The PISO-CM100U'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

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

改版資訊

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

TABLE OF CONTENTS

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.

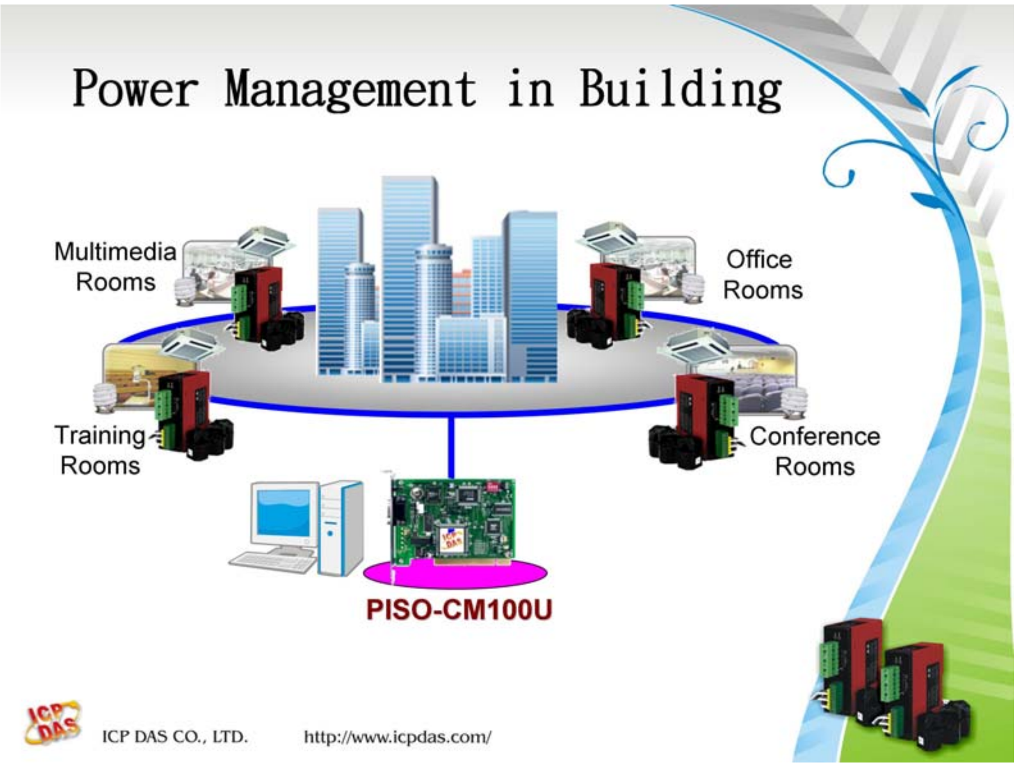
The CAN bus has perfect superiority over other serial communication buses. Here shows the features.

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.

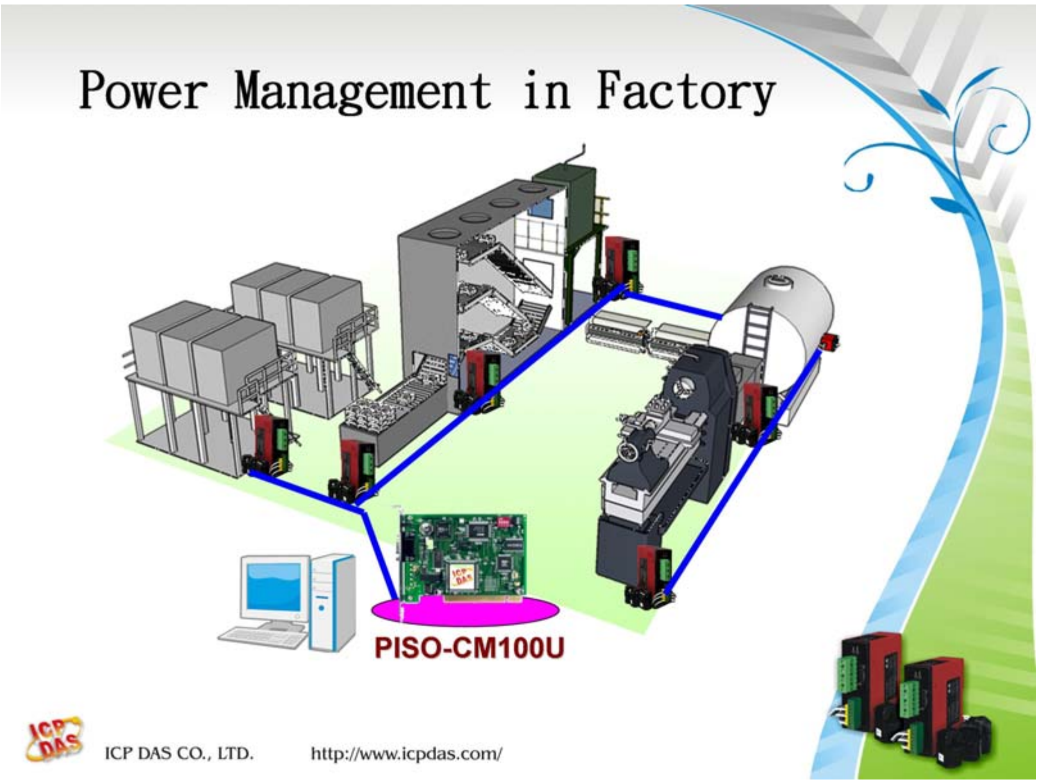
1. Industrial Control.
2. Farming Machine.
3. Aircraft electricity.
4. Vehicle Monitoring.
5. Semiconductor equipment.
6. Biotechnology and medical industry.
7. Reservoir and bridge safety.
8. Military industry.
9. Factory automation.
10. HVAC System and equipment.

