

---

# I-8123W CANopen Master Module

## User's Manual

### **Warranty**

All products manufactured by ICP DAS are warranted against defective materials for a period of one year from the date of delivery to the original purchaser.

### **Warning**

ICP DAS assume no liability for damages consequent to the use of this product. ICP DAS reserves the right to change this manual at any time without notice. The information furnished by ICP DAS is believed to be accurate and reliable. However, no responsibility is assumed by ICP DAS for its use, nor for any infringements of patents or other rights of third parties resulting from its use.

### **Copyright**

Copyright 2010 by ICP DAS. All rights are reserved.

### **Trademark**

The names used for identification only maybe registered trademarks of their respective companies.

---

## Tables of Content

|  |           |
|--|-----------|
| <b>1. General Information</b> .....                            | <b>6</b>  |
| <b>1.1. CANopen Introduction</b> .....                         | <b>6</b>  |
| <b>1.2. CANopen Applications</b> .....                         | <b>7</b>  |
| <b>1.3. I-8123 Library Characteristics</b> .....               | <b>8</b>  |
| <b>2. Hardware Specification</b> .....                         | <b>11</b> |
| <b>2.1. Hardware Structure</b> .....                           | <b>11</b> |
| <b>2.2. Wire Connection</b> .....                              | <b>12</b> |
| <b>2.3. Power LED</b> .....                                    | <b>14</b> |
| <b>2.4. Tx/Rx LED</b> .....                                    | <b>14</b> |
| <b>2.5. ERR LED</b> .....                                      | <b>14</b> |
| <b>3. I-8123 Function Library</b> .....                        | <b>15</b> |
| <b>3.1. Function List</b> .....                                | <b>15</b> |
| <b>3.2. Function Return Code</b> .....                         | <b>18</b> |
| <b>3.3. CANopen Master Library Application Flowchart</b> ..... | <b>20</b> |
| <b>3.4. Communication Services Introduction</b> .....          | <b>22</b> |
| <b>3.5. Function Description</b> .....                         | <b>25</b> |
| 3.5.1. I8123_GetVersion .....                                  | 25        |
| 3.5.2. I8123_SetFunctionTimeout .....                          | 26        |
| 3.5.3. I8123_GetCANStatus .....                                | 27        |
| 3.5.4. I8123_InitMaster .....                                  | 28        |
| 3.5.5. I8123_ShutdownMaster.....                               | 29        |
| 3.5.6. I8123_MasterSendBootupMsg .....                         | 30        |
| 3.5.7. I8123_SetMasterMode.....                                | 31        |
| 3.5.8. I8123_GetMasterMode .....                               | 32        |
| 3.5.9. I8123_GetFirmwareVersion.....                           | 33        |
| 3.5.10. I8123_EDS_Load .....                                   | 34        |
| 3.5.11. I8123_AddNode.....                                     | 35        |
| 3.5.12. I8123_RemoveNode .....                                 | 37        |
| 3.5.13. I8123_RemoveAndResetNode .....                         | 38        |
| 3.5.14. I8123_DelayAndResponseTimeout.....                     | 39        |
| 3.5.15. I8123_ScanNode .....                                   | 40        |
| 3.5.16. I8123_GetNodeList .....                                | 41        |
| 3.5.17. I8123_NMTChangeState.....                              | 42        |
| 3.5.18. I8123_NMTGetState .....                                | 43        |
| 3.5.19. I8123_NMTGuarding.....                                 | 44        |
| 3.5.20. I8123_NMTHeartbeat .....                               | 45        |

---

|   |    |
|---|----|
| 3.5.21. I8123_SDOReadData .....           | 46 |
| 3.5.22. I8123_SDOReadFile .....           | 47 |
| 3.5.23. I8123_SDOWriteData .....          | 48 |
| 3.5.24. I8123_SDOAbortTransmit .....      | 49 |
| 3.5.25. I8123_PDOWrite .....              | 50 |
| 3.5.26. I8123_PDOWrite_Fast .....         | 51 |
| 3.5.27. I8123_PDORemote .....             | 52 |
| 3.5.28. I8123_PDORemote_Fast.....         | 53 |
| 3.5.30. I8123_SetPDORemotePolling.....    | 54 |
| 3.5.31. I8123_GetPDOLastData.....         | 55 |
| 3.5.32. I8123_GetPDOLastData_Fast.....    | 56 |
| 3.5.33. I8123_GetMultiPDOData.....        | 57 |
| 3.5.34. I8123_GetMultiPDOData_Fast.....   | 58 |
| 3.5.35. I8123_GetRxPDOID .....            | 59 |
| 3.5.36. I8123_GetTxPDOID .....            | 60 |
| 3.5.37. I8123_InstallPDO .....            | 61 |
| 3.5.38. I8123_DynamicPDO .....            | 62 |
| 3.5.39. I8123_RemovePDO.....              | 63 |
| 3.5.40. I8123_ChangePDOID .....           | 64 |
| 3.5.41. I8123_GetPDOMapInfo .....         | 65 |
| 3.5.42. I8123_InstallPDO_List.....        | 66 |
| 3.5.43. I8123_RemovePDO_List .....        | 68 |
| 3.5.44. I8123_PDOWUseEntry.....           | 69 |
| 3.5.45. I8123_PDOTxType .....             | 70 |
| 3.5.46. I8123_PDOWEventTimer.....         | 71 |
| 3.5.47. I8123_PDOWInhibitTime.....        | 72 |
| 3.5.48. I8123_ChangeSYNCCID.....          | 73 |
| 3.5.49. I8123_SetSYNCC_List.....          | 74 |
| 3.5.50. I8123_GetSYNCCID .....            | 75 |
| 3.5.51. I8123_SendSYNCCMsg .....          | 76 |
| 3.5.52. I8123_GetCyclicSYNCCInfo .....    | 77 |
| 3.5.53. I8123_ChangeEMCCYID.....          | 78 |
| 3.5.54. I8123_SetEMCCY_List.....          | 79 |
| 3.5.55. I8123_GetEMCCYID .....            | 80 |
| 3.5.56. I8123_ReadLastEMCCY .....         | 81 |
| 3.5.57. I8123_GetBootUpNodeAfterAdd ..... | 82 |
| 3.5.58. I8123_GetEMCCYData .....          | 83 |
| 3.5.59. I8123_GetNMTErrror.....           | 84 |

---

|   |            |
|---|------------|
| 3.5.60. I8123_InstallBootUpISR .....        | 85         |
| 3.5.61. I8123_RemoveBootUpISR.....          | 86         |
| 3.5.62. I8123_InstallEMCYISR .....          | 87         |
| 3.5.63. I8123_RemoveEMCYISR.....            | 88         |
| 3.5.64. I8123_InstallNMTErrISR.....         | 89         |
| 3.5.65. I8123_RemoveNMTErrISR .....         | 90         |
| 3.5.66. I8123_GetMasterReadSDOEvent .....   | 91         |
| 3.5.67. I8123_GetMasterWriteSDOEvent.....   | 92         |
| 3.5.68. I8123_ResponseMasterSDO .....       | 93         |
| 3.5.69. I8123_InstallReadSDOISR .....       | 94         |
| 3.5.70. I8123_RemoveReadSDOISR .....        | 95         |
| 3.5.71. I8123_InstallWriteSDOISR.....       | 96         |
| 3.5.72. I8123_RemoveWriteSDOISR .....       | 97         |
| 3.5.73. I8123_GetMasterRemotePDOEvent ..... | 98         |
| 3.5.74. I8123_GetMasterRxPDOEvent .....     | 99         |
| 3.5.75. I8123_ResponseMasterPDO .....       | 100        |
| 3.5.76. I8123_InstallRxPDOISR .....         | 101        |
| 3.5.77. I8123_RemoveRxPDOISR.....           | 102        |
| 3.5.78. I8123_InstallRemotePDOISR .....     | 103        |
| 3.5.79. I8123_RemoveRemotePDOISR.....       | 104        |
| 3.5.80. I8123_SyncGroupCycUpdata.....       | 105        |
| 3.5.81. I8123_SyncGroupCycSend .....        | 106        |
| 3.5.82. I8123_SyncGroupClear .....          | 107        |
| 3.5.83. I8123_GetSYNCGroupStopState .....   | 108        |
| 3.5.84. I8123_InstallSYNCGroupISR.....      | 109        |
| 3.5.85. I8123_RemoveSYNCGroupISR .....      | 110        |
| <b>4. Demo Programs .....</b>               | <b>111</b> |
| <b>4.1. Brief of the demo programs.....</b> | <b>111</b> |
| 4.1.1. Listen_Mode .....                    | 112        |
| 4.1.2. NMT_Protocol.....                    | 113        |
| 4.1.3. PDO_Parameter .....                  | 114        |
| 4.1.4. PDO_Protocol.....                    | 115        |
| 4.1.5. Scan_Node .....                      | 116        |
| 4.1.6. SDO_PDO_ISR .....                    | 117        |
| 4.1.7. SDO_Read .....                       | 118        |
| 4.1.8. SDO_Write .....                      | 119        |
| 4.1.9. SYNC_Protocol .....                  | 120        |
| 4.1.10. PDO_MultiData.....                  | 121        |

---

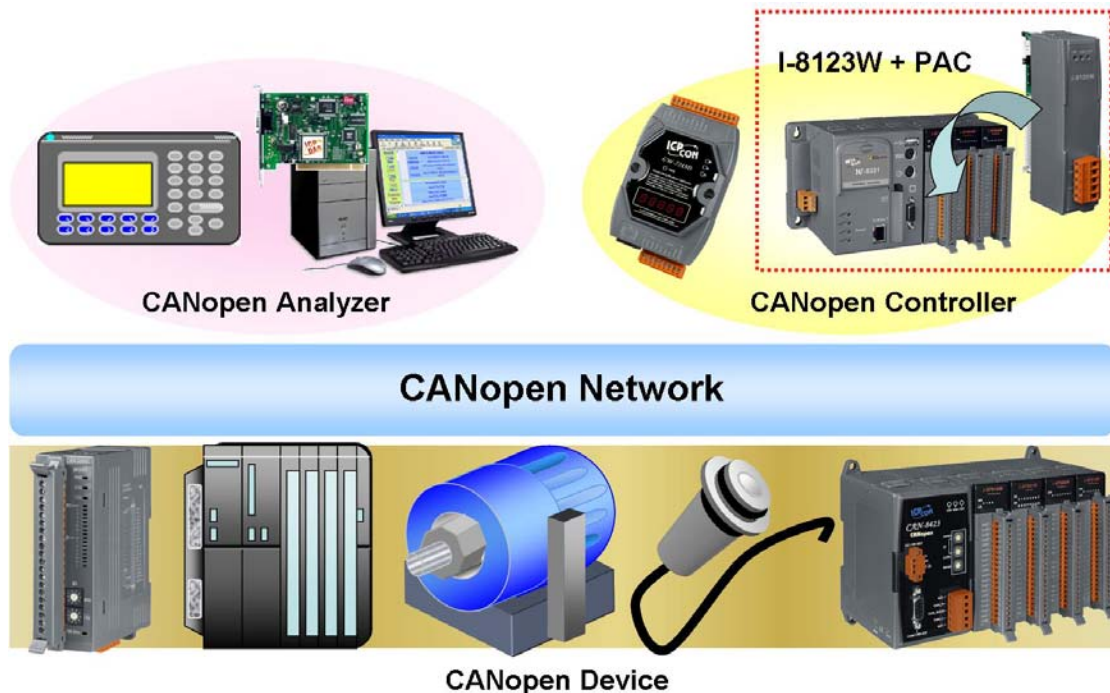
|                                 |            |
|---------------------------------|------------|
| <b>5. Update Firmware .....</b> | <b>122</b> |
|---------------------------------|------------|

---

# 1. General Information

## 1.1. CANopen Introduction

The CAN (Controller Area Network) is a kind of serial communication protocols, 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 the connections between simple industrial devices (sensors, actuators) and higher-level devices (controllers), as shown in the 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 for 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>):

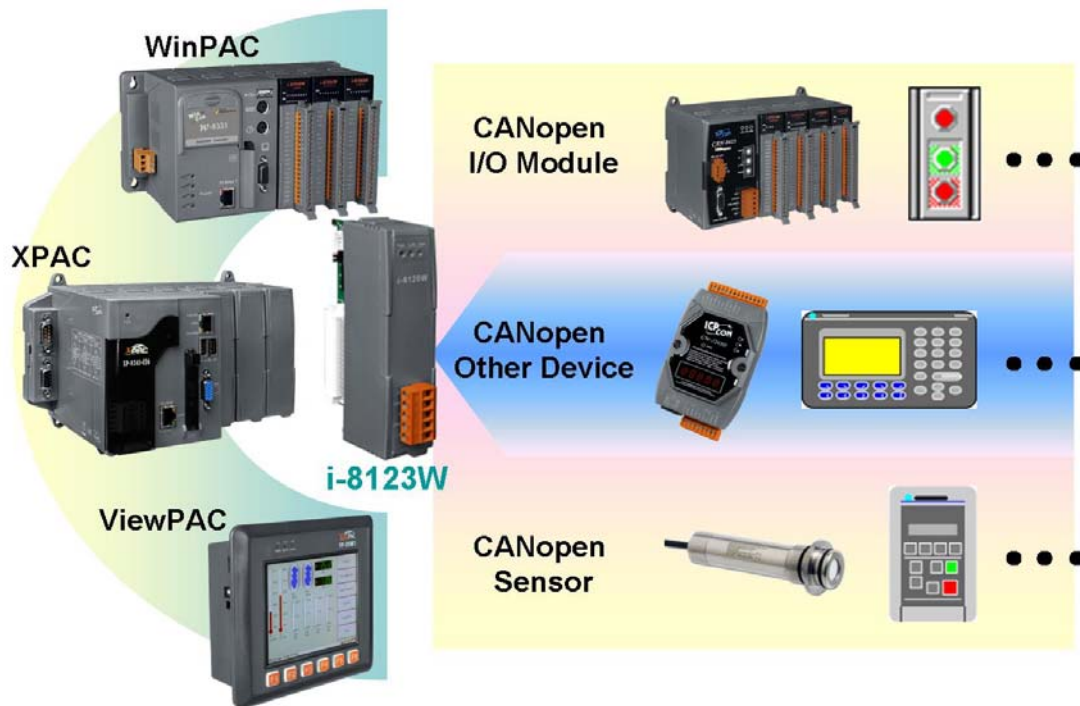
- Truck-based superstructure control systems
- Off-highway and off-road vehicles
- Passenger and cargo trains
- Maritime electronics
- Factory automation
- Industrial machine control
- Lifts and escalators
- Building automation
- Medical equipment and devices
- Non-industrial control
- Non-industrial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



---

### 1.3. I-8123 Library Characteristics

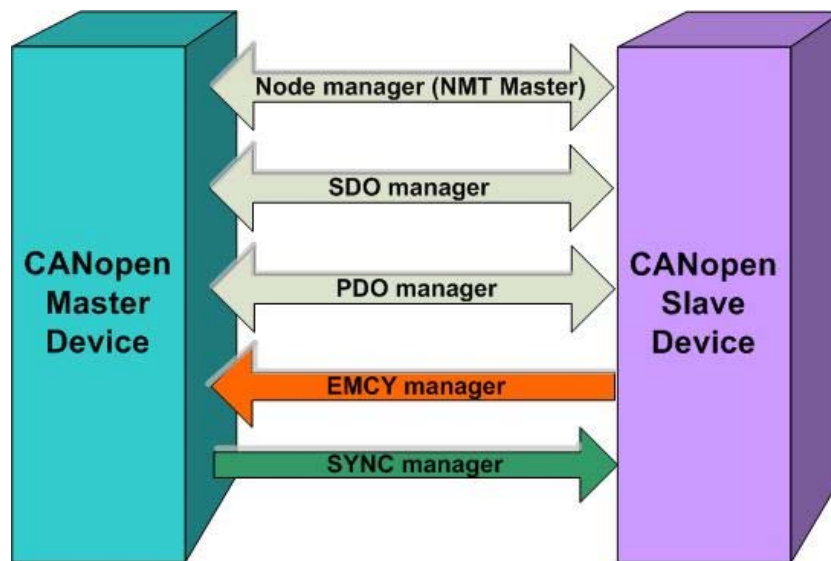
In order to use the I-8123W module, we provide the I-8123W library for the ViewPAC-2xWx, WinPAC-8000, XPAC-8000 and XPAC-8000-CE6 series main control unit, and users can use it establish CANopen communication network rapidly. Most of the CANopen communication protocols, such as PDO, SDO and NMT, will be handled by the library 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 device in the CANopen network. The following figure describes the general application architecture of the I-8123W.



**Figure 1.2 Application architecture**

The I-8123W follows the CiA CANopen specification DS-301 V4.02, 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 and Heartbeat 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 more information about the CANopen functions described above, please refer to the function descriptions and demo programs shown in the chapter 3 and chapter 4.

---

## Specifications

- CPU:80186, 80 MHz
- NXP SJA1000 CAN controller with 16MHz clock
- NXP 82C250 CAN transceiver
- Power LED, Tx/Rx LED, Error LED
- Switch for 120 $\Omega$  terminal resistor of CAN bus.
- CAN bus interface: ISO 11898-2, 5-pin screw terminal
- 2500 Vrms photo-coupled isolation on CAN side
- 1000 Vdc voltage protection
- Power Consumption: 2W
- Operating Temperature: -25 $^{\circ}$ C to +75 $^{\circ}$ C
- Storage Temperature: -30 $^{\circ}$ C to +80 $^{\circ}$ C
- Humidity: 10% ~ 90%, non-condensing

## Features

- One CAN communication port.
- Support PAC series MCU (main control unit) with Windows CE5, CE6, XPe.
- Library provides eVC++, VC++2005, C#.Net2005, and VB.Net2005 developments.
- Support InduSoft and ISaGRAF developments.
- Three indication LEDs (Pwr, Tx/Rx and Err LEDs).
- Support 8 kinds of baud rates: 10 kbps, 20 kbps, 50 kbps, 125 kbps, 250 kbps, 500 kbps, 800 kbps, and 1 Mbps.
- Support the node id range from 1 ~ 127.
- Follow CiA DS-301 V4.02.
- Support upload and download SDO Segment protocol.
- Support Node Guarding and Heartbeat protocol.
- Support EDS file.
- Provide Master Listen Mode to listen the slave status of the CANopen network.
- Block-function and non-block-function selected.
- Demos and utility are provided.

