I-8123W-CPS CANopen Slave Module

User Manual

Tables of Content

1. General Information 3
- **1.1. CANopen Introduction 3**
- **1.2. CANopen Applications 4**
- **1.3. I-8123W-CPS Library Characteristics 5**

2. Hardware Specification 8
- **2.1. Hardware Structure 8**
- **2.2. Wire Connection 9**
- **2.3. Power LED 11**
- **2.4. Tx/Rx LED 11**
- **2.5. ERR LED 11**

3. I8123W-CPS Function Library 12
- **3.1. Function List 12**
- **3.2. Function Return Code Troubleshooting 13**
- **3.3. CANopen Slave Library Application Flowchart 14**
- **3.4. Function Description 15**
  - 3.4.1 I8123CPS_GetFWVer 15
  - 3.4.2 I8123CPS_GetLib 15
  - 3.4.3 I8123CPS_CANopenInit 16
  - 3.4.4 I8123CPS_IOInit 16
  - 3.4.5 I8123CPS_InstallUserISR 17
  - 3.4.6 I8123CPS_WriteDI 17
  - 3.4.7 I8123CPS_WriteAI 18
  - 3.4.8 I8123CPS_ReadDO 18
  - 3.4.9 I8123CPS_ReadAO 19
  - 3.4.10 I8123CPS_CheckDOAOCmd 19
  - 3.4.11 I8123CPS_ReadAllDOAO 20

4. Demo Programs 21
- **4.1. Brief of the Demo Programs 21**

5. Update Firmware 26

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.

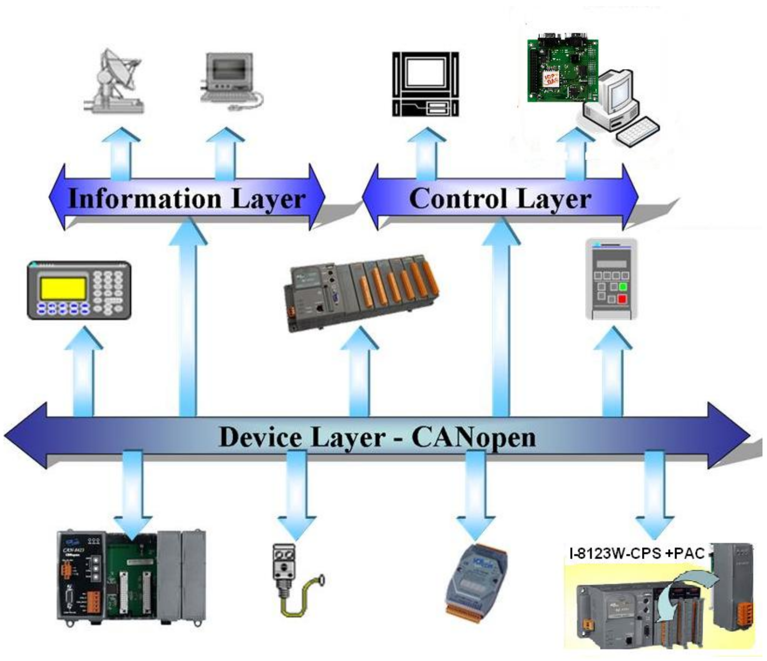


Figure 1.1 Example of the CANopen network

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-8123W-CPS Library Characteristics

In order to use the I-8123W-CPS module, we provide I8123WCPS library for the ViewPAC、WinPAC、XPAC series main control unit, and user can use it establish CANopen communication network rapidly. Most of the CANopen communication protocols, such as PDO, SDO and NMT, would be handled by the library automatically. Therefore, it is helpful to reduce the complexity of developing a CANopen slave interface, and let users ignore the detail CANopen protocol technology. This library mainly supports connection sets of master-slave architecture. The following figure describes the general application architecture of I-8123W-CPS.

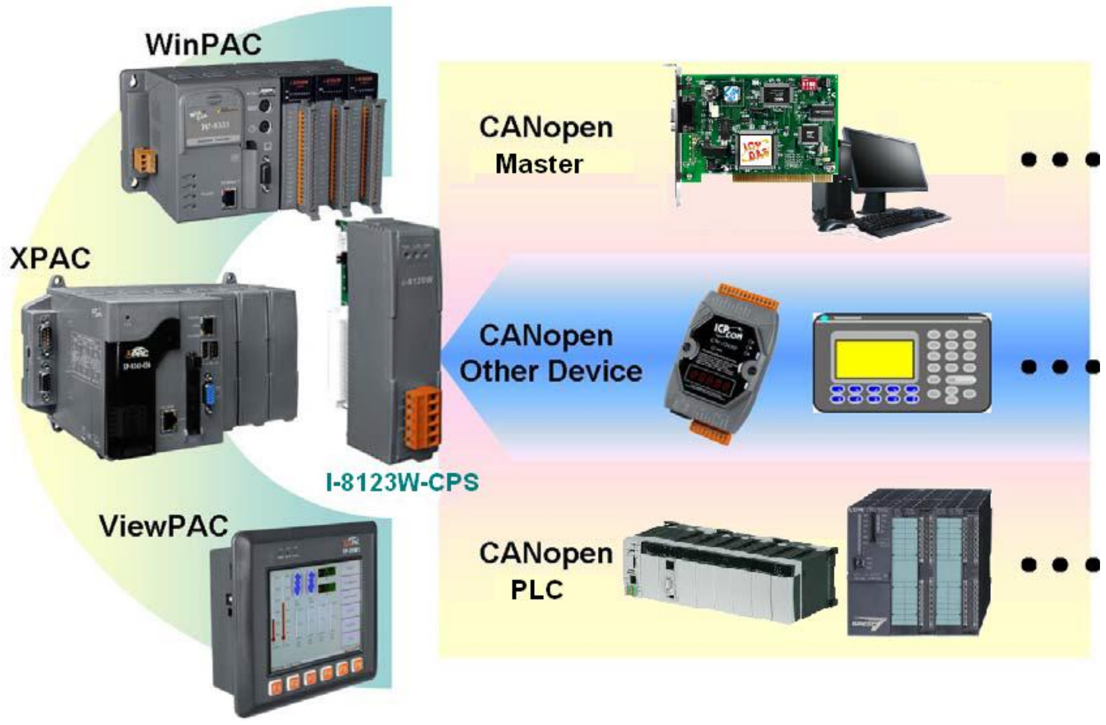


Figure 1.2 Example of application architecture

The I-8123W-CPS follows the CANopen specification CiA-301 V4.02, and supports the several CANopen features. The CANopen communication general concept is shown as Figure 1.3.