1.2 CAN bus Power Meter Applications



Power Management in Building

ICP DAS CO., LTD. http://www.icpdas.com/



Power Management in Factory

ICP DAS CO., LTD. http://www.icpdas.com/

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 ℃ ~ +70 ℃

Illustration of the Polling mode:

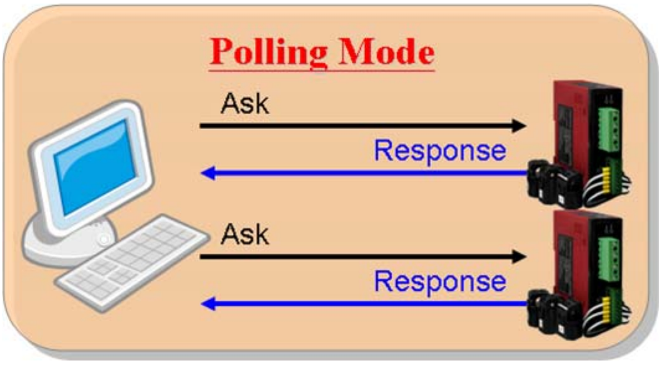
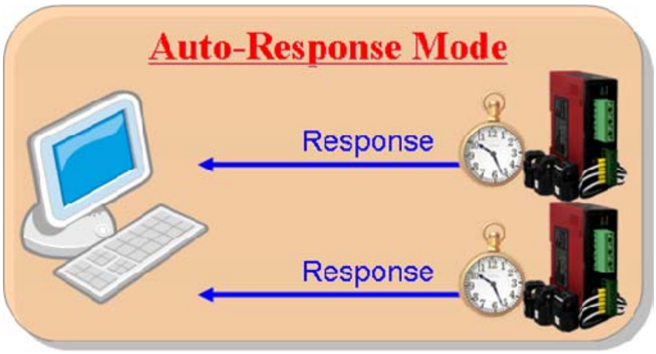


Illustration of the "Auto-Response" mode



PISO-CM100U 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 PISO-CM100U.**
- **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. PISO-CM100U Hardware Configuration

This section will describe the hardware settings of the PISO-CM100U. This information includes the wire connection and terminal resistance configuration.

2.1 PISO-CM100U Board Layout

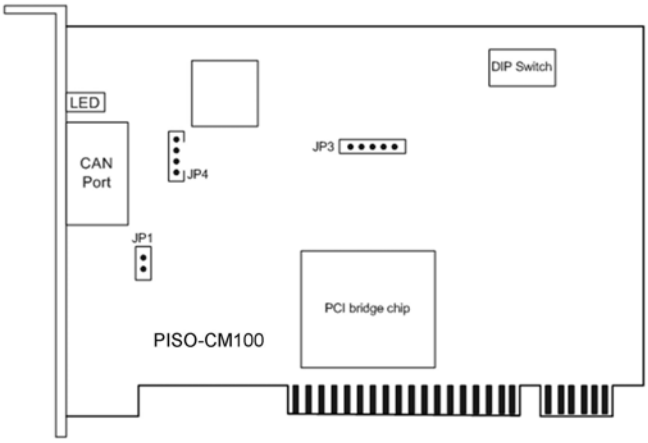


Figure2.1 PISO-CM100 Board Layout

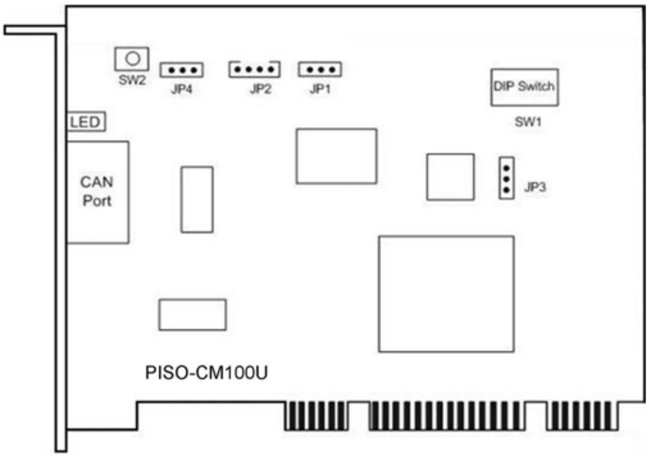


Figure 2.2 PISO-CM100U Board Layout

Note: PISO-CM100-T layout is similar with PISO-CM100-D and PISO-CM100U-T layout is similar with PISO-CM100U-D. The only difference is the position of CAN port connector. The positions of jumper or DIP switch are the same. Therefore, users can also refer to the PISO-CM100/100U-D layout to configure the jumper or DIP switch if they use PISO-CM100/100U-T.

2.2 Jumper Selection

The following table shows the definition of jumpers or DIP switch. Users need to refer to this table to configure the PISO-CM100-D/T and PISO-CM100U-D/T hardware.

PISO-CM100 :

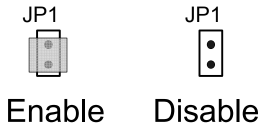
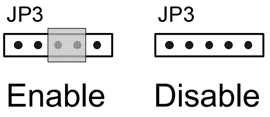
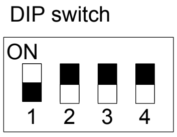
Jumper	Description	Status
JP1	CAN Port 120Ω terminal resistance. "Enable" means CAN port with 120Ω resistance. "Disable" means CAN port without resistance.	
JP3	Reset pin for download error. If users want to update firmware but the process is fail, users can enable this jumper to reset the PISO-CM100-D/T into download mode.	
JP4	Reserved	Reserved
DIP switch	DIP switch is used to set the PISO-CM100 board No. Switch1 is for bit0, switch2 is for bit1 and so forth. For example, if the left-hand-side switch (switch 1) is ON, the board No. is set to 1. The range of board No. is from 0 to 15. Be careful that the board No. for each PISO-CM100-D/T and PISO-CM100U-D/T must be unique in the PC.	 This situation indicates the board No. 1.