---

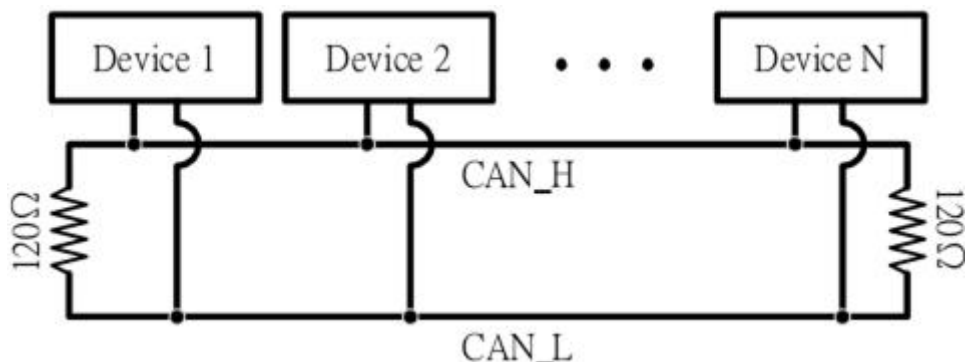
## 2. Hardware Specification

### 2.1. Hardware Structure



## 2.2. Wire Connection

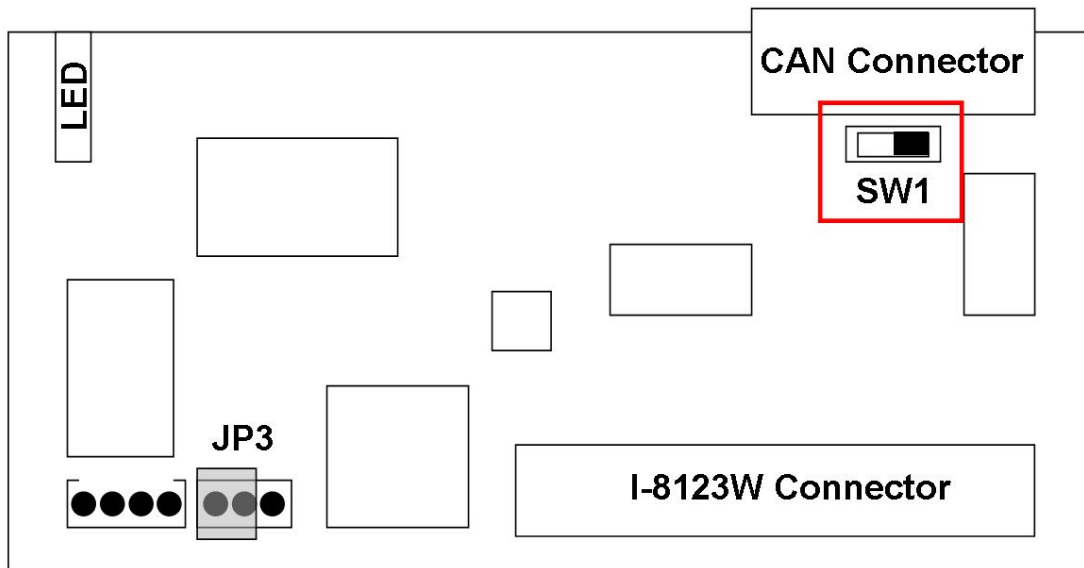
In order to minimize the reflection effects on the CAN bus line, the CAN bus line has to be terminated at both ends by two terminator resistors as in the following figure. According to the ISO 11898-2 spec, each terminator resistor is  $120\Omega$  (or between  $108\Omega\sim 132\Omega$ ). The length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



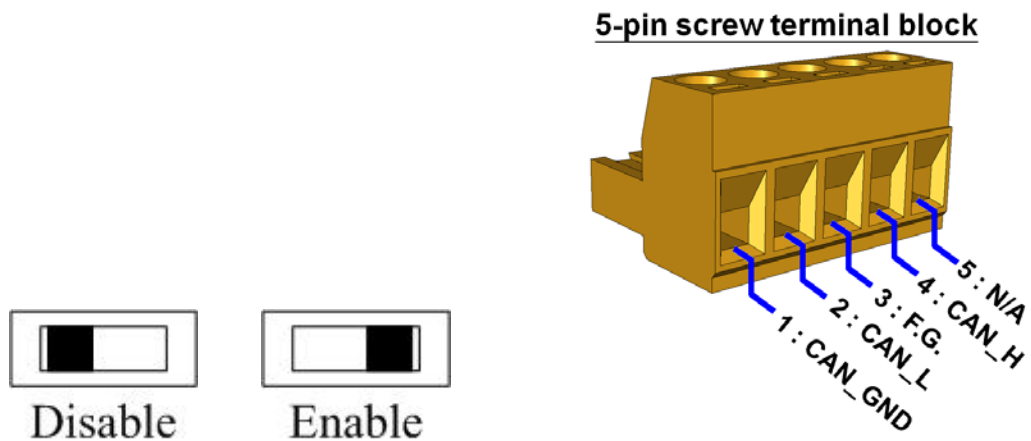
Moreover, to minimize the voltage drop over long distances, the terminator resistor should be higher than the value defined in the ISO 11898-2. The following table can be used as a good reference.

| Bus Length<br>(meter) | Bus Cable Parameters  |   | Terminator<br>Resistor ( $\Omega$ ) |
|-----------------------|---|---|-------------------------------------|
|                       | Length Related<br>Resistance<br>( $\text{m}\Omega/\text{m}$ ) | Cross Section<br>(Type)                   |                                     |
| 0~40                  | 70  | 0.25(23AWG)~0.34<br>$\text{mm}^2$ (22AWG) | 124 (0.1%)                          |
| 40~300                | <60   | 0.34(22AWG)~0.6<br>$\text{mm}^2$ (20AWG)  | 127 (0.1%)                          |
| 300~600               | <40   | 0.5~0.6 $\text{mm}^2$<br>(20AWG)          | 150~300                             |
| 600~1K                | <20   | 0.75~0.8 $\text{mm}^2$<br>(18AWG)         | 150~300                             |

In the I-8123W, the 120Ω terminator resistor is supplied. The SW1 of the I-8123W is for the terminator resistor. Its location is shown in the following figure.



The following connection statuses are presented for the condition if the terminator resistor is enabled or disabled.



SW1 Switch selection.

Pin assignment of I-8123W connector

| Pin No. | Signal         | Description                           |
|---------|----------------|---------------------------------------|
| 1       | <b>CAN_GND</b> | <b>Ground / 0 V / V-</b>              |
| 2       | <b>CAN_L</b>   | <b>CAN_L bus line (dominant low)</b>  |
| 3       | <b>F.G.</b>    | <b>CAN Shield</b>                     |
| 4       | <b>CAN_H</b>   | <b>CAN_H bus line (dominant high)</b> |
| 5       | N/A            | Non-available                         |

---

## **2.3. Power LED**

The I-8123W needs 2W power consumption. If the electric power is supplied normally, the Power LED will be turn on always. If any other situation, please check the power supply or contact to your distributor.

## **2.4. Tx/Rx LED**

Each I-8123W provides Tx/Rx LED to check the situations of the CAN messages transmission and reception. If the I-8123W is transmitting or receiving a CAN message, the Tx/Rx LED will blink. If the bus loading of the I-8123W is heavy, the Tx/Rx LED will be always turned on.

## **2.5. ERR LED**

The ERR LED indicates the error status of the CAN physical layer and indicates the errors due to missing any CAN message.

## 3. I-8123 Function Library

### 3.1. Function List

In order to use the I-8123W more easily, we provide some useful and easy-to-use functions in the I-8123W library. There are several kinds of library versions for the ViewPAC-2xWx, WinPAC-8000, and XPAC-8000 and XPAC-8000-CE6 MCUs. When using the library, the library version must be confirmed. The following table shows the all functions provided by the I-8123W library.

| Listen Mode | Function Name                 | Description   |
|-------------|-------------------------------|---|
| O           | I8123_GetVersion              | Get the version of the I-8123W library                                  |
| O           | I8123_SetFunctionTimeout      | Set the max. timeout value of all functions                             |
| O           | I8123_GetCANStatus            | Get CAN register status of the I-8123W                                  |
| O           | I8123_InitMaster              | Activate the I-8123W module   |
| O           | I8123_ShutdownMaster          | Remove all nodes and stop the master                                    |
| X           | I8123_MasterSendBootupMsg     | Let I-8123W send a NMT boot up message                                  |
| O           | I8123_SetMasterMode           | Set the master to normal mode or listen mode                            |
| O           | I8123_GetMasterMode           | Get operation mode of the master  |
| O           | I8123_GetFirmwareVersion      | Get the version of the I-8123W firmware                                 |
| O           | I8123_EDS_Load                | Add a slave node from the EDS file                                      |
| O           | I8123_AddNode                 | Add a slave node into the I-8123W master manager                        |
| O           | I8123_RemoveNode              | Remove a node from the I-8123W master manager                           |
| X           | I8123_RemoveAndResetNode      | Remove the slave and then reset it from I-8123W                         |
| O           | I8123_DelayAndResponseTimeout | Change minimum interval time and response timeout value of CAN messages |
| X           | I8123_ScanNode                | Scan all nodes on the CANopen network                                   |
| O           | I8123_GetNodeList             | Get the node list of the I-8123W master manager                         |
| X           | I8123_NMTChangeState          | Change the CANopen node state   |
| O           | I8123_NMTGetState             | Get the CANopen node state  |
| O           | I8123_NMTGuarding             | Start to the node guarding function                                     |
| O           | I8123_NMTHeartbeat            | Start to the node heartbeat function                                    |
| X           | I8123_SDOReadData             | Read data by using the upload SDO protocol                              |
| X           | I8123_SDOReadFile             | Read the huge SDO data for specific slave node                          |
| X           | I8123_SDOWriteData            | Write data by using the download SDO protocol                           |
| X           | I8123_SDOAbortTransmit        | Send the SDO abort message  |
| X           | I8123_PDOWrite                | Use the PDO to write data to the CANopen node                           |
| X           | I8123_PDOWrite_Fast           | Only send out the RxPDO but not do any check                            |

|   |                             |  |
|---|-----------------------------|--|
| X | I8123_PDORemote             | Use the PDO to get data from the CANopen node      |
| X | I8123_PDORemote_Fast        | Only send the Rtr message but don't check response |
| X | I8213_SetPDORemotePolling   | Set PDO polling list table and poll them           |
| O | I8123_GetPDOLastData        | Get the last input or output PDO data              |
| O | I8123_GetPDOLastData_Fast   | Get the last PDO data from DPRAM directly          |
| O | I8123_GetMultiPDOData       | Get the several input or output PDO data once      |
| O | I8123_GetMultiPDOData_Fast  | Get the several PDO data from DPRAM directly       |
| O | I8123_GetRxPDOID            | Get all COB-ID of the RxPDO of the specific slave  |
| O | I8123_GetTxPDOID            | Get all COB-ID of the TxPDO of the specific slave  |
| X | I8123_InstallPDO            | Install and enable a specific PDO                  |
| X | I8123_DynamicPDO            | New or change a PDO mapping to the slave           |
| X | I8123_RemovePDO             | Remove a specific PDO mapping entry or object      |
| X | I8123_ChangePDOID           | Change the PDO COB-ID of a specific slave          |
| O | I8123_GetPDOMapInfo         | Obtain all the PDO-related information             |
| O | I8123_InstallPDO_List       | Install a PDO manually without check if it exists  |
| O | I8123_RemovePDO_List        | Remove PDO object without check real states        |
| X | I8123_PDOWUseEntry          | Change the valid entry number of the PDO objects   |
| X | I8123_PDOWTxType            | Set transmission type of the specific TxPDO        |
| X | I8123_PDOWEventTimer        | Set event timer of the specific TxPDO              |
| X | I8123_PDOWInhibitTime       | Set inhibit time of the specific TxPDO             |
| X | I8123_ChangeSYNCOID         | Change the SYNC COB-ID                             |
| O | I8123_SetSYNC_List          | Set the SYNC COB-ID without check if it exists     |
| O | I8123_GetSYNCOID            | Get the SYNC COB-ID                                |
| X | I8123_SendSYNCMsg           | Send the SYNC message                              |
| X | I8123_GetCyclicSYNCInfo     | Get all the cyclic sending SYNC information        |
| X | I8123_ChangeEMCYID          | Change the EMCY COB-ID                             |
| O | I8123_SetEMCY_List          | Set the EMCY COB-ID and don't check if it exists   |
| O | I8123_GetEMCYID             | Get the EMCY COB-ID                                |
| O | I8123_ReadLastEMCY          | Get the last EMCY message of the slave             |
| O | I8123_GetBootUpNodeAfterAdd | Get boot up message of a slave that had been added |
| O | I8123_GetEMCYData           | Get the EMCY message from the EMCY buffer          |
| O | I8123_GetNMTError           | Get the NMT Error event from the NMT event buffer  |
| O | I8123_InstallBootUpISR      | Install user-defined slave boot up process         |
| O | I8123_RemoveBootUpISR       | Remove the slave boot up process                   |
| O | I8123_InstallEMCYISR        | Install user-defined EMCY process                  |
| O | I8123_RemoveEMCYISR         | Remove the EMCY process                            |



---

|   |                               |   |
|---|-------------------------------|---|
| O | I8123_InstallNMTErrISR        | Install user-defined Guarding/Heartbeat event process |
| O | I8123_RemoveNMTErrISR         | Remove the Guarding/Heartbeat event process           |
| O | I8123_GetMasterReadSDOEvent   | Get the read SDO message sent to the I-8123W          |
| O | I8123_GetMasterWriteSDOEvent  | Get the write SDO message sent to the I-8123W         |
| X | I8123_ResponseMasterSDO       | Response the SDO message to the SDO sender            |
| O | I8123_GetMasterRemotePDOEvent | Get the remote PDO message send to the I-8123W        |
| O | I8123_GetMasterRxPDOEvent     | Get the RxPDO RTR sent to the I-8123W                 |
| X | I8123_ResponseMasterPDO       | Response the RxPDO RTR to the RTR sender              |
| O | I8123_InstallReadSDOISR       | Install the user-defined read master SDO process      |
| O | I8123_RemoveReadSDOISR        | Remove the read master SDO process                    |
| O | I8123_InstallWriteSDOISR      | Install the user-defined write master SDO process     |
| O | I8123_RemoveWriteSDOISR       | Remove the write master SDO process                   |
| O | I8123_InstallRxPDOISR         | Install the user-defined write master PDO process     |
| O | I8123_RemoveRxPDOISR          | Remove the write master PDO process                   |
| O | I8123_InstallTxPDOISR         | Install the user-defined remote master PDO process    |
| O | I8123_RemoveTxPDOISR          | Remove the remote master PDO process                  |
| X | CPM100_SyncGroupCycUpdata     | Update PDO group data for SYNC protocol               |
| X | CPM100_SyncGroupCycSend       | Start to send SYNC signal and followed PDO data       |
| X | CPM100_SyncGroupClear         | Clear SYNC and PDO group data in update buffer        |
| X | CPM100_GetSYNCGroupStopState  | Get the state of SYNC and PDO group send              |
| X | CPM100_InstallSYNCGroupISR    | Install the SYNC and PDO group data process           |
| X | CPM100_RemoveSYNCGroupISR     | Remove the SYNC and PDO group data process            |

**Table 3.1 Description of functions**

## 3.2. Function Return Code

The following table interprets all the return code returned by I-8123W.

| Return Code | Error ID           | Description   |
|-------------|--------------------|---|
| 0           | CPM_NoError        | OK  |
| 3           | CPM_SlotNumberErr  | There is no I-8123W on the specific slot No.          |
| 5           | CPM_ConfigErr      | The I-8123W hasn't been configured successfully       |
| 6           | CPM_MasterInitErr  | The I-8123W initialization error.                     |
| 7           | CPM_MasterNotInit  | The I-8123W hasn't been initialized                   |
| 8           | CPM_ListenMode     | The I-8123W is in listen mode now                     |
| 9           | CPM_NodeErr        | Set node number error                                 |
| 10          | CPM_NodeExist      | The node had been added to the Master                 |
| 12          | CPM_TxBusy         | Tx buffer is busy, please wait a minute to send again |
| 13          | CPM_UnknowCmd      | This version of firmware doesn't support the function |
| 14          | CPM_CmdReceErr     | I-8123W receive command of wrong length               |
| 15          | CPM_DataEmpty      | There is no data to receive                           |
| 16          | CPM_MemAllocErr    | I-8123W has not enough memory                         |
| 17          | CPM_SendCycMsgErr  | Cyclic message send error                             |
| 18          | CPM_StatusErr      | NMT state of CANopen slave is error                   |
| 20          | CPM_SetGuardErr    | Set Guarding and LifeTime parameter error             |
| 21          | CPM_SetHbeatErr    | Set Heartbeat parameter error                         |
| 22          | CPM_SegLenErr      | SDO Segment receive error length                      |
| 23          | CPM_SegToggleErr   | SDO Segment receive error toggle                      |
| 24          | CPM_SegWriteErr    | SDO write segment error                               |
| 25          | CPM_Abort          | The return message is an Abort message                |
| 26          | CPM_PDOLenErr      | PDO output data error                                 |
| 27          | CPM_COBIDErr       | The COB-ID isn't exist or isn't correct one           |
| 28          | CPM_PDOPDOInstErr  | Install the PDO object error                          |
| 29          | CPM_PDODynaErr     | The PDO mapping data is setting error                 |
| 30          | CPM_PDONumErr      | The PDO number and COB-ID is not match                |
| 31          | CPM_PDOSetErr      | PDO parameter is setting error                        |
| 32          | CPM_PDOPDOEntryErr | The PDO entry parameter is more then useful entry     |
| 33          | CPM_SetCobIdErr    | The EMCY or SYNC COB-ID is setting error              |
| 34          | CPM_CycFullErr     | There are already 5 cyclic message running            |
| 35          | CPM_Timeout        | Message response timeout                              |
| 36          | CPM_DataLenErr     | Data length setting error                             |

---

|    |                     |  |
|----|---------------------|--|
| 37 | CPM_SyncIdDifferent | SYNC ID is not the same for all slave group      |
| 40 | CPM_Wait            | Command is uncompleted (only for non-block mode) |
| 41 | CPM_Processing      | Command is running (only for non-block mode)     |
| 50 | CPM_LoadEDSErr      | Loading the EDS file fails                       |
| 51 | CPM_EDSFormatErr    | The format of the EDS file is incorrect          |

---

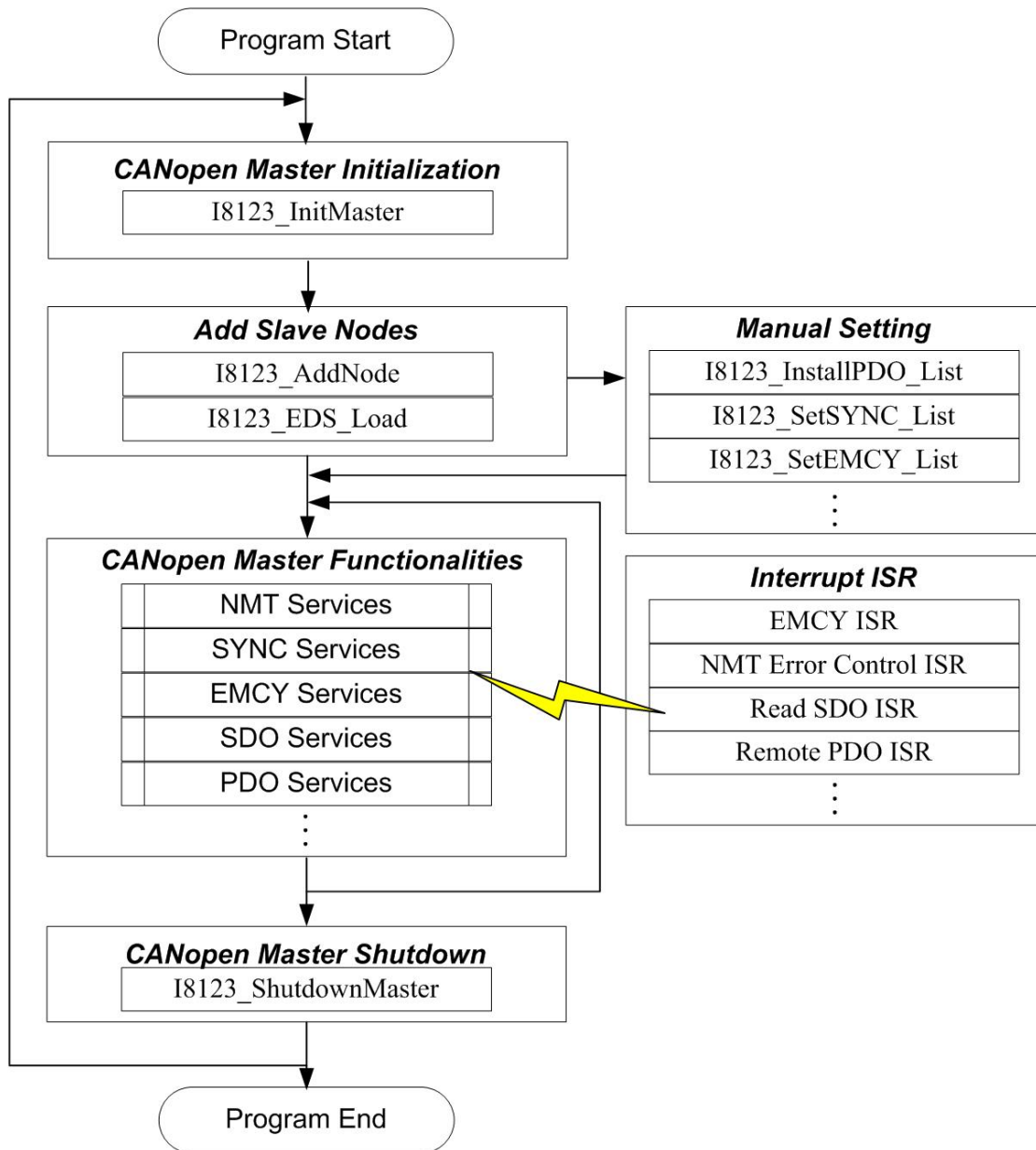
### 3.3. CANopen Master Library Application Flowchart

In this section, it describes that the operation procedure about how to use the CANopen Master Library to build users applications. This information is helpful for users 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 detail information, please refer to the demo programs in the section 4.