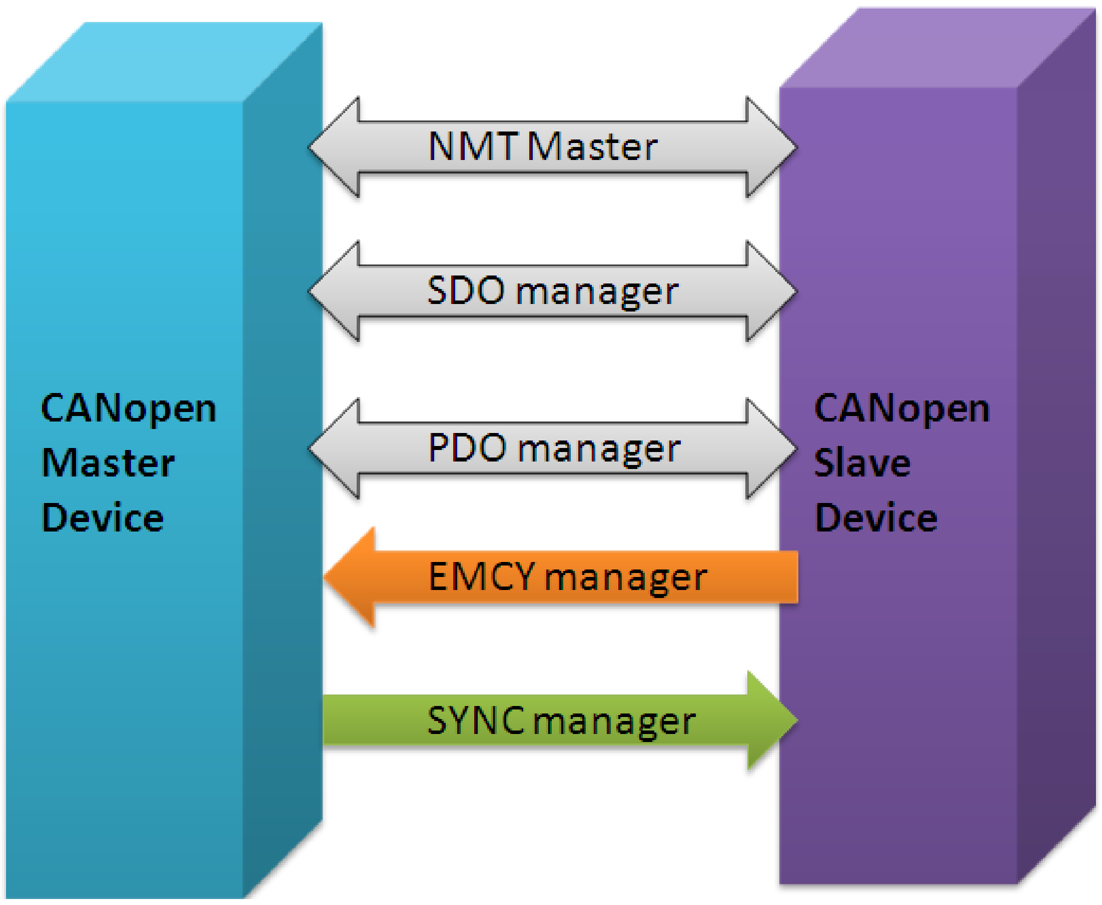


Figure 1.3 CANopen communication general concept

- **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 0.1ms 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

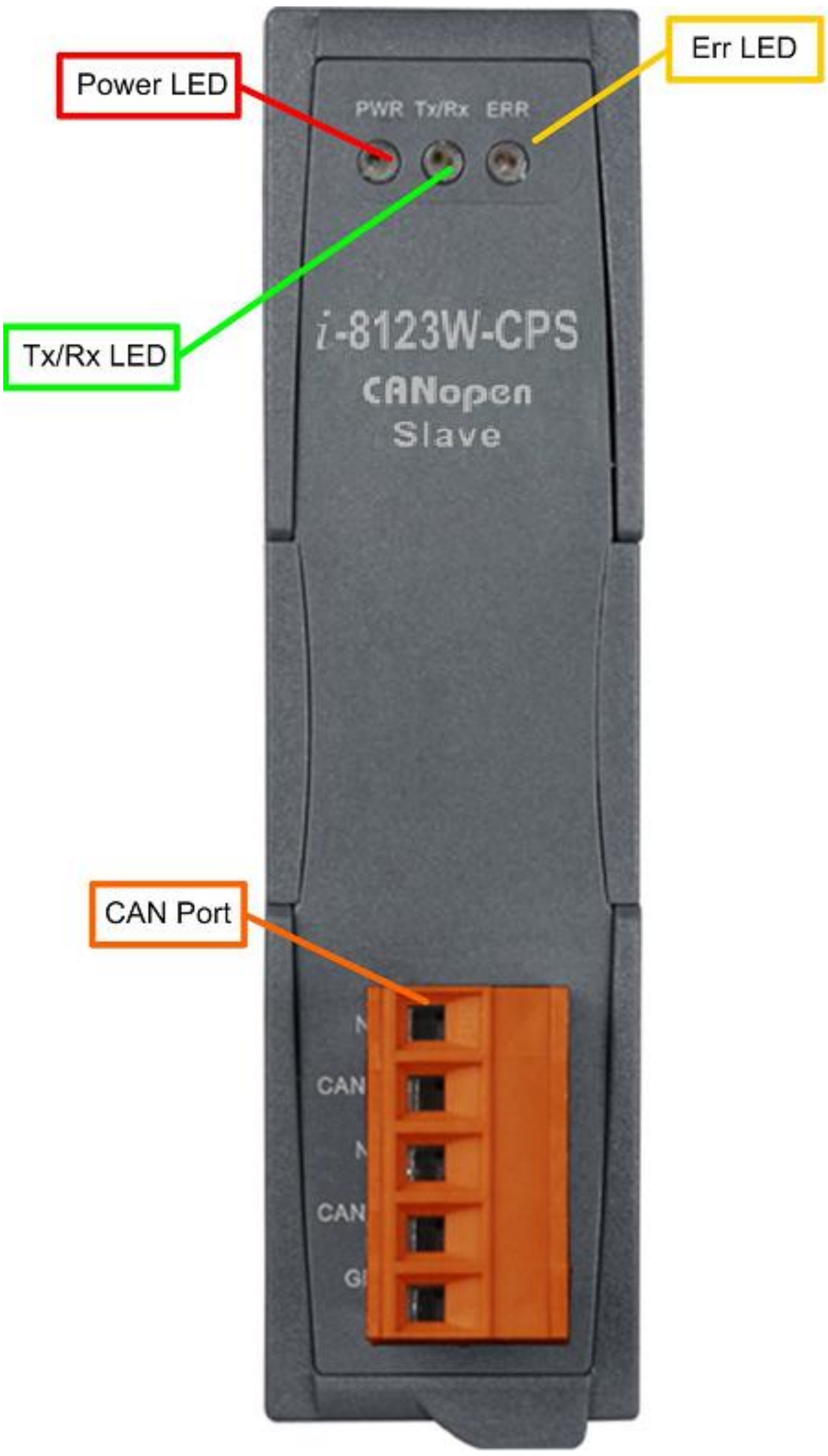
- CPU:80186, 80 MHz
- CAN controller: NXP SJA1000T with 16 MHz clock
- CAN transceiver: NXP 82C250
- Power LED, Tx/Rx LED, Error LED
- Switch for 120Ω terminal resistor of CAN bus
- CAN bus interface: Follow ISO 11898-2, 5-pin screw terminal
- 2500 Vrms photo-coupled isolation on CAN side
- 1000 Vdc voltage protection
- Power Consumption: 2W
- Operating Temperature: -25℃ ~ +75 ℃
- Storage Temperature: -30℃ ~ +80 ℃
- Humidity: 10 ~ 90% non-condensing.

Features

- One CAN communication port.
- Support PAC series MCU (main control unit) with Windows CE5, CE6
- Library provides eVC++, C#. Net2005, and VB.Net2005 developments
- Three indication LEDs (Pwr, Tx/Rx and Err LEDs).
- Support 8 kinds baud: 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.
- Support Node Guarding protocol and Heartbeat protocol.

