
The I-8120W's Application for PM-213x-CAN Power Meter SDK User Manual

Warranty

All products manufactured by ICP DAS are under warranty regarding defective materials for a period of one year, beginning from the date of delivery to the original purchaser.

Warning

ICP DAS assumes no liability for any damage resulting from 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, not for any infringements of patents or other rights of third parties resulting from its use.

Copyright

Copyright © 2009 by ICP DAS Co., Ltd. All rights are reserved.

Trademark

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

Contact US

If you have any problem, please feel free to contact us. You can count on us for quick response.

Email: service@icpdas.com

Revision

Manual Version	Firmware Version	Date	Author	Content
1.0	1.0	2011 11/17	Johney	First Version

Table of Contents

REVISION	2
1. THE CAN BUS POWER METER OVERVIEW	5
1.1 CAN BUS INTRODUCTION	5
1.2 CAN BUS POWER METER APPLICATIONS	6
1.3 FEATURES OF THE CAN BUS AND POWER METER	7
2. I-8120W HARDWARE CONFIGURATION	9
2.1 I-8120W BOARD LAYOUT	9
2.2 JUMPER SELECTION	10
2.3 CAN BUS PIN ASSIGNMENT.....	11
2.4 CAN BUS WIRE CONNECTION	12
2.5 INDICATOR LED.....	13
2.5.1 PWR LED.....	13
2.5.2 Tx/Rx LED.....	13
2.5.3 ERR LED.....	13
2.6 UPDATE THE I-8120W'S FIRMWARE	14
3. I-8120W POWER METER SDK DESCRIPTION	17
3.1 DLL LIBRARY DEFINITION AND DESCRIPTION	17
3.2 FUNCTION RETURN CODE	18
3.3 POWER METER MANAGEMENT SDK DESCRIPTION	20
3.3.1 I8120PM_TotalModule	20
3.3.2 I8120PM_ActiveModule	21
3.3.3 I8120PM_CloseModule.....	22
3.3.4 I8120PM_GetDLLVersion	23
3.3.5 I8120PM_GetFirmwareVersion	24
3.3.6 I8120PM_ResetFirmware.....	25
3.3.7 I8120PM_GetPMIDs.....	26
3.3.8 I8120PM_SetPMIDs.....	27
3.3.9 I8120PM_GetBaudRate.....	28
3.3.10 I8120PM_SetBaudRate	29
3.3.11 I8120PM_SetAutoRespTime.....	30
3.3.12 I8120PM_GetAutoRespTime	31
3.3.13 I8120PM_StartAutoResp.....	32
3.3.14 I8120PM_StopAutoResp.....	33

3.3.15	I8120PM_GetCANStatus	34
3.3.16	I8120PM_GetPowerMeterStatus	35
3.3.17	I8120PM_ReadData	36
3.3.18	I8120PM_ReadAllData_BYTE	38
3.3.19	I8120PM_ReadAllData_FLOAT	39
3.3.20	I8120PM_GetRatioCMD	40
3.3.21	I8120PM_GetRatioValue	41
3.3.22	I8120PM_SetRatioCMD	42
3.3.23	I8120PM_SetRatioResp	43
3.3.24	I8120PM_GetPMFirmVerCMD	44
3.3.25	I8120PM_GetPMFirmVerValue	45
3.3.26	I8120PM_SetNewPMIDCMD	46
3.3.27	I8120PM_SetNewPMIDResp	47
3.4	<i>THE FLOW CHART OF THE DEVELOPMENT</i>	48
3.4.1	The Flow Chart of the Fast Development	48
3.4.2	The Flow Chart of the General Development	49
4.	<i>PAC_CAN_POWER_METER SOFTWARE UTILITY</i>	50
4.1	<i>ACTIVATE I-8120W MODULE</i>	51
4.2	<i>RESET THE FIRMWARE IN THE I-8120W</i>	52
4.3	<i>START/STOP AUTOMATIC RESPONSE</i>	53
4.4	<i>PARAMETER CONFIGURATION OF POWER METER</i>	54
4.4.1	CAN bus Baud Rate	55
4.4.2	The Time of Automatic Response	56
4.4.3	The List of the Power Meter	57
4.5	<i>DISPLAY THE REAL-TIME POWER INFORMATION</i>	58
4.6	<i>ADVANCED PARAMETER OF POWER METER</i>	59
4.6.1	The Ratio Parameters of the Power Meter	60
4.6.2	The firmware version of the Power Meter	61
4.6.3	Change the ID of the Power Meter	62

1. The CAN bus Power Meter Overview

1.1 CAN bus Introduction

The Controller Area Network (CAN) is a serial communication way, which efficiently supports distributed real-time control with a very high level of security. It provides the error process mechanisms and message priority concepts. These features can improve the network reliability and transmission efficiency. Furthermore, CAN supplies the multi-master capabilities, and is especially suited for networking “intelligent” devices as well as sensors and actuators within a system or sub-system.

The CAN bus has perfect superiority over other serial communication buses. Here shows the features.

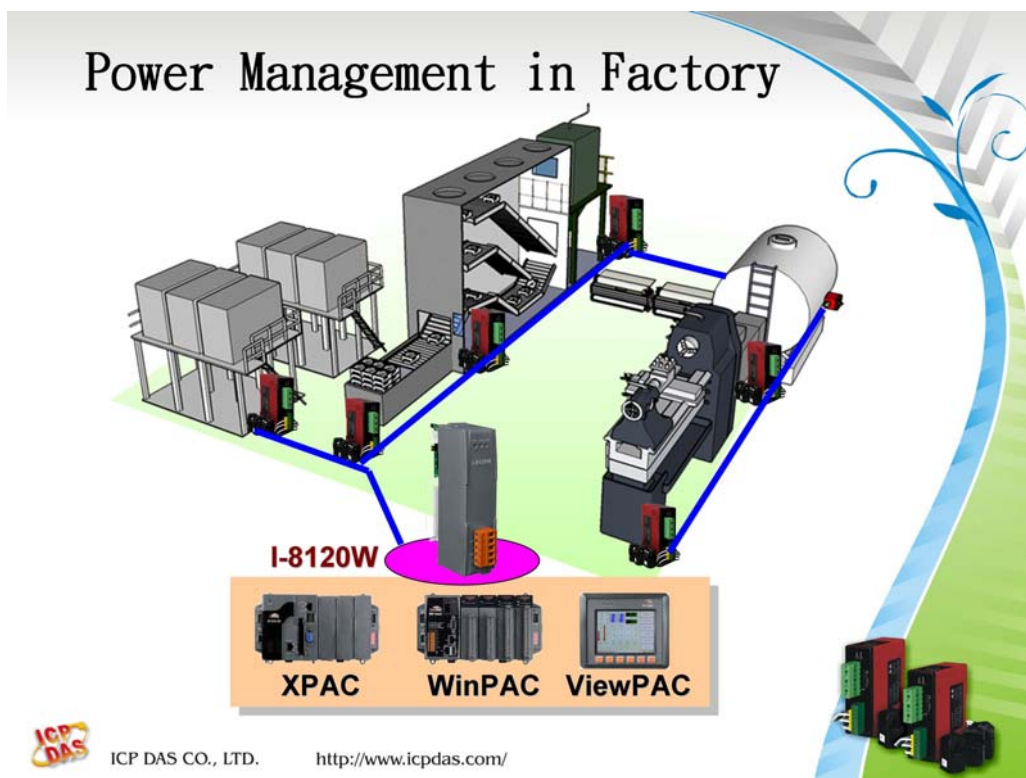
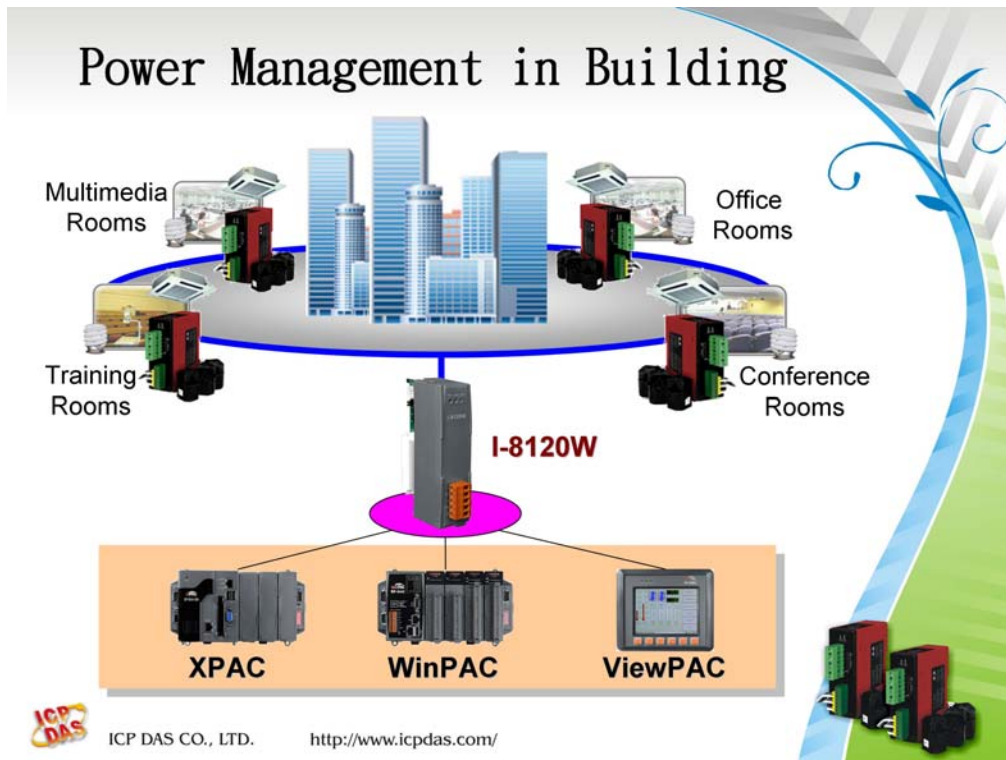
1. Physical layer with fault tolerance and anti-noise.
2. Hardware built-in five fast and stable error detecting mechanisms.
3. The arbitration guarantees the prioritization of collided messages.
4. Support Multi-master communication mechanism
5. The CAN bus has longer communication distance.
6. Good ability in integrating different CAN protocol.

Based on the superior features, those industries focusing on higher safety apply a lot of CAN bus systems. The common industries are listed as below.

1. Industrial Control.
2. Farming Machine.
3. Aircraft electricity.
4. Vehicle Monitoring.
5. Semiconductor equipment.
6. Biotechnology and medical industry.
7. Reservoir and bridge safety.
8. Military industry.
9. Factory automation.
10. HVAC System and equipment.



1.2 CAN bus Power Meter Applications



1.3 Features of the CAN bus and Power Meter

CAN bus Power Meter Features

- NXP TJA1042 CAN transceiver
- Follow CiA CAN 2.0B specification
- Support Auto-Response mode and Polling mode
- Support baud rate 125 kbps, 250 kbps and 500 kbps
- Support up to 255 nodes on the CAN network
- 2500 Vrms isolation on CAN port.
- 4 kV ESD protection on CAN port.
- CAN bus provides better capability in noise suppression
- Support arbitration mechanism to avoid collision.
- Support error detection and error correction on CAN bus.
- Operating Temperature is $-10\text{ }^{\circ}\text{C} \sim +70\text{ }^{\circ}\text{C}$

Illustration of the Polling mode:

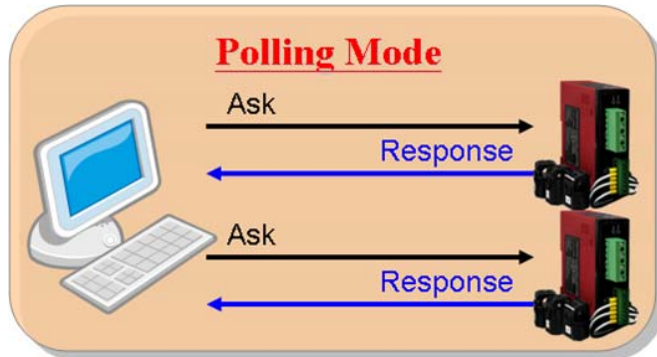
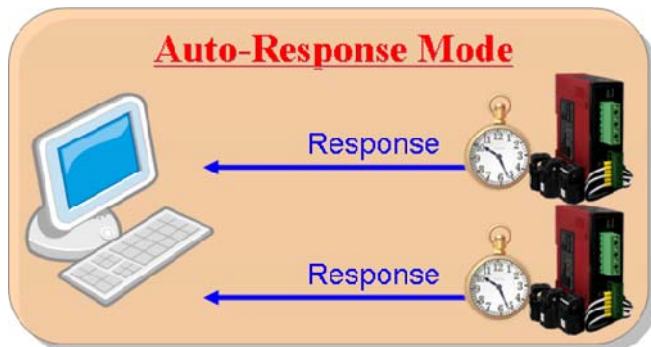


Illustration of the “Auto-Response” mode



I-8120W Power Meter Firmware Features

- Philip SJA1000 CAN controller.
- NXP 82C250 CAN transceiver.
- 2500 Vrms photo-isolation protection on CAN side.
- Support both PM-2133-CAN and PM-2134-CAN series power meter.
- Support programmable baud rate 125 kbps, 250 kbps and 500 kbps
- Managing up to 45 power meters with single I-8120W.
- Collecting power information of all power meters automatically.
- The scan time of 45 power meters is only 368ms @500Kbps.
- Monitoring the communicating status of the power meters.
- CAN bus provides better capability in noise suppression
- Support arbitration mechanism to avoid collision.
- Support error detection and error correction on CAN bus.