When users programs apply the CANopen Master Library functions, the function I8123\_InitMaster must be called first. The function is used to initialize I-8123W and configure the CAN port.

After initializing the CAN interface successfully, users need to use the function I8123\_AddNode or I8123\_EDS\_Load to install at least one CANopen device into the node list. The function I8123\_AddNode can install slave node automatically or manually. When the user applies the function to install node manually, the PDO ID, SYNC ID, and EMCY ID of the node must be added manually by the functions I8123\_InstallPDO\_List, I8123\_SetSYNC\_List, and I8123\_SetEMCY\_List.

If the function I8123\_InitMaster and I8123\_AddNode / I8123\_EDS\_Load has been executed, the communication services (NMT, SYNC, EMCY, SDO, and PDO services) can be used at any time before calling the function I8123\_ShutdownMaster, because the function I8123\_ShutdownMaster will stop all process created by the function I8123\_InitMaster.



**Figure 3.1 Main programming flow chart**

---

## 3.4. Communication Services Introduction

### NMT Services

The CANopen Master Library provides several NMT service functions, such as the functions I8123\_AddNode, I8123\_EDS\_Load, I8123\_RemoveNode, I8123\_NMTChangeState, I8123\_NMTGetState, I8123\_NMTHeartbeat and I8123\_NMTGuarding. As the prerequisite for the master, the slave nodes have to be registered by the function I8123\_AddNode or I8123\_EDS\_Load with providing its Node-ID. The registered slave nodes can be individually removed from the node list by the function I8123\_RemoveNode. Through NMT services, the NMT Master controls the state of the slaves. Table 3.3 is the command value and corresponding NMT command for the input parameters of the function I8123\_NMTChangeState. When using the function I8123\_NMTGetState, the slave status values and their descriptions are shown in the table 3.4. The Node Guarding and Heartbeat protocol are implemented via the function I8123\_NMTGuarding and the function I8123\_NMTHeartbeat. If the slave nodes are in the node list, users can change the node guarding or heartbeat parameters defined in the slave nodes by calling the function I8123\_NMTGuarding or I8123\_NMTHeartbeat.

| 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 slaves**

---

## **SDO Services**

“Initiate SDO download protocol” or “Initiate SDO upload protocol” is used when the object data length  $\leq$  4 bytes. If the object data length  $>$  4 bytes, “Download SDO segment protocol” or “Upload SDO segment protocol” will be used. Calling these two functions, I8123\_SDOReadData and I8123\_SDOWriteData, the initiate protocol and segment protocol will be selected automatically according to the object data length.

I8123\_SDOAbortTransmit function can abort a pending SDO transfer at any time. Applying the abort service will have no confirmation from the slave device.

## **PDO Services**

After using the I8123\_AddNode to add a slave, the default TxPDOs and RxPDOs (TxPDO 1 ~ 10, RxPDO 1 ~ 10) will also be added to the I-8123W’s control list. If there are PDOs other than the default setting, the function I8123\_InstallPDO is used for enabling these TxPDOs or RxPDOs object. After installing the PDOs, the function I8123\_DynamicPDO can add or change the PDOs’ mapping objects. Each PDO object supports 0~8 application objects. These application objects defined in the CANopen specification DS401, and they are mapped to the relative parameters of the DI/DO/AI/AO channels. After calling the function I8123\_InstallPDO and I8123\_DynamicPDO, the PDO communication object will be mapped and activated. If the PDO communication object is not needed no more, use the function I8123\_RemovePDO to remove it.

When you want to output data via PDO, the function I8123\_PDOWriteData is useful. This function can write all PDO 8-byte data or write some parts of PDO 8-byte data. Calling this function will send the specific data to the corresponding node via PDO protocol, and put the output data into PDO buffer at the same time. Therefore, you can check the output history of the PDO. But, if the connection between the I-8123W and the slave is lost, the history output information may be not the same with the real status on the slave.

In CANopen specification, you can get the TxPDOs data by applying the remote transmit request CAN frame. The function I8123\_PDORemote is needed in this case. Or you can use I8123\_GetPDOLastData to get the last RxPDO and TxPDO data from the PDO buffer.

---

### **SYNC Services**

Calling the function I8123\_SendSYNCMsg starts the SYNC object transmission. This function supports single SYNC message transmission and cyclic SYNC message transmission. The parameter "Timer" of the function I8123\_SendSYNCMsg can adjust the cyclic period of the SYNC COB-ID sent by master. And the parameter "Times" can set the sending times of the SYNC message. If the timer parameter is set to 0, the SYNC object transmission will be stopped. When the times parameter is set to non-zero value, the function will send the SYNC message until the timer is set to 0 or the parameter "Times" is achieved.

### **EMCY Services**

The Emergency messages are triggered by the occurrence of a device internal error situation. Users can call the function I8123\_ReadLastEMCY to receive the EMCY message or the function I8123\_InstallEMCYISR to install user-defined EMCY interrupt process. In this case, if there are CAN slaves sending the EMCY, these EMCY messages will be processed by the user-defined EMCY interrupt process.



---

## 3.5. Function Description

### 3.5.1. I8123\_GetVersion

- **Description:**

This function is used to obtain the version information of the I8123W.lib library.

- **Syntax:**

**WORD** I8123\_GetVersion(**void**)

- **Parameter:**

None

- **Return:**

The library version information.

---

### 3.5.2. I8123\_SetFunctionTimeout

- **Description:**

Sometimes, some function cost more time to complete its task, such as I8123\_ScanNode. If some API of the I-8123W library never gets the feedback from the I-8123W until timeout value goes by, the error code "CPM\_Timeout" will be returned. Through this function, the user can adjust the suitable maximum timeout value of all functions for application. Default timeout value is 1 second.

- **Syntax:**

**void** I8123\_SetFunctionTimeout(**BYTE** SlotNo, **WORD** Timeout)

- **Parameter:**

**SlotNo:** [input] I-8123W slot number (0~7).

**Timeout:** [input] Maximum timeout value of all functions (ms).

---

### 3.5.3. I8123\_GetCANStatus

- **Description:**

The function I8123\_ShutdownMaster removes all the slaves that had added to master and stop all the functions of the I-8123W. The function must be called before exit the users' application programs.

- **Syntax:**

**WORD** I8123\_ShutdownMaster (**BYTE** SlotNo, **BYTE** \*Status)

- **Parameter:**

**SlotNo:** [input] I-8123W slot number (0~7).

**\*Status:** [output] The pointer for obtaining the CAN status. It indicates the value of status register of SJA1000. Its meanings are described below.

| Bit NO. | Description  |
|---------|--|
| 7 (MSB) | Bus status. 1 for bus off, 0 for bus on.   |
| 6       | Error status. 1 for at least one error, 0 for OK.  |
| 5       | SJA1000 Transmit status. 1 for transmitting, 0 for idle  |
| 4       | SJA1000Receive status. 1 for receiving, 0 for idles.   |
| 3       | SJA1000 Transmit completes status. 1 for complete, 0 for incomplete.                                   |
| 2       | SJA1000 Transmit buffer status. 1 for released, 0 for locked.  |
| 1       | Data overrun status. 1 for SJA1000 reception buffer overrun, 0 for OK.                                 |
| 0 (LSB) | Receive buffer status. 1 for at least one message stored in the SJA1000 reception buffer, 0 for empty. |

---

### 3.5.4. I8123\_InitMaster

- **Description:**

The function must be applied when configuring the CAN controller and initialize the I-8123W. It must be called once before using other functions of the I8123W.lib.

- **Syntax:**

**WORD** I8123\_InitMaster(**BYTE** SlotNo, **BYTE** Node, **BYTE** BaudRate, **BYTE** BlockMode)

- **Parameter:**

**SlotNo:** [input] I-8123W slot number (0~7).

**Node:** [input] Master's node ID. If the parameter is 0, the master will be a normal master, and no other master can control it. If the parameter is 1 ~ 127 (different from other slave), other master can do some control to it through some ISR function.

**BaudRate:** [input] The baudrate of the I-8123W

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

**BlockMode:** [input] 0 means this function is non-block-function, and 1 means this function is block-function. If set this parameter to 1, the running procedure of the users' application will be held in the function until return. If set to 0, this function will return "CPM\_Processing" directly. This function will return its process status while users apply it with the same "SlotNo" and "Node" again. If the procedure is still not complete, it will return "CPM\_Wait".

---

### 3.5.5. I8123\_ShutdownMaster

- **Description:**

The function I8123\_ShutdownMaster removes all the slaves that had added to master and stop all the functions of the I-8123W. The function must be called before exit the users' application programs.

- **Syntax:**

**WORD** I8123\_ShutdownMaster (**BYTE** SlotNo)

- **Parameter:**

**SlotNo:** [input] I-8123W slot number (0~7).

---

### 3.5.6. I8123\_MasterSendBootupMsg

- **Description:**

To use the function I8123\_MasterSendBootupMsg can let I-8123W sends a boot up message after I8123\_InitMaster is called.

**Note: The function is valid while the Node parameter of the function I8123\_InitMaster is > 0.**

- **Syntax:**

**WORD** I8123\_MasterSendBootupMsg(**BYTE** SlotNo,  
**BYTE** BlockMode)

- **Parameter:**

**SlotNo:** [input] I-8123W slot number (0~7).

**BlockMode:** [input] 0 means this function is non-block-function, and 1 means this function is block-function. If set this parameter to 1, the running procedure of the users' application will be held in the function until return. If set to 0, this function will return "CPM\_Processing" directly. This function will return its process status while users apply it with the same "SlotNo" again. If the procedure is still not complete, it will return "CPM\_Wait".

---

### 3.5.7. I8123\_SetMasterMode

- **Description:**

This function can configure if the master is into listen mode or normal mode (default mode). In listen mode, the I-8123W can't send any CANopen message, and some functions will be useless in this mode. User can select normal mode or listen mode at any time after calling function I8123\_InitMaster.

- **Syntax:**

**WORD** I8123\_SetMasterMode(**BYTE** SlotNo, **BYTE** Mode,  
**BYTE** BlockMode)

- **Parameter:**

**SlotNo:** [input] I-8123W slot number (0~7).

**Mode:** [input] 1 is listen mode, and others are normal mode (default mode).

**BlockMode:** [input] 0 means this function is non-block-function, and 1 means this function is block-function. If set this parameter to 1, the running procedure of the users' application will be held in the function until return. If set to 0, this function will return "CPM\_Processing" directly. This function will return its process status while users apply it with the same "SlotNo" again. If the procedure is still not complete, it will return "CPM\_Wait".

---

### 3.5.8. I8123\_GetMasterMode

- **Description:**

If user want to know what operation mode of the master is, call the function to get it.

- **Syntax:**

**WORD** I8123\_GetMasterMode(**BYTE** SlotNo, **BYTE** \*Mode,  
**BYTE** BlockMode)

- **Parameter:**

**SlotNo:** [input] I-8123W slot number (0~7).

**\*Mode:** [output] 0 is normal mode (default mode), and 1 is listen mode.

**BlockMode:** [input] 0 means this function is non-block-function, and 1 means this function is block-function. If set this parameter to 1, the running procedure of the users' application will be held in the function until return. If set to 0, this function will return "CPM\_Processing" directly. This function will return its process status while users apply it with the same "SlotNo" again. If the procedure is still not complete, it will return "CPM\_Wait".



---

### 3.5.9. I8123\_GetFirmwareVersion

- **Description:**

The function can let users know what is the version number of the I-8123W's firmware.

- **Syntax:**

**WORD** I8123\_GetFirmwareVersion (**BYTE** SlotNo, **WORD** \*Fir\_Ver, **BYTE** BlockMode)

- **Parameter:**

**SlotNo:** [input] I-8123W slot number (0~7).

**\*Fir\_Ver:** [output] I-8123W firmware version information.

**BlockMode:** [input] 0 means this function is non-block-function, and 1 means this function is block-function. If set this parameter to 1, the running procedure of the users' application will be held in the function until return. If set to 0, this function will return "CPM\_Processing" directly. This function will return its process status while users apply it with the same "SlotNo" again. If the procedure is still not complete, it will return "CPM\_Wait".

---

### 3.5.10. I8123\_EDS\_Load

#### ● Description:

The function I8123\_EDS\_Load can let users load EDS file for adding a CANopen slave with specified Node ID into the master node list. Using this function will not send any message to check if the slave is existent or not.

#### ● Syntax:

**WORD** I8123\_EDS\_Load (**BYTE** SlotNo, **BYTE** Node, **char** \*FilePath, **WORD** DelayTime, **WORD** ResTimeout, **BYTE** BlockMode)

#### ● Parameter:

**SlotNo:** [input] I-8123W slot number (0~7).

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

**\*FilePath:** [input] Absolute or relative file path of the EDS file.

**DelayTime:** [input] The parameter defines the time interval between sending two messages from the I-8123W. It can avoid the master frequently sending messages that may overrun the buffer of the slave. Too large value of this parameter reduces the performance of the I-8123W. The unit of the parameter is ms. This parameter will be applied to the specified slave after finishing the ESD loading.

**ResTimeout:** [input] The timeout value of the response of the CANopen slaves. When the master sends a CANopen message to the slave, it will wait the response until timeout if there is a response. The unit of this parameter is millisecond. This parameter will be applied to the specified slave after finishing the ESD loading.

**BlockMode:** [input] 0 means this function is non-block-function, and 1 is block-function. If set this parameter to 1, the running procedure of the users' application is held until finishing this function. If 0, this function returns "CPM\_Processing" directly. While users apply it with the same "SlotNo" again, it returns the process status. If the procedure is still not complete, it returns "CPM\_Wait".

---

### 3.5.11. I8123\_AddNode

#### ● Description:

The function I8123\_AddNode can add a CANopen slave with specified Node ID into the master node list. There are three mode of the function. Mode 1 is adding node automatically, mode 2 is adding node manually, and mode 3 is waiting node's boot up message for add. In the automatic mode, after calling this function to add a slave, the slave will be into the operational state directly. And master will scan the TxPDO1 ~ TxPDO10 and RxPDO1 ~ RxPDO10 and install them into the firmware of the I-8123W if the slave supports these PDO objects. In the manual mode, this function will add a CANopen slave into the master node list only, and will not send any message to check if the slave is existent or not. Besides, the manual mode doesn't install the SYNC ID, EMCY ID, and any PDO object into the firmware of the I-8123W. Users must call I8123\_SetSYNC\_List, I8123\_SetEMCY\_List, and I8123\_InstallPDO\_List to complete the object installation to finish the adding node process.

The added node can be removed from the master node list by the function I8123\_RemoveNode.

#### ● Syntax:

**WORD** I8123\_AddNode(**BYTE** SlotNo, **BYTE** Node, **BYTE** AddMode,  
**WORD** DelayTime, **WORD** ResTimeout,  
**BYTE** BlockMode)

#### ● Parameter:

**SlotNo:** [input] I-8123W slot number (0~7).

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

**AddMode:** [input] 1: Automatic mode. 2: Manual mode.

**DelayTime:** [input] The parameter is the shortest time interval between sending two messages from the I-8123W. It can avoid the master to send too much CANopen messages in a short time that may let some slaves occur the errors. But if the delay time is too long, the performance of the I-8123W is

---

down. The unit of the parameter is ms.

**ResTimeout:** [input] The timeout value of the responded messages from the CANopen slaves. When the master sends a CANopen message to the slave, it will wait the feedback until timeout if there is a feedback. The unit of this parameter is millisecond.

**BlockMode:** [input] 0 means this function is non-block-function, and 1 means this function is block-function. If set this parameter to 1, the running procedure of the users' application will be held in the function until return. If set to 0, this function will return "CPM\_Processing" directly. This function will return its process status while users apply it with the same "SlotNo" and "Node" again. If the procedure is still not complete, it will return "CPM\_Wait".

---

### 3.5.12. I8123\_RemoveNode

- **Description:**

The function I8123\_RemoveNode removes the slave with the specified Node-ID from node list of the master. It requires a valid Node-ID, which has installed by the function I8123\_AddNode before.

- **Syntax:**

**WORD** I8123\_RemoveNode(**BYTE** SlotNo, **BYTE** Node,  
**BYTE** BlockMode)

- **Parameter:**

**SlotNo:** [input] I-8123W slot number (0~7).

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

**BlockMode:** [input] 0 means this function is non-block-function, and 1 means this function is block-function. If set this parameter to 1, the running procedure of the users' application will be held in the function until return. If set to 0, this function will return "CPM\_Processing" directly. This function will return its process status while users apply it with the same "SlotNo" and "Node" again. If the procedure is still not complete, it will return "CPM\_Wait".

---

### 3.5.13. I8123\_RemoveAndResetNode

- **Description:**

The function I8123\_RemoveAndResetNode removes the slave with the specified Node-ID from node list of the master and reset the slave. It requires a valid Node-ID, which has installed by the function I8123\_AddNode before.

- **Syntax:**

**WORD** I8123\_RemoveAndResetNode(**BYTE** SlotNo, **BYTE** Node,  
**BYTE** BlockMode)

- **Parameter:**

**SlotNo:** [input] I-8123W slot number (0~7).

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

**BlockMode:** [input] 0 means this function is non-block-function, and 1 means this function is block-function. If set this parameter to 1, the running procedure of the users' application will be held in the function until return. If set to 0, this function will return "CPM\_Processing" directly. This function will return its process status while users apply it with the same "SlotNo" and "Node" again. If the procedure is still not complete, it will return "CPM\_Wait".

---

### 3.5.14. I8123\_DelayAndResponseTimeout

- **Description:**

Call the function to change CAN message minimum interval time and response timeout value of the slave node at any time.

- **Syntax:**

**WORD** I8123\_DelayAndResponseTimeout(**BYTE** SlotNo, **BYTE** Node, **WORD** DelayTime, **WORD** ResTimeout, **BYTE** BlockMode)

- **Parameter:**

**SlotNo:** [input] I-8123W slot number (0~7).

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

**DelayTime:** [input] The parameter is the same as "DelayTime" of the function I8123\_AddNode.

**ResTimeout:** [input] The parameter is the same as "ResTimeout" of the function I8123\_AddNode.

**BlockMode:** [input] 0 means this function is non-block-function, and 1 means this function is block-function. If set this parameter to 1, the running procedure of the users' application will be held in the function until return. If set to 0, this function will return "CPM\_Processing" directly. This function will return its process status while users apply it with the same "SlotNo" and "Node" again. If the procedure is still not complete, it will return "CPM\_Wait".

---

### 3.5.15. I8123\_ScanNode

- **Description:**

User can use the function I8123\_ScanNode to know how many slave nodes are on the CANopen network.

- **Syntax:**

**WORD** I8123\_ScanNode(**BYTE** SlotNo, **BYTE** S\_Node, **BYTE** E\_Node, **BYTE** \*NodeList, **BYTE** BlockMode)

- **Parameter:**

**SlotNo:** [input] I-8123W slot number (0~7).

**S\_Node:** [input] Start node ID.

**E\_Node:** [input] End node ID. This function will scan node ID from S\_Node to E\_Node. If S\_Node >= E\_Node, it will scan all slave node ID (1 ~ 127) on the CANopen network.

**\*NodeList:** [output] This is a 16-byte array parameter. Each bit of the parameter means a node ID, the bit is 0 means that the node ID doesn't exist and 1 means the node ID is on the CANopen network. For example, if the NodeList[0] is 0x16 (0001 0110), it means that the nodes with ID 1, 2, and 4 exist, and the nodes with ID 0, 3, 5, 6, and 7 don't exist. If the NodeList[1] is 0x41 (0100 0001), it means that the nodes with ID 8 and 14 exist, and the nodes with ID 9, 10, 11, 12, 13, and 15 don't exist.

**BlockMode:** [input] 0 means this function is non-block-function, and 1 means this function is block-function. If set this parameter to 1, the running procedure of the users' application will be held in the function until return. If set to 0, this function will return "CPM\_Processing" directly. This function will return its process status while users apply it with the same "SlotNo" again. If the procedure is still not complete, it will return "CPM\_Wait".