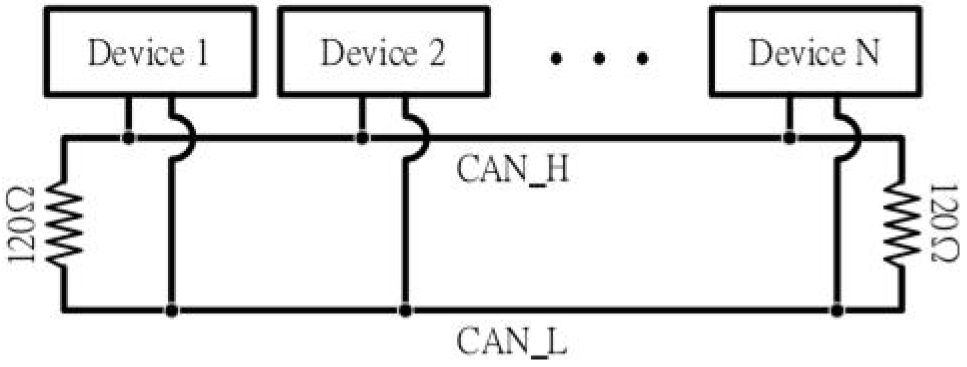
2. *Hardware Specification*

2.1. Hardware Structure



2.2. Wire Connection

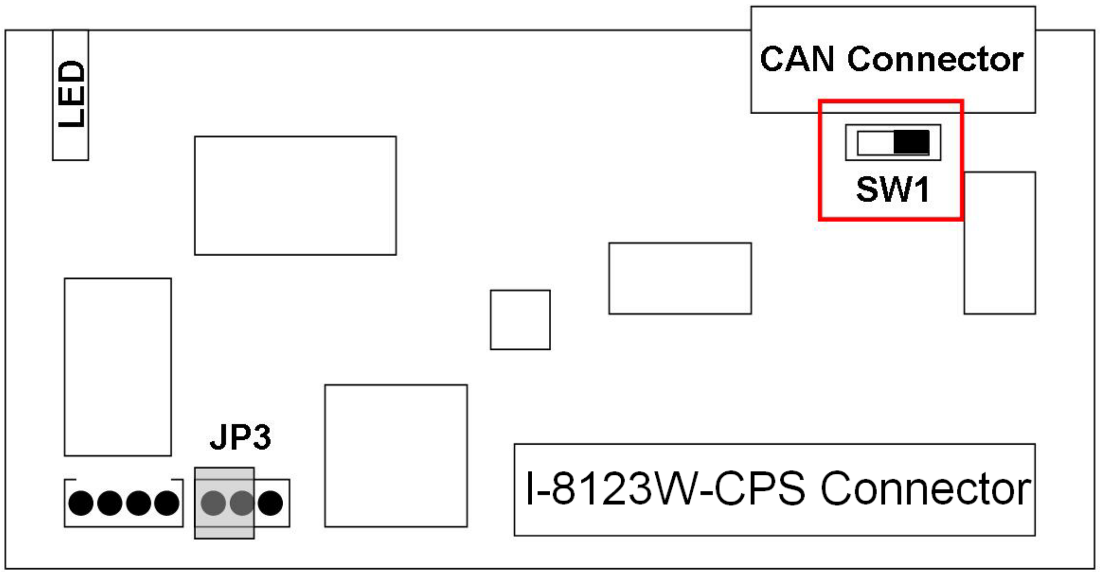
In order to minimize the reflection effects on the CAN bus line, the CAN bus line has to be terminated at both ends by two terminator resistors as in the following figure. According to the ISO-11898-2 specification, each terminator resistor is 120Ω (or between 108Ω~132Ω). The length related resistance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



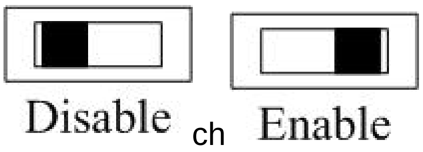
Moreover, to minimize the voltage drop over long distance, the terminator resistor should be higher than the value defined in the ISO-11898-2. The following table can be used as a good reference.

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

In the I-8123W-CPS, the 120Ω terminator resistor is supplied. The SW1of the I-8123W-CPS is for the terminator resistor. Its location is shown in the following figure.

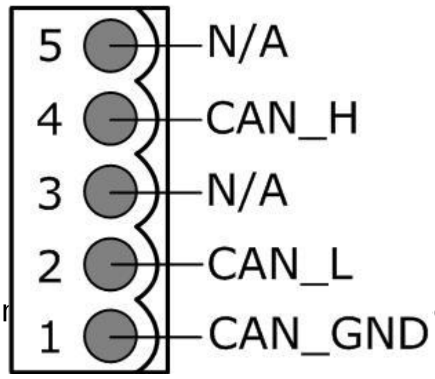


The following connection statuses are presented for the condition if the terminator resistor is enabled or disabled.



ch

Pin assignr



ctor

Pin No.	Signal	Description
1	CAN_GND	Ground
2	CAN_L	CAN_L bus line (dominant low)
3	N/A	No use
4	CAN_H	CAN_H bus line (dominant high)
5	N/A	No use

2.3. Power LED

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

2.4. Tx/Rx LED

Each I-8123W-CPS provides Tx/Rx LED to check the situations of the CAN messages transmission and reception. If the I-8123W-CPS is transmitting or receiving a CAN message, the Tx/Rx LED will blink. If the bus loading of the I-8123W-CPS 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 error due to missing any CAN message.

3. I8123W-CPS Function Library

3.1.Function List

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

Function Name	Description
I8123CPS_GetFWVer	Get version of the I-8123W-CPS firmware
I8123CPS_GetLib	Get version of the I8123CPS.dll library
I8123CPS_CANopenInit	Set baud rate of the I-8123W-CPS
I8123CPS_IOInit	Set Node ID, DI , AI, DO and AO channels of the I-8123W-CPS
I8123CPS_InstallUserISR	Apply ISR to handle the interrupts from the firmware of I-8123W-CPS
I8123CPS_WriteDI	Write DI data into the I-8123W-CPS.
I8123CPS_WriteAI	Write AI data into the I-8123W-CPS.
I8123CPS_ReadDO	Read DO data from the I-8123W-CPS.
I8123CPS_ReadAO	Read AO data from the I-8123W-CPS.
I8123CPS_CheckDOAOCmd	Check what kind of the interrupt is produced from the firmware of the I-8123W-CPS
I8123CPS_ReadAllDOAO	Read all DO and AO data from I-8123W-CPS

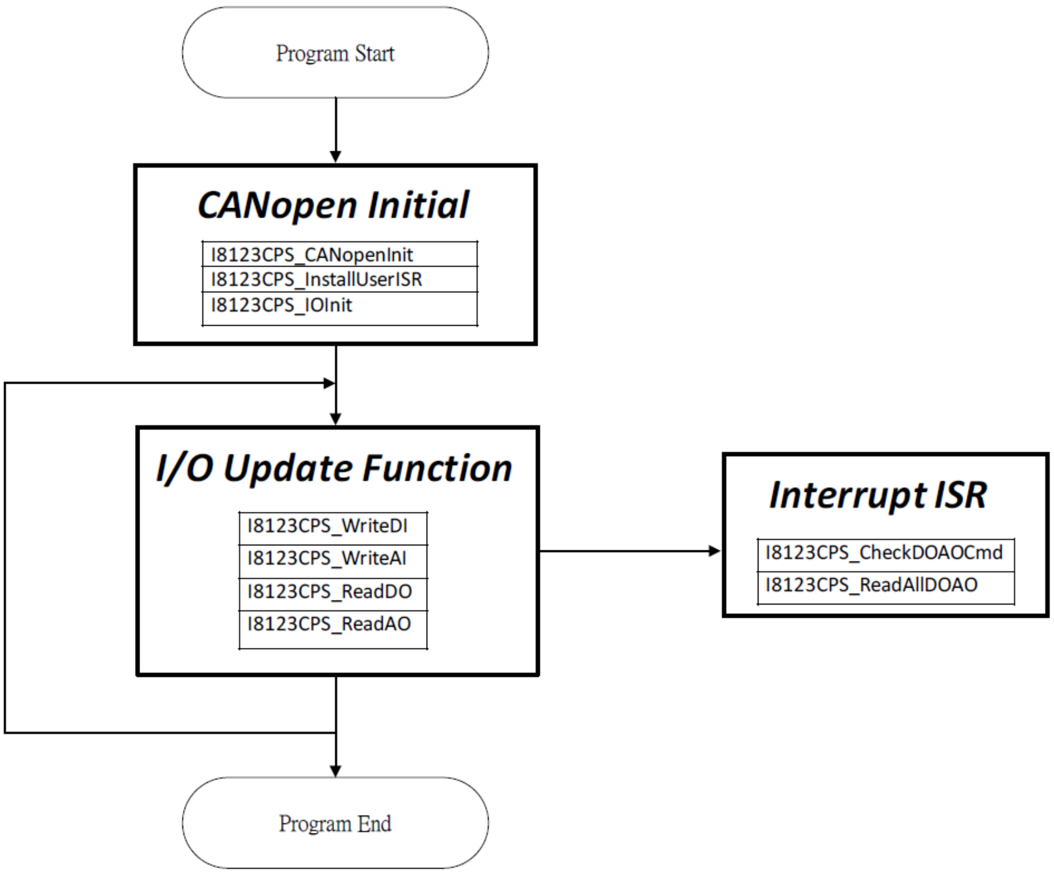
3.2.Function Return Code Troubleshooting