2. I-8120W Hardware Configuration

This section will describe the hardware settings of the I-8120W. This information includes the wire connection and terminal resistance configuration.

2.1 I-8120W Board Layout

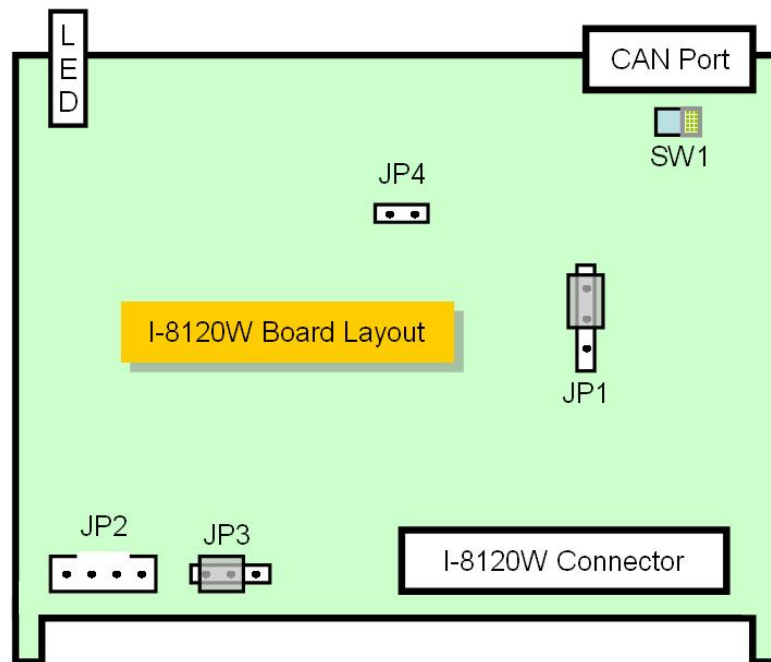


Figure2.1 I-8120W Board Layout



2.2 Jumper Selection

The following table shows the definition of jumpers. Users need to refer to this table to configure the I-8120W hardware.

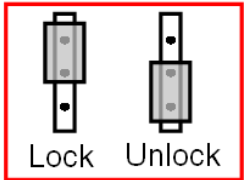


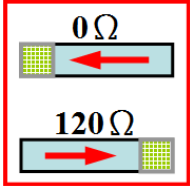
Jumper	Description	Status
JP1	Flash Protection. If you need to update the firmware, you should unlock it before you download the firmware. The default is lock.	<p>JP1</p>  <p>Lock Unlock</p>
JP4	When users download the firmware which is not proper for I-8120W so that the I-8120W can't work normally. Set this jumper to reset status until the green and yellow LEDs of I-8120W interlace flash once per second. Then, set this jumper to normal status. Afterwards, the I-8120W is in download mode. Users can overwrite the old firmware with the new one by using Utility tool.	 <p>JP4 Reset</p>  <p>JP4 Normal</p>
SW1	CAN Port 120Ω terminal resistor.	<p>Resistor Switch</p> 

Table 2.1 I-8120W Jumper selections

2.3 CAN bus Pin Assignment

The I-8120W is equipped with one **5-pin screwed terminal connector** for wire connection of the CAN bus. The connector's pin assignment is specified as follows:

The 5-pin screwed terminal connector of the CAN bus interface is shown in Figure 2.5 and the corresponding pin assignments are given in Table 2.2.

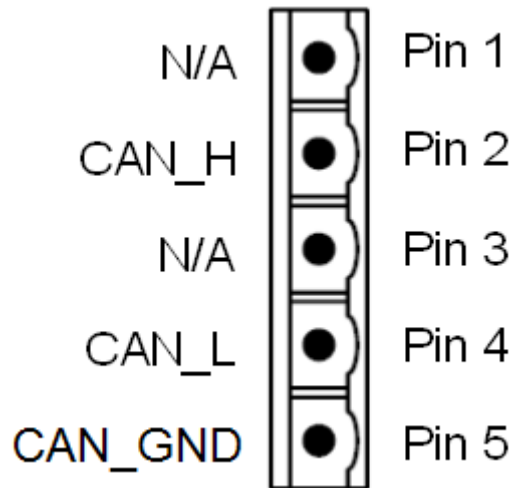


Figure2.5 5-pin terminal connector

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

Table 2.2 Pin assignment of the 5-pin screwed terminal connector

2.4 CAN bus Wire Connection

In order to minimize the reflection effects on the CAN bus line, the CAN bus line has to be terminated at both ends by two terminal resistances as in the following figure. According to the ISO 11898-2 spec, each terminal resistance is 120Ω (or between 108Ω~132Ω).

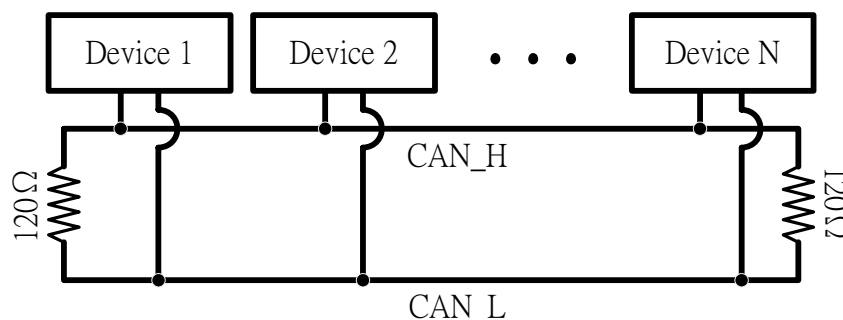


Figure 2.6 CAN bus network topology

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

Bus Length (meter)	Bus Cable Parameters		Terminal Resistance (Ω)
	Length Related Resistance (mΩ/m)	Cross Section (Type)	
0~40	70	0.25(23AWG)~ 0.34mm ² (22AWG)	124 (0.1%)
40~300	< 60	0.34(22AWG)~ 0.6mm 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

Table 2.4 Relationship between cable characteristics and terminal resistance

2.5 Indicator LED



2.5.1 PWR LED

After the I-8120W powering on, the PWR LED will also turn on.

2.5.2 Tx/Rx LED

[Tx/Rx] LED represents the status of the remote power meters.

(1). [Tx/Rx] LED solid on :

It represents that all remote power meters are normal.

(2). [Tx/Rx] LED twinkle :

It represents that at least one power meters occur errors. This situation sometime represents that the power meter has no response or is out of order. You can use I-8120W function call to realize what happens with those power meters.

2.5.3 ERR LED

[ERR] LED represents the CAN bus status. When it turns on or twinkling, the CAN bus occurs some errors.

(1). [ERR] LED twinkle :

It represents the CAN in error situation. This error sometime caused by the following reasons.

- (a) Check if the baud rate between I-8120W and power meters is unequal.
- (b) Check if the terminal resistances only added on both the end of the CAN bus.
- (c) Check if the CAN High and CAN Low has wrong connection.
- (d) Check if the CAN bus is connecting with the power meters.

2.6 Update the I-8120W's Firmware

I-8120W module should be loaded the power meter management firmware [PMPAC_xx.exe]. This module will manage and collect the data of the power meter. When first time using I-8120W or upgrade the power meter firmware, the module need to update the firmware. You can follow the simple instructions below. You will complete updating the firmware.

A. Prepare the I-8120W software utility [I8120W_UTILITY.exe]

You can download it from the FTP site of ICP DAS as below :

ftp://ftp.icpdas.com/pub/cd/fieldbus_cd/can/slotmodule/i_8120w/tools/wince5/

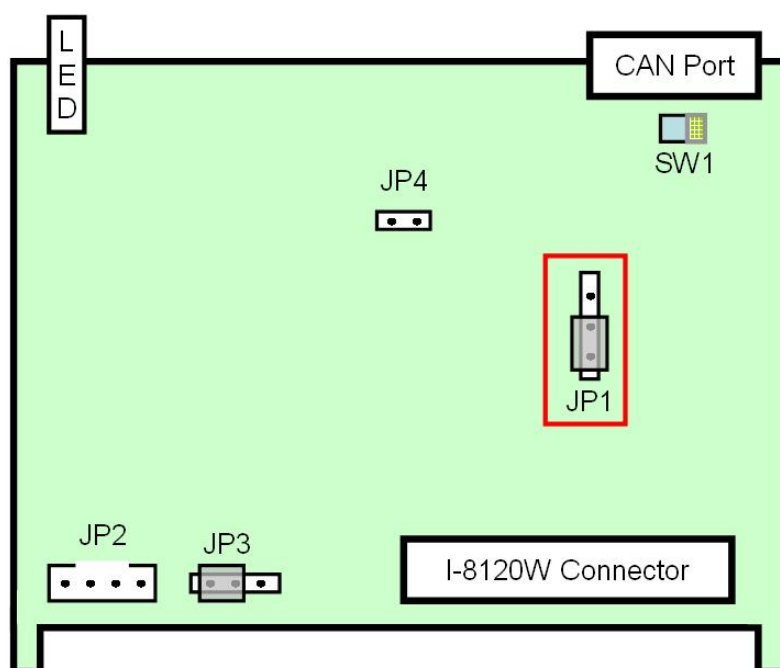
B. Prepare the power meter management firmware [PMPAC_xx.exe] (xx is the version number.)

You can download it form the FTP site of ICP DAS as below :

ftp://ftp.icpdas.com/pub/cd/fieldbus_cd/can/slotmodule/i_8120w/powermeter/firmware/

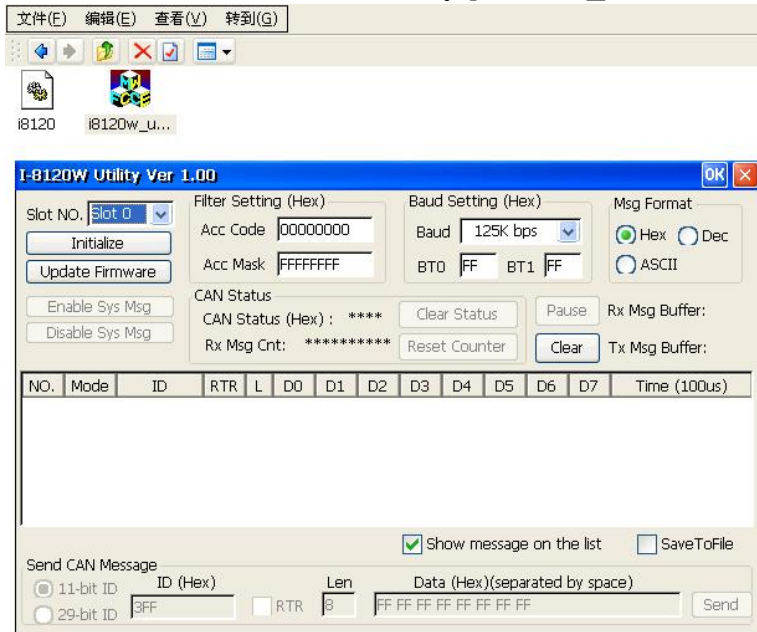
C. Change the Jumper of the I-8120W.

You need to disable the Flash protection functionality as shown below.

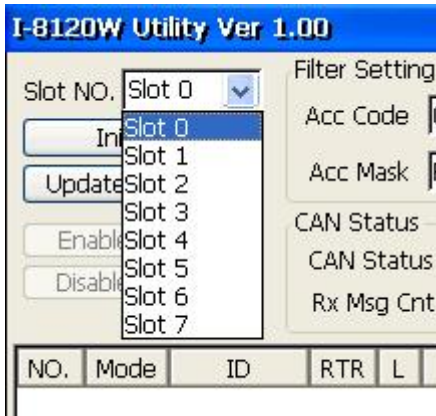


D. Update the firmware within the I-8120W.

- (1) Run the I-8120W software utility [I8120W_UTILITY.exe] .



