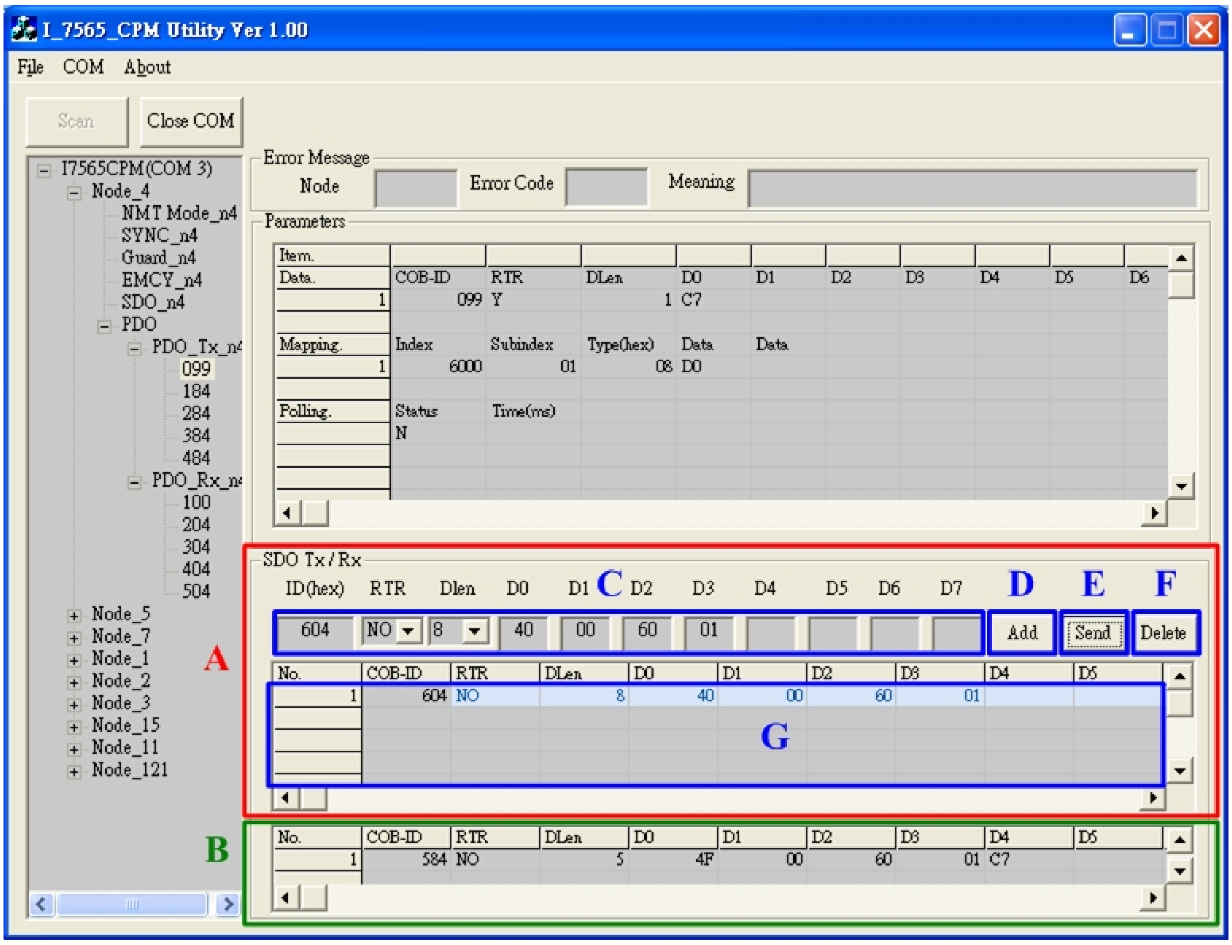
Item.							
Data.	COB-ID						
	100						
Mapping.	Index	Subindex	Type(hex)	D0	D1	D2	D3
1	6200	01	08	00			

4.4.9. SDO Function

I-7565-CPM provides function to access object by SDO protocol in the field marked with the red character "A" and the green character "B". Users can input data of commands of the SDO protocol at the position marked with the blue character "C", and click the "Add" button at the position marked with the blue character "D" to add the command into the list marked with the blue character "G".

If users want to send the SDO command, please select the No. of the SDO command in the list marked with the blue character "G", and click the "Send" button marked with the blue character "E". Then the command will be sent. Afterwards, the SDO response data will be shown in the field marked with the green character "B".

If users want to delete the SDO command, select the No. of the SDO command in the list marked with the blue character "G", and click the "Delete" button marked with the blue character "F". Then, the SDO command will be removed from the list marked the blue character "G" .



Item.										
Data.	COB-ID	RTR	DLen	D0	D1	D2	D3	D4	D5	D6
1	099	Y	1	C7						
Mapping.	Index	Subindex	Type(hex)	Data	Data					
1	6000	01	08	D0						
Polling.	Status	Time(ms)								
	N									