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

Return Code	Error ID	Description
0	I8123CPS_NoError	OK
3	I8123CPS_SlotNumberError	1. Set the SlotNo parameter of function to match the module. 2. Unplug the I-8123W-CPS, and plug it again and turn on your PC until find it in the list of hardware management of Windows.
7	I8123CPS_InitError	1. Retry the function again. 2. Call the function I8123CPS_CANopenInit() and configure I-8123W-CPS again.
23	I8123CPS_TimeOut	1. Wait for a while and call the function again. 2. Call the function I8123CPS_CANopenInit() and configure I-8123W-CPS again.
26	I8123CPS_NoDpramCmd	Wait for a while and call the function again.
41	I8123CPS_SlotNotInit	Call the function I8123CPS_CANopenInit() at the start of application.

3.3. CANopen Slave Library Application Flowchart

In this section, it describes that the operation procedure about how to use the CANopen Slave Library to build user application.



3.4.Function Description

3.4.1 I8123CPS_GetFWVer

- **Description:**

  This function is used to obtain the version information of the I-8123W-CPS firmware.

- **Syntax:**

  int I8123CPS_GetFWVer(**BYTE** SlotNo)

- **Parameter:**

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

- **Return:**

  Firmware version information.

3.4.2 I8123CPS_GetLib

- **Description:**

  This function is used to obtain the version information of the I8123CPS.dll library.

- **Syntax:**

  int I8123CPS_GetLib(**void**)

- **Parameter:**

  None

- **Return:**

  DLL version information.

3.4.3 I8123CPS_CANopenInit

- **Description:**

  This function is used to set the baud rate of the I-8123W-CPS.

- **Syntax:**

  int I8123CPS_CANopenInit(**BYTE** SlotNo, **BYTE** BaudRate)

- **Parameter:**

  SlotNo: [input] I-8123W-CPS slot number (0~7).
  BaudRate: [input] The baudrate of the I-8123W-CPS

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

-

3.4.4 I8123CPS_IOInit

- **Description:**

  This function is used to set the Node ID, DI channels, AI channels, DO channels and AO channels of the I-8123W-CPS.

- **Syntax:**

  int I8123CPS_IOInit(**BYTE** SlotNo**, BYTE** NodeNo**, WORD** DI_Ch**, WORD** AI_Ch**, WORD** DO_Ch**, WORD** AO_Ch)

- **Parameter:**

  SlotNo: [input] I-8123W-CPS slot No. (0~7).
  NodeNo: [input] The CANopen node ID is used by I-8123W-CPS.
  DI_Ch: [input] Sum of DI channels in the PAC with I-8123W-CPS.
  AI_Ch: [input] Sum of AI channels in the PAC with I-8123W-CPS.
  DO_Ch: [input] Sum of DO channels in the PAC with I-8123W-CPS.
  AO_Ch: [input] Sum of AO channels in the PAC with I-8123W-CPS.

3.4.5 I8123CPS_InstallUserISR

- **Description:**

  Using this function can allow users to apply ISR (Interrupt Service Routine).User must apply the ISR to handle the interrupts from the firmware of I-8123W-CPS.

- **Syntax:**

  int I8123CPS_InstallUserISR(**BYTE** SlotNo **,**
  void(CALLBACK *UserISR**)(void)**)

- **Parameter:**

  SlotNo: [input] I-8123W-CPS slot No. (0~7).
  (CALLBACK *UserISR)(void)): [input] The pointer which points a function with format "void CALLBACK XXX(void)". The XXX is the function name of users' ISR. After calling this function, users can use the function I8123CPS_CheckDOAOCmd in the users' ISR to check what kind of the interrupt is produced from the firmware of the I-8123W-CPS.

3.4.6 I8123CPS_WriteDI

- **Description:**

  Write DI data into specified address of DPRAM of the I-8123W-CPS.The DPRAM space can be applied is from 0 to (maximum byte of DI channels)-1. For example, the DI channels in the PAC are 32 then the DPRAM space is from 0 to 3.

- **Syntax:**

  int I8123CPS_WriteDI(**BYTE** SlotNo, **BYTE** *DI_Data,
  WORD DataNum)

- **Parameter:**

  SlotNo: [input] I-8123W-CPS slot No. (0~7).
  ***DI_Data**: [input] The start address of a byte array written to the DPRAM of I-8123W-CPS.
  DataNum: [input] The byte number of an data array written to the DPRAM of I-8123W-CPS.

3.4.7 I8123CPS_WriteAI

- **Description:**

  Write AI data into specified address of DPRAM of the I-8123W-CPS.The DPRAM space can be applied is from 0 to (AI channels *2) -1. For example, the AI channels in the PAC are 8 then the DPRAM space is from 0 to 15.

- **Syntax:**

  int I8123CPS_WriteAI(**BYTE** SlotNo**, BYTE** *AI_Data**,**
  WORD DataNum)

- **Parameter:**

  SlotNo: [input] I-8123W-CPS slot No. (0~7).
  ***AI_Data**: [input] The start address of a byte array written to the DPRAM of I-8123W-CPS.
  DataNum: [input] The byte number of an data array written to the DPRAM of I-8123W-CPS.

3.4.8 I8123CPS_ReadDO

- **Description:**

  Read DO data from specified address of DPRAM of the I-8123W-CPS.The DPRAM space can be applied is from 0 to (maximum byte of DI channels)-1. For example, the DI channels in the PAC are 32 then the DPRAM space is from 0 to 3.

- **Syntax:**

  int I8123CPS_ReadDO(**BYTE** SlotNo**, WORD** DO_Address**,**
  BYTE *DO_Data**, WORD** DataNum)

- **Parameter:**

  SlotNo: [input] I-8123W-CPS slot No. (0~7).
  ***DO_Address**: [input] The start address to read from the DPRAM of I-8123W-CPS.
  ***DO_Data**: [output] A DO byte array read from the DPRAM of I-8123W-CPS.
  DataNum: [input] The byte number of an data array read from the DPRAM of I-8123W-CPS

3.4.9 I8123CPS_ReadAO

- **Description:**

  Read AO data from specified address of DPRAM of the I-8123W-CPS.The DPRAM space can be applied is from 0 to (AI channels *2) -1. For example, the AI channels in the PAC are 8 then the DPRAM space is from 0 to 15.