Table 2.1 PISO-CM100 Jumper or DIP switch selections

PISO-CM100U :

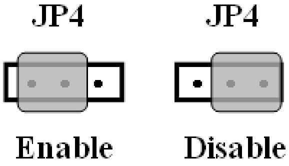
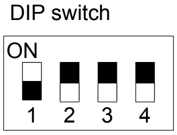
Jumper	Description	Status
JP1 JP2 JP3	Reserved	Reserved
SW2	Reset button for download error. If users want to update firmware but the process is fail, users can click this button to reset the PISO-CM100U into download mode.	
JP4	CAN Port 120Ω terminal resistance. "Enable" means CAN port with 120Ω resistance. "Disable" means CAN port without resistance.	
DIP switch	DIP switch is used to set the PISO-CM100 board No. Switch1 is for bit0, switch2 is for bit1 and so forth. For example, if the left-hand-side switch (switch 1) is ON, the board No. is set to 1. The range of board No. is from 0 to 15. Be careful that the board No. for each PISO-CM100-D/T and PISO-CM100U-D/T must be unique in the PC.	 This situation indicates the board No. 1.

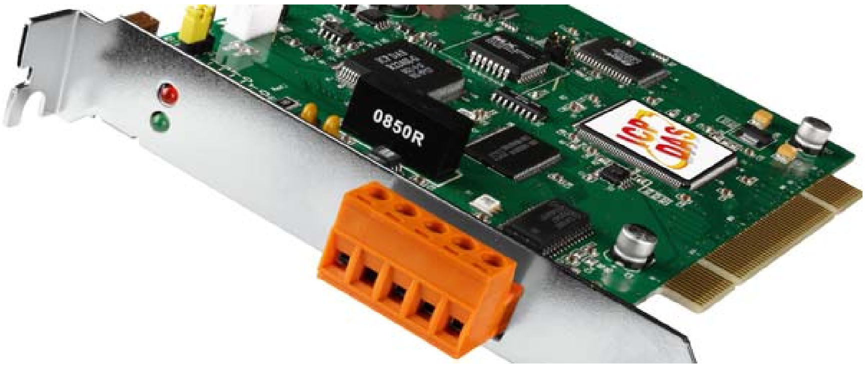
Table 2.2 PISO-CM100U Jumper or DIP switch selections

2.3 CAN Connector Pin Assignment

The PISO-CM100U-T is equipped with one **5-pin screw terminal connector** and the PISO-CM100U-D is equipped with one **9-pin D-sub male connector** for wire connection of the CAN bus. The connector's pin assignment is specified as following:

2.3.1 5-pin screw terminal connector

The 5-pin screw terminal connector of the CAN bus interface is shown in Figure 2.4. The details for the pin assignment are presented in Table 2.2.



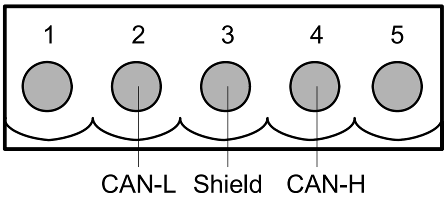


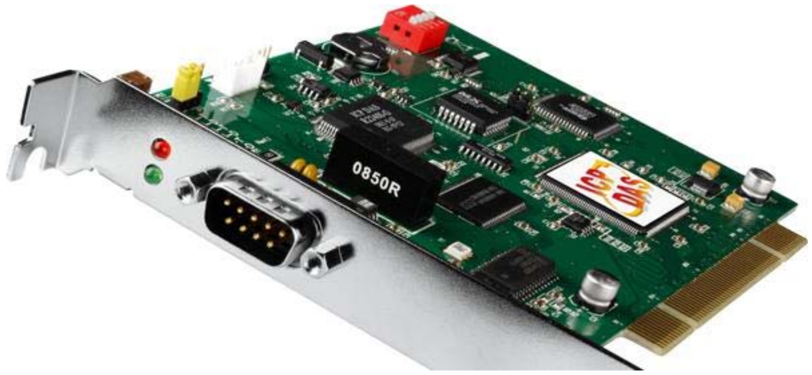
Figure2.4 5-pin screw terminal connector

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

Table 2.2: Pin assignment of 5-pin screw terminal connector

2.3.2 9-pin D-sub male connectors

The 9-pin D-sub male connector of the CAN bus interface is shown in Figure 2.5 and the corresponding pin assignments are given in Table 2.3.



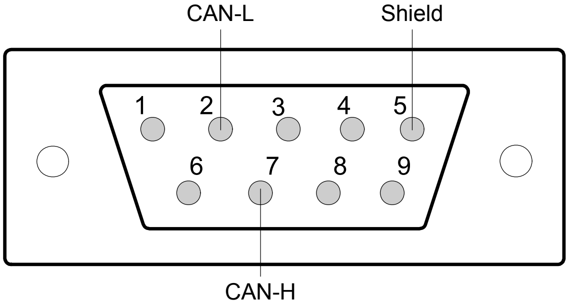


Figure2.5 9-pin D-sub male connector

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

Table 2.3 Pin assignment of the 9-pin D-sub male connector

2.3.3 CAN 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). The length related resistance should have 70 mΩ/m. Users should check the resistances of the CAN bus, before they install a new CAN network.