ID(hex)	RTR	Dlen	D0	D1	D2	D3	D4	D5	D6	D7
604	NO	8	40	00	60	01				

No.	COB-ID	RTR	DLen	D0	D1	D2	D3	D4	D5
1	604	NO	8	40	00	60	01		

No.	COB-ID	RTR	DLen	D0	D1	D2	D3	D4	D5
1	584	NO	5	4F	00	60	01	C7	

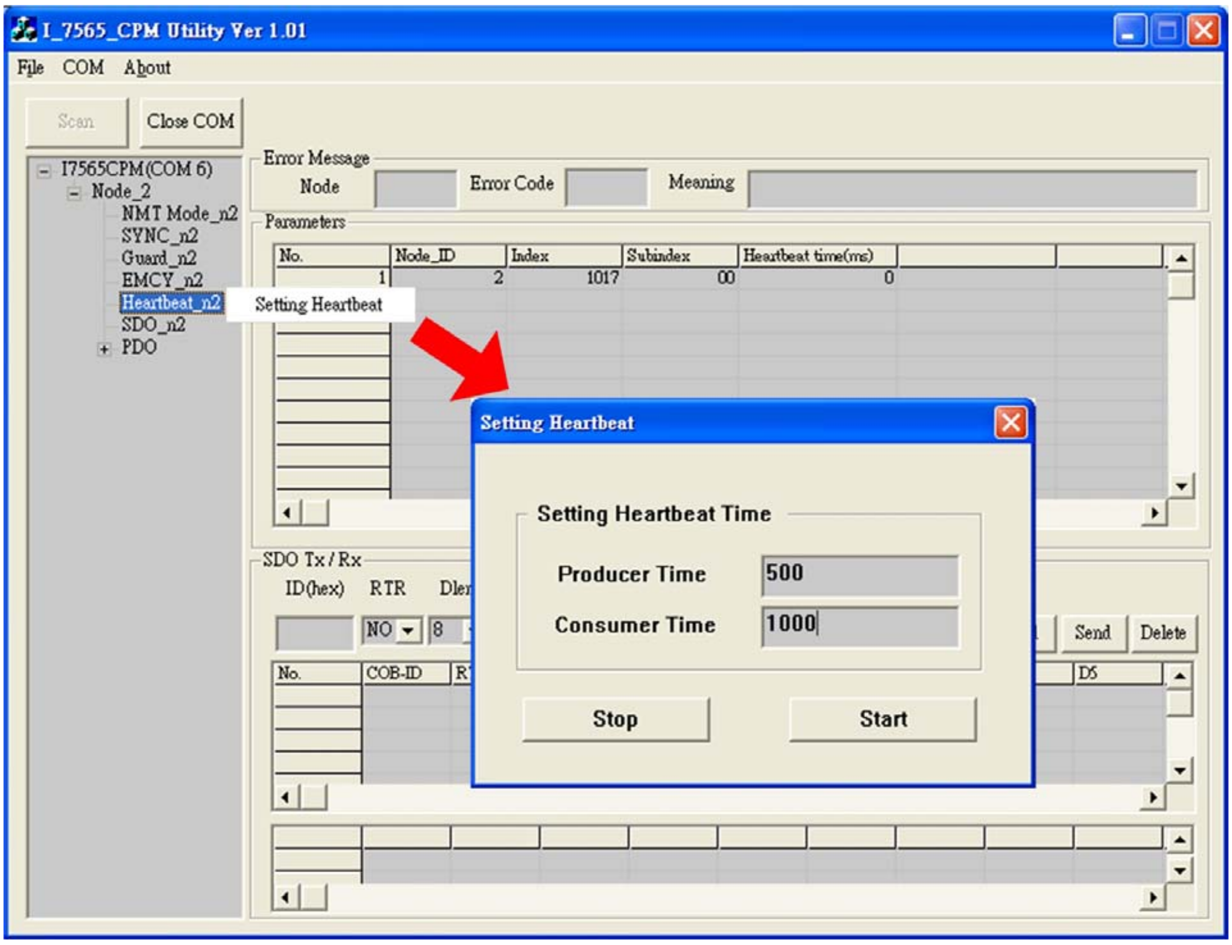
4.4.10.Heartbeat Function

When users click the item “Heartbeat_nx” in the list, the parameters will be shown in the “Parameters” field. Right click to select “Setting Heartbeat” item. The “setting Heartbeat” dialog will pop up. Users can configure the “Producer time (1~65535 ms)” and “Consumer time (1~4294967295)” to CANopen node, as follows.

Heartbeat_nx

n → node.

x → ID Number(1, 2, … or 127).



No.	Node_ID	Index	Subindex	Heartbeat time(ms)
1	2	1017	00	0

Producer Time: 500
Consumer Time: 1000

Chapter 5 Demo Programs

We provide some demos to help user using the lib of the I-7565-CPM. Users can find these demos in the CAN CD or on the web site.

The path of CAN CD

canopen/master/ I-7565-CPM

The address of the web site

http://www.icpdas.com/products/Remote_IO/can_bus/i-7565-cpm.htm

5.1. Brief of the demo programs

These demo programs are developed for demonstrating how to use the CANopen master library to apply the general CANopen communication protocol. These demo programs provide the SDO, PDO, NMT, SYNC communication applications. Each communication protocol is achieved by using different functions of CANopen master library. The relationship between CANopen master library functions and CANopen communication protocols are displayed in the following description.

I7565CPM_NMT_DEMO:

(1). I7565C_Guarding
(2). I7565C_SendSYNC
(3). I7565C_ChangeSYNCID
(4). I7565C_ChangeEMCYID
(5). I7565C_DLL_ver
(6). I7565C_Heartbeat

I7565CPM_PDO_DEMO:

(1). I7565C_InstallPDO
(2). I7565C_SetPDOResponse
(3). I7565C_RemovePDO
(4). I7565C_WritePDO
(5). I7565C_RemotePDO
(6). I7565C_PDOTxType
(7). I7565C_AddPDOPolling
(8). I7565C_DelPDOPolling
(9). I7565C_GetAllUsefulCOBID
(10). I7565C_SavePollingPara
(11). I7565C_LoadPollingPara
(12). I7565C_GetPDOInfo
(13). I7565C_GetPDOPollingTime

I7565CPM_SDO_DEMO:

(1). I7565C_AbortSDO
(2). I7565C_ReadSDO
(3). I7565C_WriteSDO
(4). I7565C_AddSDOPolling
(5). I7565C_DelSDOPolling
(6). I7565C_GetPollingNodeSDO
(7). I7565C_GetSDOPollingTime

I7565CPM_DO_DI_AO_AI_DEMO:

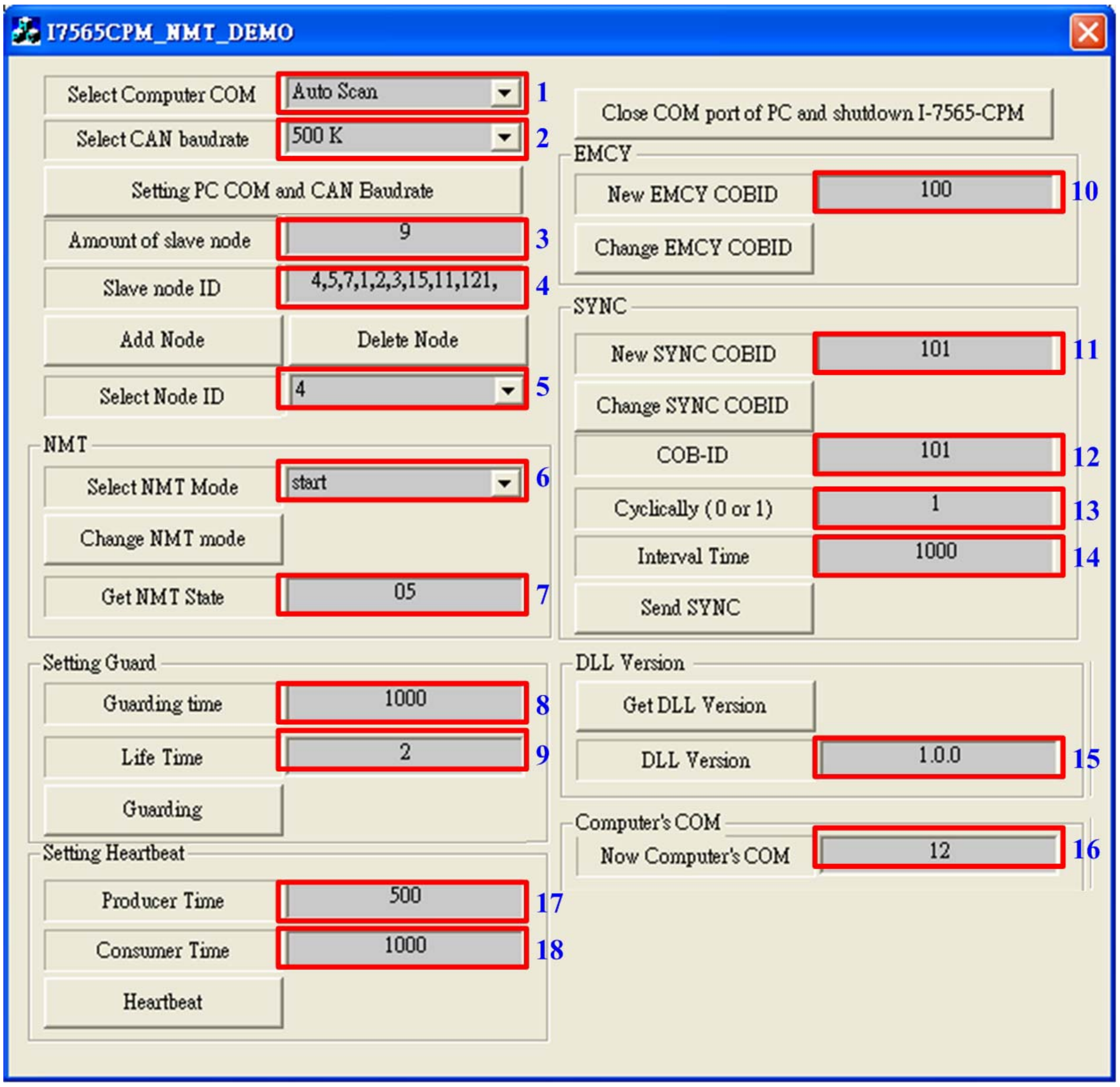
(1). I7565C_WriteDO
(2). I7565C_ReadDI
(3). I7565C_ReadAO
(4). I7565C_ReadAI

Beside, there are some functions which are used to control the I-7565-CPM or configure the CANopen master library. These functions appear in the most of all the application program, they are listed as follows.

(1). I7565C_OpenCom
(2). I7565C_Configure
(3). I7565C_AddNode
(4). I7565C_ShutdownMaster
(5). I7565C_CloseCom
(6). I7565C_RemoveNode
(7). I7565C_ChangeState
(8). I7565C_GetState
(9). I7565C_ReceiveThread
(11). I7565C_ScanAllNode

I7565CPM_NMT_DEMO

When the demo runs, the user interface of the demo is shown below.