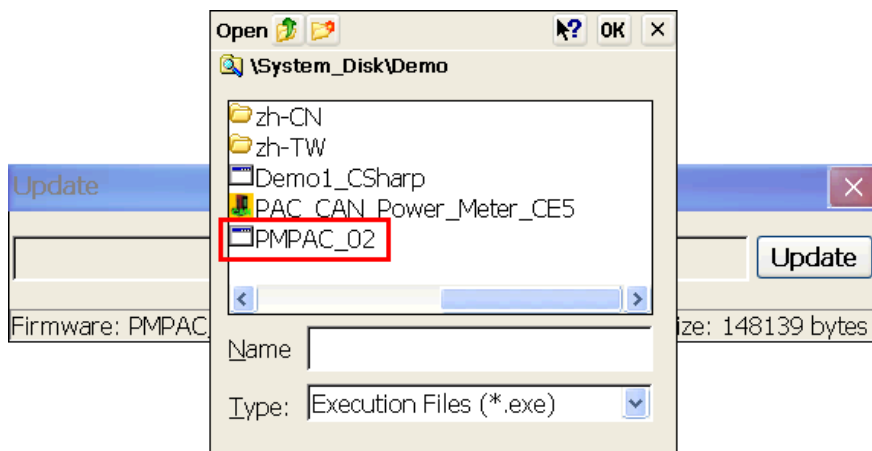
- (2) Select the slot number of the I-8120W.



- (3) Click the “Update Firmware” button. It would show the following dialog.



(4) Click the “Update” button and select the firmware [PMPAC_xx.exe] (xx is the version number).

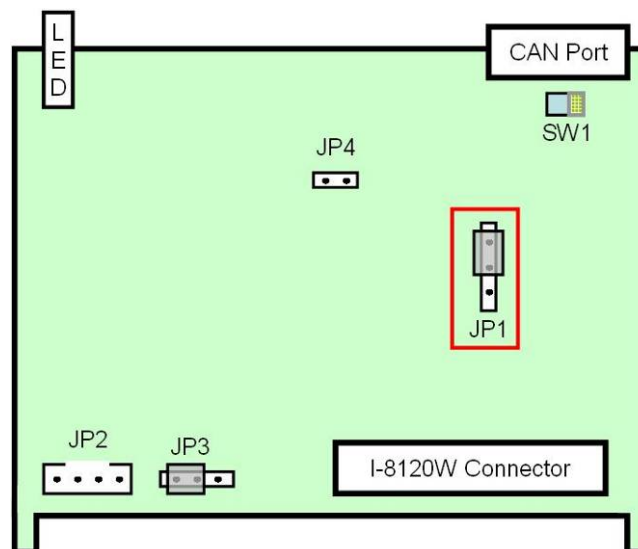


(5) After clicking the “OK” button, it will load the firmware automatically.



E. Chang the Jumper of the I-8120W again.

You need to enable the Flash protection functionality. It would protect the firmware against illegal writing. It illustrate as below.



3. I-8120W Power Meter SDK Description

3.1 DLL Library Definition and Description

All the functions provided in the I812W_PM.DLL or I8120PM_DotNET.DLL have detail description in CH3.3. However, in order to make the descriptions more simply and clearly, the attributes for the both input and output parameter functions are given as **[input]** and **[output]** respectively, as shown in the following table.

Keyword	Set parameter by user before calling this function?	Get the data from this parameter after calling this function?
[input]	Yes	No
[output]	No	Yes

3.2 Function Return Code

Table 3.2.1 Firmware Return Code

Return Code	Error ID	Comment
0	I8120PM_NoError	No Error
1000	I8120PM_PMidNotExist	Power Meter ID does not exist
1001	I8120PM_DataNameError	Data Name is error
1051	PMXS_BaudRateError	Baud Rate is error
1010	RatioError_NoGetCMD	No "GetRatio" command was sent.
1011	RatioError_NoSetCMD	No "SetRatio" command was sent.
1012	RatioError_NoResp	There is no response message.
1020	FirmError_NoGetCMD	No "GetFirm" command was sent.
1021	FirmError_NoResp	There is no response message.
1030	PMError_NoSetCMD	No "SetNewID" command was sent.
1031	PMError_NoResp	There is no response message.

Table 3.2.2 Hardware Return Code

Return Code	Error ID	Comment
10003	I8120PM_SlotNumberError	Slot No. error
10007	I8120PM_InitError	Initialize error
10021	I8120PM_SoftBufferIsEmpty	CAN buffer is empty
10022	I8120PM_SoftBufferIsFull	CAN buffer is full
10023	I8120PM_TimeOut	Firmware responses timeout
10024	I8120PM_SetCyclicMsgFailure	Cyclic Message fail
10025	I8120PM_DpramOverRange	DPRAM out of range
10026	I8120PM_NoDpramCmd	DPRAM has no command
10027	I8120PM_ModeError	Firmware is not running
10030	I8120PM_NoFileInside	Firmware does not exist
10031	I8120PM_DownloadFailure	Loading firmware is fail.
10032	I8120PM_EEPROMDamage	EEPROM is out of order
10033	I8120PM_NotEnoughSpace	EEPROM is full.
10034	I8120PM_StillDownloading	It is downloading firmware
10035	I8120PM_BoardModeError	The module is under download mode
10036	I8120PM_SetDateTimeFailure	Setting date is fail
10040	I8120PM_SlotNotConfig	Module does not configure
10041	I8120PM_SlotNotInit	Module does not initialize.
10042	I8120PM_ReplyError	Firmware replies error
10043	I8120PM_WaitForReply	Waiting for reply.

3.3 Power Meter Management SDK Description

3.3.1 I8120PM_TotalModule

- **Description :**

The function can get the count of total I-8120W modules in the user's WP-8000 or XPAC-8000 series MCU.

- **Syntax :**

```
DWORD I8120PM_TotalModule (BYTE *TotalModules ,  
                           BYTE *SlotNoList)
```

- **Parameter :**

TotalModules: [output] Return the amount of total modules.

SlotNoList: [output] Return the list of all slot No. of each modules.

- **Return :**

I8120PM_NoError(0) : OK

3.3.2 I8120PM_ActiveModule

- **Description :**

The function is used to activate the entire I-8120W module , It must be called once before using the other functions of I-8120W APIs.

- **Syntax :**

DWORD I8120PM_ActiveModule (BYTE SlotNo)

- **Parameter :**

SlotNo: [input] The slot number which the I-8120W is plugged in.

- **Return :**

I8120PM_NoError(0) : OK

I8120PM_SlotNumberError(10003) : Slot No. is error.

I8120PM_InitError(10007) : I-8120W initialization is fail.

I8120PM_NoFileInside(10030) : No firmware inside the I-8120W.

3.3.3 I8120PM_CloseModule

- **Description :**

The function is used to stop and close the I-8120W driver. This method must be called once to release the resource before exiting the user's application program.

- **Syntax :**

DWORD I8120PM_CloseModule (BYTE SlotNo)

- **Parameter :**

SlotNo: [input] The slot number which the I-8120W is plugged in.

- **Return :**

I8120PM_NoError(0) : OK

I8120PM_SlotNumberError(10003) : Slot No. is error.

3.3.4 I8120PM_GetDLLVersion

- **Description :**

By using this function, the user can query the version of theI8120W_PM.DLL.

- **Syntax :**

DWORD I8120PM_GetDLLVersion (void)

- **Parameter :**

None

- **Return :**

Return the DLL version information. For example: If 100(hex) is returned, it means DLL version is 1.00. If 123(hex) is returned, it means DLL version is 1.23.

3.3.5 I8120PM_GetFirmwareVersion

- **Description :**

The function can obtain the version information of the firmware inside the I-8120W module.

- **Syntax :**

DWORD I8120PM_GetFirmwareVersion (BYTE SlotNo)

- **Parameter :**

SlotNo: [input] The slot number which the I-8120W is plugged in.

- **Return :**

Return the firmware version information. For example: If 100(hex) is returned, it means firmware version is 1.00. If 123(hex) is returned, it means firmware version is 1.23.

- **Error Return :**

I8120PM_SlotNumberError(10003) : Slot No. is error.

3.3.6 I8120PM_ResetFirmware

- **Description :**

The function is used to reset the I-8120W module. When users have changed the baud rate of CAN bus or changed other settings, the function must be called to make the modification enable.

- **Syntax :**

DWORD I8120PM_ResetFirmware (BYTE SlotNo)

- **Parameter :**

SlotNo: [input] The slot number which the I-8120W is plugged in.

- **Return :**

I8120PM_NoError(0) : OK

I8120PM_SlotNumberError(10003) : Slot No. is error.

3.3.7 I8120PM_GetPMIDs

- **Description :**

This function is used to get the ID list of all the power meters which are in I-8120W. The users can use this function to know which power meters are in the I-8120W.

- **Syntax :**

DWORD I8120PM_GetPMIDs (BYTE SlotNo, BYTE *PMIDCount,
BYTE *PMIDArray)

- **Parameter :**

SlotNo: [input] The slot number which the I-8120W is plugged in.

PMIDCount: [output] Return the amount of all power meter in the I-8120W.

PMIDArray: [output] Return the ID array of all power meter in the I-8120W.

- **Return :**

I8120PM_NoError(0) : OK

I8120PM_SlotNumberError(10003) : Slot No. is error.

3.3.8 I8120PM_SetPMIDs

- **Description :**

This function is used to set the power meters into the I-8120W. The users can save certain power meters into the EEPROM of the I-8120W. The setting does not lose when power is off. One I-8120W could manage at most 45 power meters. If the array is larger than 45 elements, this function would retrieve the forefront 45 elements.

- **Syntax :**

DWORD I8120PM_SetPMIDs (BYTE SlotNo, BYTE PMIDCount,
BYTE *PMIDArray)

- **Parameter :**

SlotNo: [input] The slot number which the I-8120W is plugged in.

PMIDCount: [input] The amount of power meter which will be saved into the I-8120W.

PMIDArray: [input] The ID array of all power meter which will be saved into the I-8120W.

- **Return :**

I8120PM_NoError(0) : OK

I8120PM_SlotNumberError(10003) : Slot No. is error.

3.3.9 I8120PM_GetBaudRate

- **Description :**

This function can help you to get the CAN bus baud rate information of the I-8120W.

- **Syntax :**

DWORD I8120PM_GetBaudRate(BYTE SlotNo, BYTE *BaudRate)

- **Parameter :**

SlotNo: [input] The slot number which the I-8120W is plugged in.

BaudRate: [output] Return the baud rate of the CAN bus.

BaudRate = 0 , means that the baud rate is 125 kbps

BaudRate = 1 , means that the baud rate is 250 kbps

BaudRate = 2 , means that the baud rate is 500 kbps

- **Return :**

I8120PM_NoError(0) : OK

I8120PM_SlotNumberError(10003) : Slot No. is error.

3.3.10 I8120PM_SetBaudRate

- **Description :**

This function can set the CAN bus baud rate of the I-8120W. After calling this function, you must call I8120PM_ResetFirmware to reset the firmware to make change enabled.

- **Syntax :**

DWORD I8120PM_SetBaudRate (BYTE SlotNo, BYTE BaudRate)

- **Parameter :**

SlotNo: [input] The slot number which the I-8120W is plugged in.

BaudRate: [input]The baud rate of the CAN bus.

BaudRate = 0 , means that the baud rate is 125 kbps

BaudRate = 1 , means that the baud rate is 250 kbps

BaudRate = 2 , means that the baud rate is 500 kbps

- **Return :**

I8120PM_NoError(0) : OK

I8120PM_SlotNumberError(10003) : Slot No. is error.

PMXS_BaudRateError(1051) : Baud rate parameter is error.

3.3.11 I8120PM_SetAutoRespTime

- **Description :**

This function is use to set the automatic response time of the power meter. The users can change the frequency of the automatic response. After setting new response time, the users should call I8120PM_StartAutoResp or I8120PM_ResetFirmware to make the new setting enable. The I-8120W will send command to all power meter for this new setting.

- **Syntax :**

```
DWORD I8120PM_SetAutoRespTime(BYTE SlotNo,  
                               DWORD AutoRespTime)
```

- **Parameter :**

SlotNo: [input] The slot number which the I-8120W is plugged in.