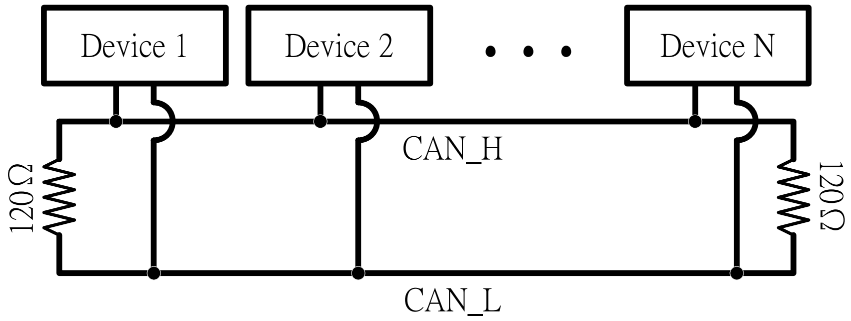


Figure 2.6 CAN bus network topology

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

2.4 LED Indicator

2.4.1 Green LED Indicator

[Green] LED represents the status of the remote power meters.

(1). Green LED solid on：

It represents that all remote power meters are normal.

(2). Green 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 PISO-CM100U function call to realize what happens with those power meters.

2.4.2 Red LED Indicator

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

(1). [Red] LED twinkle：

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

(a) Check if the baud rate between PISO-CM100U 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.

2.5 PISO-CM100U Hardware Installation

When users want to use PISO-CM100U-D/T, the hardware installation needs to be finished as following steps.

1. Shutdown your personal computer.
2. Configure the DIP switch and JP4 of the PISO-CM100U-D/T for the board No. and the terminal resistance. The more detail information could be found on the figure 2.1 and table 2.1.
3. Check JP3 and JP2 status of PISO-CM100U-D/T. If necessary, enable them.
4. Find an empty PCI slot for the PISO-CM100U-D/T on the mother board of the personal computer. Plug the configured PISO-CM100U-D/T into this empty PCI slot.
5. Plug the CAN bus cable(s) into the 5-pin screw terminal connector or the 9-pin D-sub connector.

When the procedure described above is completed, turn on the PC.

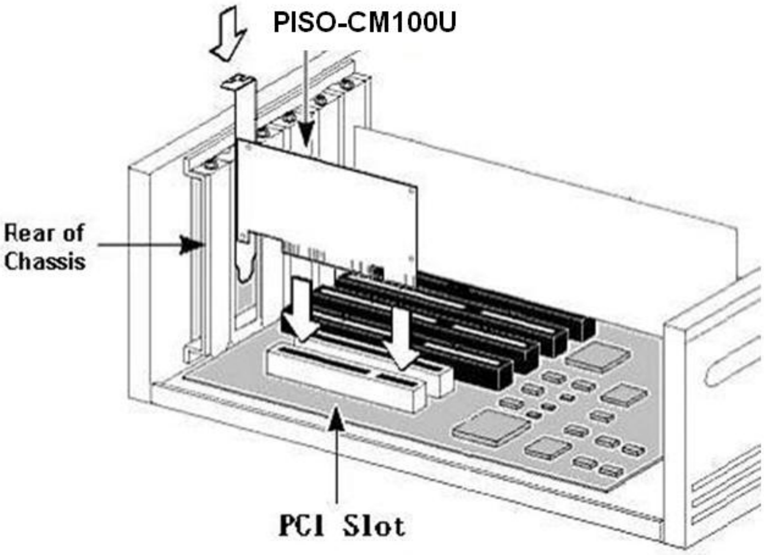


Figure 2.5 PISO-CM100U Installation

3. Software and Firmware Installation

3.1 Install PISO-CM100U Software

The PISO-CM100U DLL for power meter is CM100U_PM.dll. It contains a lot of functionalities of power meter. Based on the driver of the PISO-CM100U, the dll library can work on the Windows 2000/XP and Win 7. The architecture of the driver as show in the Figure 3.1. The users can use the most popular development tools like VB6, Delphi, BCB6, C#, VB.NET and etc. By using those tools and CMP100U_PM.dll, the users can develop efficient and stable power management system.

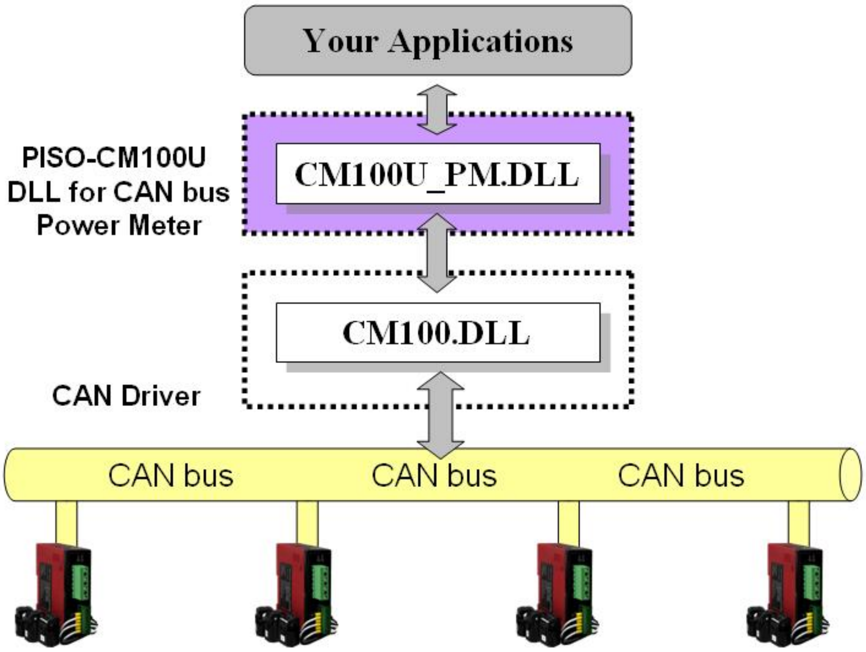


Figure 3.1 The architecture of the windows driver

Users need to get proper driver for their operation system. These drivers are in Field Bus CD in the PISO-CM100U-D/T package. The path is as follow.