I7565CPM_NMT_DEMO

Select Computer COM: Auto Scan 1
Select CAN baudrate: 500 K 2
Setting PC COM and CAN Baudrate
Amount of slave node: 9 3
Slave node ID: 4,5,7,1,2,3,15,11,121, 4
Add Node | Delete Node
Select Node ID: 4 5

NMT
Select NMT Mode: start 6
Change NMT mode
Get NMT State: 05 7

Setting Guard
Guarding time: 1000 8
Life Time: 2 9
Guarding

Setting Heartbeat
Producer Time: 500 17
Consumer Time: 1000 18
Heartbeat

Close COM port of PC and shutdown I-7565-CPM

EMCY
New EMCY COBID: 100 10
Change EMCY COBID

SYNC
New SYNC COBID: 101 11
Change SYNC COBID
COB-ID: 101 12
Cyclically (0 or 1): 1 13
Interval Time: 1000 14
Send SYNC

DLL Version
Get DLL Version
DLL Version: 1.0.0 15

Computer's COM
Now Computer's COM: 12 16

Setting PC COM port and CAN baud rate:

User can select the PC COM port and CAN baud rate at the location marked with blue number 1 and 2, and then push the "Setting PC COM and CAN baud rate" button. If the PC COM port has be opened successfully, the "Amount of slave node" and "Slave node ID" will present the number of CANopen slaves and list all the ID of the CANopen slaves at the location marked with the blue number 3 and 4. The used PC COM port will be presented at the location marked with the blue number 16.

Add Node and Delete Node:

After the steps described above, users have to click the “Add Node” button to add all of the scanned CANopen slaves into the I-7565-CPM management list. Afterwards, the NMT states of these CANopen slaves will be changed to operational mode, and the “Select Node ID” at the position marked with blue number 5 will list the node ID of the CANopen slaves which are in the I-7565-CPM management list. Before users want to do any configuration or setting, select the target slave in this field firstly. If users click “Delete Node” button the I-7565-CPM module will remove all nodes in the I-7565-CPM management list.

Change NMT, SYNC, EMCY:

After selecting the target node, users can firstly set the new parameters of NMT, SYNC, EMCY at the position marked with the blue number 6, 10 and 11. Then click the “Change NMT mode”, “Change SYNC COBID” or “Change EMCY COBID” button to change them (note: After user click the “Change NMT mode”, the NMT response of the CANopen slave will be shown in the “Get NMT State” at the position marked with the blue number 7).

Setting Guarding:

If users want to configure the node guarding parameters of a CANopen slave, users have to set the “Guarding time” and “Life Time” at the position marked with the blue number 8 and 9. Then click the “Guarding” button to set these parameters into the CANopen slave immediately

If user wants to stop node guarding of a CANopen slave, please set the “Life Time” value to 0 and click “Guarding” button to apply it.

Sending SYNC:

If user wants to send SYNC message, set the COB-ID of SYNC, Cyclically (note: Cyclically value = 0 just send SYNC once. Cyclically value = 1 send SYNC cyclically) and Interval Time (note: the time range is from 1 to 65535 ms) at the position marked with the blue number 12, 13 and 14.

Close COM and shutdown I-7565-CPM:

Users can click the “Close COM port of PC and shutdown the I-7565-CPM” button to close the computer’s COM port and shutdown the I-7565-CPM.

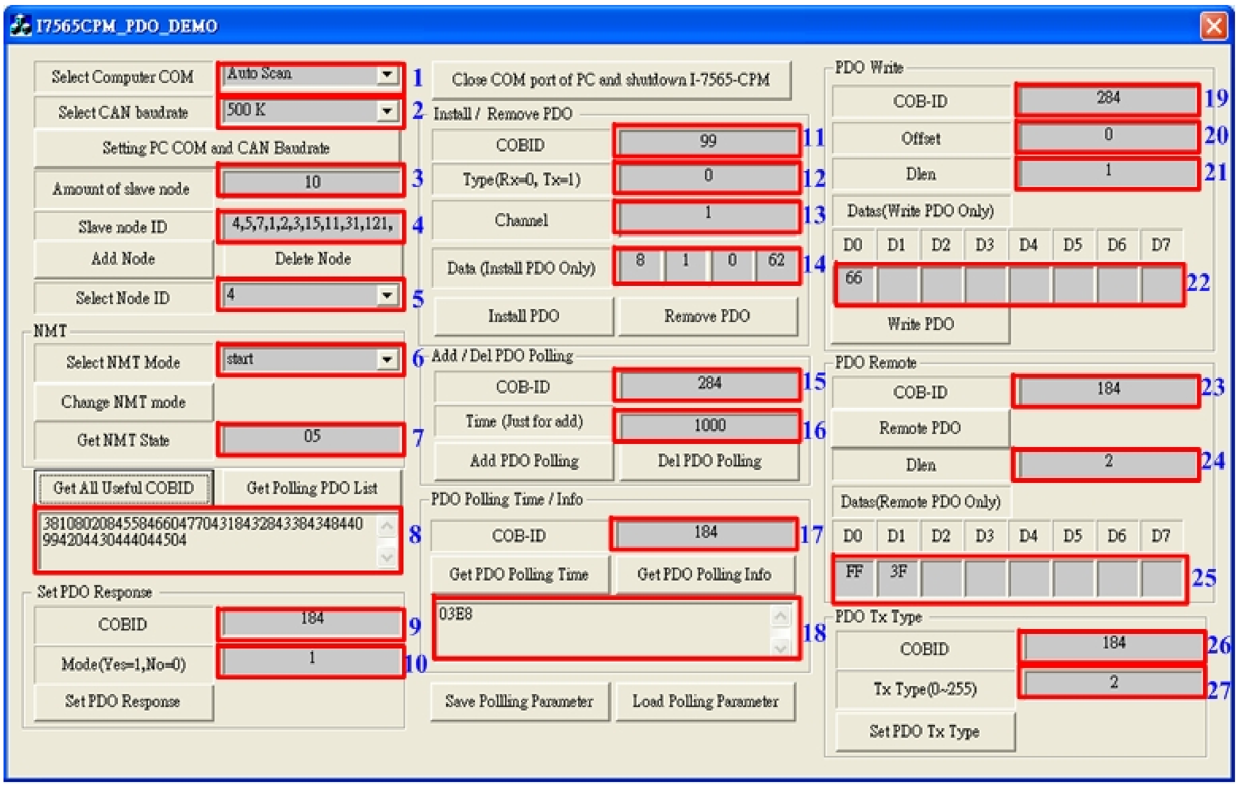
Setting Heartbeat:

If users want to configure the node Heartbeat parameters of a CANopen slave, users have to set the “Producer time” and “Consumer Time” at the position marked with the blue number 17 and 18. Then click the “Heartbeat” button to set these parameters into the CANopen slave immediately

If user wants to stop node Heartbeat of a CANopen slave, please set the “Producer Time” and “Consumer Time” value to 0 and click “Heartbeat” button to apply it.

I7565CPM_PDO_DEMO

When the demo runs, the user interface of the demo is shown below.


I7565CPM_PDO_DEMO

Select Computer COM: Auto Scan (1)
Select CAN baudrate: 500 K (2)
Amount of slave node: 10 (3)
Slave node ID: 4,5,7,1,2,3,15,11,31,121, (4)
Select Node ID: 4 (5)
Select NMT Mode: start (6)
Get NMT State: 05 (7)
Get All Useful COBID / Get Polling PDO List: 381080208455846604770431843284338434844099420443044404450 4 (8)
Set PDO Response — COBID: 184 (9); Mode(Yes=1,No=0): 1 (10)
Install / Remove PDO — COBID: 99 (11); Type(Rx=0, Tx=1): 0 (12); Channel: 1 (13); Data (Install PDO Only): 8 1 0 62 (14)
Add / Del PDO Polling — COB-ID: 284 (15); Time (Just for add): 1000 (16)
PDO Polling Time / Info — COB-ID: 184 (17); 03E8 (18)
PDO Write — COB-ID: 284 (19); Offset: 0 (20); Dlen: 1 (21); Datas(Write PDO Only) D0: 66 (22)
PDO Remote — COB-ID: 184 (23); Dlen: 2 (24); Datas(Remote PDO Only) D0: FF, D1: 3F (25)
PDO Tx Type — COBID: 184 (26); Tx Type(0~255): 2 (27)

Setting PC COM port and CAN baud rate:

User can select the PC COM port and CAN baud rate at the location marked with blue number 1 and 2, and then push the “Setting PC COM and CAN Baudrate” button. If the PC COM port has be opened successfully, the “Amount of slave node” and “Slave node ID” will present the amount of CANopen slaves and list all the ID of the CANopen slaves at the location marked with the blue number 3 and 4.