---

### 3.5.16. I8123\_GetNodeList

- **Description:**

User can use the function I8123\_GetNodeList to know how many slave nodes are added to the node list of the I-8123W.

- **Syntax:**

**WORD** I8123\_GetNodeList(**BYTE** SlotNo, **BYTE** \*NodeList,  
**BYTE** BlockMode)

- **Parameter:**

**SlotNo:** [input] I-8123W slot number (0~7).

**\*NodeList:** [output] This is a 16-byte array parameter. Each bit of the parameter means a node ID, the bit is 0 means the node ID not exist and 1 means the node ID now on the CANopen bus. For example, if the NodeList[0] is 0x16 (0001 0110), it means that the nodes with ID 1, 2, and 4 have been added into node list, and the nodes with ID 0, 3, 5, 6, and 7 don't. If the NodeList[1] is 0x41 (0100 0001), it means that the nodes with ID 8 and 14 have been added into node list, and the nodes with ID 9, 10, 11, 12, 13, and 15 don't.

**BlockMode:** [input] 0 means this function is non-block-function, and 1 means this function is block-function. If set this parameter to 1, the running procedure of the users' application will be held in the function until return. If set to 0, this function will return "CPM\_Processing" directly. This function will return its process status while users apply it with the same "SlotNo" again. If the procedure is still not complete, it will return "CPM\_Wait".

---

### 3.5.17. I8123\_NMTChangeState

- **Description:**

The function I8123\_NMTChangeState 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 CANopen network will be changed.

- **Syntax:**

**WORD** I8123\_NMTChangeState(**BYTE** SlotNo, **BYTE** Node,  
**BYTE** State, **BYTE** BlockMode)

- **Parameter:**

**SlotNo:** [input] I-8123W slot number (0~7).

**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

2: stop

128: enter PRE-OPERATIONAL

129: Reset\_Node

130: Reset\_Communication

**BlockMode:** [input] 0 means this function is non-block-function, and 1 means this function is block-function. If set this parameter to 1, the running procedure of the users' application will be held in the function until return. If set to 0, this function will return "CPM\_Processing" directly. This function will return its process status while users apply it with the same "SlotNo" and "Node" again. If the procedure is still not complete, it will return "CPM\_Wait".

---

### 3.5.18. I8123\_NMTGetState

- **Description:**

The function I8123\_NMTGetState can get the NMT state from slaves.

- **Syntax:**

**WORD** I8123\_NMTGetState(**BYTE** SlotNo, **BYTE** Node, **BYTE** \*State,  
**BYTE** BlockMode)

- **Parameter:**

**SlotNo:** [input] I-8123W slot number (0~7).

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

**\*State:** [output] The NMT state of the slave.

**4:** STOPPED

**5:** OPERATIONAL

**127:** PRE-OPERATIONAL

**BlockMode:** [input] 0 means this function is non-block-function, and 1 means this function is block-function. If set this parameter to 1, the running procedure of the users' application will be held in the function until return. If set to 0, this function will return "CPM\_Processing" directly. This function will return its process status while users apply it with the same "SlotNo" and "Node" again. If the procedure is still not complete, it will return "CPM\_Wait".

---

### 3.5.19. I8123\_NMTGuarding

#### ● Description:

Use the function I8123\_NMTGuarding to set guard time and life time factor of the slave with the specified node ID. Then, the master will automatically send the guarding message to slave device according to the “GuardTime” parameters of this function. If the master doesn’t receive the confirmation of guarding message from the slave, the I-8123W will produce a Node\_Guarding\_Event event to users. Users may use the function I8123\_InstallNMTErrISR to install a user-defined process to get the guarding fail event and process the guarding fail procedure. However, if the slave doesn’t receive the guarding message during the Node Life time period (Node Life time = “GuardTime” \* “LifeTime”), it will be triggered to send the EMCY message. It is recommended that “LifeTime” value is set to more than 1.

Take a note that following the CANopen specification, it is not allowed for one slave device to use both error control mechanisms Guarding Protocol and Heartbeat Protocol at the same time.

#### ● Syntax:

**WORD** I8123\_NMGuarding(**BYTE** SlotNo, **BYTE** Node,  
**WORD** GuardTime, **BYTE** LifeTime, **BYTE** BlockMode)

#### ● Parameter:

**SlotNo:** [input] I-8123W slot number (0~7).

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

**GuardTime:** [input] Guard Time (1 ~ 65535 ms).

**LifeTime:** [input] Life Time Factor (1 ~ 255).

**BlockMode:** [input] 0 means this function is non-block-function, and 1 means this function is block-function. If set this parameter to 1, the running procedure of the users’ application will be held in the function until return. If set to 0, this function will return “CPM\_Processing” directly. This function will return its process status while users apply it with the same “SlotNo” and “Node” again. If the procedure is still not complete, it will return “CPM\_Wait”.

---

### 3.5.20. I8123\_NMTHeartbeat

#### ● Description:

Use the function I8123\_NMTHeartbeat to set heartbeat time of the slave with the specified node ID and consume time with the I-8123W. Then, the slave will automatically send the heartbeat message to master according to the “ProduceTime” parameters of this function. If the master doesn’t receive the heartbeat message from the slave until the “ConsumeTime” time (unit is millisecond) is up, the I-8123W will produce a “Heartbeat\_Event” event to users. Users may use the function I8123\_InstallNMTErrISR to install a user-defined process to get the heartbeat fail event and process the heartbeat fail procedure. It is recommended that “ConsumeTime” value is set to bigger than the “ProduceTime” value.

Take a note that following the CANopen specification, it is not allowed for one slave device to use both error control mechanisms Guarding Protocol and Heartbeat Protocol at the same time.

#### ● Syntax:

**WORD** I8123\_NMHartbeat(**BYTE** SlotNo, **BYTE** Node,  
**WORD** ProduceTime, **WORD** ConsumeTime,  
**BYTE** BlockMode)

#### ● Parameter:

**SlotNo:** [input] I-8123W slot number (0~7).

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

**ProduceTime:** [input] Produce Time of slave device (1 ~ 65535 ms).

**ConsumeTime:** [input] Consume Time of master (1 ~ 65535 ms).

**BlockMode:** [input] 0 means this function is non-block-function, and 1 means this function is block-function. If set this parameter to 1, the running procedure of the users’ application will be held in the function until return. If set to 0, this function will return “CPM\_Processing” directly. This function will return its process status while users apply it with the same “SlotNo” and “Node” again. If the procedure is still not complete, it will return “CPM\_Wait”.

---

### 3.5.21. I8123\_SDOReadData

- **Description:**

The function I8123\_SDOReadData is useful to the SDO upload from a specified slave. When users use this function, pass the slave device node ID, object index and object subindex into this function. This function supports both expedition mode (less than 4-byte data length) and segment mode (more than 4-byte data length).

- **Syntax:**

**WORD** I8123\_SDOReadData (**BYTE** SlotNo, **BYTE** Node,  
**WORD** Index, **BYTE** SubIndex,  
**DWORD** \*RDLen, **BYTE** \*RData,  
**BYTE** BlockMode)

- **Parameter:**

**SlotNo:** [input] I-8123W slot number (0~7).

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

**Index:** [input] Object index of object dictionary of slave devices.

**SubIndex:** [input] Object subindex of object dictionary of slave devices.

**\*RDLen:** [output] Total data length.

**\*RData:** [output] SDO data respond from the specified slave device.

**BlockMode:** [input] 0 means this function is non-block-function, and 1 means this function is block-function. If set this parameter to 1, the running procedure of the users' application will be held in the function until return. If set to 0, this function will return "CPM\_Processing" directly. This function will return its process status while users apply it with the same "SlotNo" and "Node" again. If the procedure is still not complete, it will return "CPM\_Wait".

---

### 3.5.22. I8123\_SDOReadFile

- **Description:**

The function I8123\_SDOReadFile is useful as uploading the SDO data more than 1024 bytes. While users use the I8123\_ReadData to read the SDO data and the return data length is more than 1024 byte. The SDO data are stored in a file. Users can use the function I8123\_SDOReadFile for reading the SDO data from the file.

- **Syntax:**

**WORD** I8123\_SDOReadFile (**BYTE** SlotNo, **BYTE** Node,  
**WORD** Index, **BYTE** SubIndex,  
**DWORD** Start, **DWORD** Len,  
**DWORD** \*RLen, **BYTE** \*RData)

- **Parameter:**

**SlotNo:** [input] I-8123W slot number (0~7).

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

**Index:** [input] Object index of object dictionary of slave devices.

**SubIndex:** [input] Object subindex of object dictionary of slave devices.

**Start:** [input] Start position for reading the SDO data from file. The range is from 0 to maximum length.

**Len:** [input] Data length for reading the SDO data.

**\*RDLen:** [output] Returned data length in reality.

**\*RData:** [output] The SDO data respond from the specified slave device.

---

### 3.5.23. I8123\_SDOWriteData

#### ● Description:

The function I8123\_SDOWriteData can send out a SDO message to the specified slave device. This procedure is also called download SDO protocol. The parameter node of the function I8123\_SDOWriteData is used to point which slave device will receive this SDO message. Because the data length of each object stored in 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 (less than 4-byte data length) and segment mode (more than 4-byte data length)

#### ● Syntax:

**WORD** I8123\_SDOWriteData (**BYTE** SlotNo, **BYTE** Node, **WORD** Index, **BYTE** SubIndex, **DWORD** TDLen, **BYTE** \*TData, **WORD** \*RDLen, **BYTE** \*RData, **BYTE** BlockMode)

#### ● Parameter:

**SlotNo:** [input] I-8123W slot number (0~7).

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

**Index:** [input] The index value of object of the object dictionary.

**SubIndex:** [input] The subindex value of object of the object dictionary.

**TDLen:** [input] Total data size to be written.

**\*TData:** [input] The SDO data written to slave device.

**\*RLen:** [output] Total data size of responded data.

**\*RData:** [output] SDO data responded from the specified slave device.

**BlockMode:** [input] 0 means this function is non-block-function, and 1 means this function is block-function. If set this parameter to 1, the running procedure of the users' application will be held in the function until return. If set to 0, this function will return "CPM\_Processing" directly. This function will return its process status while users apply it with the same "SlotNo" and "Node" again. If the procedure is still not complete, it will return "CPM\_Wait".



---

### 3.5.24. I8123\_SDOAbortTransmit

- **Description:**

Call the function I8123\_SDOAbortTransmit to cancel the SDO transmission. The parameter node of this function is used to specify which SDO communication will be terminated between the master and the specified slave device.

- **Syntax:**

**WORD** I8123\_SDOAbortTransmit(**BYTE** SlotNo, **BYTE** Node,  
**WORD** Index, **BYTE** SubIndex, **DWORD** \*AData,  
**BYTE** BlockMode)

- **Parameter:**

**SlotNo:** [input] I-8123W slot number (0~7).

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

**Index:** [input] The object index value of the object dictionary.

**SubIndex:** [input] The object subindex value of the object dictionary.

**\*AData:** [input] Abort data to be send to slave.

**BlockMode:** [input] 0 means this function is non-block-function, and 1 means this function is block-function. If set this parameter to 1, the running procedure of the users' application will be held in the function until return. If set to 0, this function will return "CPM\_Processing" directly. This function will return its process status while users apply it with the same "SlotNo" and "Node" again. If the procedure is still not complete, it will return "CPM\_Wait".

---

### 3.5.25. I8123\_PDOWrite

#### ● Description:

Call the function I8123\_PDOWrite to send out a PDO message to the specified slave device. Before using this function, users need to use the function I8123\_InstallPDO to install the PDO object into I-8123W if users want to use non-default PDO. Then, change the NMT state of the target slave device to operational mode by using the function I8123\_NMTChangeState if the slave is not in the operational mode. Use the parameter offset to set the start position of the PDO data, and use the parameters “\*TData” and “DLen” to point the data and data length. Then, this function will follow the data length to cover the slave PDO buffer of the I-8123W with the data from the specified start position, and send the data to the specified slave via PDO protocol at the same time.

#### ● Syntax:

**WORD** I8123\_PDOWrite (**BYTE** SlotNo, **WORD** Cobid, **BYTE** Offset,  
**BYTE** DLen, **BYTE** \*TData,  
**BYTE** BlockMode)

#### ● Parameter:

**SlotNo:** [input] I-8123W slot number (0~7).

**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)).

**\*TData:** [output] The data pointer point to the PDO data.

**BlockMode:** [input] 0 means this function is non-block-function, and 1 means this function is block-function. If set this parameter to 1, the running procedure of the users' application will be held in the function until return. If set to 0, this function will return “CPM\_Processing” directly. This function will return its process status while users apply it with the same “SlotNo” and “Cobid” again. If the procedure is still not complete, it will return “CPM\_Wait”.

---

### 3.5.26. I8123\_PDOWrite\_Fast

- **Description:**

The function is like I8123\_PDOWrite but does not check whether the PDO message really sends to CAN bus or not. So I8123\_PDOWrite\_Fast is about twice as faster than I8123\_PDOWrite at high speed baud rate (greater than or equal to 250 kbps).

- **Syntax:**

**WORD** I8123\_PDOWrite\_Fast (**BYTE** SlotNo, **WORD** Cobid, **BYTE** Offset, **BYTE** DLen, **BYTE** \*TData)

- **Parameter:**

**SlotNo:** [input] I-8123W slot number (0~7).

**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  $\leq$  8 (total length of the PDO)).

**\*TData:** [output] The data pointer point to the PDO data.

---

### 3.5.27. I8123\_PDORemote

- **Description:**

Use the function I8123\_PDORemote to send a RTR (remote-transmit-request) PDO message to the slave device.

- **Syntax:**

**WORD** I8123\_PDORemote (**BYTE** SlotNo, **WORD** Cobid,  
**BYTE** \*DLen, **BYTE** \*TData,  
**BYTE** BlockMode)

- **Parameter:**

**SlotNo:** [input] I-8123W slot number (0~7).

**Cobid:** [input] COB-ID used by the PDO object.

**\*DLen:** [output] The data length of the RTR PDO message.

**\*TData:** [output] The PDO message received from the slave device.

**BlockMode:** [input] 0 means this function is non-block-function, and 1 means this function is block-function. If set this parameter to 1, the running procedure of the users' application will be held in the function until return. If set to 0, this function will return "CPM\_Processing" directly. This function will return its process status while users apply it with the same "SlotNo" and "Cobid" again. If the procedure is still not complete, it will return "CPM\_Wait".

---

### 3.5.28. I8123\_PDORemote\_Fast

- **Description:**

Use the function I8123\_PDORemote\_Fast only to send a RTR (remote-transmit-request) PDO message to the slave device but not wait for the response message.

- **Syntax:**

**WORD** I8123\_PDORemote\_Fast (**BYTE** SlotNo, **WORD** Cobid)

- **Parameter:**

**SlotNo:** [input] I-8123W slot number (0~7).

**Cobid:** [input] COB-ID used by the PDO object.

### 3.5.29.

---

### 3.5.30. I8123\_SetPDORemotePolling

- **Description:**

If the CANopen slaves do not support the event timer function of the TxPDOs, using the function I8123\_SetPDORemotePolling can config the most 125 TxPDO objects into the remote polling list. Then, the I-8123W will poll the configured TxPDOs and update the data into buffer automatically. Users can use I8123\_GetMultiPDOData to get these TxPDOs data from the buffer faster and easily.

- **Syntax:**

**WORD** I8123\_SetPDORemotePolling (**BYTE** SlotNo, **BYTE** PDOCnt,  
**WORD** \*Cobid, **WORD** PollingTime,  
**BYTE** BlockMode)

- **Parameter:**

**SlotNo:** [input] I-8123W slot number (0~7).

**PODCnt:** [input] The number of the \*Cobid array.

**\*Cobid:** [input] COB-ID array used by the TxPDO objects.

**PollingTime:** [input] The minimum polling timer for the COB-ID array.

**BlockMode:** [input] 0 means this function is non-block-function, and 1 means this function is block-function. If set this parameter to 1, the running procedure of the users' application will be held in the function until return. If set to 0, this function will return "CPM\_Processing" directly. This function will return its process status while users apply it with the same "SlotNo" and "Cobid" again. If the procedure is still not complete, it will return "CPM\_Wait".

---

### 3.5.31. I8123\_GetPDOLastData

- **Description:**

Using the function I8123\_GetPDOLastData can get the last data of the RxPDO and TxPDO from the PDO data buffer. The last PDO data is saved in PDO buffer, so it may not be the same with the real situation.

- **Syntax:**

**WORD** I8123\_GetPDOLastData (**BYTE** SlotNo, **WORD** Cobid, **BYTE** \*IsNew, **BYTE** \*DLen, **BYTE** \*RData, **BYTE** BlockMode)

- **Parameter:**

**SlotNo:** [input] I-8123W slot number (0~7).

**Cobid:** [input] COB-ID used by the PDO object.

**\*IsNew:** [output] Check the data if had been read before. 0 is been read before, and 1 is new one.

**\*DLen:** [output] The data length of the PDO message.

**\*RData:** [output] The PDO message gets from the PDO buffer.

**BlockMode:** [input] 0 means this function is non-block-function, and 1 means this function is block-function. If set this parameter to 1, the running procedure of the users' application will be held in the function until return. If set to 0, this function will return "CPM\_Processing" directly. This function will return its process status while users apply it with the same "SlotNo" and "Cobid" again. If the procedure is still not complete, it will return "CPM\_Wait".

---

### 3.5.32. I8123\_GetPDOLastData\_Fast

- **Description:**

The function is like I8123\_GetPDOLastData but gets PDO data from DPRAM instead from firmware. So I8123\_GetPDOLastData\_Fast is about twice as fast than I8123\_GetPDOLastData.

- **Syntax:**

**WORD** I8123\_GetPDOLastData\_Last (**BYTE** SlotNo, **WORD** Cobid,  
**BYTE** \*DLen, **BYTE** \*RData)

- **Parameter:**

**SlotNo:** [input] I-8123W slot number (0~7).

**Cobid:** [input] COB-ID used by the PDO object.

**\*DLen:** [output] The data length of the PDO message.

**\*RData:** [output] The PDO message gets from the PDO buffer.



---

### 3.5.33. I8123\_GetMultiPDOData

#### ● Description:

This can get the last data of the RxPDO and TxPDO from the PDO data buffer such as the function I8123\_GetPDOLastData. But the difference between these two functions is that user can use the function I8123\_GetMultiPDOData to get maximum 50 PDO data at the same time.

#### ● Syntax:

**WORD** I8123\_GetMuliPDOData (**BYTE** SlotNo, **BYTE** PDOCnt,  
**WORD** \*Cobid, **BYTE** \*IsNew,  
**BYTE** \*DLen, **BYTE** \*RData,  
**BYTE** BlockMode)

#### ● Parameter:

**SlotNo:** [input] I-8123W slot number (0~7).

**PDOPCnt:** [input] Total PDO number that want to get.

**\*Cobid:** [input] Maximum 50 COB-ID used by the PDO objects.

**\*IsNew:** [output] Check these PDO data if they have been read before.  
0 is to be read before, and 1 is new one.

**\*DLen:** [output] The total data length obtained from the PDO buffer.

**\*RData:** [output] The PDO messages get from the PDO buffer.

**BlockMode:** [input] 0 means this function is non-block-function, and 1 means this function is block-function. If set this parameter to 1, the running procedure of the users' application will be held in the function until return. If set to 0, this function will return "CPM\_Processing" directly. This function will return its process status while users apply it with the same "SlotNo" and "Cobid" again. If the procedure is still not complete, it will return "CPM\_Wait".

---

### 3.5.34. I8123\_GetMultiPDOData\_Fast

- **Description:**

The function is like I8123\_GetMultiPDOData but gets PDO data from DPRAM instead from firmware. So I8123\_GetMultiPDOData\_Fast is about twice as fast than I8123\_GetMultiPDOData .