AutoRespTime: [input] The automatic response time in millisecond.

- **Return :**

I8120PM_NoError(0) : OK

I8120PM_SlotNumberError(10003) : Slot No. is error.

3.3.12 I8120PM_GetAutoRespTime

- **Description :**

This function could help to obtain the value of the automatic response time in the I-8120W.

- **Syntax :**

```
DWORD I8120PM_GetAutoRespTime(BYTE SlotNo,  
                                DWORD *AutoRespTime)
```

- **Parameter :**

SlotNo: [input] The slot number which the I-8120W is plugged in.

AutoRespTime: [output] Return the automatic response time in millisecond.

- **Return :**

I8120PM_NoError(0) : OK

I8120PM_SlotNumberError(10003) : Slot No. is error.

3.3.13 I8120PM_StartAutoResp

- **Description :**

This function could make all power meters to response power information automatically. When calling this function, the I-8120W will send command to all power meters. If the power meter is responding information automatically, the power meter will response information with new frequency.

- **Syntax :**

DWORD I8120PM_StartAutoResp(BYTE SlotNo)

- **Parameter :**

SlotNo: [input] The slot number which the I-8120W is plugged in.

- **Return :**

I8120PM_NoError(0) : OK

I8120PM_SlotNumberError(10003) : Slot No. is error.

3.3.14 I8120PM_StopAutoResp

- **Description :**

This function would make all the power meters stopping to response information. If needing to re-start response information automatically, the users can call I8120PM_StartAutoResp again.

- **Syntax :**

DWORD I8120PM_StopAutoResp(BYTE SlotNo)

- **Parameter :**

SlotNo: [input] The slot number which the I-8120W is plugged in.

- **Return :**

I8120PM_NoError(0) : OK

I8120PM_SlotNumberError(10003) : Slot No. is error.

3.3.15 I8120PM_GetCANStatus

- **Description :**

This function is used to get the current status of the I-8120W.

- **Syntax :**

DWORD I8120PM_GetCANStatus(BYTE SlotNo, BYTE *CANStatus)

- **Parameter :**

SlotNo: [input] The slot number which the I-8120W is plugged in.

CANStatus: [output] Return the status of the I-8120W.

The data would be the following value.

CANSTA_NoError(0x00) : CAN bus is OK.

CANSTA_BusOff(0x01) : The wrong CAN baud-rate or wrong wire connection causes this error.

CANSTA_Error(0x02) : The transmission and reception is failure.

CANSTA_DataOverRun(0x03) : Too much reception causes the buffer overwriting.

CANSTA_TxIncomplete(0x04) : The transmission is incomplete.

CANSTA_TxLocked(0x05) : The transmission is locked.

- **Return :**

I8120PM_NoError(0) : OK

I8120PM_SlotNumberError(10003) : Slot No. is error.

3.3.16 I8120PM_GetPowerMeterStatus

- **Description :**

This function is used to get the communication status of certain power meter.

- **Syntax :**

DWORD I8120PM_GetPowerMeterStatus(BYTE SlotNo, BYTE PMID, BYTE *Status)

- **Parameter :**

SlotNo: [input] The slot number which the I-8120W is plugged in.

PMID: [input] The ID of the power meter.

Status: [output] Return the status of the power meter

The data would be the following value.

PMSTA_NoError(0x00) : The communication of this power meter is OK.

PMSTA_Timeout(0x01) : The communication of this power meter is timeout.

- **Return :**

I8120PM_NoError(0) : OK

I8120PM_SlotNumberError(10003) : Slot No. is error.

I8120PM_PMIDNotExist(1000) : The ID of the power meter does not exist in the I-8120W.

3.3.17 I8120PM_ReadData

- **Description :**

This function is used to read certain power information.

- **Syntax :**

DWORD I8120PM_ReadData(BYTE SlotNo, BYTE PMID,
WORD DataName, float *fData)

- **Parameter :**

SlotNo: [input] The slot number which the I-8120W is plugged in.

PMID: [input] The ID of the power meter.

DataName: [input] The power meter data name. Please refer to the following description.

fData: [output] Return the value of the power information.

- **Return :**

I8120PM_NoError(0) : OK

I8120PM_SlotNumberError(10003) : Slot No. is error.

I8120PM_PMIDNotExist(1000) : The ID of the power meter does not exist in the I-8120W.

- **Power Meter Data Name :**

DataName_V_a (0x0000) : The voltage of the channel A.

DataName_I_a (0x0001) : The current of the channel A.

DataName_kW_a (0x0002) : The kW of the channel A.

DataName_kvar_a (0x0003) : The kvar of the channel A.

DataName_kVA_a (0x0004) : The kVA of the channel A.

DataName_PF_a (0x0005) : The power factor of the channel A.

DataName_kWh_a (0x0006) : The kWh of the channel A.

DataName_kvarh_a(0x0007) : The kvarh of the channel A.

DataName_kVAh_a(0x0008) : The kVAh of the channel A.

DataName_V_b (0x0009) : The voltage of the channel B.

DataName_I_b (0x000A) : The current of the channel B.

DataName_kW_b (0x000B) : The kW of the channel B.
DataName_kvar_b (0x000C) : The kvar of the channel B.
DataName_kVA_b (0x000D) : The kVA of the channel B.
DataName_PF_b (0x000E) : The power factor of the channel B.
DataName_kWh_b (0x000F) : The kWh of the channel B.
DataName_kvarh_b(0x0010) : The kvarh of the channel B.
DataName_kVAh_b(0x0011) : The kVAh of the channel B.

DataName_V_c (0x0012) : The voltage of the channel C.
DataName_I_c (0x0013) : The current of the channel C.
DataName_kW_c (0x0014) : The kW of the channel C.
DataName_kvar_c (0x0015) : The kvar of the channel C.
DataName_kVA_c (0x0016) : The kVA of the channel C.
DataName_PF_c (0x0017) : The power factor of the channel C.
DataName_kWh_c (0x0018) : The kWh of the channel C.
DataName_kvarh_c(0x0019) : The kvarh of the channel C.
DataName_kVAh_c(0x001A) : The kVAh of the channel C.

DataName_V_d (0x001B) : The voltage of the channel D.
DataName_I_d (0x001C) : The current of the channel D.
DataName_kW_d (0x001D) : The kW of the channel D.
DataName_kvar_d (0x001E) : The kvar of the channel D.
DataName_kVA_d (0x001F) : The kVA of the channel D.
DataName_PF_d (0x0020) : The power factor of the channel D.
DataName_kWh_d (0x0021) : The kWh of the channel D.
DataName_kvarh_d(0x0022) : The kvarh of the channel D.
DataName_kVAh_d(0x0023) : The kVAh of the channel D.

3.3.18 I8120PM_ReadAllData_BYTE

- **Description :**

This function is used to read all power information in byte format.

- **Syntax :**

DWORD I8120PM_ReadAllData_BYTE(BYTE SlotNo, BYTE PMID,
WORD *DataLen, BYTE *DataArray)

- **Parameter :**

SlotNo: [input] The slot number which the I-8120W is plugged in.

PMID: [input] The ID of the power meter.

DataLen: [output] Return the size of all power meter information in byte.
At present, the value is always 144 Bytes.

DataArray: [output] Return all power meter information. Please refer to the following illustration for the arrangement of all the data.

- **Return :**

I8120PM_NoError(0) : OK

I8120PM_SlotNumberError(10003) : Slot No. is error.

I8120PM_PMIIDNotExist(1000) : The ID of the power meter does not exist in the I-8120W.

- **The Arrangement of all power data in byte :**

	0x00	0x04	0x08	0x0C
0x00	V_a	I_a	kW_a	kvar_a
0x10	kVA_a	PF_a	kWh_a	Kvarh_a
0x20	kVAh_a	V_b	I_b	kW_b
0x30	kvar_b	kVA_b	PF_b	kWh_b
0x40	Kvarh_b	kVAh_b	V_c	I_c
0x50	kW_c	kvar_c	kVA_c	PF_c
0x60	kWh_c	Kvarh_c	kVAh_c	V_d
0x70	I_d	kW_d	kvar_d	kVA_d
0x80	PF_d	kWh_d	Kvarh_d	kVAh_d
0x90				

3.3.19 I8120PM_ReadAllData_FLOAT

- **Description :**

This function is used to read all power information in float format.

- **Syntax :**

DWORD I8120PM_ReadAllData_FLOAT(BYTE SlotNo, BYTE PMID, WORD *DataLen, float *fdataArray)

- **Parameter :**

SlotNo: [input] The slot number which the I-8120W is plugged in.

PMID: [input] The ID of the power meter.

DataLen: [output] Return the size of all power meter information in float.
At present, the value is always 36 floats.

fdataArray: [output] Return all power meter information. Please refer to the following illustration for the arrangement of all the data.

- **Return :**

I8120PM_NoError(0) : OK

I8120PM_SlotNumberError(10003) : Slot No. is error.

I8120PM_PmidNotExist(1000) : The ID of the power meter does not exist in the I-8120W.

- **The Arrangement of all power data in float :**

	0x00	0x01	0x02	0x03
0x00	V_a	I_a	kW_a	kvar_a
0x04	kVA_a	PF_a	kWh_a	Kvarh_a
0x08	kVAh_a	V_b	I_b	kW_b
0x0C	kvar_b	kVA_b	PF_b	kWh_b
0x10	Kvarh_b	kVAh_b	V_c	I_c
0x14	kW_c	kvar_c	kVA_c	PF_c
0x18	kWh_c	Kvarh_c	kVAh_c	V_d
0x1C	I_d	kW_d	kvar_d	kVA_d
0x20	PF_d	kWh_d	Kvarh_d	kVAh_d

3.3.20 I8120PM_GetRatioCMD

- **Description :**

This function will send a "Get Ratio" command to the specific power meter. The returned value of this function indicates if the command has been sent successfully or not. The users can use the I8120PM_GetRatioValue function to read the ratio value replied from the power meter.

- **Syntax :**

DWORD I8120PM_GetRatioCMD (BYTE SlotNo, BYTE PMID)

- **Parameter :**

SlotNo: [input] The slot number which the I-8120W is plugged in.

PMID: [input] The ID of the power meter.

- **Return :**

I8120PM_NoError(0) : OK

I8120PM_SlotNumberError(10003) : Slot No. is error.

I8120PM_PMIIDNotExist(1000) : The ID of the power meter does not exist in the I-8120W.

3.3.21 I8120PM_GetRatioValue

- **Description :**

The function can read the ratio values which have been replied from the power meter. Before using this function, users need to call the function I8120PM_GetRatioCMD first. Then, users can get the last ratio values correctly.

- **Syntax :**

```
DWORD I8120PM_GetRatioValue (BYTE SlotNo,  BYTE PMID,  
                             WORD *MeterRatio, WORD *PTRatio, WORD *CTRatio)
```

- **Parameter :**

SlotNo: [input] The slot number which the I-8120W is plugged in.

PMID: [input] The ID of the power meter.

MeterRatio: [output] Return the meter ratio value of the power meter.

PTRatio: [output] Return the PT ratio value of the power meter.

CTRatio: [output] Return the CT ratio value of the power meter.

- **Return :**

I8120PM_NoError(0) : OK

I8120PM_SlotNumberError(10003) : Slot No. is error.

I8120PM_PMIDNotExist(1000) : The ID of the power meter does not exist in the I-8120W.

RatioError_NoGetCMD(1010) : No "Get Ratio" command was sent.

RatioError_NoResp(1012) : There is no response message from the power meter.

3.3.22 I8120PM_SetRatioCMD

- **Description :**

This function will send a "Set Ratio" command to the specific power meter. The returned value of this function indicates if the command has been sent successfully or not. Users can use the I8120PM_SetRatioResp function to read the response replied from the power meter to confirm if the setting is ok or not.

- **Syntax :**

DWORD I8120PM_SetRatioCMD (BYTE SlotNo, BYTE PMID,
WORD MeterRatio, WORD PTRatio, WORD CTRatio)

- **Parameter :**

SlotNo: [input] The slot number which the I-8120W is plugged in.

PMID: [input] The ID of the power meter.

MeterRatio: [input] The meter ratio value of the power meter.

PTRatio: [input] The PT ratio value of the power meter.

CTRatio: [input] The CT ratio value of the power meter.

- **Return :**

I8120PM_NoError(0) : OK

I8120PM_SlotNumberError(10003) : Slot No. is error.

I8120PM_PMIDNotExist(1000) : The ID of the power meter does not exist in the I-8120W.