Fieldbus CD: CAN\PCI\PISO-CM100U

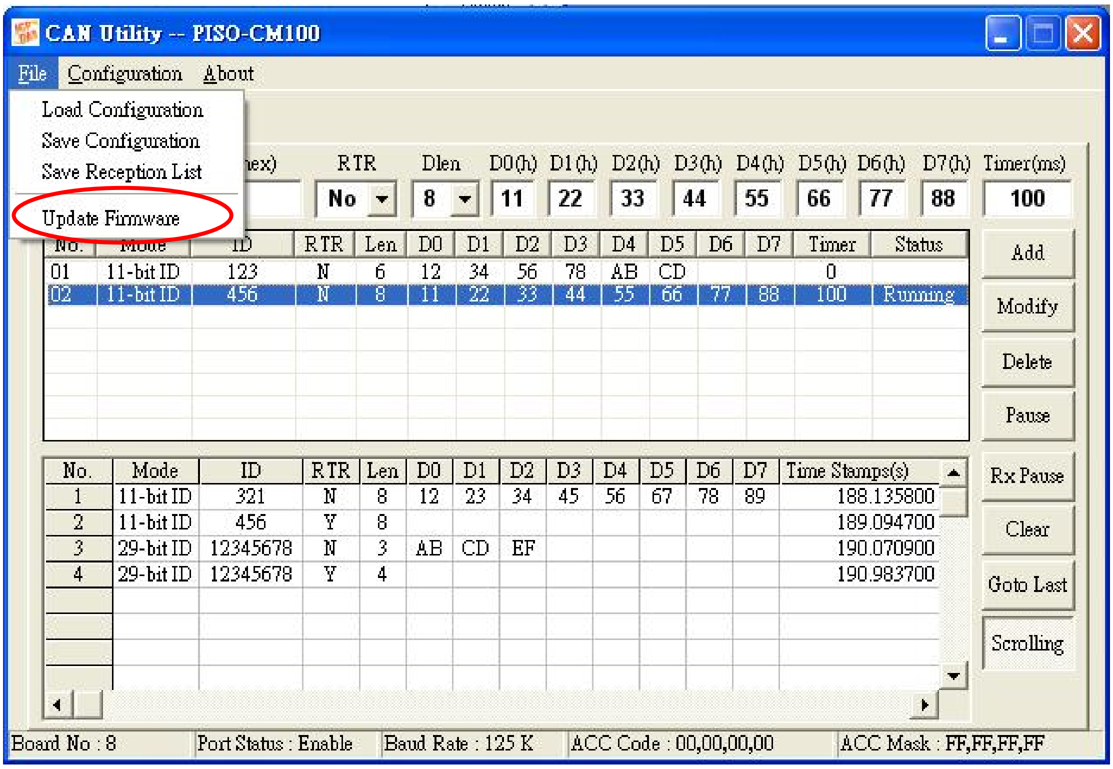
The path is CAN\PCI\PISO-CM100U. Also, users can find them from our website as follow.

http://www.icpdas.com/products/Remote_IO/can_bus/piso-cm100u.htm

The users can refer to the manual of the PISO-CM100U for installation.

3.1 Download the Power Meter Management Firmware

PISO-CM100U software tool which is CANUtility.exe is located in the path C:\ICPDAS\PISO-CM100\CANUtility.exe. After running the ”CANUtility”, the users can find the menu of “File” which is in the left and upper side of the dialog. There is a “Update Firmware” item as shown below. By utilizing this downloading firmware functionality, we can download the power meter management firmware into the PISO-CM100U.



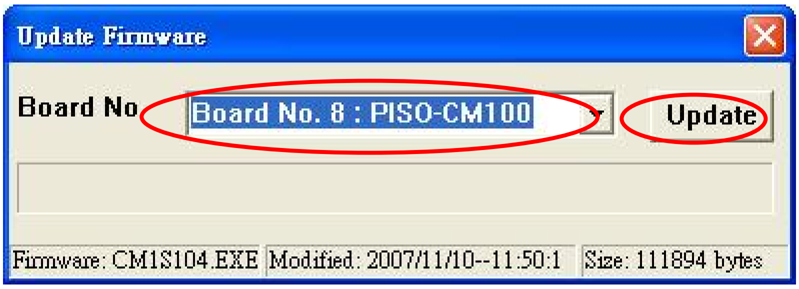
ex)	RTR	Dlen	D0(h)	D1(h)	D2(h)	D3(h)	D4(h)	D5(h)	D6(h)	D7(h)	Timer(ms)
	No	8	11	22	33	44	55	66	77	88	100

No.	Mode	ID	RTR	Len	D0	D1	D2	D3	D4	D5	D6	D7	Timer	Status
01	11-bit ID	123	N	6	12	34	56	78	AB	CD			0	
02	11-bit ID	456	N	8	11	22	33	44	55	66	77	88	100	Running

No.	Mode	ID	RTR	Len	D0	D1	D2	D3	D4	D5	D6	D7	Time Stamps(s)
1	11-bit ID	321	N	8	12	23	34	45	56	67	78	89	188.135800
2	11-bit ID	456	Y	8									189.094700
3	29-bit ID	12345678	N	3	AB	CD	EF						190.070900
4	29-bit ID	12345678	Y	4									190.983700