Add Node and Delete Node:

After the steps described above, users have to click the “Add Node” button to add all of the scanned CANopen slaves into the I-7565-CPM management list. Afterwards, the NMT states of these CANopen slaves will be changed to operational mode, and the “Select Node ID” at the position marked with blue number 5 will list the node ID of the CANopen slaves which are in the I-7565-CPM management list. **Before users want to do any configuration or setting, select the target slave in this field firstly.** If users click “Delete Node” button the I-7565-CPM module will remove all nodes in the I-7565-CPM management list.

Change NMT:

After selecting the target node, users can firstly set the new parameters of NMT at the position marked with the blue number 6. Then click the “Change NMT mode” to change it (note: After user click the “Change NMT mode”, the NMT response of the CANopen slave will be shown in the “Get NMT State” at the position marked with the blue number 7).

Get All Useful COBID:

After selecting the target node, users can click “Get All Useful COBID” to get the node ID information. This button is at the position marked with the blue number 8. The response may be as follows:

3810802084558466047704318432843384348440994204430444044504

Numbe	Description
1	SYNC COB-ID
2	EMCY COB-ID
3	TxPDO COB-ID
4	RxPDO COB-ID
5	TxSDO COB-ID
6	RxSDO COB-ID
7	NMT COB-ID

38(Hex) $\longrightarrow$ Total byes

1 $\longrightarrow$ SYNC COBID = 080

2 $\longrightarrow$ EMCY COBID = 084

3 $\longrightarrow$ TxPDO COBID = 184, 284, 384, 484

4 $\longrightarrow$ RxPDO COBID = 099, 204, 304, 404, 504

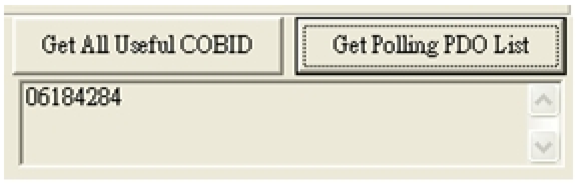
5 $\longrightarrow$ TxSDO COBID = 584

6 $\longrightarrow$ RxSDO COBID = 604

7 $\longrightarrow$ NMT COBID = 704

Get Polling PDO List:

After selecting the target node, users can click the “Get Polling PDO List” button at the position marked with the blue number 8. The response may be as follows:



```
06184284
```

06184286

06(Hex) $\longrightarrow$ Total byes

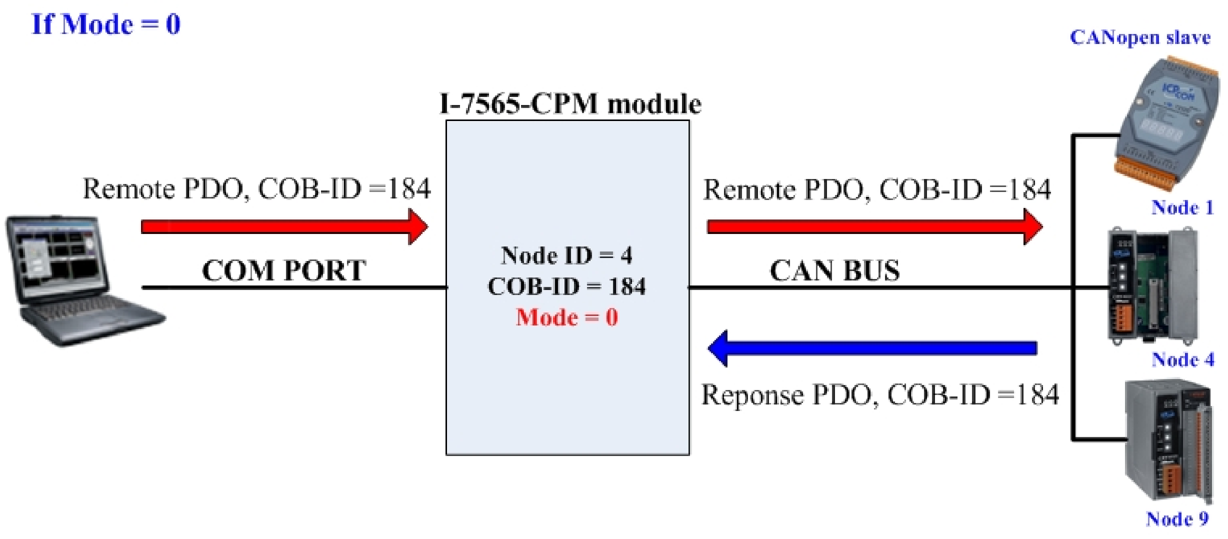
184 $\longrightarrow$ COBID of PDO Polling

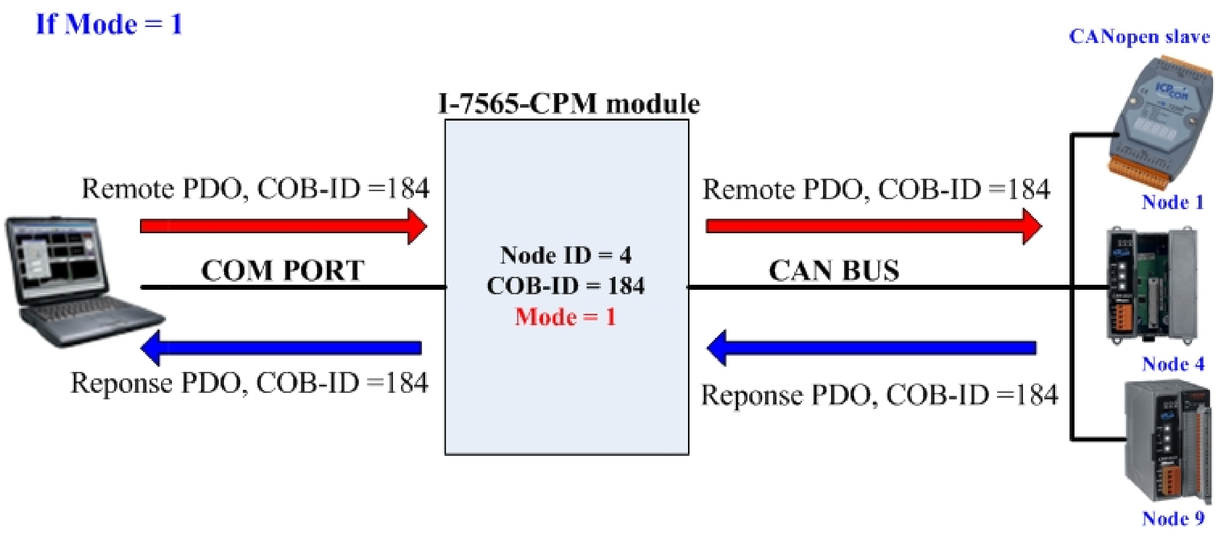
284 $\longrightarrow$ COBID of PDO Polling

Set PDO Response:

If users want to control the response mode of the PDO, use this function to do it. If set the mode to 0, there is no response if users send a remote transmit request PDO. To use this function, set COB-ID and mode at the position marked with the blue number 9 and 10. Then click the "Set PDO Response" button to apply the setting. The different modes of the response are shown as following figure.