3.3.23 I8120PM_SetRatioResp

- **Description :**

After calling I8120PM_SetRatioCMD, the function can read the response values which have been replied from the power meter. Before using this function, users need to call I8120PM_SetRatioCMD first. Then, the users can get the last response correctly.

- **Syntax :**

DWORD I8120PM_SetRatioResp (BYTE SlotNo, BYTE PMID,
WORD *RespValue)

- **Parameter :**

SlotNo: [input] The slot number which the I-8120W is plugged in.

PMID: [input] The ID of the power meter.

RespValue: [output] The meter ratio value of the power meter.

- **Return :**

I8120PM_NoError(0) : OK

I8120PM_SlotNumberError(10003) : Slot No. is error.

I8120PM_PMIDNotExist(1000) : The ID of the power meter does not exist in the I-8120W.

RatioError_NoSetCMD(1011) : No "Set Ratio" command was sent.

RatioError_NoResp(1012) : There is no response message from the power meter.

3.3.24 I8120PM_GetPMFirmVerCMD

- **Description :**

This function will send a "Get Firmware Version" command to the specific power meter. The returned value of this function indicates if the command has been sent successfully or not. Users can use the I8120PM_GetPMFirmVerValue function to read the response firmware version replied from the power meter.

- **Syntax :**

DWORD I8120PM_GetPMFirmVerCMD (BYTE SlotNo, BYTE PMID)

- **Parameter :**

SlotNo: [input] The slot number which the I-8120W is plugged in.

PMID: [input] The ID of the power meter.

- **Return :**

I8120PM_NoError(0) : OK

I8120PM_SlotNumberError(10003) : Slot No. is error.

I8120PM_PMIIDNotExist(1000) : The ID of the power meter does not exist in the I-8120W.

3.3.25 I8120PM_GetPMFirmVerValue

- **Description :**

After calling the function I8120PM_GetPMFirmVerCMD, the specific power meter will reply the result of the processed command. This response will be kept in the I-8120W. Therefore, calling the function I8120PM_GetPMFirmVerValue can read the response stored in the I-8120W to confirm if the setting is ok or not. Before using this function, users need to call the function I8120PM_GetPMFirmVerCMD first. Then, users can get the firmware version of the specific power meter correctly.

- **Syntax :**

```
DWORD I8120PM_GetPMFirmVerValue (BYTE SlotNo,  BYTE PMID,  
                                WORD *PMFirmVer)
```

- **Parameter :**

SlotNo: [input] The slot number which the I-8120W is plugged in.

PMID: [input] The ID of the power meter.

PMFirmVer: [output] Return the firmware version of the power meter.

- **Return :**

I8120PM_NoError(0) : OK

I8120PM_SlotNumberError(10003) : Slot No. is error.

I8120PM_PMIDNotExist(1000) : The ID of the power meter does not exist in the I-8120W.

FirmError_NoGetCMD(1020) : No "Get Firm Ver" command was sent.

FirmError_NoResp(1021) : There is no response message from the power meter.

3.3.26 I8120PM_SetNewPMIDCMD

- **Description :**

This function will send a "Change ID" command to the specific power meter. The returned value of this function indicates if the command has been sent successfully or not. Users can use the I8120PM_SetNewPMIDResp function to read the response replied from the power meter.

- **Syntax :**

DWORD I8120PM_SetNewPMIDCMD (BYTE SlotNo, BYTE OldPMID, BYTE NewPMID)

- **Parameter :**

SlotNo: [input] The slot number which the I-8120W is plugged in.

OldPMID: [input] The original ID of the power meter.

NewPMID: [input] The desired new ID of the power meter.

- **Return :**

I8120PM_NoError(0) : OK

I8120PM_SlotNumberError(10003) : Slot No. is error.

I8120PM_PMIDNotExist(1000) : The ID of the power meter does not exist in the I-8120W.

3.3.27 I8120PM_SetNewPMIDResp

- **Description :**

After calling I8120PM_SetNewPMIDCMD, the specific power meter will reply the result of the processed command. This response will be kept in the I-8120W. Therefore, calling the function I8120PM_SetNewPMIDResp can read the response stored in the I-8120W to confirm if the setting is ok or not. Before using this function, users need to call the function I8120PM_SetNewPMIDCMD first.

- **Syntax :**

```
DWORD I8120PM_SetNewPMIDResp(BYTE SlotNo, BYTE NewPMID,  
                              DWORD *RespValue)
```

- **Parameter :**

SlotNo: [input] The slot number which the I-8120W is plugged in.

NewPMID: [input] The new ID of the power meter.

RespValue: [output] Return the response value which has been replied from the power meter.

- **Return :**

I8120PM_NoError(0) : OK

I8120PM_SlotNumberError(10003) : Slot No. is error.

I8120PM_PMIDNotExist(1000) : The ID of the power meter does not exist in the I-8120W.

PMError_NoSetCMD(1030) : No "Set New ID" command was sent.

PMError_NoResp(1031) : There is no response message from the power meter.

3.4 The Flow Chart of the Development

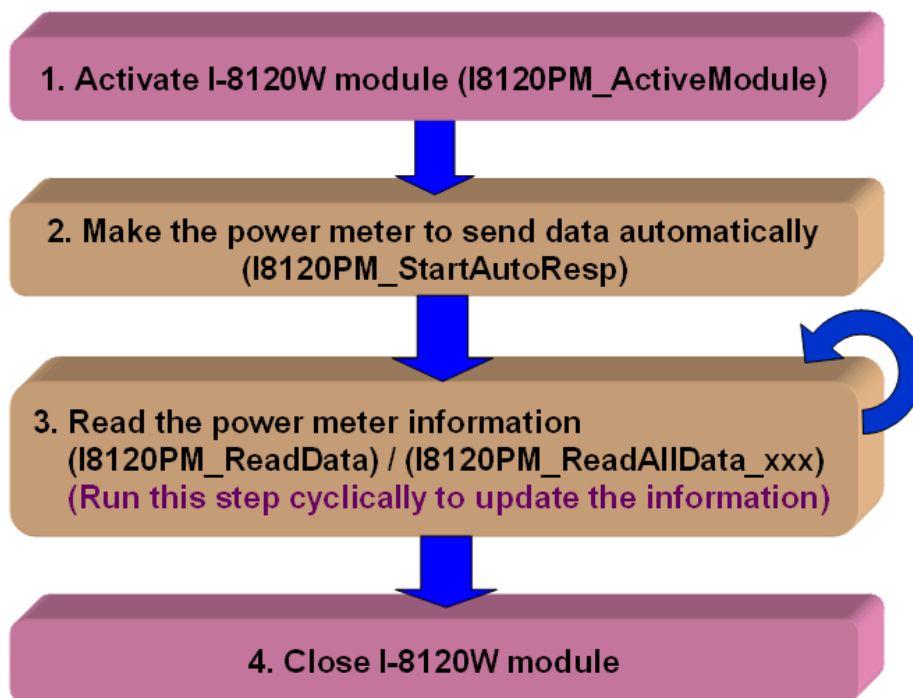
3.4.1 The Flow Chart of the Fast Development

The users can use PAC_CAN_Power_Meter software utility to configure the response time and other parameters. After configuring those parameters, the settings would be saved into the I-8120W. Then, the users can use your familiar development tools to collect all power meter information. We illustrate the steps described previously.

The Flow Chart of the Fast Development

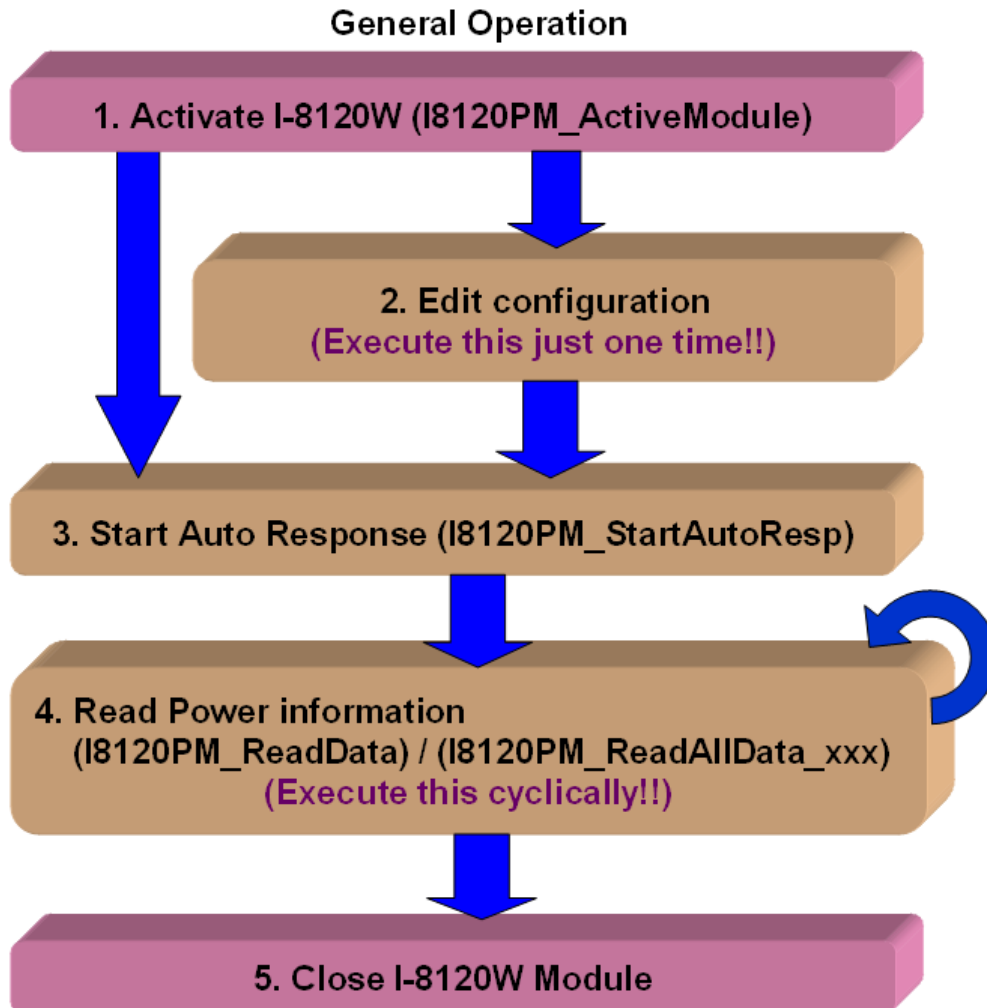


After configuring by PAC_CAN_Power_Meter, You can save the setting into the I-8120W.



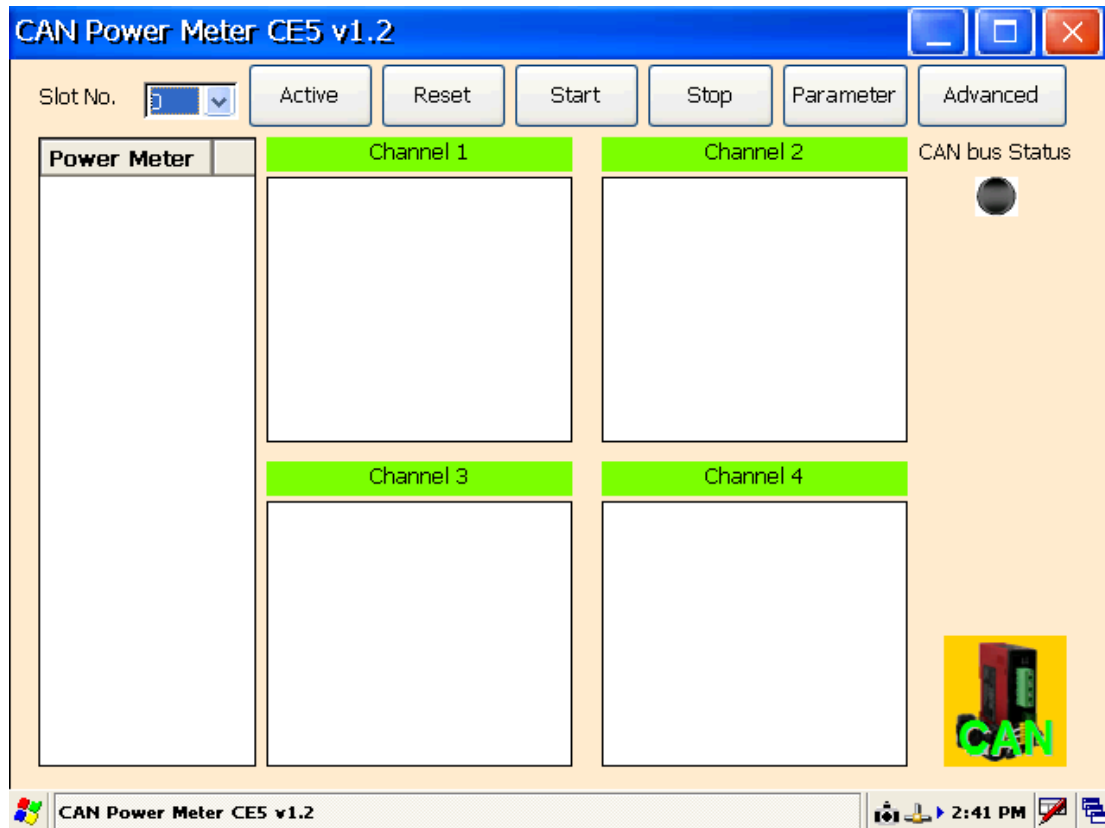
3.4.2 The Flow Chart of the General Development

The users can design your own software including the configuration of the automatic response time and those parameters. The users can design your own interface and operation method. We show the flow chart of the general development as below.