- **Syntax:**

  int I8123CPS_ReadAO(**BYTE** SlotNo**, WORD** AO_Address**,**
  BYTE *AO_Data**, WORD** DataNum)

- **Parameter:**

  SlotNo: [input] I-8123W-CPS slot No. (0~7).
  ***AO_Address**: [input] The start address to read from the DPRAM of I-8123W-CPS.
  ***AO_Data**: [output] An AO byte array read from the DPRAM of I-8123W-CPS.
  DataNum: [input] The byte number of a data array read from the DPRAM of I-8123W-CPS

3.4.10 I8123CPS_CheckDOAOCmd

- **Description:**

  This function must be used in the users' ISR to check what kind of the interrupt is produced from the firmware of the I-8123W-CPS.

- **Syntax:**

  int I8123CPS_CheckDOAOCmd(**BYTE SlotNo, BYTE** *DO_Address**,**
  BYTE *DO_DataNum**, BYTE** *AO_Address**, BYTE** *AO_DataNum)

- **Parameter:**

  SlotNo: [input] I-8120W slot No. (0~7).
  ***DO_Address**: [output] The start address of DO data to read from the DPRAM of I-8123W-CPS.
  ***DO_DataNum**: [output] The byte number of a DO data array read from the DPRAM of I-8123W-CPS.
  ***AO_Address**: [output] The start address of AO data to read from the DPRAM of I-8123W-CPS.
  ***AO_DataNum**: [output] The byte number of a AO data array read from the DPRAM of I-8123W-CPS.

3.4.11 I8123CPS_ReadAllDOAO

- **Description:**

  This function is used to read DO and AO data. It is usually called after I8123CPS_CheckDOAOCmd to update the DO and AO data for PAC.

- **Syntax:**

  int I8123CPS_ReadAllDOAO(**BYTE** SlotNo**, BYTE** *DO_Data**,**
  WORD DO_DataNum**, BYTE** *AO_Data**, WORD** AO_DataNum)

- **Parameter:**

  SlotNo: [input] I-8123W-CPS slot No. (0~7).
  ***DO_Data**: [output] An DO byte array read from the DPRAM of I-8123W-CPS.
  DO_DataNum: [input] The byte number of a DO data array read from the DPRAM of I-8123W-CPS.
  ***AO_Data**: [output] An AO byte array read from the DPRAM of I-8123W-CPS.
  AO_DataNum: [input] The byte number of a AO data array read from the DPRAM of I-8123W-CPS.

- **Return:**

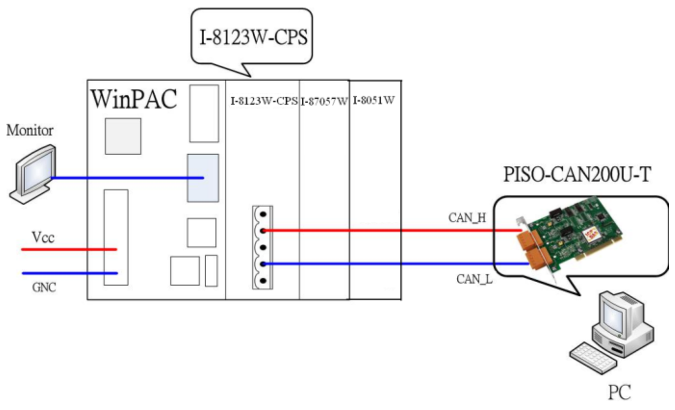
4. Demo Programs

4.1. Brief of the Demo Programs

In order to use the I-8123W-CPS more easily, we provide some useful and easy-to-use functions in I8123CPS library. There are three function libraries for different compiler, such as eVC++, VB.net and C#. Users can use these functions to control the I-8123W-CPS by using the functions. The following shows one of the demos.

Before following the steps below, users need to prepare some hardware, an I-8123W-CPS, an I8K DI module, an I87K DO module and a WinPAC or ViewPAC series MCU.

Step 1: Put the I-8123W-CPS in slot 0 of WinPAC series MCU and connect the CAN port of the I-8123W-CPS with the CAN port of a CANopen master device as following figure. Then power on these hardware.



Step 3: Download the I8120.dll and I8123CPS.dll into the same folder of the WinPAC. Then select a demo execute file and download it into the same folder. **Take a note that if you select a C#.net demo or a VB.net demo, the I8123CPS_Net.dll also needs to be downloaded into the same folder.** (About how to download file to WinPAC, please refer to the WinPAC user manual) The paths for these files are as follows:

CD path:

I8120.dll:

CAN/SlotModule/I_8120W/Demos/WinCE5_Lib/Ver_200/ or CANopen/Master/I-8123W/Drivers/CE5/

I8123CPS.dll:

CANopen/Slave/I-8123WCPS/Drivers/CE5/

Demos:

CANopen/Slave/I-8123WCPS/Demos/CE5/

FTP path:

I8120.dll:

ftp://ftp.icpdas.com/pub/cd/fieldbus_cd/can/slotmodule/i_8120w/demos/wince5_lib/ver_200/

or

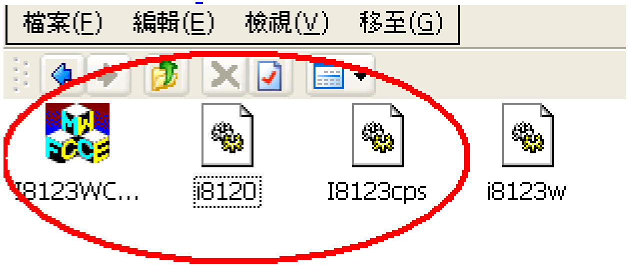
ftp://ftp.icpdas.com/pub/cd/fieldbus_cd/canopen/master/i-8123w/drivers/ce5/

I8123CPS.dll:

ftp://ftp.icpdas.com/pub/cd/fieldbus_cd/canopen/slave/i-8123wcps/drivers/ce5/

Demos:

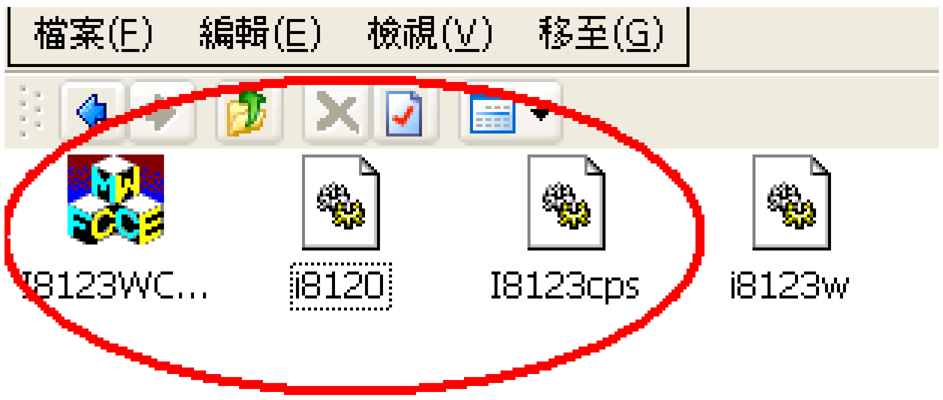
ftp://ftp.icpdas.com/pub/cd/fieldbus_cd/canopen/slave/i-8123wcps/demos/ce5/