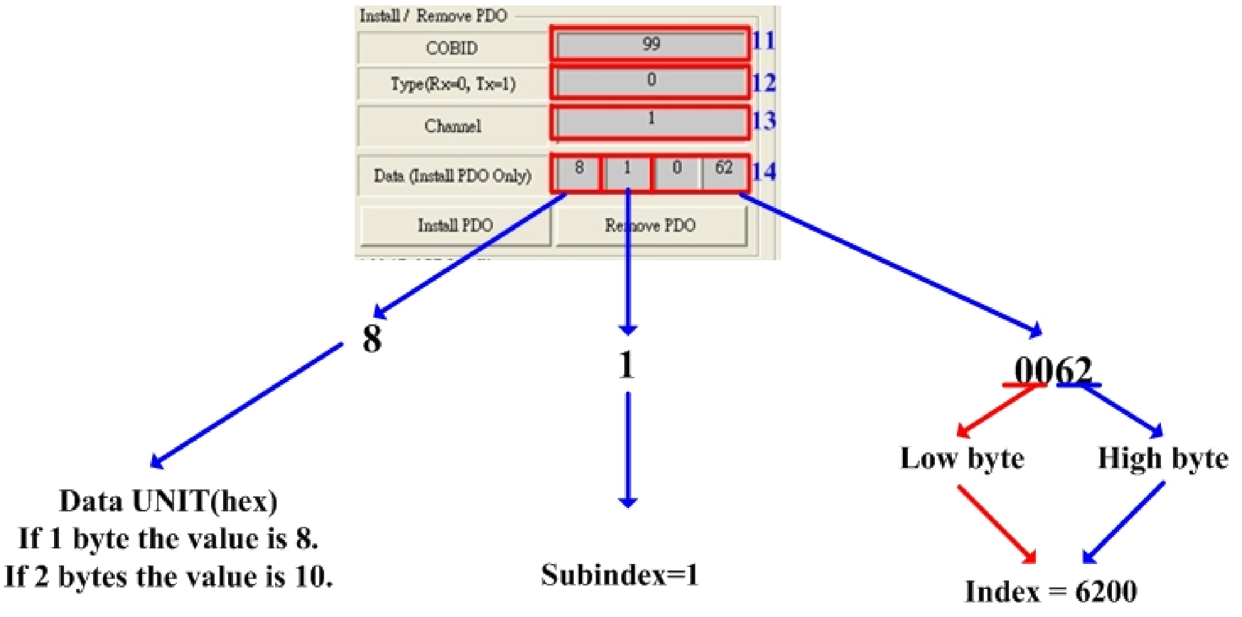
Set PDO Response





Install PDO:

If user wants to dynamically add a PDO, users can use this function to create it. User have to input the COB-ID, PDO type, the channel value and the Data at the position marked with the blue number 11, 12, 13 and 14 respectively, then click the "Install PDO" button to apply these parameters. The channel value is the sub-index of the PDO mapping object. The range of this value is from 1 to 8. For example, if you want to add an object into the sub-index 1 of the PDO mapping object, set this value to 1. The format of the data filled in the Data filed is hexadecimal format, and users must follow the following rules. For example, the first number "8" of the data field indicates 8 bits (1 byte). If users want to set the data format to 2 bytes, the value is 10 (that is 16 bits.).

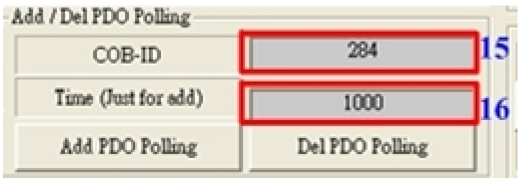


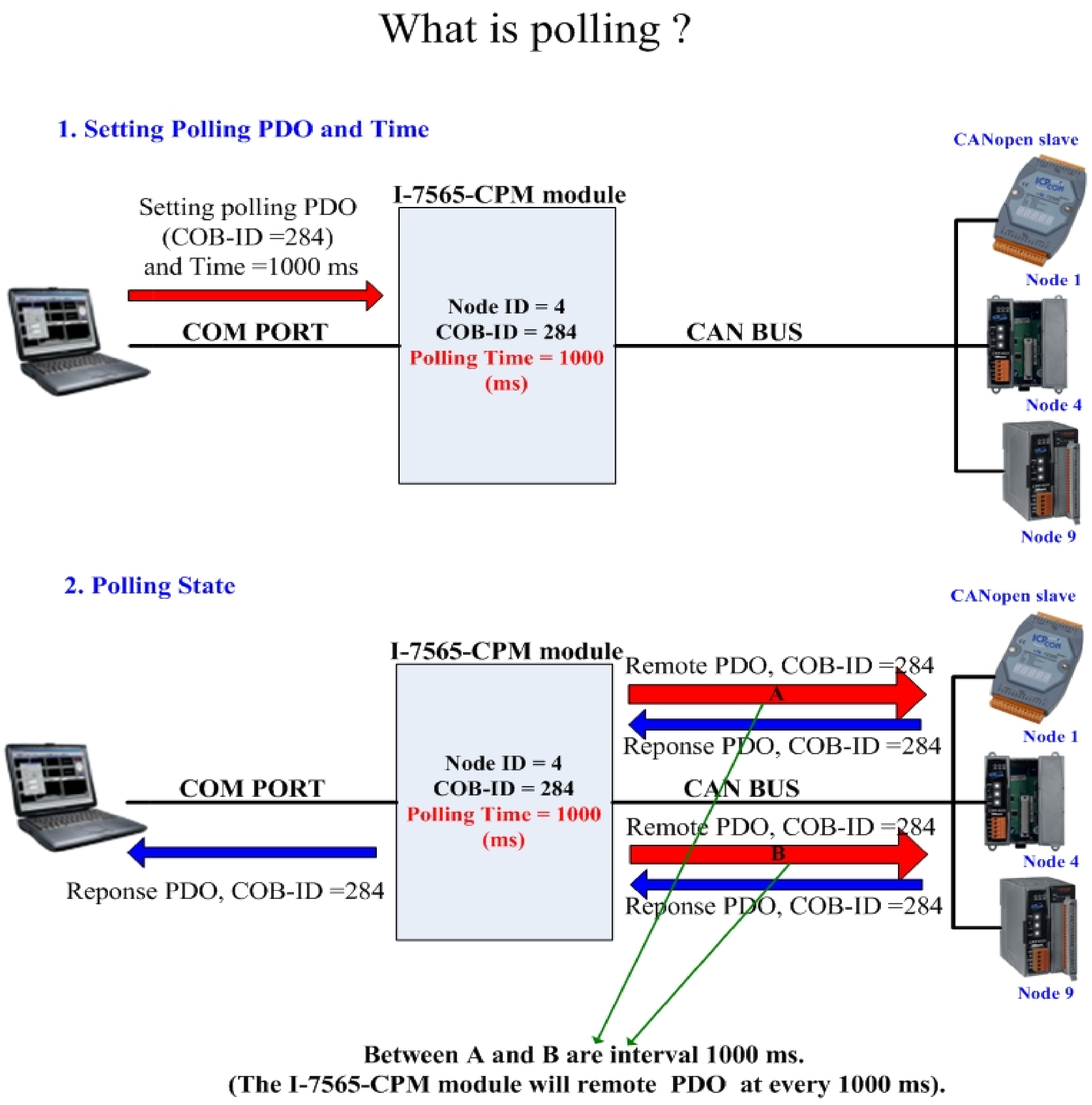
Remove PDO:

If user wants to remove a PDO, users can use this function to delete it. User have to keep them the node ID, the COBID, the PDO type and the channel's value at the position marked with the blue number 11, 12, 13 and 14 respectively, then click the "Remove PDO" button. The channel value is the sub-index of the PDO mapping object. The range of this value is from 0 to 8. The value 0 means that this PDO will be removed. The other means that only the specified sub-index of the PDO mapping objects is removed. The Data field is useless in this case.

Add PDO Polling:

If users want to dynamically add a PDO into polling mode, users can use this function to add it. User have to input the COBID and the time period(the value unit is ms) at the position marked with blue number 15 and 16, then click the "Add PDO Polling" button. (Note: when the I-7565-CPM reboots, the polling mode will be stop. Users can use I7565C_LoadPollingPara function to load it from the EEPROM of the I-7565-CPM if users had used the function I7565C_SavePollingPara to save the polling object of PDO or SDO before)



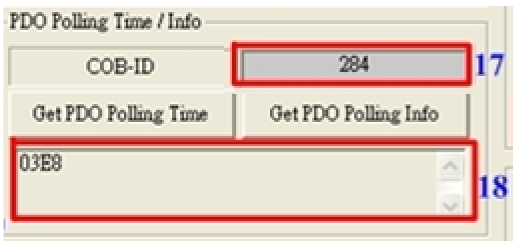


Delete PDO Polling:

If users want to remove the polling mode of a PDO, user can use this function to remove it. Users have to input the COBID at the position marked with the blue number 15. Then click the “Del PDO Polling” button to delete it.