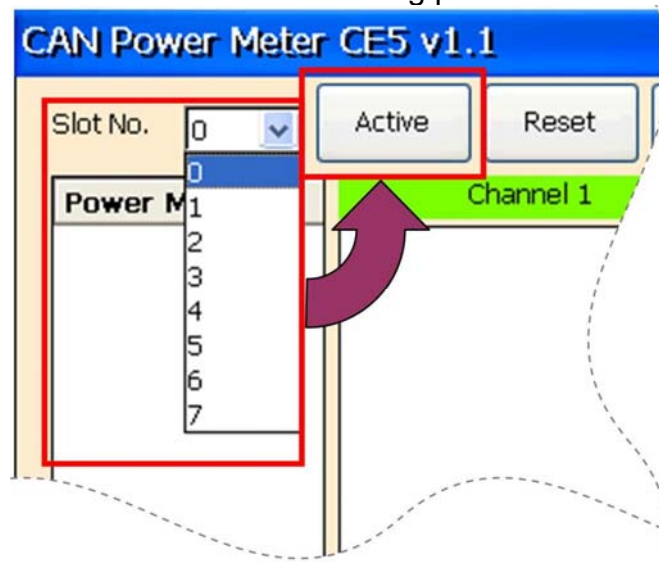
4. PAC_CAN_Power_Meter Software Utility

PAC_CAN_Power_Meter manages CAN bus power meters. The utility need to work with the I-8120W module which has loaded power meter management firmware. The utility interface looks like the following picture when it has started.

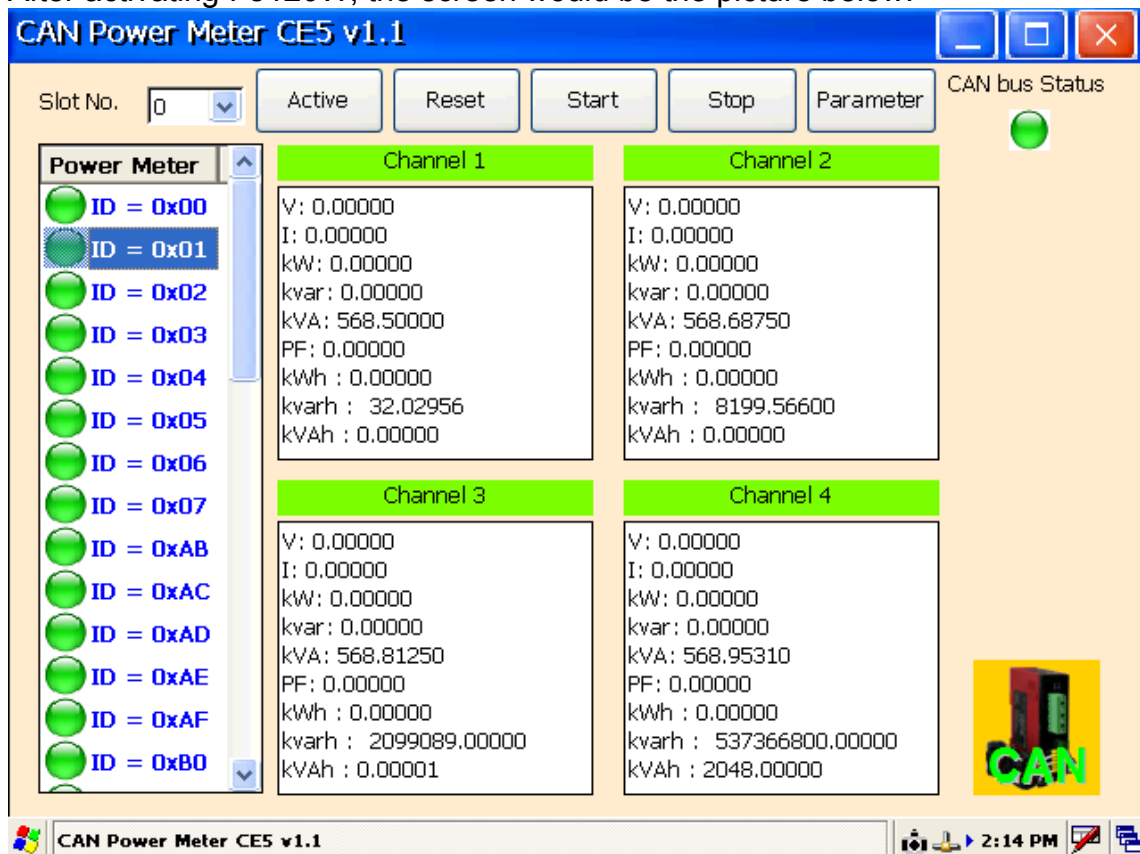


4.1 Activate I-8120W Module

Please check the slot No. which the I-8120W is plugged in. The users can select the correct number in the field of "Slot No.". And then, click the "Active" button to activate the I-8120W. The following picture shows the detail steps.

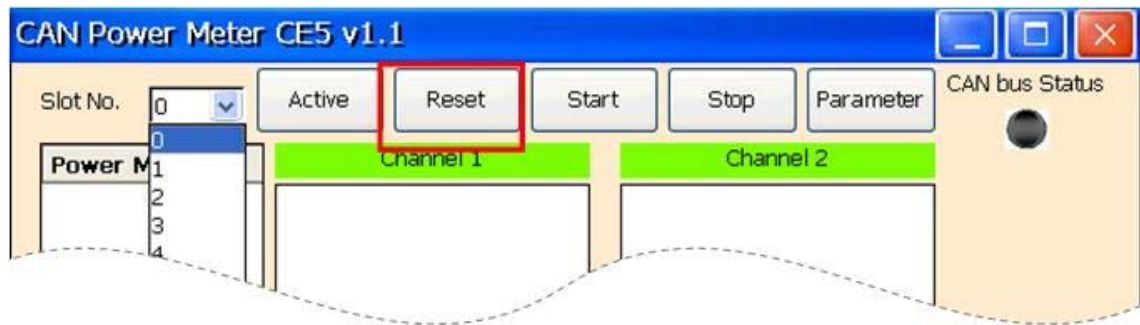


After activating I-8120W, the screen would be the picture below.



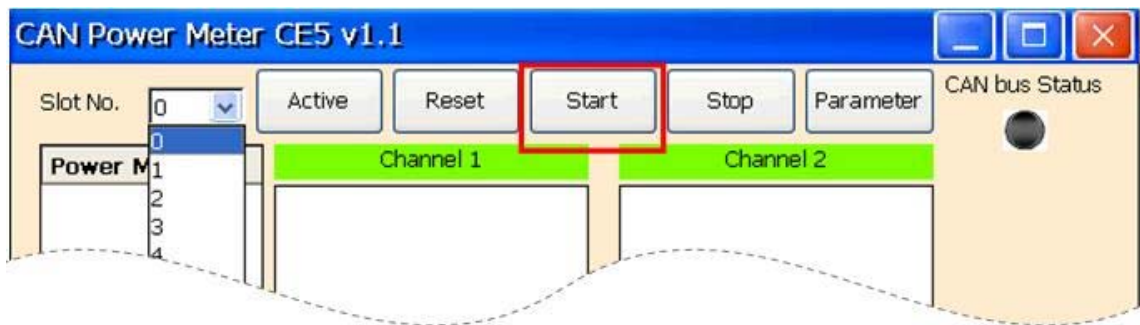
4.2 Reset the Firmware in the I-8120W

When changing the important setting, the users need to restart the firmware to make the new setting enable. The users can click the “Reset” button to reset the firmware. Here shows the detail steps.

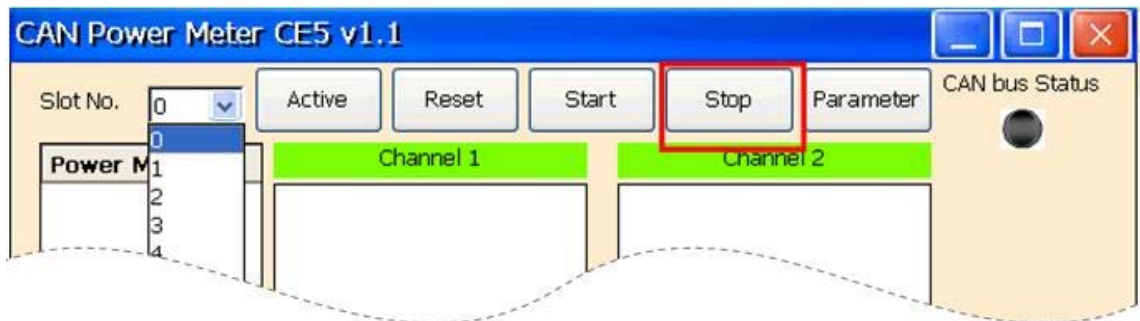


4.3 Start / Stop Automatic Response

When completing the wire connection of power meter, the users can make all power meters to reply information automatically. The users can click the “Start” button to achieve this functionality. The picture below shows the detail steps.

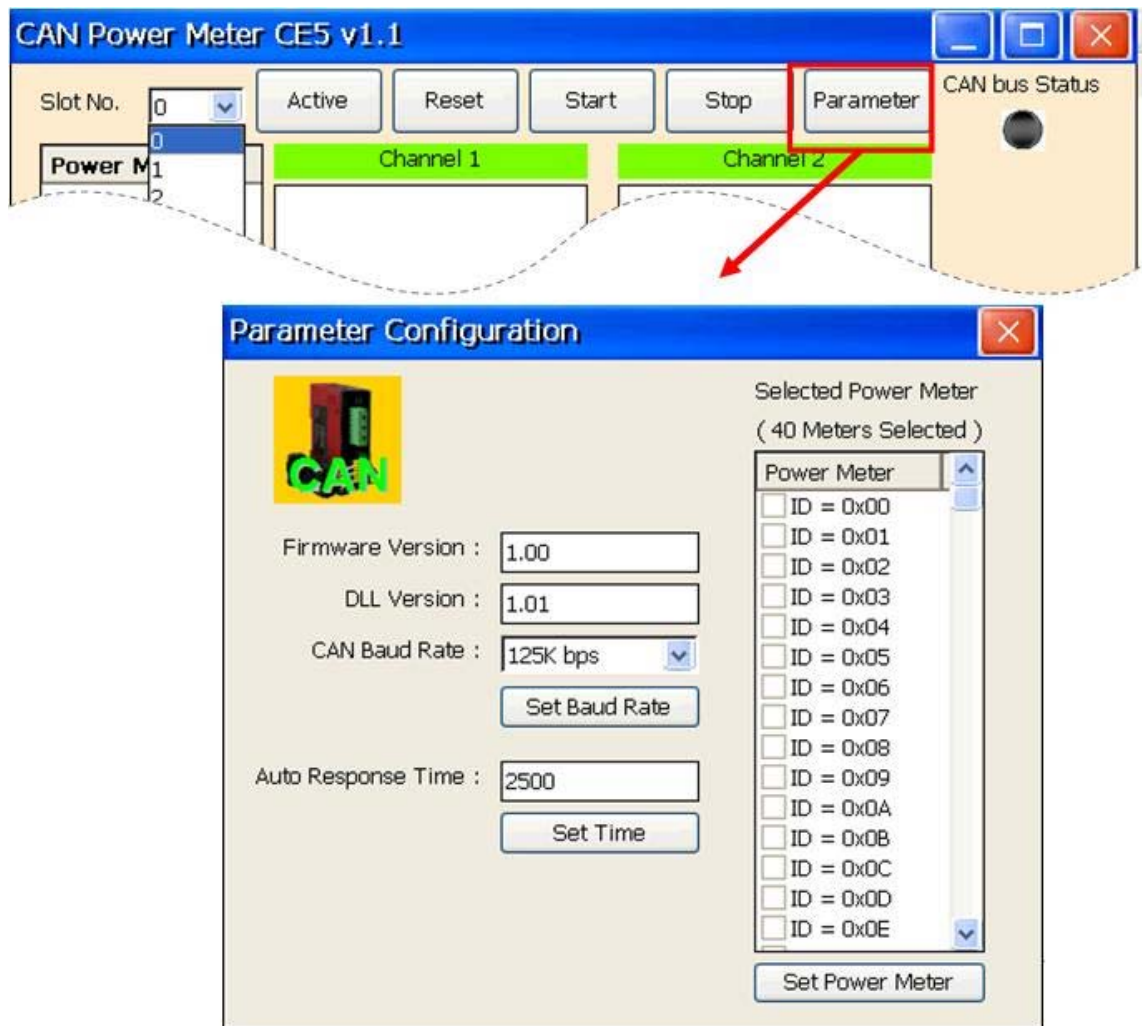


If pausing to reply is necessary, the users can click the “Stop” button to make all the power meters break off. The following picture shows the detail steps.