- **Syntax:**

**WORD** I8123\_GetMultiPDOData\_Fast (**BYTE** SlotNo, **BYTE** PDOCnt,  
**WORD** \*Cobid, **BYTE** \*IsNew,  
**BYTE** \*DLen, **BYTE** \*RData)

- **Parameter:**

**SlotNo:** [input] I-8123W slot number (0~7).

**PDOPCnt:** [input] Total PDO number that want to get.

**\*Cobid:** [input] Maximum 50 COB-ID used by the PDO objects.

**\*IsNew:** [output] Check these PDO data if they have been read before.  
0 is to be read before, and 1 is new one.

**\*DLen:** [output] The total data length obtained from the PDO buffer.

**\*RData:** [output] The PDO messages get from the PDO buffer.

---

### 3.5.35. I8123\_GetRxPDOID

- **Description:**

Use the function I8123\_GetRxPDOID to get all the RxPDO COB-IDs of the specified slave, which have been installed to the master.

- **Syntax:**

**WORD** I8123\_GetRxPDOID (**BYTE** SlotNo, **BYTE** Node,  
**BYTE** \*PDO\_Cnt, **WORD** \*ID\_List,  
**BYTE** BlockMode)

- **Parameter:**

**SlotNo:** [input] I-8123W slot number (0~7).

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

**\*PDO\_Cnt:** [output] The number of installed RxPDO.

**\*ID\_List:** [output] The RxPDO COB-ID list.

**BlockMode:** [input] 0 means this function is non-block-function, and 1 means this function is block-function. If set this parameter to 1, the running procedure of the users' application will be held in the function until return. If set to 0, this function will return "CPM\_Processing" directly. This function will return its process status while users apply it with the same "SlotNo" and "Node" again. If the procedure is still not complete, it will return "CPM\_Wait".

---

### 3.5.36. I8123\_GetTxPDOID

- **Description:**

Use the function I8123\_GetTxPDOID to get all the TxPDO COB-IDs of the specified slave, which have been installed to the master.

- **Syntax:**

**WORD** I8123\_GetTxPDOID (**BYTE** SlotNo, **BYTE** Node,  
**BYTE** \*PDO\_Cnt, **WORD** \*ID\_List,  
**BYTE** BlockMode)

- **Parameter:**

**SlotNo:** [input] I-8123W slot number (0~7).

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

**\*PDO\_Cnt:** [output] The number of installed TxPDO.

**\*ID\_List:** [output] The TxPDO COB-ID list.

**BlockMode:** [input] 0 means this function is non-block-function, and 1 means this function is block-function. If set this parameter to 1, the running procedure of the users' application will be held in the function until return. If set to 0, this function will return "CPM\_Processing" directly. This function will return its process status while users apply it with the same "SlotNo" and "Node" again. If the procedure is still not complete, it will return "CPM\_Wait".

---

### 3.5.37. I8123\_InstallPDO

- **Description:**

After calling the I8123\_InstallPDO function, a PDO COB-ID will be installed in the PDO object list of the CANopen Master Library stack. If the slave device has defined the default PDO object in RxPDO1 ~ RxPDO10 and TxPDO1 ~ TxPDO10, in this case, these default PDO will be installed automatically while using the function I8123\_AddNode with automatic mode. Otherwise, the TxPDOs or RxPDOs need to be installed manually by calling the function I8123\_InstallPDO.

- **Syntax:**

**WORD** I8123\_InstallPDO(**BYTE** SlotNo, **BYTE** Node, **WORD** Cobid,  
**BYTE** RxTx, **WORD** PDO\_No,  
**BYTE** BlockMode)

- **Parameter:**

**SlotNo:** [input] I-8123W slot number (0~7).

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

**Cobid:** [input] COB-ID used by the PDO object.

**RxTx:** [input] PDO type (0: RxPDO, 1: TxPDO).

**PDO\_No:** [input] PDO mapping object No (1 ~ 512).

**BlockMode:** [input] 0 means this function is non-block-function, and 1 means this function is block-function. If set this parameter to 1, the running procedure of the users' application will be held in the function until return. If set to 0, this function will return "CPM\_Processing" directly. This function will return its process status while users apply it with the same "SlotNo" and "Node" again. If the procedure is still not complete, it will return "CPM\_Wait".

---

### 3.5.38. I8123\_DynamicPDO

#### ● Description:

This function can modify the mapping data of PDO object in the PDO object list of the CANopen Master Library stack. Take a note that before calling this function user must check if the PDO had been installed in the I-8123W.

#### ● Syntax:

**WORD** I8123\_DynamicPDO(**BYTE** SlotNo, **BYTE** Node, **WORD** Cobid, **BYTE** RxTx, **BYTE** Entry, **DWORD** EntryData, **BYTE** BlockMode)

#### ● Parameter:

**SlotNo:** [input] I-8123W slot number (0~7).

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

**Cobid:** [input] COB-ID used by the PDO object.

**RxTx:** [input] PDO type (0: RxPDO, 1: TxPDO).

**Entry:** [input] PDO mapping object subindex value (1 ~ 8).

**EntryData:** [input] A Double Word (4-byte) information of mapped application object. Users need to look up the user manual of the CANopen slave device to find the information of the application object, and obey the following example format to fill this parameter.

**For Example, EntryData = 0x64110310:** Mapping to index 0x6411 and subindex 0x03 with data length 0x10 bit (2-byte).

If the function parameters are as following, **Cobid = 0x333**, **RxTx = 0**, **Entry = 2**, **EntryData = 0x64110310**. This example will map the 16-bit data of index 0x6411 and subindex 0x03 object to the 2<sup>nd</sup> entry of COB-ID 0x333 RxPDO.

**BlockMode:** [input] 0 means this function is non-block-function, and 1 means this function is block-function. If set this parameter to 1, the running procedure of the users' application will be held in the function until return. If set to 0, this function will return "CPM\_Processing" directly. This function will return its process status while users apply it with the same "SlotNo" and "Node" again. If the procedure is still not complete, it will return "CPM\_Wait".

---

### 3.5.39. I8123\_RemovePDO

- **Description:**

The function I8123\_RemovePDO can remove a TxPDO or RxPDO object had installed by the I8123\_InstallPDO or I8123\_AddNode. This function also can remove single object mapped in TxPDO or RxPDO.

- **Syntax:**

**WORD** I8123\_RemovePDO(**BYTE** SlotNo, **BYTE** Node,  
**WORD** Cobid, **BYTE** Entry,  
**BYTE** BlockMode)

- **Parameter:**

**SlotNo:** [input] I-8123W slot number (0~7).

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

**Cobid:** [input] COB-ID used by the PDO object.

**Entry:** [input] PDO mapping object subindex value (0 ~ 8). If this value is set to 0, the specified PDO object will be removed. If others (1 ~ 8), the specified subindex content of the PDO will be removed.

**BlockMode:** [input] 0 means this function is non-block-function, and 1 means this function is block-function. If set this parameter to 1, the running procedure of the users' application will be held in the function until return. If set to 0, this function will return "CPM\_Processing" directly. This function will return its process status while users apply it with the same "SlotNo" and "Node" again. If the procedure is still not complete, it will return "CPM\_Wait".

---

### 3.5.40. I8123\_ChangePDOID

- **Description:**

Use the function I8123\_ChangePDOID to change the PDO COB-ID from old "Old\_Cobid" to new "New\_Cobid" of a slave device.

- **Syntax:**

**WORD** I8123\_ChangePDOID (**BYTE** SlotNo, **WORD** Old\_Cobid,  
**WORD** New\_Cobid, **BYTE** BlockMode)

- **Parameter:**

**SlotNo:** [input] I-8123W slot number (0~7).

**Old\_Cobid:** [input] Old COB-ID used by the PDO object.

**New\_Cobid:** [input] New COB-ID used by the PDO object.

**BlockMode:** [input] 0 means this function is non-block-function, and 1 means this function is block-function. If set this parameter to 1, the running procedure of the users' application will be held in the function until return. If set to 0, this function will return "CPM\_Processing" directly. This function will return its process status while users apply it with the same "SlotNo" and "Old\_Cobid" again. If the procedure is still not complete, it will return "CPM\_Wait".



---

### 3.5.41. I8123\_GetPDOMapInfo

- **Description:**

The function I8123\_GetPDOMapInfo can get the mapping data information of the PDO object.

- **Syntax:**

**WORD** I8123\_GetPDOMapInfo (**BYTE** SlotNo, **WORD** Cobid,  
**WORD** \*PDONo, **BYTE** \*RxTx, **BYTE** \*Tx\_Type,  
**WORD** \*Event\_Timer, **BYTE** \*Entry\_Cnt,  
**DWORD** \*Map\_Data, **BYTE** BlockMode)

- **Parameter:**

**SlotNo:** [input] I-8123W slot number (0~7).

**Cobid:** [input] COB-ID used by the PDO object.

**\*PDONo:** [output] PDO mapping object No (1 ~ 512).

**\*RxTx:** [output] PDO type (0: RxPDO, 1: TxPDO).

**\*Tx\_Type:** [output] Transmission type.

**\*Event\_Timer:** [output] PDO event timer.

**\*Entry\_Cnt:** [output] Useful PDO entry number of the PDO object.

**\*Map\_Data:** [output] Double Word array parameter. Response the mapping data of the PDO object's every useful entry.

**BlockMode:** [input] 0 means this function is non-block-function, and 1 means this function is block-function. If set this parameter to 1, the running procedure of the users' application will be held in the function until return. If set to 0, this function will return "CPM\_Processing" directly. This function will return its process status while users apply it with the same "SlotNo" and "Cobid" again. If the procedure is still not complete, it will return "CPM\_Wait".

---

### 3.5.42. I8123\_InstallPDO\_List

#### ● Description:

This function is similar with the I8123\_InstallPDO function. It can install the old or new PDO object in the PDO object list of the I-8123W. It is the same as I8123\_InstallPDO. But the I8123\_InstallPDO\_List doesn't send any message to check if the PDO object exists in the real slave. It just changes the list in the memory of the I-8123W. It means that user can use this function to install PDO and change PDO mapping data arbitrarily without disturbing the CANopen network. After using this function, the I-8123W will process the slave PDOs which have the same IDs configured by the function I8123\_InstallPDO\_List. It is very useful when the I-8123W is running in listen mode. User can use the function I8123\_RemovePDO\_List to remove the PDO object which is installed by I8123\_InstallPDO\_List.

#### ● Syntax:

**WORD** I8123\_InstallPDO\_List(**BYTE** SlotNo, **BYTE** Node, **WORD** Cobid, **BYTE** RxTx, **WORD** PDO\_No, **BYTE** Tx\_Type, **WORD** EventTimer, **BYTE** EntryUse, **DWORD** \*EntryData, **BYTE** BlockMode)

#### ● Parameter:

**SlotNo:** [input] I-8123W slot number (0~7).

**Node:** [input] Slave device Node-ID (1~127). If the "Node" parameter is the I-8123W Node-ID, the parameters, EntryUse & EntryData, will be useless. In this case, users can use the function I8123\_InstallRxPDOISR or I8123\_InstallRemotePDOISR to install the users' callback function to process the received PDOs of the I-8123W.

**Cobid:** [input] COB-ID used by the PDO object.

**RxTx:** [input] PDO type (0: RxPDO, 1: TxPDO).

**PDO\_No:** [input] PDO mapping object No (1 ~ 512).

**Tx\_Type:** [input] Transmission type.

---

**Event\_Timer:** [input] PDO event timer.

**EntryUse:** [input] Total entry of mapping object that will be installed.

**\*EntryData:** [input] Double Word array information of mapped application object. For example:

If the configuration is “**Cobid = 0x333, RxTx = 0, PDO\_No = 10, Entry = 2, EntryData[0] = 0x64110310, EntryData[1] = 0x62000108**”, it will map the RxPDO10 with COB-ID 0x333. The 1<sup>st</sup> entry is 16-bit data of index 0x6411 and subindex 0x03 object and the 2<sup>nd</sup> entry is 8-bit data of index 0x6200 and subindex 0x01 object.

**BlockMode:** [input] 0 means this function is non-block-function, and 1 means this function is block-function. If set this parameter to 1, the running procedure of the users' application will be held in the function until return. If set to 0, this function will return “CPM\_Processing” directly. This function will return its process status while users apply it with the same “SlotNo” and “Node” again. If the procedure is still not complete, it will return “CPM\_Wait”.

---

### 3.5.43. I8123\_RemovePDO\_List

- **Description:**

The function I8123\_RemovePDO\_List can remove a TxPDO or RxPDO object had installed by the I8123\_InstallPDO\_List. This function also can remove single object mapped in the TxPDO or RxPDO.

- **Syntax:**

**WORD** I8123\_RemovePDO\_List(**BYTE** SlotNo, **BYTE** Node,  
**WORD** Cobid, **BYTE** Entry,  
**BYTE** BlockMode)

- **Parameter:**

**SlotNo:** [input] I-8123W slot number (0~7).

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

**Cobid:** [input] COB-ID used by the PDO object.

**Entry:** [input] PDO mapping object subindex value (0 ~ 8). If this parameter is set to 0, the specified PDO object will be removed. If others (1 ~ 8), the specified subindex content of the PDO will be removed.

**BlockMode:** [input] 0 means this function is non-block-function, and 1 means this function is block-function. If set this parameter to 1, the running procedure of the users' application will be held in the function until return. If set to 0, this function will return "CPM\_Processing" directly. This function will return its process status while users apply it with the same "SlotNo" and "Node" again. If the procedure is still not complete, it will return "CPM\_Wait".

---

### 3.5.44. I8123\_PDUseEntry

- **Description:**

Use this function to change the useful object mapping entry of PDO object. The useful entry starts from 1 to the parameter Entry. Therefore, if the parameter Entry is 0, it means that the PDO have no useful object mapping entry.

- **Syntax:**

**WORD** I8123\_PDUseEntry(**BYTE** SlotNo, **WORD** Cobid, **BYTE** Entry, **BYTE** BlockMode)

- **Parameter:**

**SlotNo:** [input] I-8123W slot number (0~7).

**Cobid:** [input] COB-ID used by the PDO object.

**Entry:** [input] Useful entry number of PDO mapping object.

**BlockMode:** [input] 0 means this function is non-block-function, and 1 means this function is block-function. If set this parameter to 1, the running procedure of the users' application will be held in the function until return. If set to 0, this function will return "CPM\_Processing" directly. This function will return its process status while users apply it with the same "SlotNo" and "Cobid" again. If the procedure is still not complete, it will return "CPM\_Wait".

---

### 3.5.45. I8123\_PDOTxType

● **Description:**

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

● **Syntax:**

**WORD** I8123\_PDOTxType(**BYTE** SlotNo, **WORD** Cobid,  
**BYTE** Tx\_Type, **BYTE** BlockMode)

● **Parameter:**

**SlotNo:** [input] I-8123W slot number (0~7).

**Cobid:** [input] COB-ID used by the PDO object.

**Tx\_Type:** [input] Transmission type of TxPDO (0 ~ 255).

**Description of transmission type**

| Transmission type | PDO transmission |         |             |              |          |
|-------------------|------------------|---------|-------------|--------------|----------|
|                   | cyclic           | acyclic | synchronous | asynchronous | RTR only |
| 0                 |                  | X       | X           |              |          |
| 1-240             | X                |         | X           |              |          |
| 241-251           | - reserved -     |         |             |              |          |
| 252               |                  |         | X           |              | X        |
| 253               |                  |         |             | X            | X        |
| 254               |                  |         |             | X            |          |
| 255               |                  |         |             | X            |          |

**BlockMode:** [input] 0 means this function is non-block-function, and 1 means this function is block-function. If set this parameter to 1, the running procedure of the users' application will be held in the function until return. If set to 0, this function will return "CPM\_Processing" directly. This function will return its process status while users apply it with the same "SlotNo" and "Cobid" again. If the procedure is still not complete, it will return "CPM\_Wait".

---

### 3.5.46. I8123\_PDOEventTimer

- **Description:**

Use this function to change the event timer of the TxPDO. The default event timer is 0. When the event timer of the PDO object of the slave is more than 0, the PDO will be sent to master due to the parameter "Timer" automatically.

- **Syntax:**

**WORD** I8123\_PDOEventTimer(**BYTE** SlotNo, **WORD** Cobid,  
**WORD** Timer, **BYTE** BlockMode)

- **Parameter:**

**SlotNo:** [input] I-8123W slot number (0~7).

**Cobid:** [input] COB-ID used by the PDO object.

**Timer:** [input] Event timer of TxPDO. The unit is millisecond.

**BlockMode:** [input] 0 means this function is non-block-function, and 1 means this function is block-function. If set this parameter to 1, the running procedure of the users' application will be held in the function until return. If set to 0, this function will return "CPM\_Processing" directly. This function will return its process status while users apply it with the same "SlotNo" and "Cobid" again. If the procedure is still not complete, it will return "CPM\_Wait".

---

### 3.5.47. I8123\_PDOInhibitTime

- **Description:**

Use this function to set the inhibit time to the TxPDO. This time is a minimum interval for PDO transmission.

- **Syntax:**

**WORD** I8123\_PDOInhibitTime(**BYTE** SlotNo, **WORD** Cobid,  
**WORD** Time, **BYTE** BlockMode)

- **Parameter:**

**SlotNo:** [input] I-8123W slot number (0~7).

**Cobid:** [input] COB-ID used by the PDO object.

**Time:** [input] Inhibit time of TxPDO. The unit is millisecond.

**BlockMode:** [input] 0 means this function is non-block-function, and 1 means this function is block-function. If set this parameter to 1, the running procedure of the users' application will be held in the function until return. If set to 0, this function will return "CPM\_Processing" directly. This function will return its process status while users apply it with the same "SlotNo" and "Cobid" again. If the procedure is still not complete, it will return "CPM\_Wait".



---

### 3.5.48. I8123\_ChangeSYNCID

- **Description:**

Use the function I8123\_ChangeSYNCID to change the SYNC COB-ID of a slave device.

- **Syntax:**

**WORD** I8123\_ChangeSYNCID (**BYTE** SlotNo, **BYTE** Node,  
**WORD** Cobid, **BYTE** BlockMode)

- **Parameter:**

**SlotNo:** [input] I-8123W slot number (0~7).

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

**Cobid:** [input] COB-ID used by the SYNC object.

**BlockMode:** [input] 0 means this function is non-block-function, and 1 means this function is block-function. If set this parameter to 1, the running procedure of the users' application will be held in the function until return. If set to 0, this function will return "CPM\_Processing" directly. This function will return its process status while users apply it with the same "SlotNo" and "Cobid" again. If the procedure is still not complete, it will return "CPM\_Wait".

---

### 3.5.49. I8123\_SetSYNC\_List

- **Description:**

If the user uses I8123\_AddNode function to add the slave with manual mode, the function I8123\_SetSYNC\_List must be called while the SYNC ID of the slave needs to be changed or be set. The function I8123\_SetSYNC\_List can only change the SYNC COB-ID in the COB-ID list of the I-8123W, the real value stored in the slave device may be different from the configuration which is set by the function I8123\_SetSYNC\_List. The users need to confirm that by themselves.

- **Syntax:**

**WORD** I8123\_SetSYNC\_List (**BYTE** SlotNo, **BYTE** Node,  
**WORD** Cobid, **BYTE** BlockMode)

- **Parameter:**

**SlotNo:** [input] I-8123W slot number (0~7).

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

**Cobid:** [input] COB-ID used by the SYNC object.

**BlockMode:** [input] 0 means this function is non-block-function, and 1 means this function is block-function. If set this parameter to 1, the running procedure of the users' application will be held in the function until return. If set to 0, this function will return "CPM\_Processing" directly. This function will return its process status while users apply it with the same "SlotNo" and "Node" again. If the procedure is still not complete, it will return "CPM\_Wait".

---

### 3.5.50. I8123\_GetSYNCID

- **Description:**

This function can get the SYNC ID from the COB-ID list of the I-8123W.

- **Syntax:**

**WORD** I8123\_GetSYNCID (**BYTE** SlotNo, **BYTE** Node,  
**WORD** \*Cobid, **BYTE** BlockMode)

- **Parameter:**

**SlotNo:** [input] I-8123W slot number (0~7).

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

**\*Cobid:** [output] Return the COB-ID used by the SYNC object.

**BlockMode:** [input] 0 means this function is non-block-function, and 1 means this function is block-function. If set this parameter to 1, the running procedure of the users' application will be held in the function until return. If set to 0, this function will return "CPM\_Processing" directly. This function will return its process status while users apply it with the same "SlotNo" and "Node" again. If the procedure is still not complete, it will return "CPM\_Wait".