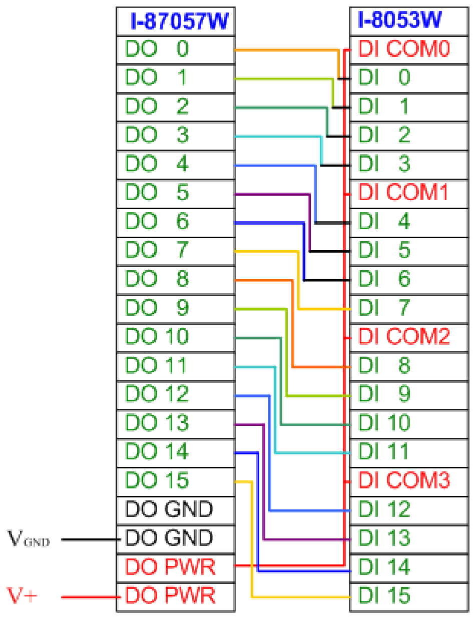
winpac_8000/download_documents.htm

23WCPSDIDODemo demo and

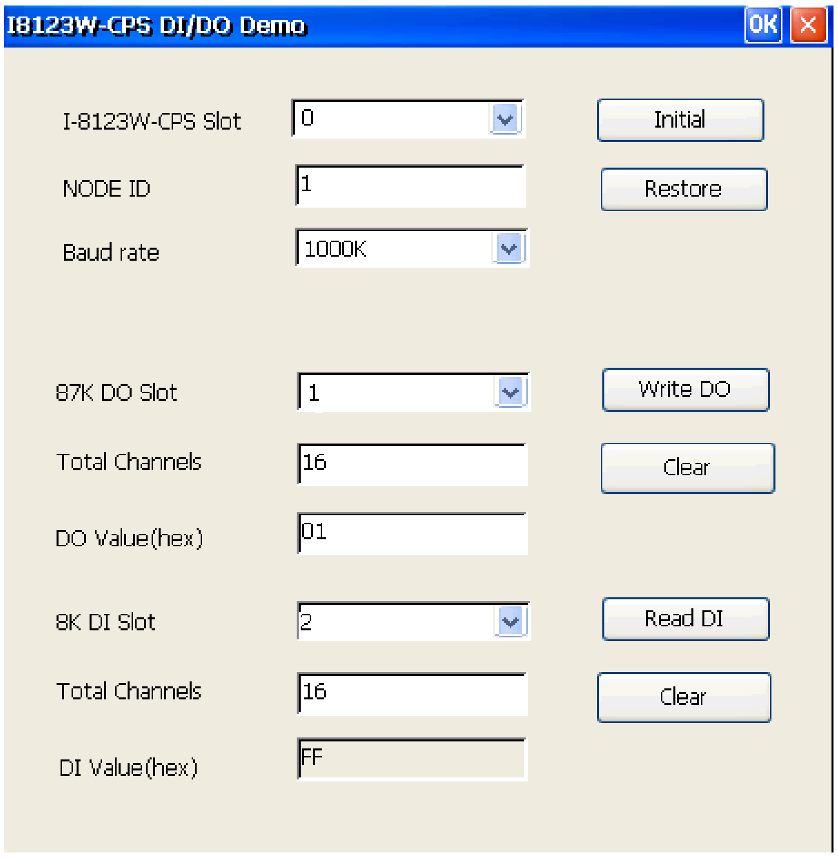
nPAC.



Step 5: Assume I-87057W and I-8053W are separately inserted in slot 1, 2. Wire connection is as below.



Step 6: Run the demo on the WinPAC. The following dialog is popped up.


I8123W-CPS DI/DO Demo

I-8123W-CPS Slot: 0

NODE ID: 1

Baud rate: 1000K

87K DO Slot: 1

Total Channels: 16

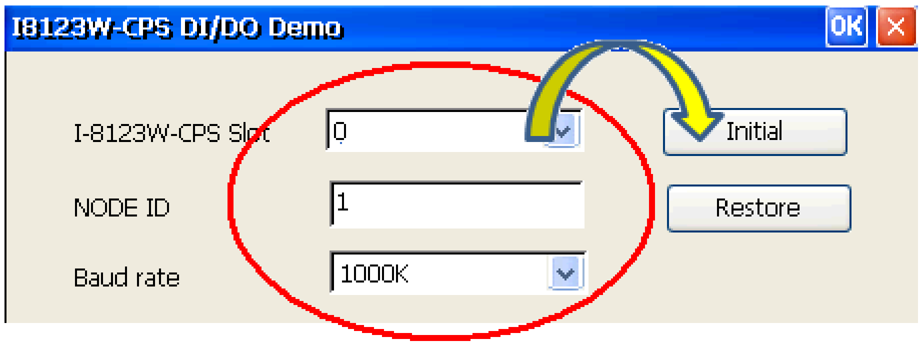
DO Value(hex): 01

8K DI Slot: 2

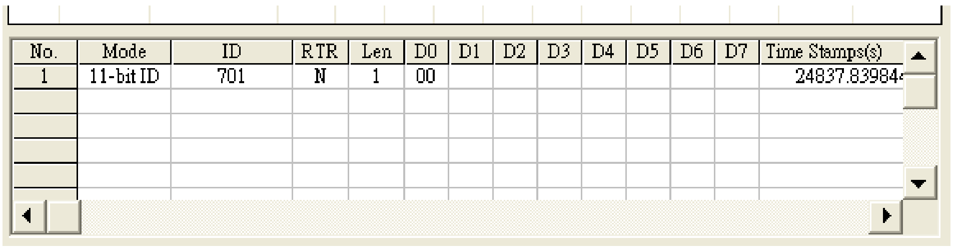
Total Channels: 16

DI Value(hex): FF

Step 7: Assume the I-8123W-CPS is plugged in slot 0 of the WinPAC, the Node ID of I-8123W-CPS is 0 and the baud of the CANopen network is 1000 kbps, set the "Slot No.",” NODE ID” and "Baud Rate" as following figure. Then click "Initial " to initialize the I-8123W-CPS. The CAN utility can receive boot up message as below.