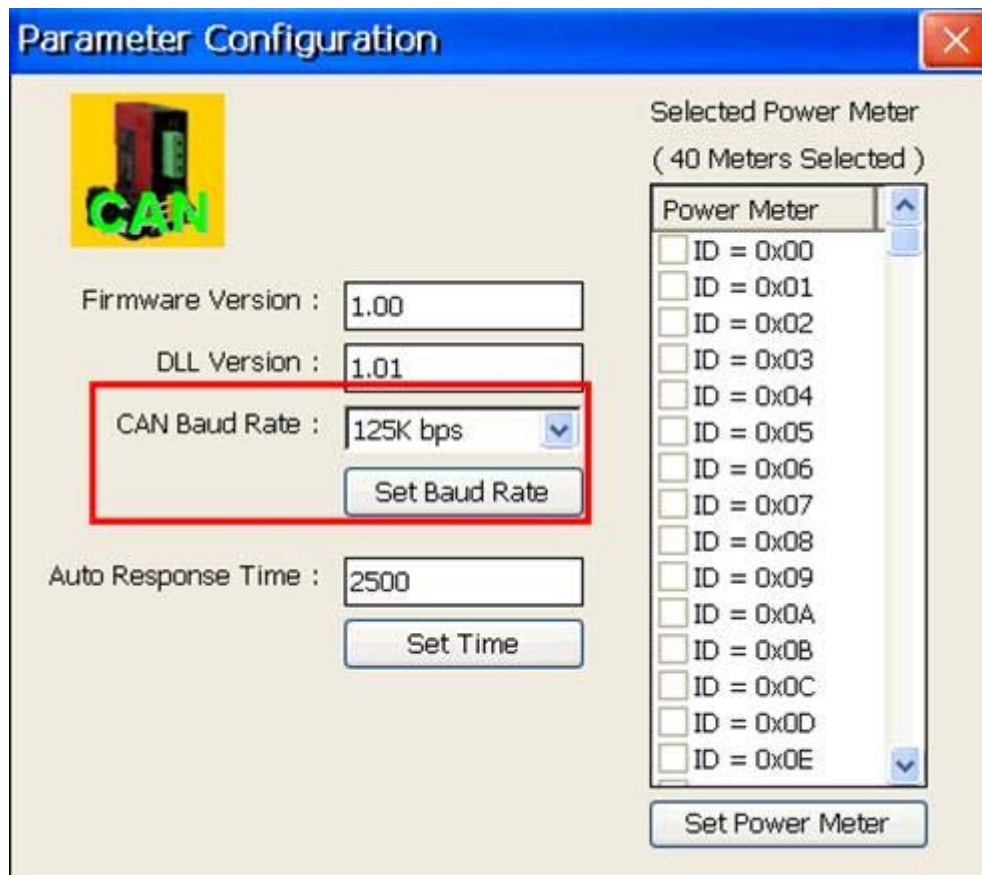
4.4 Parameter Configuration of Power Meter

The users can configure the list of power meters and automatic response time by “Parameter Configuration”. At the same time, the users can check the version of the firmware and the DLL. The users can click the “Parameter” button to show up the “Parameter Configuration” dialog. Here shows the detail steps.



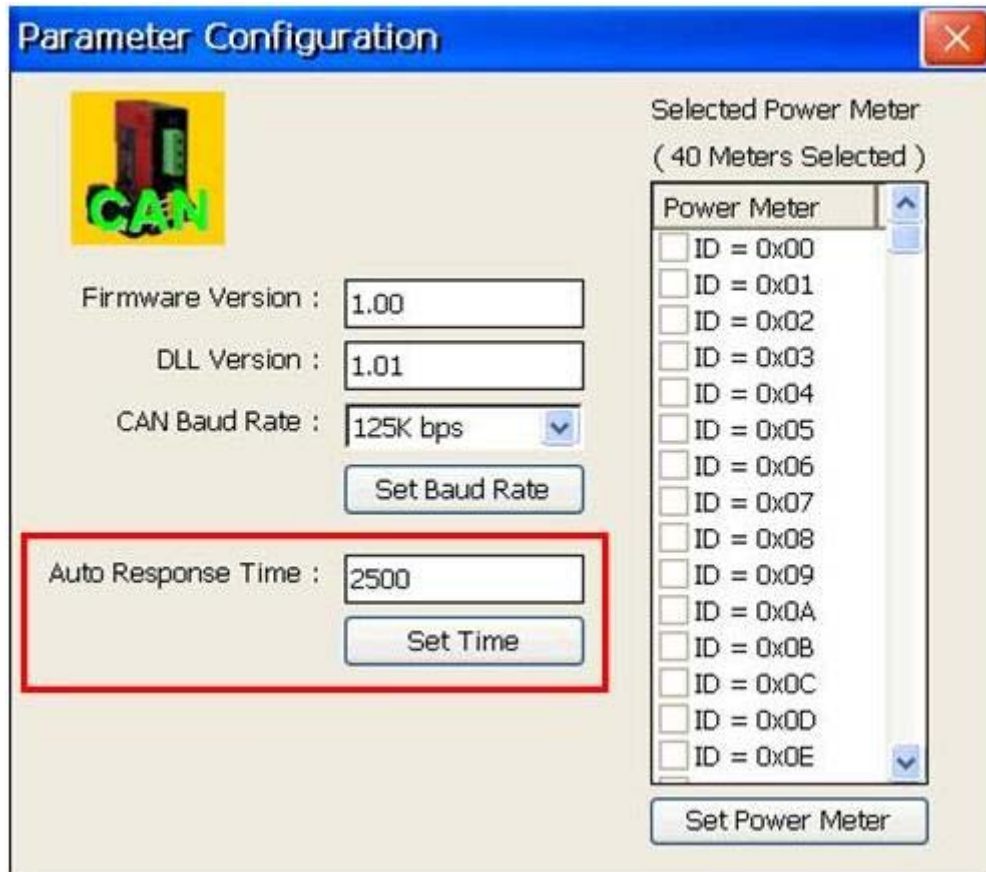
4.4.1 CAN bus Baud Rate

After clicking the “Parameter” button, the users can see the following picture and select the desired baud rate. After selecting the baud rate, the users can click “Set Baud Rate” button to save the setting into the I-8120W. The following picture shows the detail steps.



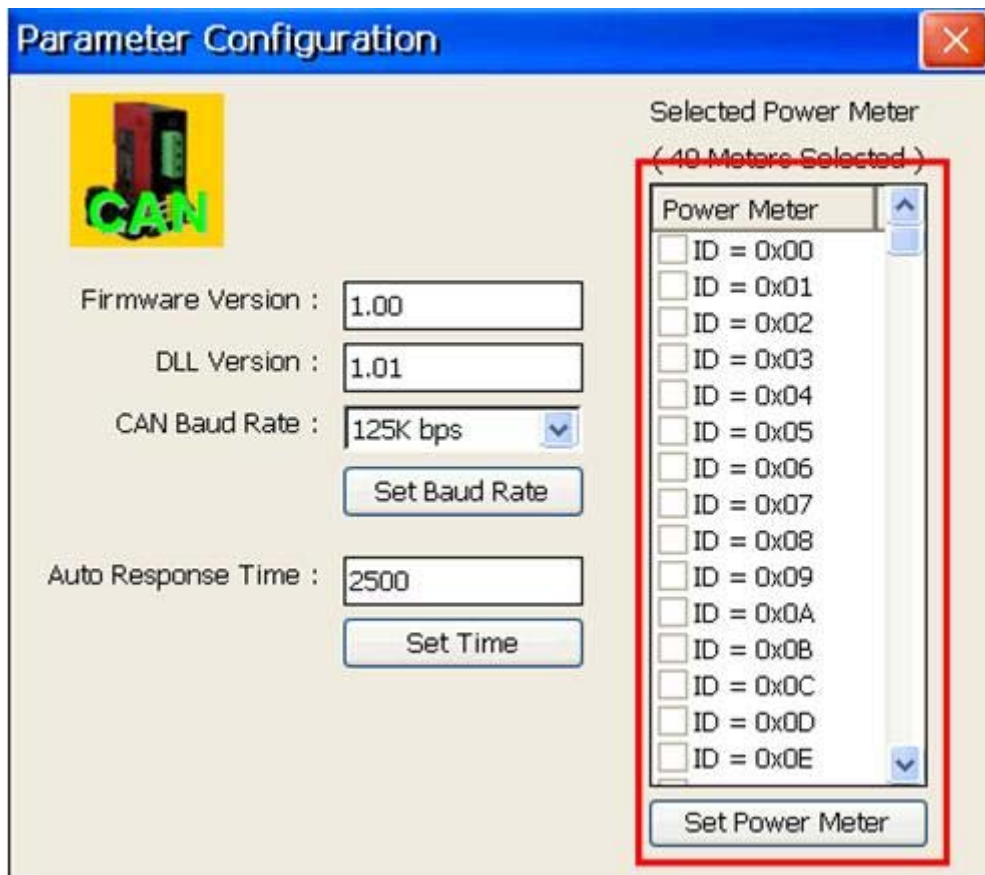
4.4.2 The Time of Automatic Response

After clicking the “Parameter” button, the users can see the following picture and input the desired time value. Notice that the input value must be decimal format in millisecond. After inputting the time value, the users can click the “Set Time” button to save the setting into the I-8120W. The picture below shows the detail steps.



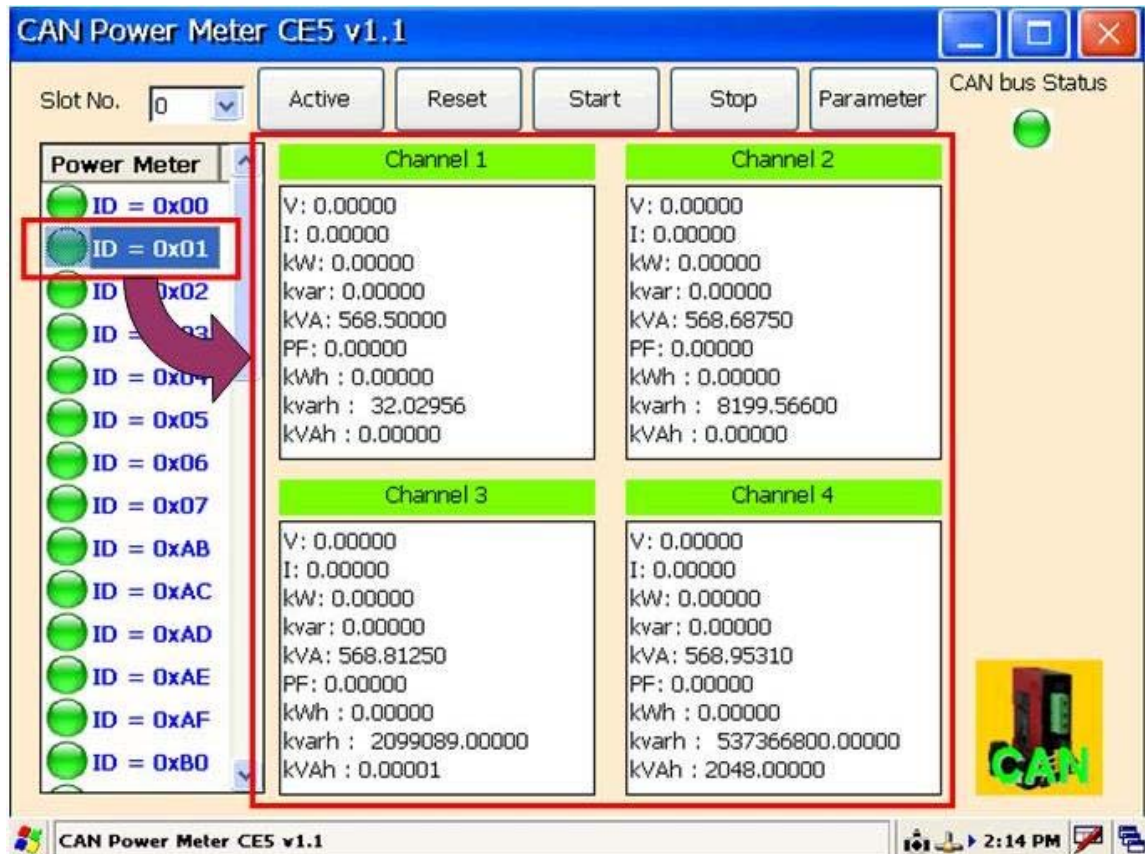
4.4.3 The List of the Power Meter

After clicking the “Parameter” button, the users can see the following picture and select the desired power meter’s ID. After selecting the power meter’s ID, the users need to click the “Set Power Meter” button to save the setting into the I-8120W. The following picture shows the detail steps.