Board No : 8 | Port Status : Enable | Baud Rate : 125 K | ACC Code : 00,00,00,00 | ACC Mask : FF,FF,FF,FF

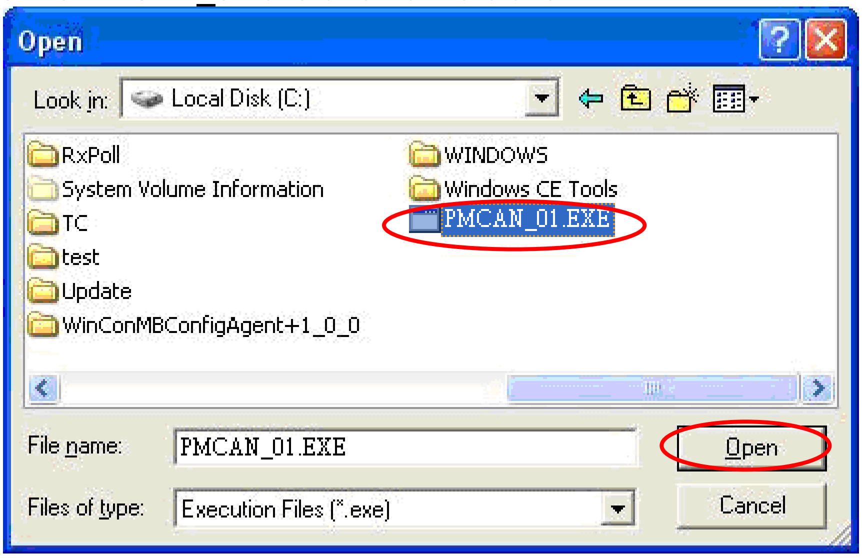
When users apply the Update Firmware function, select the specified board firstly. The PISO-CM100U are listed in the Combo box. Then, click Update button to select the proper firmware for the specified board.



Board No: Board No. 8 : PISO-CM100

Firmware: CM1S104.EXE | Modified: 2007/11/10--11:50:1 | Size: 111894 bytes

At this moment, there is a dialog to select the correct firmware. The firmware is like PMCAN_01.exe or other versions.

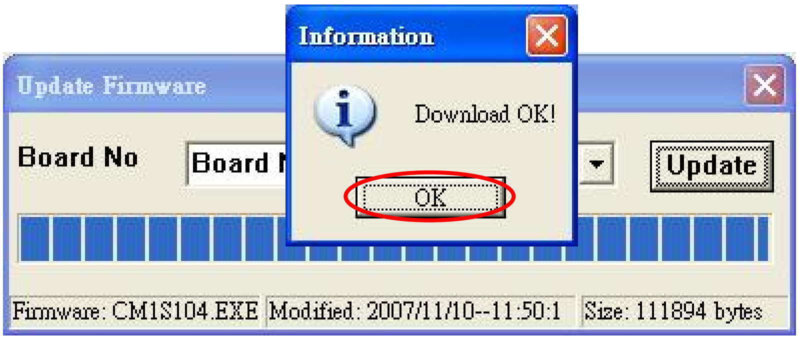


Look in: Local Disk (C:)

File name: PMCAN_01.EXE

Files of type: Execution Files (*.exe)

After selecting the correct firmware and clicking the “Open” button, the download process will start. When finishing the download procedure, the Download OK dialog is popped up. Click OK button to continue.



Download OK!

Firmware: CM1S104.EXE Modified: 2007/11/10--11:50:1 Size: 111894 bytes

After finishing those steps, the firmware has been downloaded into the PISO-CM100U.

4. PISO-CM100U Power Meter SDK Description

4.1 DLL Library Definition and Description

All the functions provided in the CM100U_PM.DLL or CM100PM_DotNET.DLL have detail description in CH4.3. 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

4.2 Function Return Code

Table 4.2.1 Return Code

Return Code	Error ID	Comment
0	CPM100PM_NoError	No Error
1000	CPM100PM_PMIDNotExist	Power Meter ID does not exist
1001	CPM100PM_DataNameError	Data Name is error
1051	PMXS_BaudRateError	Baud Rate is error
10001	CM100PM_DriverError	Driver error
10002	CM100PM_ActiveBoardError	Activate error
10003	CM100PM_BoardNumberError	Board No. error
10004	CM100PM_PortNumberError	CAN port No. error
10007	CM100PM_InitError	Initialize error
10021	CM100PM_SoftBufferIsEmpty	CAN buffer is empty
10022	CM100PM_SoftBufferIsFull	CAN buffer is full
10023	CM100PM_TimeOut	Firmware responses timeout
10024	CM100PM_SetCyclicMsgFailure	Cyclic Message fail
10025	CM100PM_DpramOverRange	DPRAM out of range
10026	CM100PM_NoDpramCmd	DPRAM has no command
10027	CM100PM_ModeError	Firmware is not running.
10030	CM100PM_NoFileInside	Firmware does not exist
10031	CM100PM_DownloadFailure	Loading firmware is fail.
10032	CM100PM_EEPROMDamage	EEPROM is out of order
10033	CM100PM_NotEnoughSpace	EEPROM is full.
10034	CM100PM_StillDownloading	It is downloading firmware
10035	CM100PM_BoardModeError	The board is under download mode.

4.3 Power Meter Management SDK Description

4.3.1 CM100PM_TotalCM100Board

- **Description：**

  The function can get the count of total boards in the PC.

- **Syntax：**

  DWORD CM100PM_TotalCM100Board (BYTE *TotalBoards ,
  BYTE *BoardIDList)

- **Parameter：**

  TotalBoards: [output] Return the amount of total boards.
  BoardIDList: [output] Return the list of all board No.