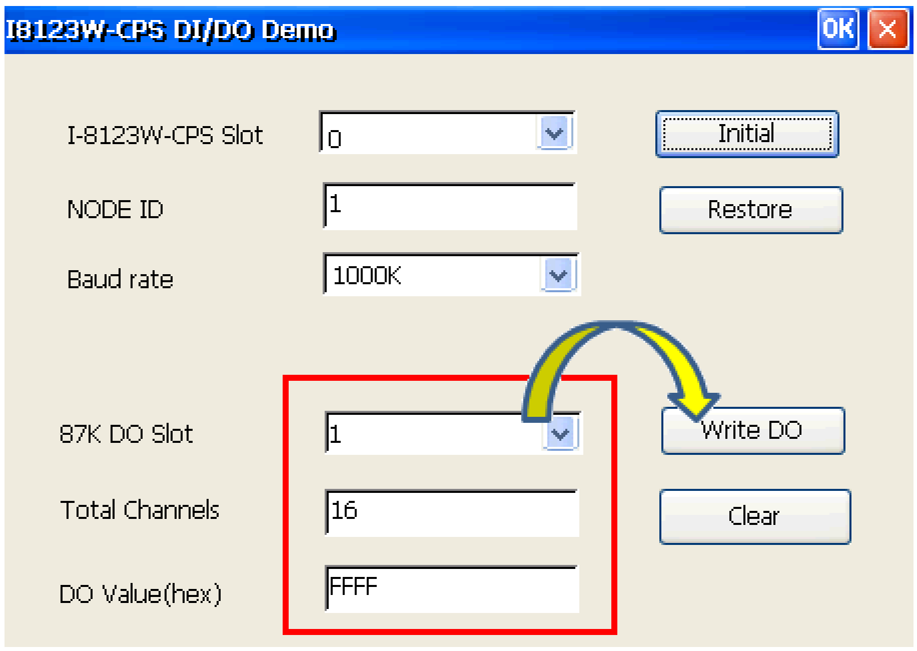


I-8123W-CPS Slot: 0
NODE ID: 1
Baud rate: 1000K



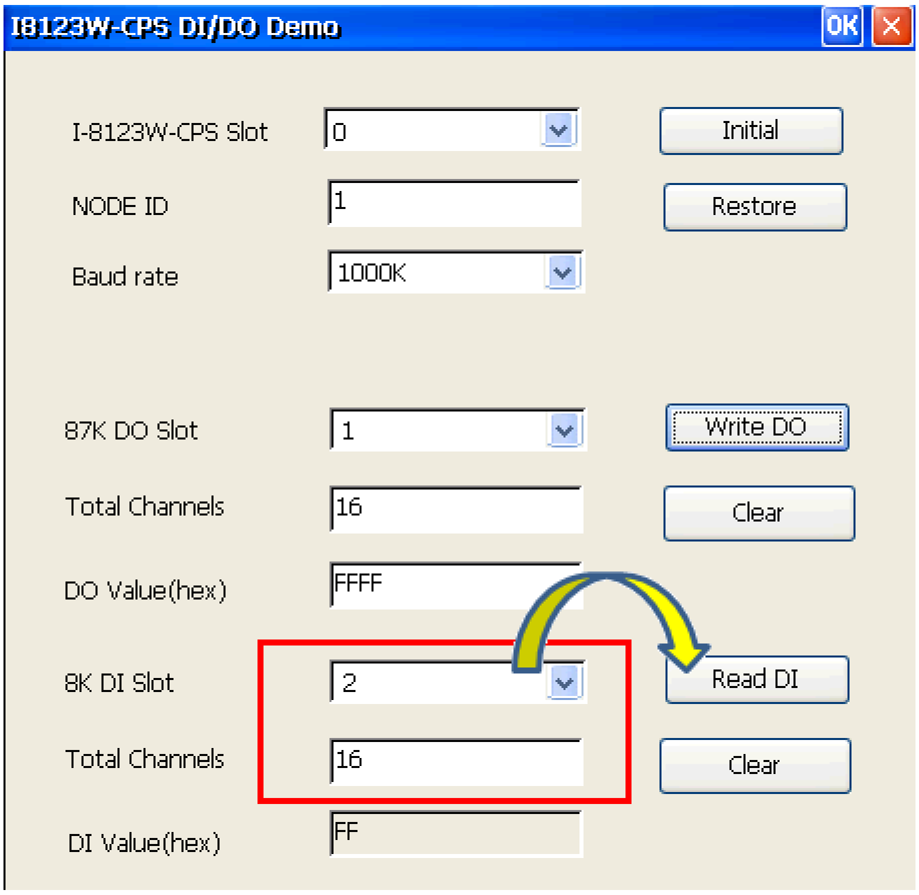
No.	Mode	ID	RTR	Len	D0	D1	D2	D3	D4	D5	D6	D7	Time Stamps(s)
1	11-bit ID	701	N	1	00								24837.83984

Step 8: Assume there is an I87K DO module with total channels is 16 on the slot 1 of the WinPAC. Select the "87k DO slot" 1 ,"Total Channels" 16 and “DO Value ” FFFF then click “ Write DO” button to set the DO value to I87K module.



I-8123W-CPS Slot: 0
NODE ID: 1
Baud rate: 1000K
87K DO Slot: 1
Total Channels: 16
DO Value(hex): FFFF

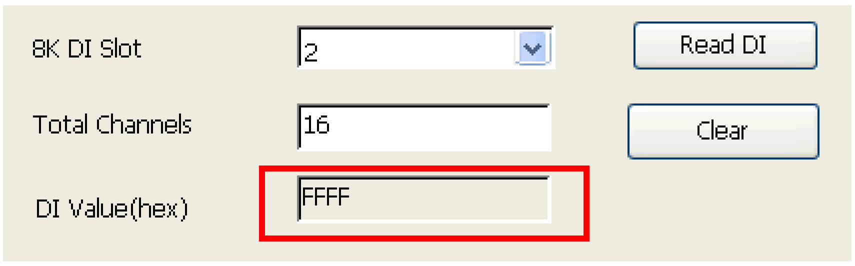
Step 9: After setting the DO values successfully, you can enter module slot inserted and total channels of an I8K DI module. Then click the “Read DI” button to read DI value of the I8K module.



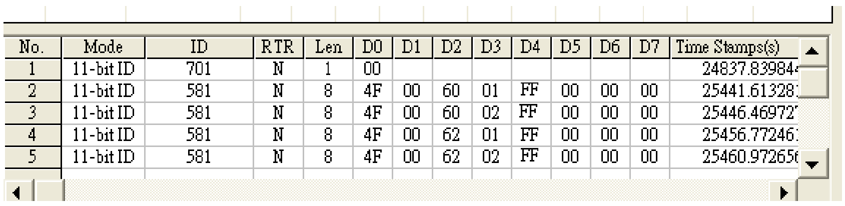
I8123W-CPS DI/DO Demo

I-8123W-CPS Slot	0	Initial
NODE ID	1	Restore
Baud rate	1000K	
87K DO Slot	1	Write DO
Total Channels	16	Clear
DO Value(hex)	FFFF	
8K DI Slot	2	Read DI
Total Channels	16	Clear
DI Value(hex)	FF	

Step 10: The DI value will be set DO value of the I87K DO module. Send SDO to read DI and DO data, the result is as below.



8K DI Slot	2	Read DI
Total Channels	16	Clear
DI Value(hex)	FFFF	



No.	Mode	ID	RTR	Len	D0	D1	D2	D3	D4	D5	D6	D7	Time Stamps(s)
1	11-bit ID	701	N	1	00								24837.83984
2	11-bit ID	581	N	8	4F	00	60	01	FF	00	00	00	25441.61328
3	11-bit ID	581	N	8	4F	00	60	02	FF	00	00	00	25446.46972
4	11-bit ID	581	N	8	4F	00	62	01	FF	00	00	00	25456.77246
5	11-bit ID	581	N	8	4F	00	62	02	FF	00	00	00	25460.97265

5. Update Firmware

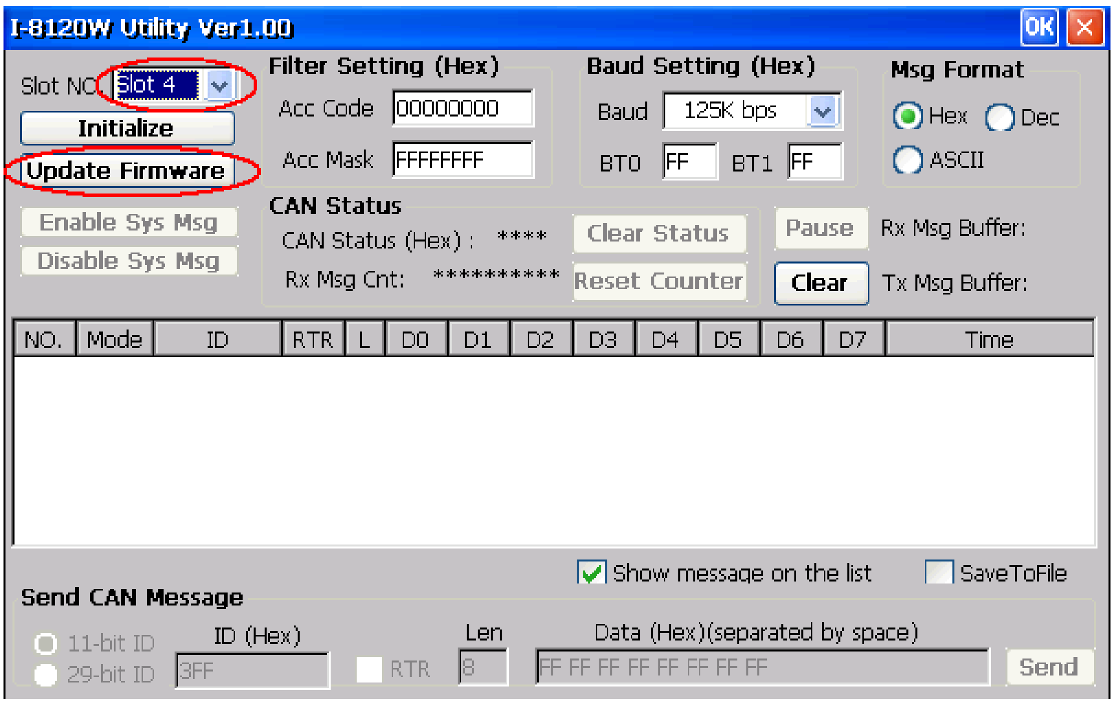
If user want to update the firmware of I-8123W-CPS, use the I-8120W utility to do it.