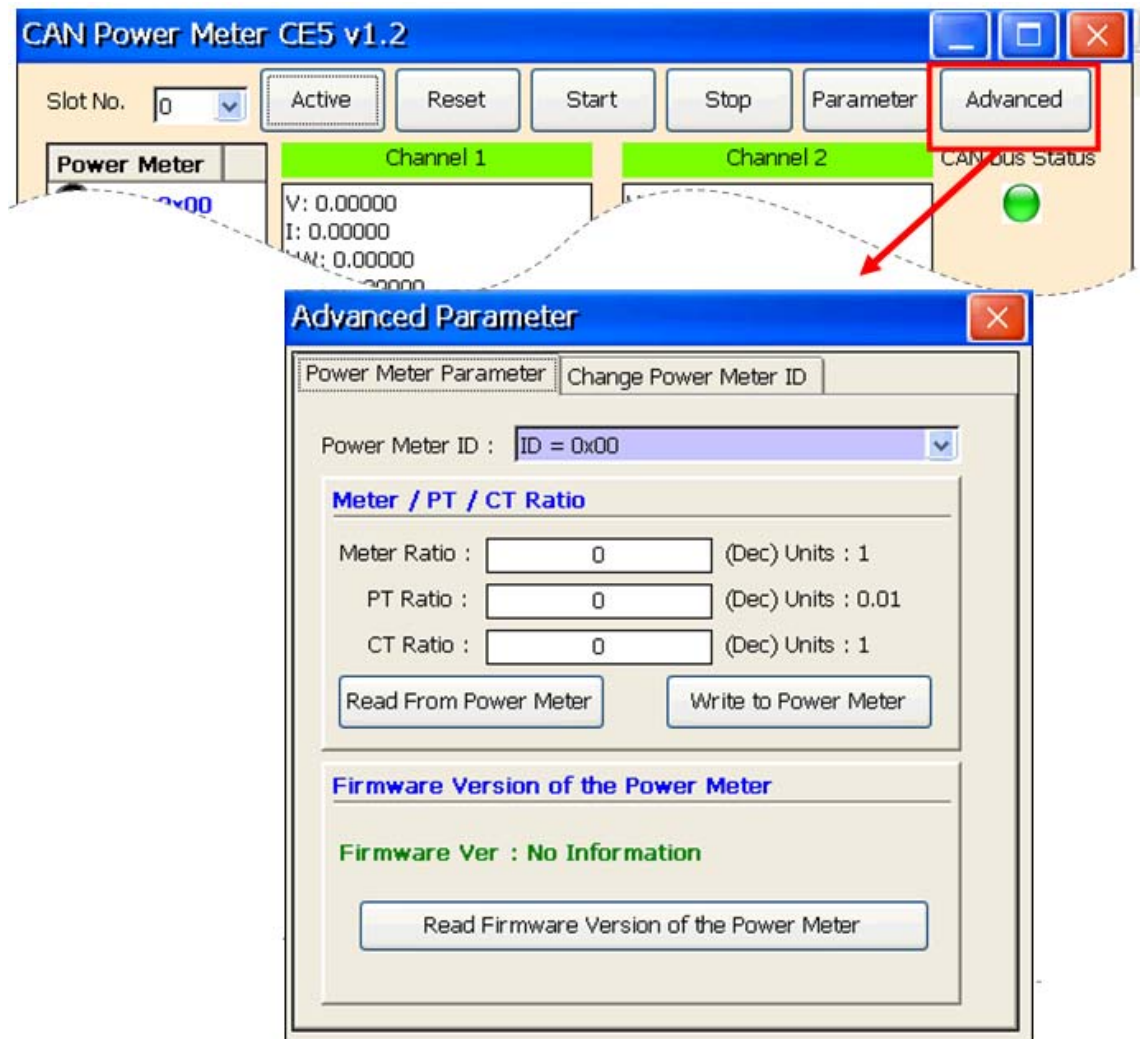
4.5 Display the Real-Time Power Information

If the users have never configured any power meter within the I-8120W, you would not see the following picture. If connecting and configuring the power meter, please select any one power meter in the “Power Meter” filed. The dialog on the right hand side would show the all power information. The picture below shows the operation.



4.6 Advanced Parameter of Power Meter

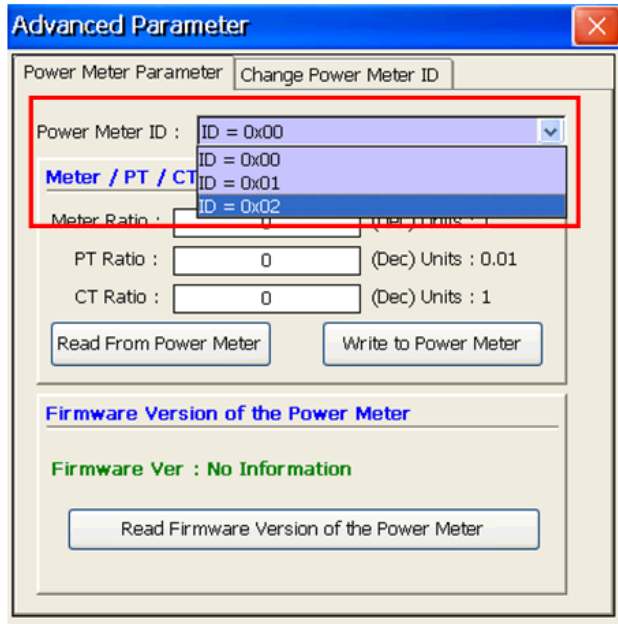
The users can configure the advanced ratio parameters in the power meter by “Advanced Parameter”. At the same time, the users can check the firmware version of the power meter. The users can click the “Advanced” button to show up the “Advanced Parameter” dialog. Here shows the detail steps.



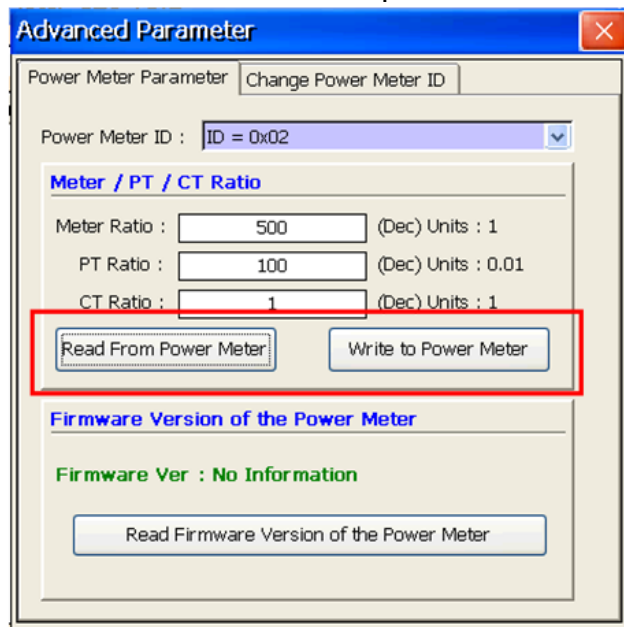
4.6.1 The Ratio Parameters of the Power Meter

After clicking the “Advanced” button, the following dialog is popped up. You can select the desired power meter’s ID to read or write the ratio parameters. Note that the ratio parameters would affect the information of the power measurement. The following picture shows the detail steps.

Step 1 : Select the correct ID of the power meter

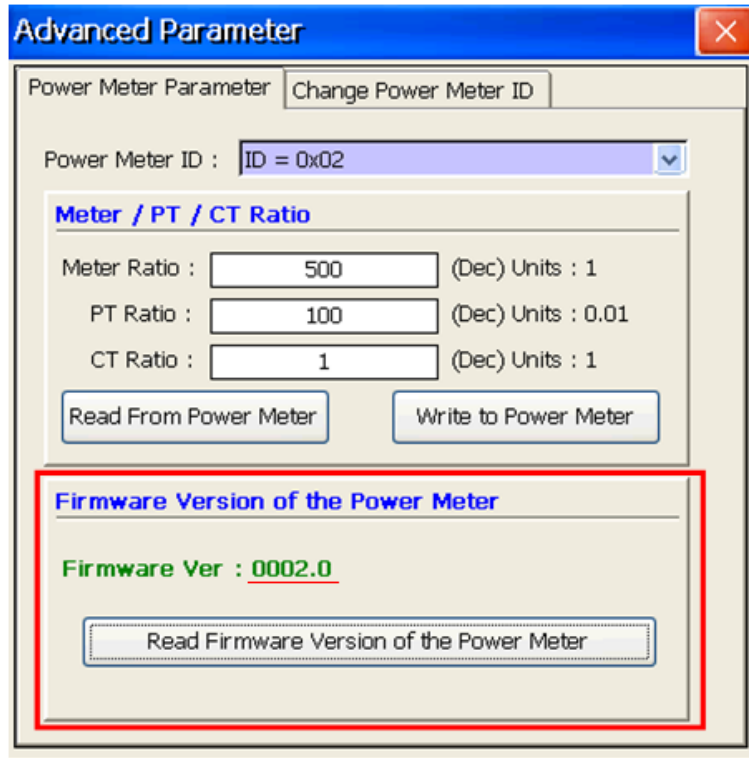


Step 2 : Read or write the ratio parameters of the power meter



4.6.2 The firmware version of the Power Meter

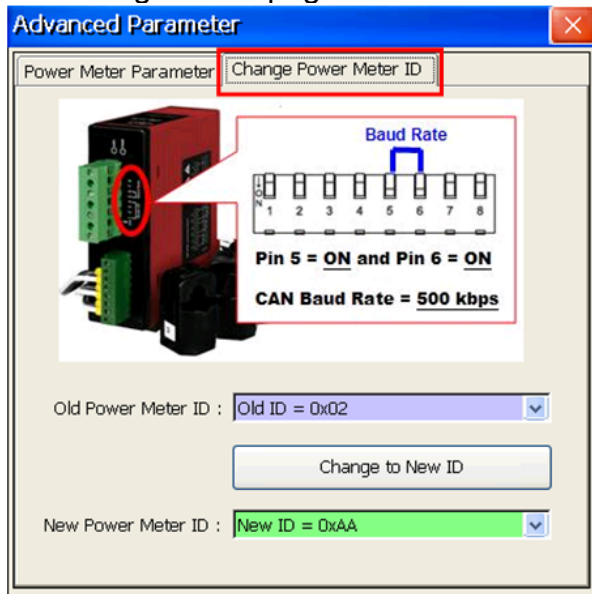
After clicking the “Advanced” button, the users can see the following picture. You can select the desired power meter’s ID to read the firmware version of the power meter. The following picture shows the detail steps.



4.6.3 Change the ID of the Power Meter

After clicking the “Advanced” button and select the “Change Power Meter ID” page, the dialog is popped up as below. Before setting the ID of the power meter, you should change the bit 5 and bit 6 of the DIP switch to be on first. Then, you can configure the ID of this power meter with the 500 kbps baud and the original software ID (or the hardware ID set by DIP switch bit 1 ~ bit 4 if you forget the original software ID). The following picture shows the detail steps.

Step 1 : Change to the page as below.



Step 2 : Click the “Change to New ID” button to change to the new ID.