Getting PDO Polling Time:

If users want to get the parameter of the PDO polling time, use this function to get it. Users have to input the COBID at the position marked with the blue number 17, and then click the “Get PDO Polling Time” button. If the PDO had set the polling time before, the value will be shown in the EDIT at the position marked with the blue number 18, as follows.



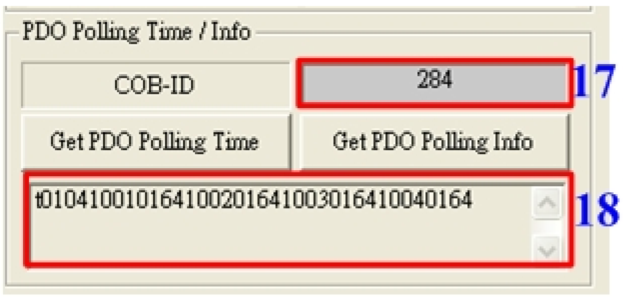
COB-ID: 284

03E8

Get PDO Polling Time = 03E8 (hex) = 1000(Dec)

Getting PDO Polling Info:

To get the PDO polling information, use this function to achieve it. Users have to input the COBID at the position marked with the blue number 19, and then click the “Get PDO Polling Info” button. If the PDO had set the polling information before, the response will be shown in the EDIT box at the position marked with the blue number 20, as follows.



284

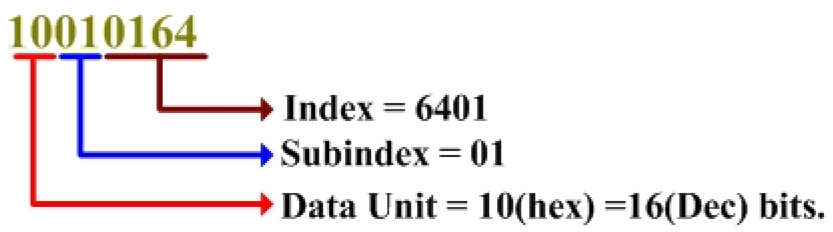
```
t0104100101641002016410030164100401 64
```

t0104100101641002016410030164100401 64

t = TxPDO Object (r = RxPDO)

01 = PDO number. If there are four PDO objects of RxPDO, the COBID respectively are 184, 284, 384, 484 then the COBID 184 = 00, 284= 01, 384= 02, 484 =03.

04 = The amount of PDO mapping objects.
If the value is 01, there is 1 object be mapped of the PDO.
If the value is 02, there are 2 objects be mapped of the PDO.
If the value is 03, there are 3 objects be mapped of the PDO.
If the value is 04, there are 4 objects be mapped of the PDO.



Save Polling Parameters:

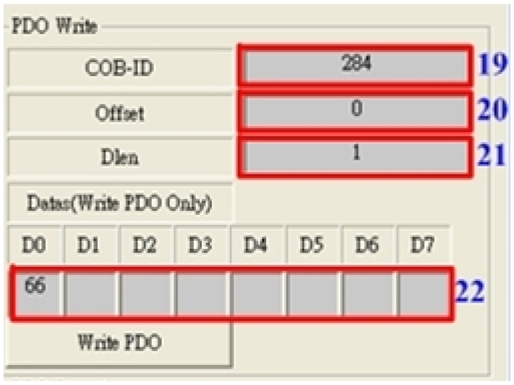
If users want to save the polling configuration of the PDO or SDO, users can click the “Save Polling Parameters” button to save them into the EEPROM of the I-7565-CPM.

Load Polling Parameters:

If users want to load the polling configuration of the PDO or SDO from the EEPROM of the I-7565-CPM, users can click the “Load Polling Parameters” button.

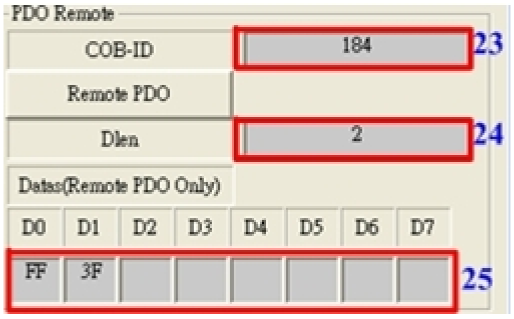
Write PDO:

If users want to write PDO objects, users can use this function to write it. Users have to input the COB-ID, the Offset, the data length, and the data value at the position marked with the blue number 19, 20, 21 and 22 respectively. Then click the "Write PDO" button to send the PDO value.



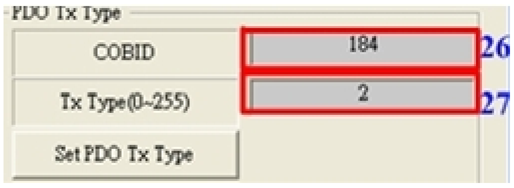
Remote PDO:

If users want to remote PDO objects, users can use this function to remote it. Users have to input the COB-ID, and then click the "Remote PDO" button to read the PDO value. If the response value is not null, the data length and data value will be shown at the position marked with the blue number 24 and 25 respectively, as follows.



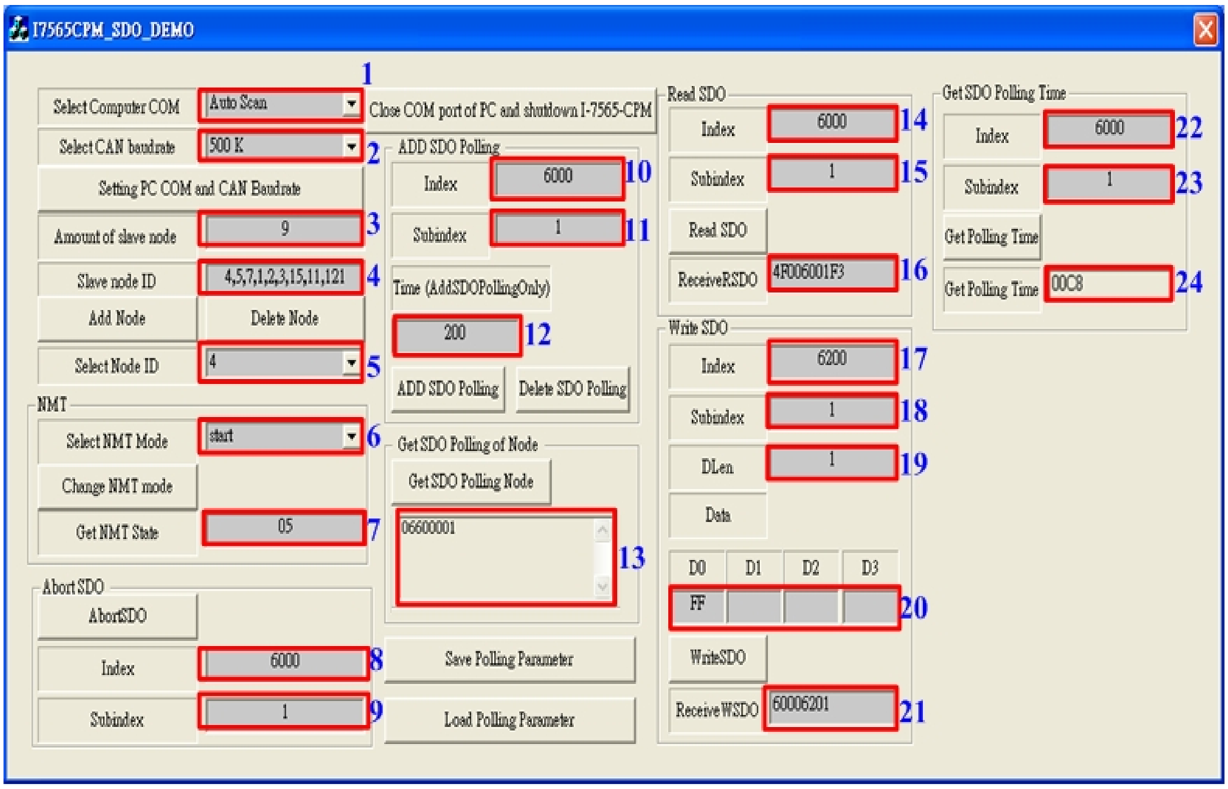
Set PDO Tx Type

If users want to set the transmission type of the TxPDO, users can use this function to set it. Users have to input the COB-ID, and the TxPDO type at the position marked the blue number 26 and 29 respectively. Then click the "Set PDO Tx Type" button to set the PDO type, as follows.