- **Return：**

  CM100PM_NoError(0)：OK

4.3.2 CM100PM_ActiveBoard

- **Description：**

  The function is used to activate the entire PISO-CM100U board，It must be called once before using the other functions of PISO-CM100U APIs.

- **Syntax：**

  DWORD CM100PM_ActiveBoard (BYTE BoardNo)

- **Parameter：**

  BoardNo: [input] The board number of the PISO-CM100U.

- **Return：**

  CM100PM_NoError(0)：OK
  CM100PM_DriverError(10001)：Driver error.
  CM100PM_ActiveBoardError (10002)：Activate error.
  CM100PM_BoardNumberError(10003)：Board No. error.
  CM100PM_NoFileInside(10030)：PISO-CM100U is without firmware.

4.3.3 CM100PM_CloseBoard

- **Description：**

  The function is used to stop and close the PISO-CM100U driver. This method must be called once to release the resource before exiting the user's application program.

- **Syntax：**

  DWORD CM100PM_CloseBoard (BYTE BoardNo)

- **Parameter：**

  BoardNo: [input] The board number of the PISO-CM100U.

- **Return：**

  CM100PM_NoError(0)：OK
  CM100PM_DriverError(10001)：Driver error.
  CM100PM_ActiveBoardError (10002)：Activate error.
  CM100PM_BoardNumberError(10003)：Board No. error.

4.3.4 CM100PM_GetDLLVersion

- **Description：**

  By using this function, the user can query the version of the CM100U_PM.DLL.

- **Syntax：**

  DWORD CM100PM_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.

4.3.5 CM100PM_GetFirmwareVersion

- **Description：**

  The function can obtain the version information of the firmware inside the PISO-CM100U.

- **Syntax：**

  DWORD CM100PM_GetFirmwareVersion (BYTE BoardNo)

- **Parameter：**

  BoardNo: [input] The board number of the PISO-CM100U.

- **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：**

  CM100PM_NoError(0)：OK
  CM100PM_DriverError(10001)：Driver error.
  CM100PM_ActiveBoardError (10002)：Activate error.
  CM100PM_BoardNumberError(10003)：Board No. error.

4.3.6 CM100PM_ResetFirmware

- **Description：**

  The function is used to reset the PISO-CM100U. 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 CM100PM_ResetFirmware (BYTE BoardNo)

- **Parameter：**

  BoardNo: [input] The board number of the PISO-CM100U.

- **Return：**

  CM100PM_NoError(0)：OK
  CM100PM_DriverError(10001)：Driver error.
  CM100PM_ActiveBoardError (10002)：Activate error.
  CM100PM_BoardNumberError(10003)：Board No. error.

4.3.7 CM100PM_GetPMIDs

- **Description：**

This function is used to get the ID list of all the power meters which are in PISO-CM100U. The users can use this function to know which power meters are in the PISO-CM100U.

- **Syntax：**

```
DWORD CM100PM_GetPMIDs (BYTE BoardNo, BYTE *PMIDCount,
                              BYTE *PMIDArray)
```

- **Parameter：**

BoardNo: [input] The board number of the PISO-CM100U.

PMIDCount: [output] Return the amount of all power meter in the PISO-CM100U.

PMIDArray: [output] Return the ID array of all power meter in the PISO-CM100U.

- **Return：**

CM100PM_NoError(0)：OK
CM100PM_DriverError(10001)：Driver error.
CM100PM_ActiveBoardError (10002)：Activate error.
CM100PM_BoardNumberError(10003)：Board No. error.

4.3.8 CM100PM_SetPMIDs

- **Description：**

This function is used to set the power meters into the PISO-CM100U. The users can save certain power meters into the EEPROM of the PISO-CM100U. The setting does not lose when power is off. One PISO-CM100U 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 CM100PM_SetPMIDs (BYTE BoardNo, BYTE PMIDCount,
                                       BYTE *PMIDArray)
```

- **Parameter：**

BoardNo: [input] The board number of the PISO-CM100U.
PMIDCount: [input] The amount of power meter which will be saved into the PISO-CM100U.
PMIDArray: [input] The ID array of all power meter which will be saved into the PISO-CM100U.

- **Return：**

CM100PM_NoError(0)：OK
CM100PM_DriverError(10001)：Driver error.
CM100PM_ActiveBoardError (10002)：Activate error.
CM100PM_BoardNumberError(10003)：Board No. error.

4.3.9 CM100PM_GetBaudRate

- **Description：**

  This function can help you to get the CAN bus baud rate information of the PISO-CM100U.

- **Syntax：**

  DWORD CM100PM_GetBaudRate (BYTE BoardNo, BYTE *BaudRate)

- **Parameter：**

  BoardNo: [input] The board number of the PISO-CM100U.
  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：**

  CM100PM_NoError(0)：OK
  CM100PM_DriverError(10001)：Driver error.
  CM100PM_ActiveBoardError (10002)：Activate error.
  CM100PM_BoardNumberError(10003)：Board No. error.

4.3.10 CM100PM_SetBaudRate

- **Description：**

  This function can set the CAN bus baud rate of the PISO-CM100U. After calling this function, you must call CM100PM_ResetFirmware to reset the firmware to make change enabled.

- **Syntax：**

  DWORD CM100PM_SetBaudRate (BYTE BoardNo, BYTE BaudRate)

- **Parameter：**

BoardNo: [input] The board number of the PISO-CM100U.
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：**

CM100PM_NoError(0)：OK
PMXS_BaudRateError(1051)：Baud rate parameter is error.
CM100PM_DriverError(10001)：Driver error.
CM100PM_ActiveBoardError (10002)：Activate error.
CM100PM_BoardNumberError(10003)：Board No. error.

4.3.11 CM100PM_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 CM100PM_StartAutoResp or CM100PM_ResetFirmware to make the new setting enable. The PISO-CM100U will send command to all power meter for this new setting.

- **Syntax：**