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

The path of field bus CD
fieldbus_cd://can/slotmodule/I_8120w/tools/wince5/

The address of the web site is
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-CPS 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-CPS has been plugged in. Then click the "Update Firmware" button.



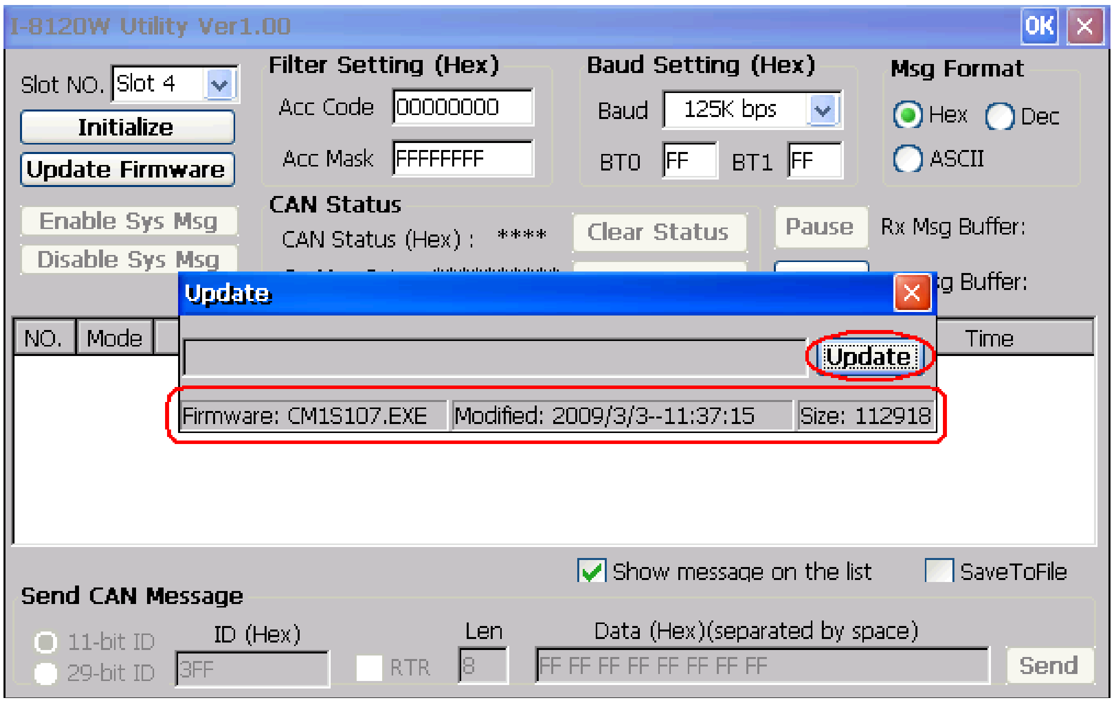
Slot NO: Slot 4
Acc Code: 00000000
Acc Mask: FFFFFFFF
Baud: 125K bps
BT0: FF BT1: FF
[✓ Hex] Dec ASCII
CAN Status (Hex) : ****
Rx Msg Cnt: **********

NO.	Mode	ID	RTR	L	D0	D1	D2	D3	D4	D5	D6	D7	Time

[✓] Show message on the list   [ ] SaveToFile

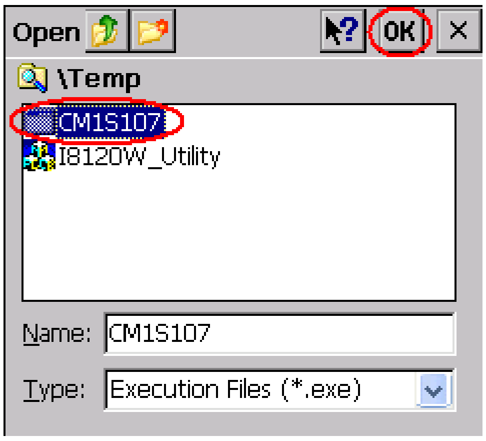
11-bit ID / 29-bit ID
ID (Hex): 3FF
RTR
Len: 8
Data (Hex)(separated by space): FF FF FF FF FF FF FF FF

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-CPS. Then click "Update" button to continue.

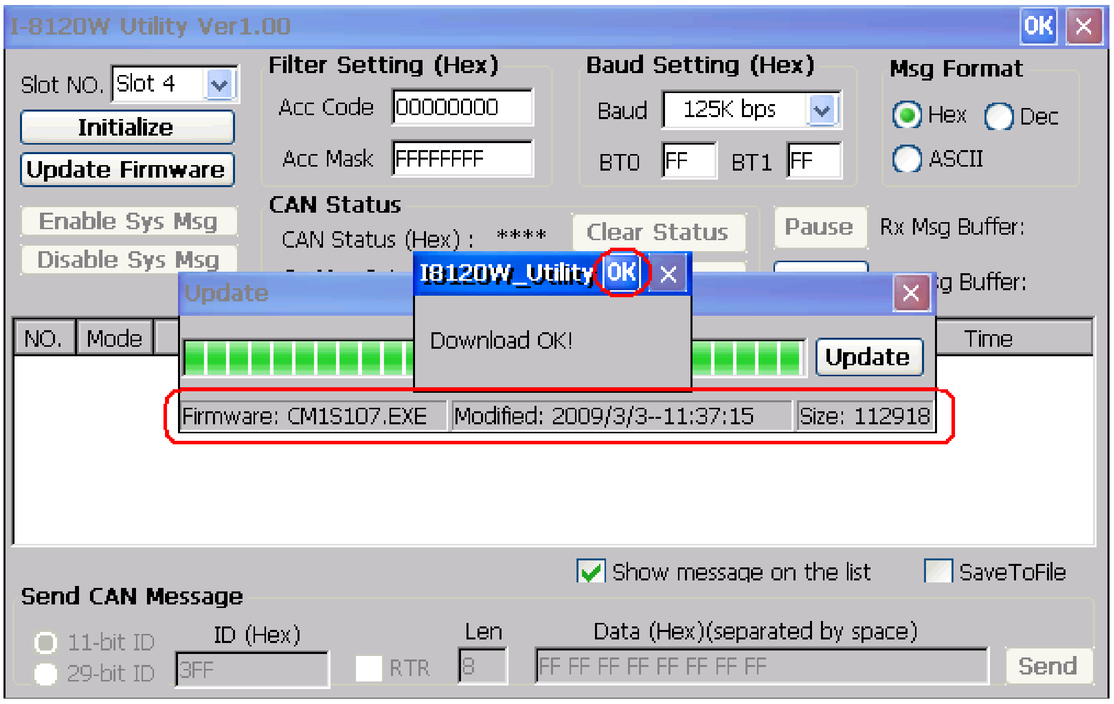


Firmware: CM1S107.EXE | Modified: 2009/3/3--11:37:15 | Size: 112918

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-CPS will be killed.



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



Acc Code 00000000
Acc Mask FFFFFFFF
Baud 125K bps
BT0 FF BT1 FF
CAN Status (Hex) : ****

Download OK!

Firmware: CM1S107.EXE | Modified: 2009/3/3--11:37:15 | Size: 112918

ID (Hex) 3FF | Len 8 | Data FF FF FF FF FF FF FF FF