I7565CPM_SDO_DEMO

When the demo runs, the user interface of the demo is shown below.


I7565CPM_SDO_DEMO

Select Computer COM: Auto Scan (1)
Select CAN baudrate: 500 K (2)
Amount of slave node: 9 (3)
Slave node ID: 4,5,7,1,2,3,15,11,121 (4)
Select Node ID: 4 (5)
NMT
Select NMT Mode: start (6)
Get NMT State: 05 (7)
Abort SDO
Index: 6000 (8)
Subindex: 1 (9)
ADD SDO Polling
Index: 6000 (10)
Subindex: 1 (11)
Time (AddSDOPollingOnly): 200 (12)
Get SDO Polling of Node: 06600001 (13)
Read SDO
Index: 6000 (14)
Subindex: 1 (15)
ReceiveRSDO: 4F006001F3 (16)
Write SDO
Index: 6200 (17)
Subindex: 1 (18)
DLen: 1 (19)
Data: D0 FF, D1, D2, D3 (20)
ReceiveWSDO: 60006201 (21)
Get SDO Polling Time
Index: 6000 (22)
Subindex: 1 (23)
Get Polling Time: 00C8 (24)

Setting PC COM port and CAN baud rate:

User can select the PC COM port and CAN baud rate at the location marked with blue number 1 and 2, and then push the "Setting PC COM and CAN Baudrate" button. If the PC COM port has be opened successfully, the "Amount of slave node" and "Slave node ID" will present the amount of CANopen slaves and list all the ID of the CANopen slaves at the location marked with the blue number 3 and 4.

Add Node and Delete Node:

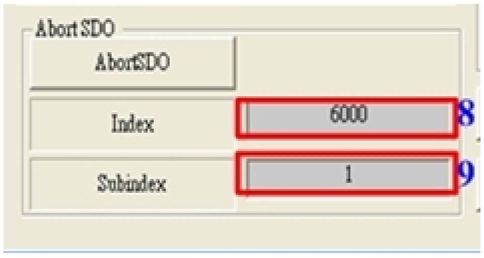
After the steps described above, users have to click the "Add Node" button to add all of the scanned CANopen slaves into the I-7565-CPM management list. Afterwards, the NMT states of these CANopen slaves will be changed to operational mode, and the "Select Node ID" at the position marked with blue number 5 will list the node ID of the CANopen slaves which are in the I-7565-CPM management list. **Before users want to do any configuration or setting, select the target slave in this field firstly.** If users click "Delete Node" button the I-7565-CPM module will remove all nodes in the I-7565-CPM management list.

Change NMT:

After selecting the target node, users can firstly set the new parameters of NMT at the position marked with the blue number 6. Then click the "Change NMT mode" to change it (note: After user click the "Change NMT mode", the NMT response of the CANopen slave will be shown in the "Get NMT State" at the position marked with the blue number 7).

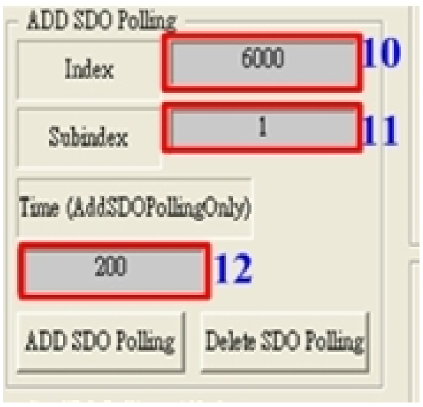
Abort SDO:

When users want to stop the communication of the TxSDO or RxSDO, user can use this function to break it. Users have to input the index and sub-index at the position marked with the blue number 8 and 9 respectively. Then click the "AbortSDO" button, as follows.



ADD SDO Polling:

If users want to add a SDO into polling mode, use this function to add it. Users have to input the index of the target object, the sub-index of the target object, and the Time (the unit is ms) value at the position marked with the blue number 10, 11and 12 respectively, and then click the "Add SDO Polling" button. (Note: when the I-7565-CPM reboots, the polling mode will be stop. Users can use I7565C_LoadPollingPara function to load it from the EEPROM of the I-7565-CPM if users had used the function I7565C_SavePollingPara to save the polling object of PDO or SDO before)

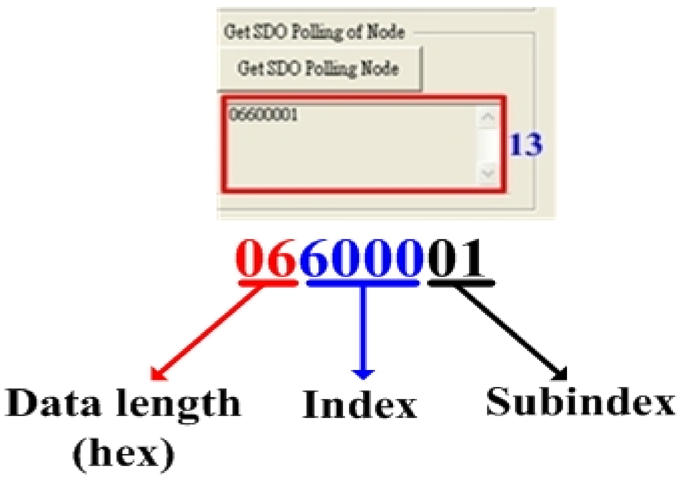


Delete SDO Polling:

If users want to remove a PDO from the polling mode, use this function to remove it. Users have to input the node ID, the index and the sub-index at the position marked with the blue number 10 and 11 respectively. Then click the “Delete SDO Polling” button to delete it, as above figure.

Getting SDO Polling Node:

If users want to get the SDO polling information of the specified node, use this function to get it. Users have to select node ID, and then click the “Get PDO Polling Node” button. If the SDO had set the polling time, the response will be shown in the EDIT box at the position marked with the blue number 13, as follows.



Save Polling Parameters:

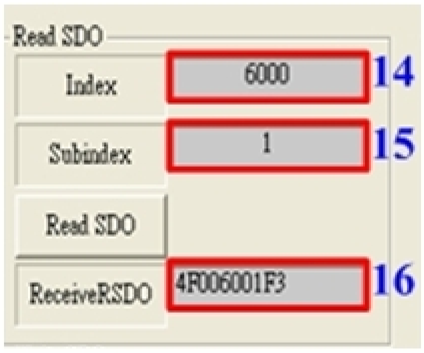
If users want to save the polling configuration of the PDO or SDO, users can click the “Save Polling Parameters” button to save them into the EEPROM of the I-7565-CPM.

Load Polling Parameters:

If users want to load the polling configuration of the PDO or SDO from the EEPROM of the I-7565-CPM, users can click the “Load Polling Parameters” button.

Read SDO:

If users want to read SDO objects of a node, use this function to get it. Users have to input the index of the target object, and the sub-index of the target object at the position marked with the blue number 14 and 15 respectively. Then click the "Read SDO" button. If the SDO is read successfully, the received data (8 bytes max) are shown at the position marked with the blue number 16, as follows.