---

### 3.5.51. I8123\_SendSYNCMsg

- **Description:**

Use the function I8123\_SendSYNCMsg to send a SYNC message with specified COB-ID cyclically. If the parameter "Timer" is 0, the SYNC message will be stopped. If the parameter "Timer" is more than 0, the function will send SYNC message per "Timer" millisecond until finish the parameter "Times". When the "Times" is set to 0, the function will send SYNC message continuously until set "Timer" to 0. Users can set at most 5 SYNC messages with different ID to be sent cyclically.

- **Syntax:**

**WORD** I8123\_SendSYNCMsg(**BYTE** SlotNo, **WORD** Cobid,  
**WORD** Timer, **DWORD** Times,  
**BYTE** BlockMode)

- **Parameter:**

**SlotNo:** [input] I-8123W slot number (0~7).

**Cobid:** [input] COB-ID used by the SYNC object.

**Timer:** [input] SYNC message transmission period. If the timer is 0, the SYNC message will be stopped.

**Times:** [input] SYNC message transmission times. If the time is 0, the SYNC message will be sending until "Timer" is set to 0.

**BlockMode:** [input] 0 means this function is non-block-function, and 1 means this function is block-function. If set this parameter to 1, the running procedure of the users' application will be held in the function until return. If set to 0, this function will return "CPM\_Processing" directly. This function will return its process status while users apply it with the same "SlotNo" and "Cobid" again. If the procedure is still not complete, it will return "CPM\_Wait".

---

### 3.5.52. I8123\_GetCyclicSYNCInfo

- **Description:**

This function can get at most 5 SYNC messages information which have been configured by the function I8123\_SendSYNCMsg. User can know what SYNC ID had been set.

- **Syntax:**

**WORD** I8123\_GetCyclicSYNCInfo(**BYTE** SlotNo, **WORD** \*Cobid,  
**WORD** \*Timer, **DWORD** \*Times,  
**BYTE** BlockMode)

- **Parameter:**

**SlotNo:** [input] I-8123W slot number (0~7).

**\*Cobid:** [input] COB-ID array. Return most 5 SYNC ID.

**\*Timer:** [input] 5 WORD array. Each value is the cyclic period of the SYNC message.

**\*Times:** [input] 5 Double WORD array. Each one is the SYNC message sending times that set before.

**BlockMode:** [input] 0 means this function is non-block-function, and 1 means this function is block-function. If set this parameter to 1, the running procedure of the users' application will be held in the function until return. If set to 0, this function will return "CPM\_Processing" directly. This function will return its process status while users apply it with the same "SlotNo" again. If the procedure is still not complete, it will return "CPM\_Wait".

---

### 3.5.53. I8123\_ChangeEMCYID

- **Description:**

Use the function I8123\_ChangeEMCYID to change the EMCY COB-ID of a specific slave device.

- **Syntax:**

**WORD** I8123\_ChangeEMCYID (**BYTE** SlotNo, **BYTE** Node,  
**WORD** Cobid, **BYTE** BlockMode)

- **Parameter:**

**SlotNo:** [input] I-8123W slot number (0~7).

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

**Cobid:** [input] COB-ID used by the EMCY object.

**BlockMode:** [input] 0 means this function is non-block-function, and 1 means this function is block-function. If set this parameter to 1, the running procedure of the users' application will be held in the function until return. If set to 0, this function will return "CPM\_Processing" directly. This function will return its process status while users apply it with the same "SlotNo" and "Node" again. If the procedure is still not complete, it will return "CPM\_Wait".

---

### 3.5.54. I8123\_SetEMCY\_List

- **Description:**

If the user uses I8123\_AddNode function to add the slave with manual mode, the function I8123\_SetEMCY\_List must be called while the EMCY ID of the slave needs to be changed or be set. I8123\_SetEMCY\_List only can change the EMCY COB-ID in the COB-ID list of the I-8123W. Afterwards, the I-8123W process the EMCY messages with the specific EMCY COB-ID which is configured by the function I8123\_SetEMCY\_List.

- **Syntax:**

**WORD** I8123\_SetEMCY\_List (**BYTE** SlotNo, **BYTE** Node,  
**WORD** Cobid, **BYTE** BlockMode)

- **Parameter:**

**SlotNo:** [input] I-8123W slot number (0~7).

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

**Cobid:** [input] COB-ID used by the EMCY object.

**BlockMode:** [input] 0 means this function is non-block-function, and 1 means this function is block-function. If set this parameter to 1, the running procedure of the users' application will be held in the function until return. If set to 0, this function will return "CPM\_Processing" directly. This function will return its process status while users apply it with the same "SlotNo" and "Node" again. If the procedure is still not complete, it will return "CPM\_Wait".

---

### 3.5.55. I8123\_GetEMCYID

- **Description:**

This function can get the EMCY ID from the COB-ID list of the I-8123W.

- **Syntax:**

**WORD** I8123\_GetEMCYID (**BYTE** SlotNo, **BYTE** Node,  
**WORD** \*Cobid, **BYTE** BlockMode)

- **Parameter:**

**SlotNo:** [input] I-8123W slot number (0~7).

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

**\*Cobid:** [output] Return the COB-ID used by the EMCY object.

**BlockMode:** [input] 0 means this function is non-block-function, and 1 means this function is block-function. If set this parameter to 1, the running procedure of the users' application will be held in the function until return. If set to 0, this function will return "CPM\_Processing" directly. This function will return its process status while users apply it with the same "SlotNo" and "Node" again. If the procedure is still not complete, it will return "CPM\_Wait".



---

### 3.5.56. I8123\_ReadLastEMCY

- **Description:**

This function can check if one slave had produced EMCY. If yes, this function will return the last EMCY message of the specific slave.

- **Syntax:**

**WORD** I8123\_ReadLastEMCY (**BYTE** SlotNo, **BYTE** Node,  
**BYTE** \*IsNew, **BYTE** \*RData,  
**BYTE** BlockMode)

- **Parameter:**

**SlotNo:** [input] I-8123W slot number (0~7).

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

**\*IsNew:** [output] Check the data if had been read before. 0 is been read before, and 1 is new one.

**\*RData:** [output] 8-byte EMCY message gets from the EMCY buffer.

**BlockMode:** [input] 0 means this function is non-block-function, and 1 means this function is block-function. If set this parameter to 1, the running procedure of the users' application will be held in the function until return. If set to 0, this function will return "CPM\_Processing" directly. This function will return its process status while users apply it with the same "SlotNo" and "Node" again. If the procedure is still not complete, it will return "CPM\_Wait".

---

### 3.5.57. I8123\_GetBootUpNodeAfterAdd

- **Description:**

If users don't know that which slave node occurred the boot-up message and which I-8123W received it. Users can use the function I8123\_GetBootUpNodeAfterAdd. This function can get not only the slave node ID but also the slot number of the I-8123W. The parameter, SlotNo, indicates the number of the I-8123W which receives the boot-up message. The parameter, Node, is the ID of the slave node which produces the boot-up message. The I8123\_GetBootUpNodeAfterAdd function is usually applied with the function I8123\_InstallBootUpISR. But note that, this function can only get the slave node ID that has already been added (I8123\_AddNode) to the I-8123W.

- **Syntax:**

**WORD** I8123\_GetBootUpNodeAfterAdd(**BYTE** \*SlotNo, **BYTE** \*Node)

- **Parameter:**

\***SlotNo:** [output] Get the slot number of the I-8123W which receives the boot-up message.

\***Node:** [output] Get the slave node ID of the received boot-up message.

---

### 3.5.58. I8123\_GetEMCYData

#### ● Description:

If users don't know that which slave node occurred the EMCY message and which I-8123W received it. Users can use the function I8123\_GetEMCYData. This function can get not only the EMCY message with the slave node ID but also the slot number of the I-8123W. The parameter, SlotNo, indicates the number of the I-8123W which receives the EMCY message. The parameter, Node, is the ID of the slave node which produces the EMCY. The parameters, \*RData, is the 8-bytes EMCY message data. The function I8123\_GetEMCYData is usually applied with the function I8123\_InstallEMCYISR. About the demo please refer to the section 4.1.2 NMT\_Protocol.

#### ● Syntax:

```
WORD I8123_GetEMCYData (BYTE *SlotNo, BYTE *Node,  
                        BYTE *RData)
```

#### ● Parameter:

\***SlotNo**: [output] Get the slot number of the I-8123W which receives the EMCY message.

\***Node**: [output] Get the slave node ID of the received EMCY message.

\***RData**: [output] 8-byte EMCY message obtained from the EMCY buffer.

---

### 3.5.59. I8123\_GetNMTErr

#### ● Description:

User can use the function I8123\_GetNMTErr to check if the I-8123W gets NMT Error Event for any slave node. The parameters of the function indicate that which I-8123W gets the NMT Error Event, which node produces this event, and what kind of event it is. The parameter, SlotNo, indicates the number of the I-8123W which indicates the Heartbeat\_Event or Node\_Guarding\_Event. The parameter, Node, is the ID of the slave node which responds the heartbeat protocol or guarding protocol. The parameter, NMTErr, is the NMTErr event mode. If the NMTErr event is Node\_Guarding\_Event, the NMTErr parameter is CPM\_Node\_Guarding\_Event or else the CPM\_Heartbeat\_Event is obtained. The function I8123\_GetNMTErr is usually applied with the function I8123\_InstallNMTErrISR. About the demo please refer to the section 4.1.2 NMT\_Protocol.

#### ● Syntax:

```
WORD I8123_GetNMTErr (BYTE *SlotNo, BYTE *Node,  
                     BYTE *NMTErr)
```

#### ● Parameter:

\***SlotNo**: [output] Get the slot number of the I-8123W which receives the NMT Error Event.

\***Node**: [output] Get the slave node ID of the NMT Error Event.

\***NMTErr**: [output] The value CPM\_Node\_Guarding\_Event indicates the Node Guarding Event, and the CPM\_Heartbeat\_Event is the Heartbeat Event.

---

### 3.5.60. I8123\_InstallBootUpISR

- **Description:**

This function allows the user to apply the slave boot-up IST (interrupt service thread). When the user puts his boot-up process into this function, all the boot-up triggered by the slaves will go to the boot-up IST. If the boot-up message of a slave which has been added to the I-8123W is happen, the I-8123W will go into the boot-up process to do some specified mechanism which follows the user's boot-up process.

- **Syntax:**

**WORD** I8123\_InstallBooUpISR(**BYTE** SlotNo, **void** (\*BOOTISR)( ))

- **Parameter:**

**SlotNo:** [input] I-8123W slot number (0~7).

**(\*BOOTISR)( ): [input]** The pointer which points a function with format "void XXX( )". The XXX is the function name of the user's boot-up process. This process is usually applied with the function I8123\_GetBootUpNodeAfterAdd.

---

### 3.5.61. I8123\_RemoveBootUpISR

- **Description:**

When the user doesn't need the boot-up IST function, call this function to remove the user's IST.

- **Syntax:**

**WORD** I8123\_RemoveBootUpISR(**BYTE** SlotNo)

- **Parameter:**

**SlotNo:** [input] I-8123W slot number (0~7).

---

### 3.5.62. I8123\_InstallEMCYISR

- **Description:**

This function allows the user to apply the EMCY IST (interrupt service thread). When the user puts his EMCY process into this function, all the EMCY triggered by the slaves will go to the EMCY IST. If the EMCY of a slave is happen, the I-8123W will go into the EMCY process to do some security mechanism which follows the user's EMCY process.

- **Syntax:**

**WORD** I8123\_InstallEMCYISR(**BYTE** SlotNo, **void** (\*EMCYISR)( ))

- **Parameter:**

**SlotNo:** [input] I-8123W slot number (0~7).

**(\*EMCYISR)( ):** [input] The pointer which points a function with format "void XXX( )". The XXX is the function name of the user's EMCY process. This process is usually applied with the function I8123\_GetEMCYData.

---

### 3.5.63. I8123\_RemoveEMCYISR

- **Description:**

When the user doesn't need the EMCY IST function, call this function to remove the user's IST.

- **Syntax:**

**WORD** I8123\_RemoveEMCYISR(**BYTE** SlotNo)

- **Parameter:**

**SlotNo:** [input] I-8123W slot number (0~7).



---

### 3.5.64. I8123\_InstallNMTErrISR

- **Description:**

This function allows the user to apply NMTErr IST (interrupt service thread). When the user puts his NMTErr process into this function, all the Heartbeat\_Event and Node\_Guarding\_Event triggered by the slaves will go to the IST. If the user had used the I8123\_NMTGuarding to enable the guarding protocol or had used the I8123\_Heartbeat to enable the heartbeat protocol, the I-8123W will go into the NMTErr IST to do the user's NMTErr process while the guarding confirms or heartbeat indicator doesn't be received.

- **Syntax:**

**WORD** I8123\_InstallNMTErrISR(**BYTE** SlotNo,  
**void** (\*NMTErrISR)( ))

- **Parameter:**

**SlotNo:** [input] I-8123W slot number (0~7).

**(\*NMTErrISR)( ): [input]** The pointer which points a function with format "void XXX( )". The XXX is the function name of the user's process. This process is usually applied with the function I8123\_GetNMTError.

---

### 3.5.65. I8123\_RemoveNMTErrISR

- **Description:**

When the user doesn't need the NMTErr IST function, call this function to remove the user's IST.

- **Syntax:**

**WORD** I8123\_RemoveNMTErrISR(**BYTE** SlotNo)

- **Parameter:**

**SlotNo:** [input] I-8123W slot number (0~7).

---

### 3.5.66. I8123\_GetMasterReadSDOEvent

- **Description:**

Using this function can get all the read SDO messages sent to the specific node ID of the I-8123W. For example, the I-8123W is initialized with node ID 2. If someone sends an SDO message with the COB-ID 0x602 to the I-8123W for reading an object, users can use the function I8123\_GetMasterReadSDOEvent for obtaining this SDO message, and respond some information to the SDO sender. The parameter, SlotNo, indicates the number of the I-8123W which receives the read SDO message. The parameters, Index and SubIndex, are the object indicator. The function I8123\_GetMasterReadSDOEvent is usually applied with the function I8123\_InstallReadSDOISR. About the demo please refer to the section 4.1.6 SDO\_PDO\_ISR.

**Note: The function is valid while the Node parameter of the function I8123\_InitMaster is > 0.**

- **Syntax:**

**WORD** I8123\_GetMasterReadSDOEvent (**BYTE** \*SlotNo,  
**WORD** \*Index, **BYTE** \*SubIndex)

- **Parameter:**

\***SlotNo:** [output] Get the slot number of the I-8123W which receives the read SDO message.

\***Index:** [output] Get the object index of the SDO message.

\***SubIndex:** [output] Get the object subindex of the SDO message.

---

### 3.5.67. I8123\_GetMasterWriteSDOEvent

#### ● Description:

Using this function can get all the write SDO messages sent to the specific node ID of the I-8123W. For example, the I-8123W is initialized with node ID 2. If someone sends an SDO message with the COB-ID 0x602 to the I-8123W for writing an object, users can use the function I8123\_GetMasterWriteSDOEvent for obtaining this SDO message. The parameter, SlotNo, indicates the number of the I-8123W which receives the write SDO message. The parameters, Index and SubIndex, are the object indicator. The parameter WLen is the data length of the parameter \*WData. The function I8123\_GetMasterWriteSDOEvent is usually applied with the function I8123\_InstallWriteSDOISR. About the demo please refer to the section 4.1.6 SDO\_PDO\_ISR.

**Note: The function is valid while the Node parameter of the function I8123\_InitMaster is > 0.**

#### ● Syntax:

```
WORD I8123_GetMasterWriteSDOEvent (BYTE *SlotNo,  
                                  WORD *Index, BYTE *SubIndex,  
                                  BYTE *WLen, BYTE *WData)
```

#### ● Parameter:

**\*SlotNo:** [output] Get the slot number of the I-8123W which receives the write SDO message.

**\*Index:** [output] Get the object index of the SDO message.

**\*SubIndex:** [output] Get the object subindex of the SDO message.

**\*WLen:** [output] The data length of the write data.

**\*WData:** [output] Return 0~4 bytes of the SDO write data.

---

### 3.5.68. I8123\_ResponseMasterSDO

#### ● Description:

Using this function can reply the SDO messages to the SDO sender. For example, the I-8123W is initialized with node ID 2. If someone sends a SDO message with the COB-ID 0x602 for reading or writing the object of the I-8123W, the I-8123W need to reply the corresponding SDO message, use the function I8123\_ResponseMasterSDO to do it. When users implement the function I8123\_ResponseMasterSDO, the I-8123W will send a SDO message with COB-ID 0x582 to the CANopen network. This function is usually applied with the SDO ISR series function .About the demo please refer to the section 4.1.6 SDO\_PDO\_ISR.

**Note1: The function is valid while the Node parameter of the function I8123\_InitMaster is > 0.**

**Note2: If the I-8123W want to reply a SDO Abort message, please use the function I8123\_SDOAbortTransmit (section 3.5.20) to do it.**

#### ● Syntax:

**WORD** I8123\_ResponseMasterSDO (**BYTE** SlotNo, **BYTE** ResType, **WORD** Index, **BYTE** SubIndex, **BYTE** Len, **BYTE** \*Data)

#### ● Parameter:

**SlotNo:** [input] I-8123W slot number (0~7).

**ResType:** [input] Response type of SDO message, 0 for replying the read SDO message and 1 for the write SDO message.

**Index:** [input] Object index of object dictionary of slave devices.

**SubIndex:** [input] Object subindex of object dictionary of slave devices.

**Len:** [input] The data length of the response data. If the ResType is 1 (write type), the Len and \*Data parameter is useless.

**\*Data:** [input] Return 0~4 bytes of the SDO response data.

---

### 3.5.69. I8123\_InstallReadSDOISR

- **Description:**

This function allows the user to apply the ReadSDO IST (interrupt service thread) of I-8123W. When the user puts his read SDO process into this function, all the read SDO messages sent to the specified I-8123W will trigger the IST. For example, the I-8123W is initialized with node ID 2. If someone sends an SDO message with the COB-ID 0x602 for reading the object of the I-8123W, the I-8123W will go into the IST if the user has installed it.

**Note: The function is valid while the Node parameter of the function I8123\_InitMaster is > 0.**

- **Syntax:**

```
WORD I8123_InstallReadSDOISR(BYTE SlotNo,  
                             void (*RSDOISR)( ))
```

- **Parameter:**

**SlotNo:** [input] I-8123W slot number (0~7).

**(\*RSDOISR)( ): [input]** The pointer which points a function with format "void XXX( )". The XXX is the function name of the user's process. This process is usually used with the function I8123\_GetMasterReadSDOEvent.

---

### 3.5.70. I8123\_RemoveReadSDOISR

- **Description:**

When the user doesn't need the ReadSDO IST function, call this function to remove the user IST.

- **Syntax:**

**WORD** I8123\_RemoveReadSDOISR(**BYTE** SlotNo)

- **Parameter:**

**SlotNo:** [input] I-8123W slot number (0~7).

---

### 3.5.71. I8123\_InstallWriteSDOISR

- **Description:**

This function allows the user to apply the WriteSDO IST (interrupt service routine) of the I-8123W. When the user puts the process into this function, all the written SDO messages sent to the specified I-8123W will trigger the IST. For example, the I-8123W is initialized with node ID 2. If someone sends an SDO message with the COB-ID 0x602 to the I-8123W for writing an object, the I-8123W will go into the IST if the user has installed it.

**Note: The function is valid while the Node parameter of the function I8123\_InitMaster is > 0.**

- **Syntax:**

```
WORD I8123_InstallWriteSDOISR(BYTE SlotNo,  
                               void (*WSDOISR)( ))
```

- **Parameter:**

**SlotNo:** [input] I-8123W slot number (0~7).

**(\*WSDOISR)( ): [input]** The pointer which points a function with format "void XXX( )". The XXX is the function name of user's process. This process is usually used with the function I8123\_GetMasterWriteSDOEvent.



---

### 3.5.72. I8123\_RemoveWriteSDOISR

- **Description:**

When the user doesn't need the WriteSDO IST function, call this function to remove the user IST.