ReceiveRSDO = Receive Read SDO Datas(8 bytes max)

4F006001F3

4F = byte 0

00 = byte 1

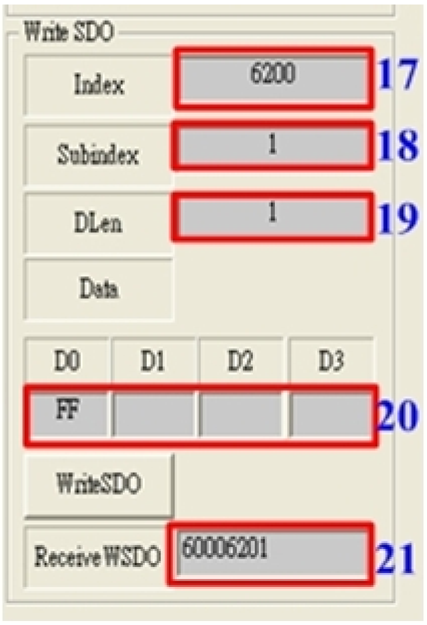
60 = byte 2

01 = byte 3

F3 = byte 4

Write SDO:

If users want to write a SDO objects to a CANopen slave, use this function to write it. Users have to input the index of the target object, the sub-index of the target object, the data length and the data at the position marked with the blue number 17, 18, 19 and 20 respectively. Then click the "Write SDO" button. If the SDO is written successfully, the response of the SDO is shown at the position marked with the blue number 21, as follows.



ReceiveWSDO = Receive Write SDO Datas(8 bytes max)

60006201

60 = byte 0

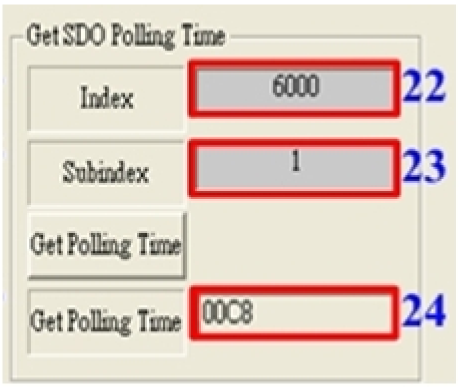
00 = byte 1

62 = byte 2

01 = byte 3

Get SDO Polling Time:

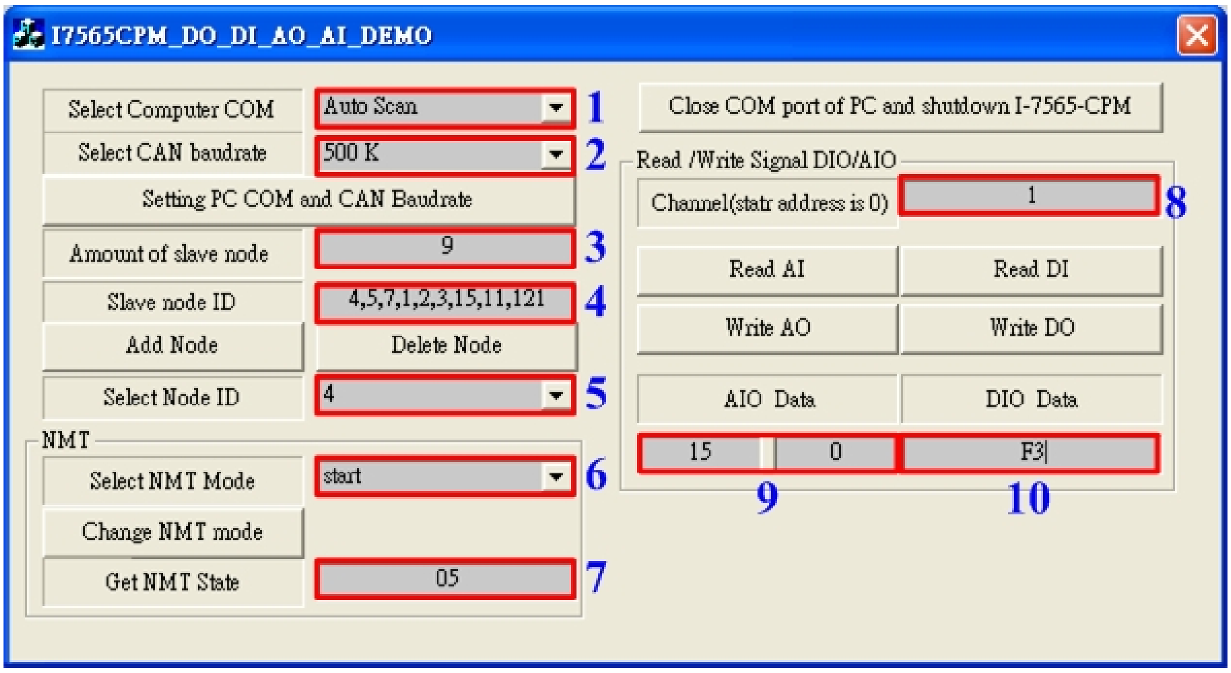
If users want to get a polling time of a SDO object of a node, use this function to get it. Users have to input the node ID, the index of the target object, and the sub-index of the target object at the position marked with the blue number 22 and 23 respectively. Then click the "Get Polling Time" button. If it is successful, the value with hexadecimal format of the polling time is displayed at the position marked with the blue number 24.



Get Polllling Time = 00C8 (hex) = 200 (ms).

I7565CPM_DO_DI_AO_AI_DEMO

When the demo runs, the user interface of the demo is shown below.



Setting PC COM port and CAN baud rate:

User can select the PC COM port and CAN baud rate at the location marked with blue number 1 and 2, and then push the “Setting PC COM and CAN Baudrate” button. If the PC COM port has be opened successfully, the “Amount of slave node” and “Slave node ID” will present the amount of CANopen slaves and list all the ID of the CANopen slaves at the location marked with the blue number 3 and 4.

Add Node and Delete Node:

After the steps described above, users have to click the “Add Node” button to add all of the scanned CANopen slaves into the I-7565-CPM management list. Afterwards, the NMT states of these CANopen slaves will be changed to operational mode, and the “Select Node ID” at the position marked with blue number 5 will list the node ID of the CANopen slaves which are in the I-7565-CPM management list. **Before users want to do any configuration or setting, select the target slave in this field firstly.** If users click “Delete Node” button the I-7565-CPM module will remove all nodes in the I-7565-CPM management list.

Change NMT:

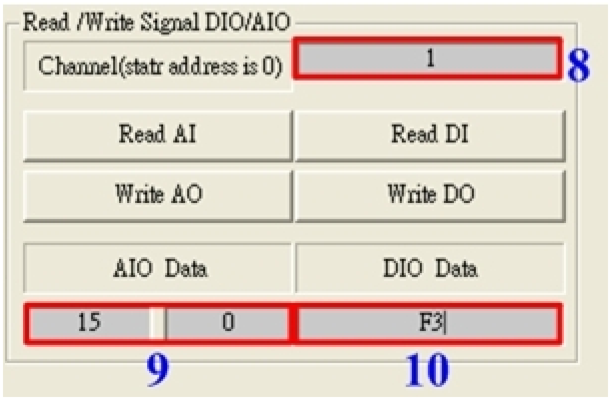
After selecting the target node, users can firstly set the new parameters of NMT at the position marked with the blue number 6. Then click the "Change NMT mode" to change it (note: After user click the "Change NMT mode", the NMT response of the CANopen slave will be shown in the "Get NMT State" at the position marked with the blue number 7).

Close COM and shutdown I-7565-CPM:

Users can click the "Close COM port of PC and shutdown the I-7565-CPM" button to close the computer's COM port and shutdown the I-7565-CPM.

Read DI / AI:

If users want to read DI or AI data from a CANopen slave, use this function to do that. Users have to input the sub-index at the position marked with the blue number 8 respectively. Then click the "Read AI" or "Read DI" button. The value of the index starts from 1. For AI channels, here is the index 0x6401. For DI channels, the index is 0x6000. If it is successful, the AI data and the DI data are shown at the position marked with the blue number 9 and 10.



Read DO / AO:

If users want to write the value to the DO or AO channel, use this function to write it. Users have to input the sub-index, and the data at the position marked with the number 8, 9 and 10 respectively. If users want to write the AO value, please fill the AO value in the EDIT box at the position marked with the blue number 9. If the value is the DI value, fill the data in the EDIT box at the position marked with the blue number 10. The sub-index starts from 1. For AO channels, the index is 0x6411. For DO channels the index is 0x6200.