- **Syntax:**

**WORD** I8123\_RemoveWriteSDOISR(**BYTE** SlotNo)

- **Parameter:**

**SlotNo:** [input] I-8123W slot number (0~7).

---

### 3.5.73. I8123\_GetMasterRemotePDOEvent

- **Description:**

Using this function can get all the Remote PDO messages sent to the I-8123W. For example, the I-8123W has used the function I8123\_InstallPDO\_List to install an I-8123W's TxPDO object with the COB-ID 0x444. If someone sends a Remote PDO message with the COB-ID 0x444 to the I-8123W, users can use the function I8123\_GetMasterRemotePDOEvent to get this PDO message. The parameter, SlotNo, indicates the number of the I-8123W which receives the Remote PDO message. The parameter, Cobid, is the PDO COB-ID sent to the I-8123W. This function is usually used with the function I8123\_InstallRemotePDOISR. About the demo please refer to the section 4.1.6 SDO\_PDO\_ISR.

**Note: The function is valid while the Node parameter of the function I8123\_InitMaster is > 0.**

- **Syntax:**

**WORD** I8123\_GetMasterRemotePDOEvent (**BYTE** \*SlotNo,  
**WORD** \*CobId)

- **Parameter:**

\***SlotNo:** [output] Get the slot number of the I-8123W which receives the Remote PDO message.

\***CobId:** [output] Return the COB-ID of the Remote PDO message.

---

### 3.5.74. I8123\_GetMasterRxPDOEvent

- **Description:**

Using this function can get all the RxPDO messages sent to the I-8123W. For example, the I-8123W has used the function I8123\_InstallPDO\_List to install an I-8123W's RxPDO object with the COB-ID 0x333. If someone sends an RxPDO message with the COB-ID 0x333 to the I-8123W, users can use this function to get this RxPDO message. The parameter, SlotNo, indicates the number of the I-8123W which receives the RxPDO message. The parameter, Cobid, is the RxPDO COB-ID ID. The two parameters, \*WLen and \*WData, are the data length and contents of the RxPDO message. This function is usually applied with the function I8123\_InstallRxPDOISR. About the demo please refer to the section 4.1.6 SDO\_PDO\_ISR.

**Note: The function is valid while the Node parameter of the function I8123\_InitMaster is > 0.**

- **Syntax:**

**WORD** I8123\_GetMasterRxPDOEvent (**BYTE** \*SlotNo, **WORD** \*CobId, **BYTE** \*WLen, **BYTE** \*WData)

- **Parameter:**

\***SlotNo:** [output] Get the slot number of the I-8123W which receives the RxPDO message.

\***CobId:** [output] Return the RxPDO COB-ID.

\***WLen:** [output] The data length of the RxPDO data.

\***WData:** [output] Return 0~4 bytes of the RxPDO data.

---

### 3.5.75. I8123\_ResponseMasterPDO

- **Description:**

Using this function can reply the Remote PDO messages to the sender. For example, the I-8123W has used I8123\_InstallPDO\_List to install a TxPDO object with the COB-ID 0x444. If someone sends a Remote PDO message with the COB-ID 0x444 to the I-8123W, and the I-8123W needs to reply a TxPDO message, users can use this function to do it. When users implement the function I8123\_ResponseMasterPDO, the I-8123W will send a TxPDO message to the CANopen network. This function is usually used with the I8123\_InstallRemotePDOISR function. About the demo please refer to the section 4.1.6 SDO\_PDO\_ISR.

**Note: The function is valid while the Node parameter of the function I8123\_InitMaster is > 0.**

- **Syntax:**

**WORD** I8123\_ResponseMasterPDO (**BYTE** SlotNo, **WORD** CobId, **BYTE** Len, **BYTE** \*Data)

- **Parameter:**

**SlotNo:** [input] I-8123W slot number (0~7).

**CobId:** [input] TxPDO COB-ID for replying the PDO message.

**Len:** [input] The data length of the response data.

**\*Data:** [input] Return the COB-ID of the TxPDO PDO message.

---

### 3.5.76. I8123\_InstallRxPDOISR

#### ● Description:

This function allows the user to apply the RxPDO IST (interrupt service routine) of the I-8123W. When the user puts his process into this function, all the RxPDO messages with the I-8123W's PDO objects will trigger the IST. For example, the I-8123W has used I8123\_InstallPDO\_List to install a PDO object with the COB-ID 0x333 of the I-8123W. If some one sends a PDO message with the COB-ID 0x333 to the I-8123W, the I-8123W will go into the IST if the user had installed it.

**Note: The function will usefully when the Node parameter of the function I8123\_InitMaster is > 0.**

#### ● Syntax:

```
WORD I8123_InstallRxPDOISR(BYTE SlotNo,  
                           void (*RXPDOISR)( ))
```

#### ● Parameter:

**SlotNo:** [input] I-8123W slot number (0~7).

**(\*RXPDOISR)( ): [input]** The pointer which points a function with format "void XXX( )". The XXX is the function name of user's process. This process is usually used with the function I8123\_GetMasterRxPDOEvent.

---

### 3.5.77. I8123\_RemoveRxPDOISR

- **Description:**

When the user doesn't need the RxPDO IST function, call this function to remove the user IST.

- **Syntax:**

**WORD** I8123\_RemoveRxPDOISR(**BYTE** SlotNo)

- **Parameter:**

**SlotNo:** [input] I-8123W slot number (0~7).

---

### 3.5.78. I8123\_InstallRemotePDOISR

- **Description:**

This function allows the user to apply the RemotePDO IST (interrupt service routine) of the I-8123W. When the user puts his process into this function, all the Remote PDO messages of the I-8123W's PDO objects will trigger the IST. For example, the I-8123W has used I8123\_InstallPDO\_List to install a TxPDO object with the COB-ID 0x444 of the I-8123W. If some one sends a Remote PDO message with the COB-ID 0x444 to the I-8123W, the I-8123W will go into the IST if the user had installed it.

**Note: The function will usefully when the Node parameter of the function I8123\_InitMaster is > 0.**

- **Syntax:**

**WORD** I8123\_InstallRemotePDOISR(**BYTE** SlotNo,  
**void** (\*REMOTEPDOISR)( ))

- **Parameter:**

**SlotNo:** [input] I-8123W slot number (0~7).

**(\*REMOTEPDOISR)( ): [input]** The pointer which points a function with format "void XXX( )". The XXX is the function name of user's process. This process is usually used with the function I8123\_GetMasterRemotePDOEvent.

---

### 3.5.79. I8123\_RemoveRemotePDOISR

- **Description:**

When the user doesn't need the RemotePDO IST function, call this function to remove the user IST.

- **Syntax:**

**WORD** I8123\_RemoveRemotePDOISR(**BYTE** SlotNo)

- **Parameter:**

**SlotNo:** [input] I-8123W slot number (0~7).



---

### 3.5.80. I8123\_SyncGroupCycUpdata

#### ● Description:

Before use I8123\_SyncGroupCycSend to send SYNC and PDO group data, user should use this function to fill the Max 15 cycle buffer. If users want their devices run more group cycles, users can still use this function to continuously fill the cycle buffer after execute I8123\_SyncGroupCycSend.

#### ● Syntax:

**WORD** I8123\_SyncGroupCycUpdata(**BYTE** SlotNo, **BYTE** RecordNum, **BYTE** StepNum, **WORD** \*PDOId, **BYTE** \*GroupData, **BYTE** BlockMode)

#### ● Parameter:

**SlotNo:** [input] I-8123W slot number (0~7).

**RecordNum:** [input] PDO numbers of this SYNC group. Every update, the PDO numbers of the group data must be the same. This parameter is < 32 (RecordNum \* StepNum < 60).

**StepNum:** [input] Cycle number of this update. This parameter is < 15.

**\*PDOId:** [input] PDO ID array of these group data. The array size is according to parameter RecordNum.

**\*GroupData:** [input] PDO group data buffer. Every PDO reserved 8 byte buffers. Array size is "RecordNum \* StepNum \* 8".

**BlockMode:** [input] 0 means this function is non-block-function, and 1 means this function is block-function. If set this parameter to 1, the running procedure of the users' application will be held in the function until return. If set to 0, this function will return "CPM\_Processing" directly. This function will return its process status while users apply it with the same "BoardNo" again. If the procedure is still not complete, it will return "CPM\_Wait".

---

### 3.5.81. I8123\_SyncGroupCycSend

- **Description:**

Use this function start to send SYNC and PDO group data from group cycle buffer (updated by I8123\_SyncGroupCycUpdata).

- **Syntax:**

**WORD** I8123\_SyncGroupCycSend(**BYTE** SlotNo, **DWORD** TotalCycle, **DWORD** CycInterval, **BYTE** BlockMode)

- **Parameter:**

**SlotNo:** [input] I-8123W slot number (0~7).

**TotalCycle:** [input] Total cycle of these SYNC group mission. TotalCycle = 0 means unlimited.

**CycInterval:** [input] Cycle timer, unit is 0.1 ms and minimum value is 5 (0.5 ms).

**BlockMode:** [input] 0 means this function is non-block-function, and 1 means this function is block-function. If set this parameter to 1, the running procedure of the users' application will be held in the function until return. If set to 0, this function will return "CPM\_Processing" directly. This function will return its process status while users apply it with the same "BoardNo" again. If the procedure is still not complete, it will return "CPM\_Wait".

---

### 3.5.82. I8123\_SyncGroupClear

- **Description:**

Use this function to clear all the SYNC and PDO group data in group cycle buffer (updated by I8123\_SyncGroupCycUpdata). If the last SYNC group is stopped for some problem, before start the next new SYNC group data, recommend use this function to clear the group data buffer first.

- **Syntax:**

**WORD** I8123\_SyncGroupClear(**BYTE** SlotNo, **BYTE** BlockMode)

- **Parameter:**

**SlotNo:** [input] I-8123W slot number (0~7).

**BlockMode:** [input] 0 means this function is non-block-function, and 1 means this function is block-function. If set this parameter to 1, the running procedure of the users' application will be held in the function until return. If set to 0, this function will return "CPM\_Processing" directly. This function will return its process status while users apply it with the same "BoardNo" again. If the procedure is still not complete, it will return "CPM\_Wait".

---

### 3.5.83. I8123\_GetSYNCGroupStopState

- **Description:**

Use this function to get stop state of the SYNC and PDO group data. This function can use with SYNCISR user-defined function (please refer to 4.5.86 I8123\_InstallSYNCGroupISR).

- **Syntax:**

**WORD** I8123\_GetSyncGroupStopState(**BYTE** \*SlotNo, **BYTE** \*State)

- **Parameter:**

**\*SlotNo:** [output] Get the slot number of the I8123 which receives the stop state.

**\*State:** [output] 1: SYNC and PDO group data send complete. 2: SYNC and PDO group data is stopped by send fail.

---

### 3.5.84. I8123\_InstallSYNCGroupISR

- **Description:**

This function allows the user to apply the SYNCGroup IST (interrupt service routine) of the I-8123W. When the user puts his process into this function, if the SYNC and PDO group data is stopped or completed, a SYNC group stop state event will trigger the IST.

- **Syntax:**

**WORD** I8123\_InstallSYNCGroupISR(**BYTE** SlotNo, **void**  
(\*SYNCISR)())

- **Parameter:**

**SlotNo:** [input] I-8123W slot number (0~7).

**\*SYNCISR:** [input] The pointer which points a function with format “void XXX( )”. The XXX is the function name of user’s process. This process is usually used with the function I8123\_GetSyncGroupStopState.

---

### 3.5.85. I8123\_RemoveSYNCGroupISR

- **Description:**

When the user doesn't need the SYNCGroup IST function, call this function to remove the user IST.

- **Syntax:**

**WORD** I8123\_RemoveSYNCGroupISR(**BYTE** SlotNo)

- **Parameter:**

**SlotNo:** [input] I-8123W slot number (0~7).

---

## 4. Demo Programs

The I-8123W provides 10 demos of the various applications for NMT protocol, SDO protocol, PDO protocol, NMT Error IST...etc. All these demos support eVC++ 4.0, VB.net 2005, and C# 2005. There is also a tool, CPMUtility, to control/monitor CANopen slaves with I-8123W easily and quickly. Users can find these demos and utility tool in the fieldbus CD or on the web site.

The path of field bus CD

**fieldbus cd://canopen/master/I 8123w**

The address of the web site

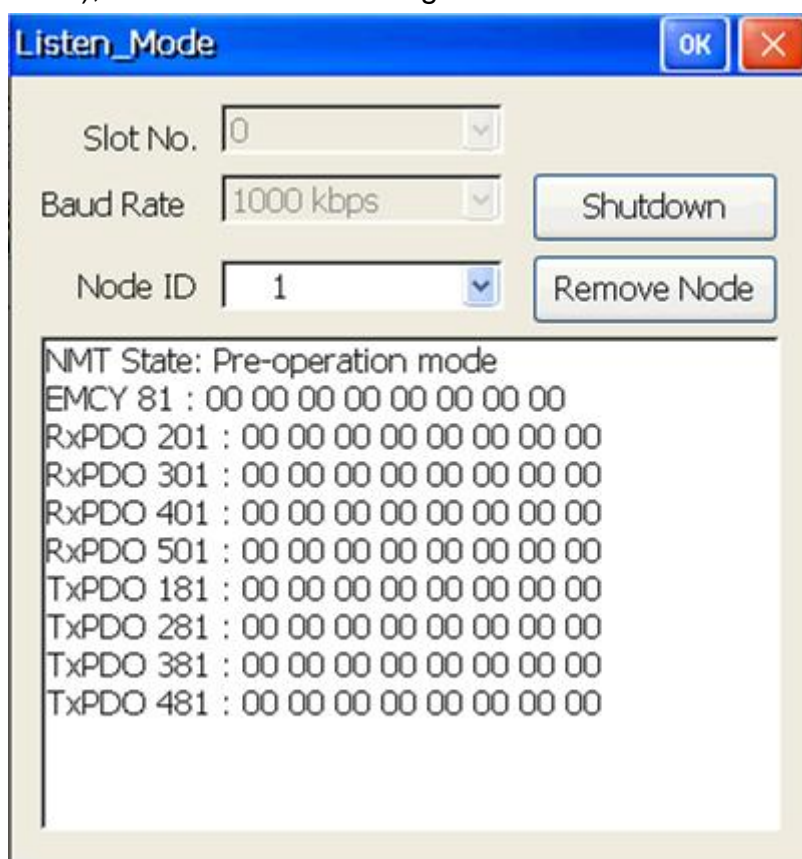
**[http://www.icpdas.com/products/Remote\\_IO/can\\_bus/I-8123w.htm](http://www.icpdas.com/products/Remote_IO/can_bus/I-8123w.htm)**

### 4.1. 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 demo program includes some functions of the CANopen master library. The relationship between CANopen master library functions and demo programs are displayed in the following description.

### 4.1.1. Listen\_Mode

Initialize the I-8123W with the “listen mode” and add slave nodes with the “manual mode”, then the I-8123W listens CANopen messages only and does not send any message to the CANopen network. In this demo, the I-8123W will listen NMT state, 4 TxPDO messages (with the COB ID 0x180+Node ID, 0x280+Node ID, 0x380+Node ID, and 0x480+Node ID), 4 RxPDO messages (with the COB ID 0x200+Node ID, 0x300+Node ID, 0x400+Node ID, and 0x500+Node ID), and the EMCY messages.



Applied function list:

[I8123\\_InitMaster](#), [I8123\\_Shutdown](#), [I8123\\_AddNode](#), [I8123\\_RemoveNode](#),  
[I8123\\_EDS\\_Load](#), [I8123\\_SetMasterMode](#), [I8123\\_NMTGetState](#),  
[I8123\\_InstallPDO\\_List](#), [I8123\\_GetPDOLastData](#), [I8123\\_GetRxPDOID](#),  
[I8123\\_GetTxPDOID](#), [I8123\\_SetEMCY\\_List](#), [I8123\\_GetEMCYID](#),  
[I8123\\_ReadLastEMCY](#).



---

## 4.1.2. NMT\_Protocol

This is a NMT network control demo. The demo not only tells users how to control the NMT status of a specific slave node, but also how to protect the slave through the “Guarding” and “Heartbeat” functions.

The screenshot shows a software window titled "NMT\_Protocol" with standard Windows window controls (OK, X). The interface includes several configuration options:

- Slot No.: 0
- Baud Rate: 1000 kbps
- Node: 1
- Node State: Operation
- Guarding Time: 1000
- Life: 2
- Heartbeat (ms): 1000
- Consumer (ms): 2500

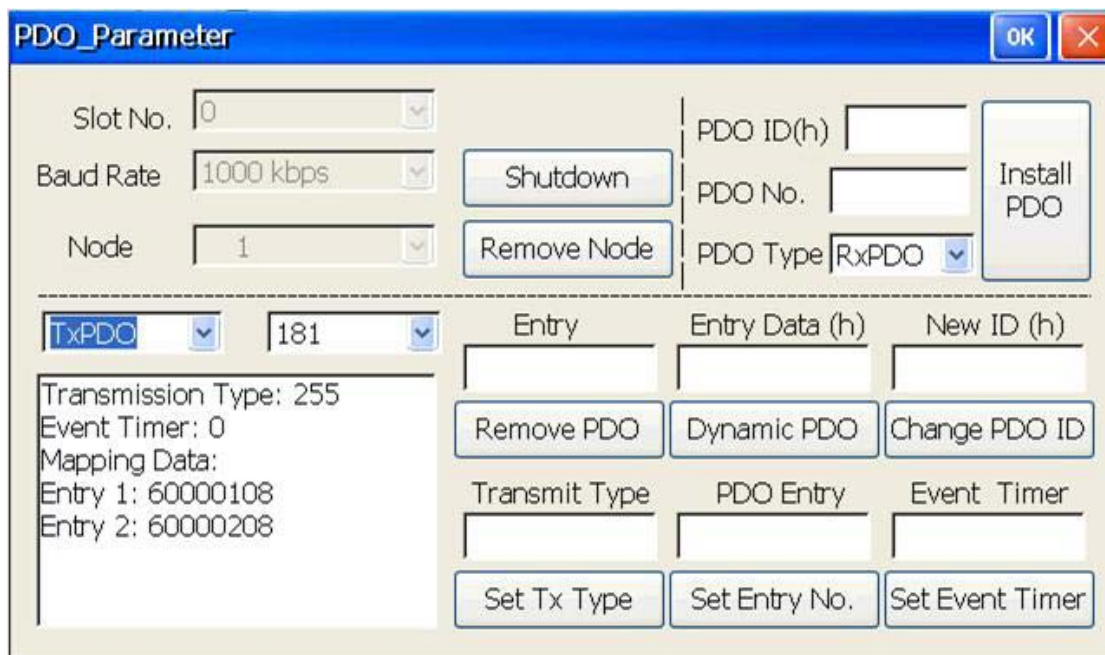
Buttons available include Shutdown, Remove Node, Set Status, Set Guarding, Set Heartbeat, and Clear. A text area at the bottom displays "EMCY: 00 00 00 00 00 00 00 00".

Applied function list:

[I8123\\_InitMaster](#), [I8123\\_Shutdown](#), [I8123\\_AddNode](#), [I8123\\_RemoveNode](#),  
[I8123\\_NMTChangeState](#), [I8123\\_NMTGuarding](#), [I8123\\_NMTHeartbeat](#),  
[I8123\\_GetEMCYData](#), [I8123\\_GetNMTError](#), [I8123\\_InstallNMTErrISR](#),  
[I8123\\_RemoveNMTErrISR](#), [I8123\\_InstallEMCYISR](#),  
[I8123\\_RemoveEMCYISR](#).

### 4.1.3. PDO\_Parameter

Sometimes, the default PDO configuration can't satisfy users. Users need to change the configuration of the PDO related parameters such as transmission type, PDO ID, event timer, dynamic PDO, and so forth. This demo will demonstrate how to change settings of these PDO parameters and show the configuration result.



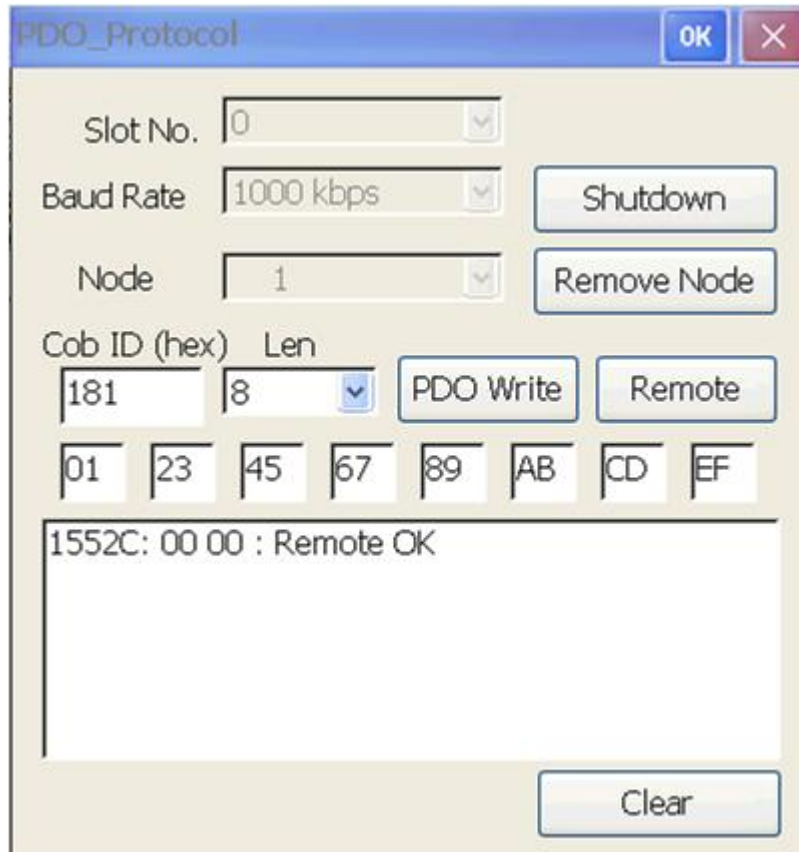
Applied function list:

[I8123\\_InitMaster](#), [I8123\\_Shutdown](#), [I8123\\_AddNode](#), [I8123\\_RemoveNode](#),  
[I8123\\_InstallPDO](#), [I8123\\_RemovePDO](#), [I8123\\_DynamicPDO](#),  
[I8123\\_ChangePDOID](#), [I8123\\_PDOTxType](#), [I8123\\_PDUseEntry](#),  
[I8123\\_PDSEventTimer](#), [I8123\\_GetTxPDOID](#), [I8123\\_GetRxPDOID](#),  
[I8123\\_GetPDOMapInfo](#).

---

#### 4.1.4. PDO\_Protocol

The PDO protocol is the main protocol to control the I/O of the specific slave device in the CANopen network. This demo shows how to read and write data to the slave device with the PDO functions.



Applied function list:

[I8123\\_InitMaster](#), [I8123\\_Shutdown](#), [I8123\\_AddNode](#), [I8123\\_RemoveNode](#),  
[I8123\\_PDOWrite](#), [I8123\\_PDORemote](#), [I8123\\_GetPDOLastData](#)

---

### 4.1.5. Scan\_Node

When users want to know which slave nodes exist on the CANopen network or which slave nodes are under the control of the I-8123W, this demo will be useful.



Applied function list:

[I8123\\_InitMaster](#), [I8123\\_Shutdown](#), [I8123\\_AddNode](#), [I8123\\_RemoveNode](#),  
[I8123\\_SetFunctionTimeout](#), [I8123\\_ScanNode](#), [I8123\\_GetNodeList](#)

---

#### 4.1.6. SDO\_PDO\_ISR

In this demo, it is allowed to configure the I-8123W as a CANopen slave. Users can use another CANopen master to read/write the users' defined object dictionary of the I-8123W by SDO protocol or to get/set the DIO status by PDO protocol when the I-8123W, the I-8053W DI module, and the I-8057W DO module are plugged in the same MCU. If the user has an application like this, this demo may be a good reference.



Applied function list:

[I8123\\_InitMaster](#), [I8123\\_Shutdown](#), [I8123\\_GetMasterReadSDOEvent](#),  
[I8123\\_GetMasterWriteSDOEvent](#), [I8123\\_GetMasterRemotePDOEvent](#),  
[I8123\\_GetMasterRxPDOEvent](#), [I8123\\_ResponseMasterSDO](#),  
[I8123\\_ResponseMasterPDO](#), [I8123\\_InstallPDO\\_List](#),  
[I8123\\_InstallReadSDOISR](#), [I8123\\_InstallWriteSDOISR](#),  
[I8123\\_InstallRxPDOISR](#), [I8123\\_InstallRemotePDOISR](#).

---

### 4.1.7. SDO\_Read

SDO protocol is a kind of the communication functions used to read/write CANopen object dictionary. You can read any object data of the object dictionary through the object address (index and sub-index) by SDO protocol. This demo is a good model to do that.



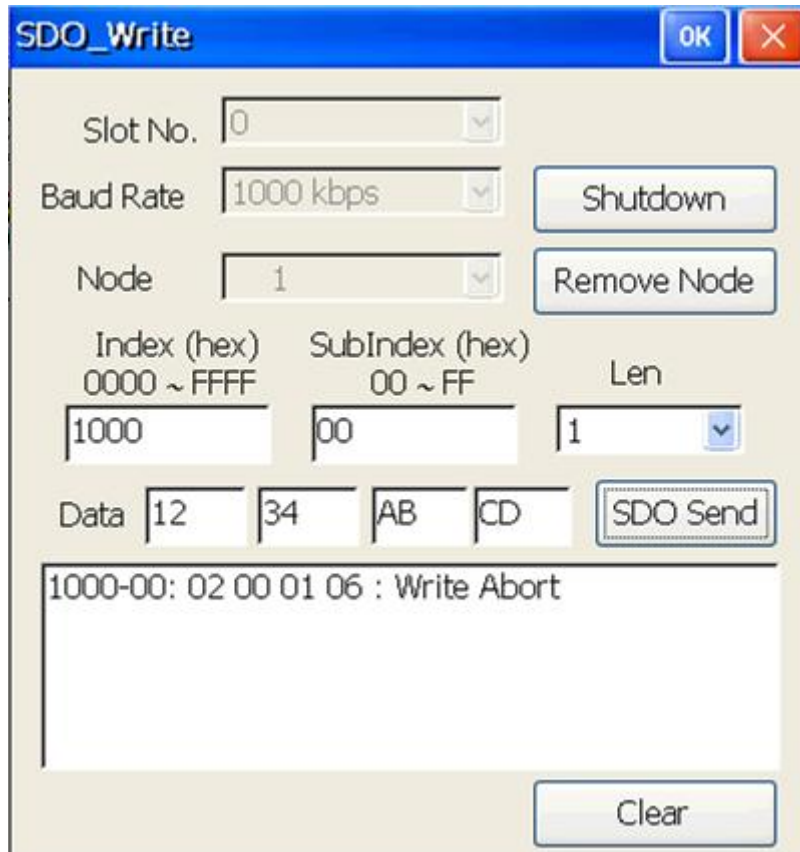
Applied function list:

[I8123\\_InitMaster](#), [I8123\\_Shutdown](#), [I8123\\_AddNode](#), [I8123\\_RemoveNode](#),  
[I8123\\_SDOReadData](#).

---

### 4.1.8. SDO\_Write

SDO protocol is a kind of the communication functions used to read/write CANopen object dictionary. You can write any data to the specific object of the object dictionary through the object address (index and sub-index) by SDO protocol. This demo is a good model to do that.



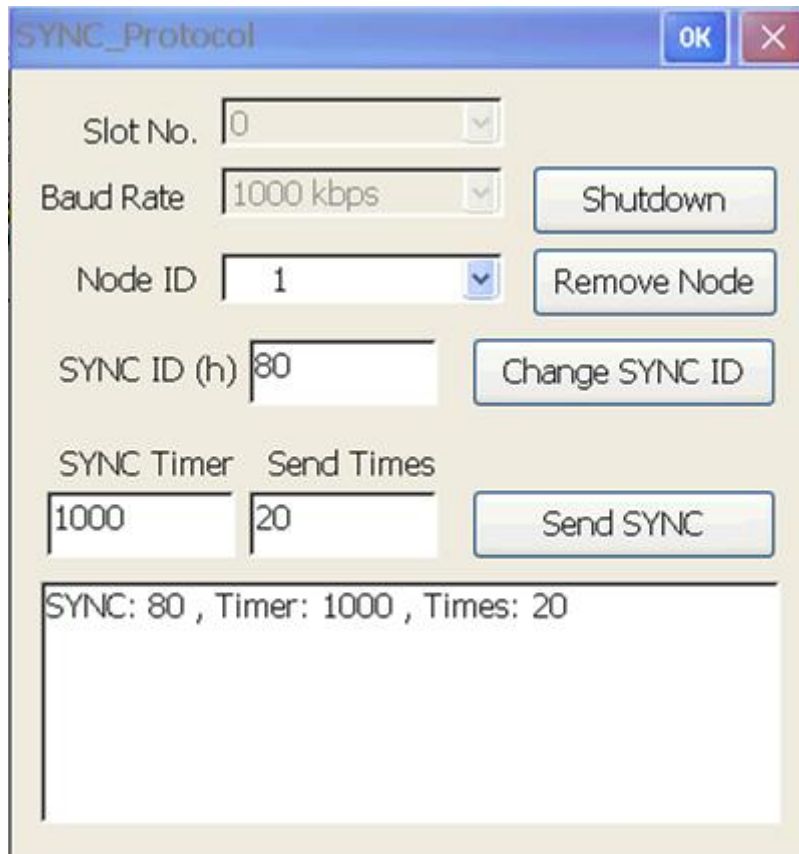
Applied function list:

[I8123\\_InitMaster](#), [I8123\\_Shutdown](#), [I8123\\_AddNode](#), [I8123\\_RemoveNode](#),  
[I8123\\_SDOWriteData](#).

---

### 4.1.9. SYNC\_Protocol

SYNC protocol is a synchronous function of the PDO communication. It is always used with the transmission type of the PDO communication. In this demo, users can know how to use the SYNC related functions.



Applied function list:

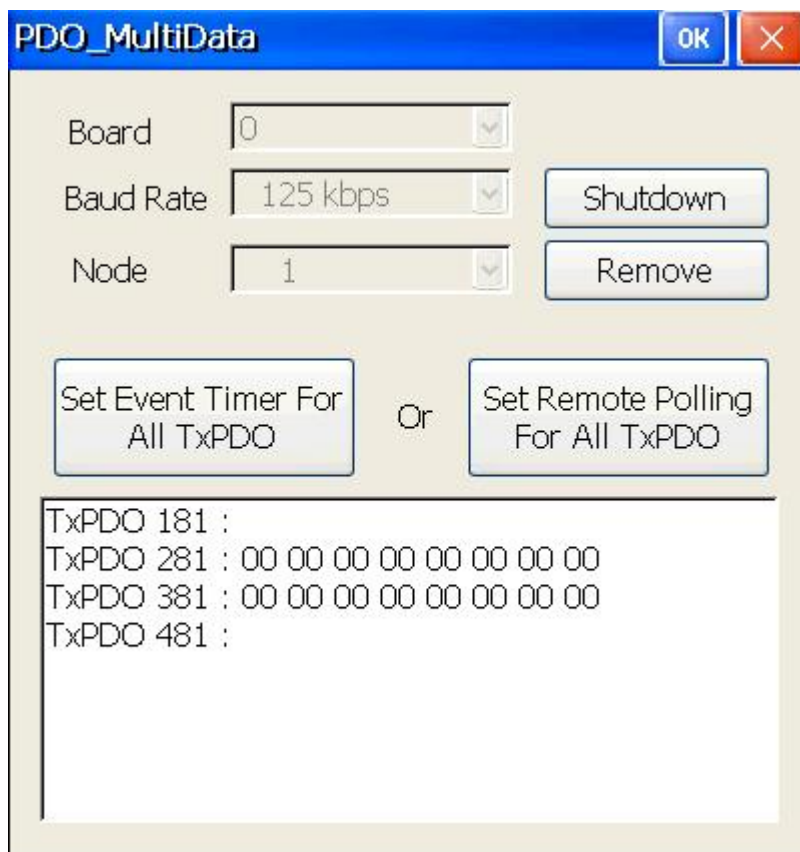
[I8123\\_InitMaster](#), [I8123\\_Shutdown](#), [I8123\\_AddNode](#), [I8123\\_RemoveNode](#),  
[I8123\\_ChangeSYNCID](#), [I8123\\_GetSYNCID](#), [I8123\\_SendSYNCMsg](#),  
[I8123\\_GetCyclicSYNCInfo](#).



---

#### 4.1.10. PDO\_MultiData

Sometimes, users want to poll several PDO objects data at the same time for increasing the performance. But it is slower than sending the Remote PDO to poll each PDO data one by one. So users can set event timer or remote list for these PDO. When the PDO data are polled by the I-8123W or are replied from slave automatically, then use the I8123\_GetMultiPDOData function to obtain these PDO data from the buffer at the same time.



Applied function list:

[I8123\\_InitMaster](#), [I8123\\_Shutdown](#), [I8123\\_AddNode](#), [I8123\\_RemoveNode](#),  
[I8123\\_GetTxPDOID](#), [I8123\\_SetPDORemotePolling](#), [I8123\\_PDOWriteEventTimer](#),  
[I8123\\_GetMultiPDOData](#)

## 5. Update Firmware

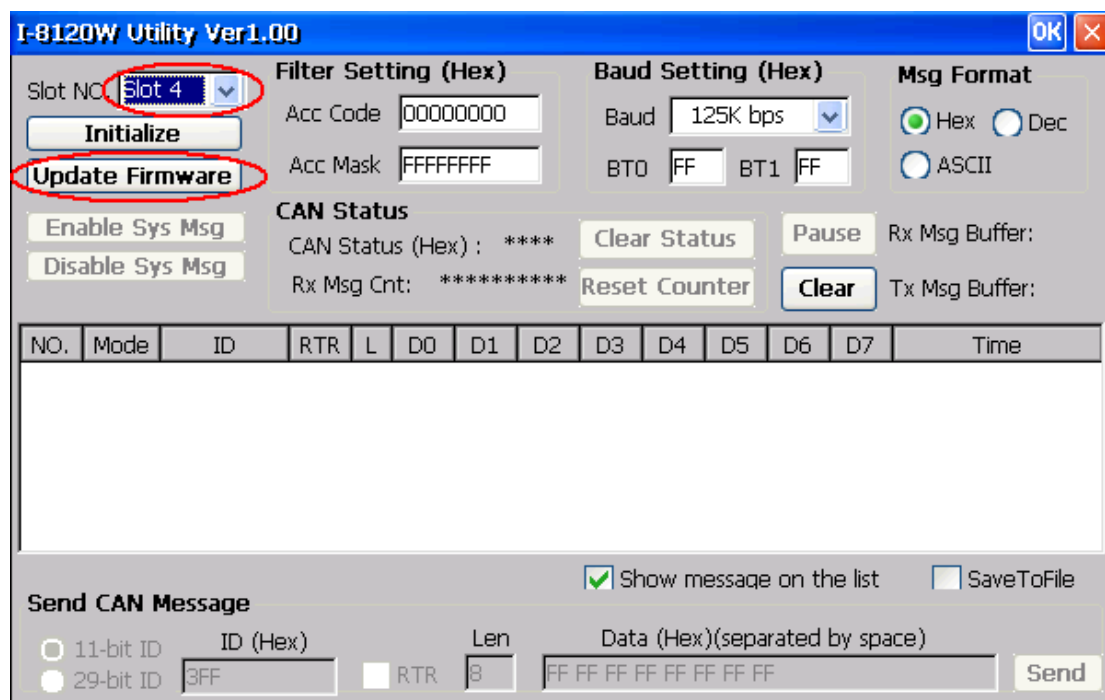
If users want to update the firmware of the I-8123W, use the I-8120W utility to do it.

\* Where to find the I-8120W utility?

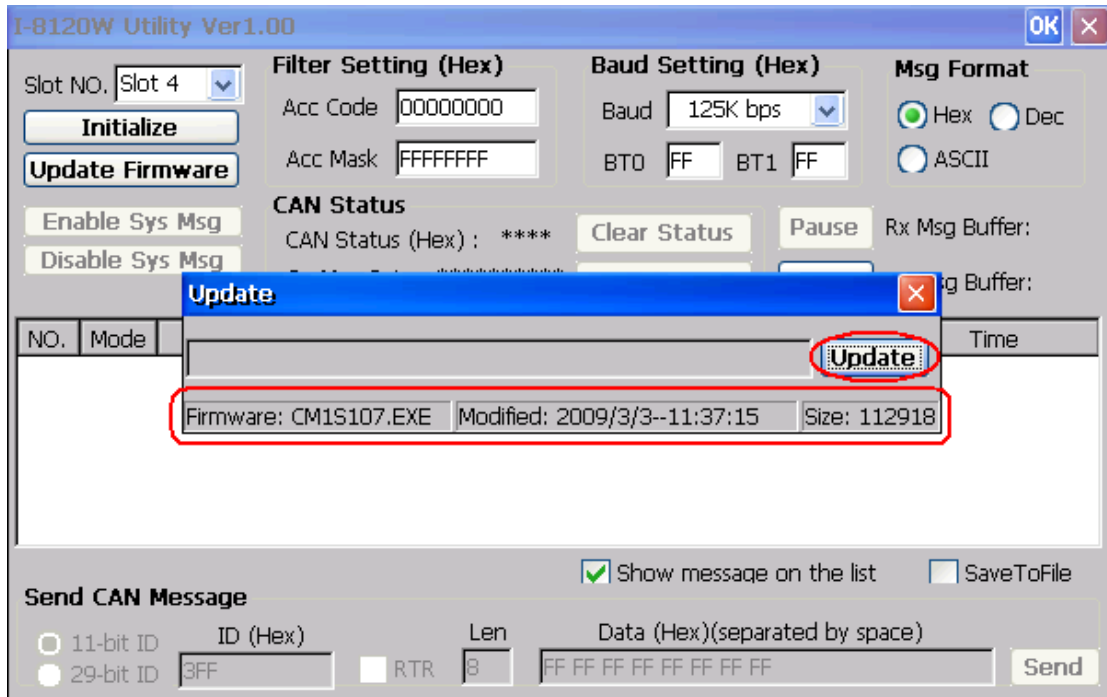
The path of field bus CD:  
[fieldbus\\_cd://can/slotmodule/I\\_8120w/tools/wince5/](ftp://ftp.icpdas.com/pub/cd/fieldbus_cd/can/slotmodule/i_8120w/tools/wince5/)

The address of the web site:  
[http://ftp.icpdas.com/pub/cd/fieldbus\\_cd/can/slotmodule/i\\_8120w/tools/wince5/](http://ftp.icpdas.com/pub/cd/fieldbus_cd/can/slotmodule/i_8120w/tools/wince5/)

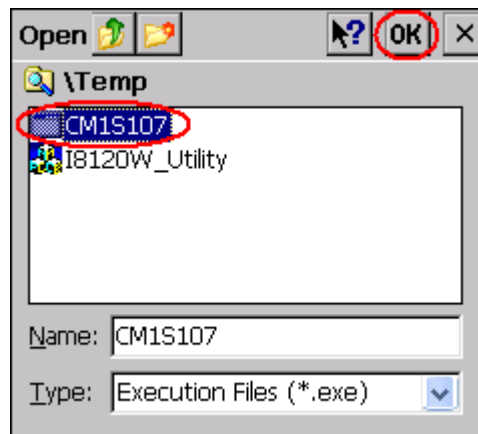
The following steps describe the method about how to update the I-8123W firmware by I-8120W utility. After copy the I-8120W utility on to MCU, choose the proper No. of the slot which the I-8123W has been plugged in. Then click the “Update Firmware” button.



The download dialog will be popped up. Users can see the firmware name, modified date, and file size of the firmware stored in the I-8123W. Then click “Update” button to continue.



In the browser, select the file which you want to download. Then click “OK” button to go on the download procedure. Take a note that when users click “OK” button, the download procedure will be started. The original firmware stored in the I-8123W will be killed.



When the procedure is finished, users can see the present firmware information of the I-8123W. Click “OK” button to close the download dialog. Afterwards, the new firmware will be run automatically.