```
DWORD CM100PM_SetAutoRespTime (BYTE BoardNo,
                                                DWORD AutoRespTime)
```

- **Parameter：**

BoardNo: [input] The board number of the PISO-CM100U.
AutoRespTime: [input] The automatic response time in millisecond.

- **Return：**

CM100PM_NoError(0)：OK
CM100PM_DriverError(10001)：Driver error.
CM100PM_ActiveBoardError (10002)：Activate error.
CM100PM_BoardNumberError(10003)：Board No. error.

4.3.12 CM100PM_GetAutoRespTime

- **Description：**

  This function could help to obtain the value of the automatic response time in the PISO-CM100U.

- **Syntax：**

  DWORD CM100PM_GetAutoRespTime (BYTE BoardNo,
  DWORD *AutoRespTime)

- **Parameter：**

  BoardNo: [input] The board number of the PISO-CM100U.
  AutoRespTime: [output] Return the automatic response time in millisecond.

- **Return：**

  CM100PM_NoError(0)：OK
  CM100PM_DriverError(10001)：Driver error.
  CM100PM_ActiveBoardError (10002)：Activate error.
  CM100PM_BoardNumberError(10003)：Board No. error.

4.3.13 CM100PM_StartAutoResp

- **Description：**

  This function could make all power meters to response power information automatically. When calling this function, the PISO-CM100U 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 CM100PM_StartAutoResp (BYTE BoardNo)

- **Parameter：**

  BoardNo: [input] The board number of the PISO-CM100U.

- **Return：**

  CM100PM_NoError(0)：OK
  CM100PM_DriverError(10001)：Driver error.
  CM100PM_ActiveBoardError (10002)：Activate error.
  CM100PM_BoardNumberError(10003)：Board No. error.

4.3.14 CM100PM_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 CM100PM_StartAutoResp again.

- **Syntax：**

```
DWORD CM100PM_StopAutoResp (BYTE BoardNo)
```

- **Parameter：**

BoardNo: [input] The board number of the PISO-CM100U.

- **Return：**

CM100PM_NoError(0)：OK
CM100PM_DriverError(10001)：Driver error.
CM100PM_ActiveBoardError (10002)：Activate error.
CM100PM_BoardNumberError(10003)：Board No. error.

4.3.15 CM100PM_GetCANStatus

- **Description：**

  This function is used to get the current status of the PISO-CM100U.

- **Syntax：**

  DWORD CM100PM_GetCANStatus (BYTE BoardNo, BYTE *CANStatus)

- **Parameter：**

  BoardNo: [input] The board number of the PISO-CM100U.
  CANStatus: [output] Return the status of the PISO-CM100U.

  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：**

  CM100PM_NoError(0)：OK
  CM100PM_DriverError(10001)：Driver error.
  CM100PM_ActiveBoardError (10002)：Activate error.
  CM100PM_BoardNumberError(10003)：Board No. error.

4.3.16 CM100PM_GetPowerMeterStatus

- **Description：**

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

- **Syntax：**

```
DWORD CM100PM_GetPowerMeterStatus (BYTE BoardNo,
                                   BYTE PMID, BYTE *Status)
```

- **Parameter：**

BoardNo: [input] The board number of the PISO-CM100U.
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：**

CM100PM_NoError(0)：OK
CM100PM_PMIDNotExist(1000)：The ID of the power meter does not exist in the PISO-CM100U.
CM100PM_DriverError(10001)：Driver error.
CM100PM_ActiveBoardError (10002)：Activate error.
CM100PM_BoardNumberError(10003)：Board No. error.

4.3.17 CM100PM_ReadData

- **Description：**

  This function is used to read certain power information.

- **Syntax：**

  DWORD CM100PM_ReadData (BYTE BoardNo, BYTE PMID,
  WORD DataName, float *fData)

- **Parameter：**

  BoardNo: [input] The board number of the PISO-CM100U.
  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：**

  CM100PM_NoError(0)：OK
  CM100PM_PMIDNotExist(1000)：The ID of the power meter does not exist in the PISO-CM100U.
  CM100PM_DriverError(10001)：Driver error.
  CM100PM_ActiveBoardError (10002)：Activate error.
  CM100PM_BoardNumberError(10003)：Board No. error.

- **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.

4.3.18 CM100PM_ReadAllData_BYTE

- **Description：**

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

- **Syntax：**

  DWORD CM100PM_ReadAllData_BYTE (BYTE BoardNo, BYTE PMID,
  WORD *DataLen, BYTE *DataArray)

- **Parameter：**

  BoardNo: [input] The board number of the PISO-CM100U.

  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：**

  CM100PM_NoError(0)：OK

  CM100PM_PMIDNotExist(1000)：The ID of the power meter does not exist in the PISO-CM100U.

  CM100PM_DriverError(10001)：Driver error.

  CM100PM_ActiveBoardError (10002)：Activate error.

  CM100PM_BoardNumberError(10003)：Board No. error.

- **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				

4.3.19 CM100PM_ReadAllData_FLOAT

- **Description：**

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

- **Syntax：**

DWORD CM100PM_ReadAllData_FLOATT(BYTE BoardNo,
BYTE PMID, WORD *DataLen, float *fDataArray)

- **Parameter：**

BoardNo: [input] The board number of the PISO-CM100U.

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：**

CM100PM_NoError(0)：OK

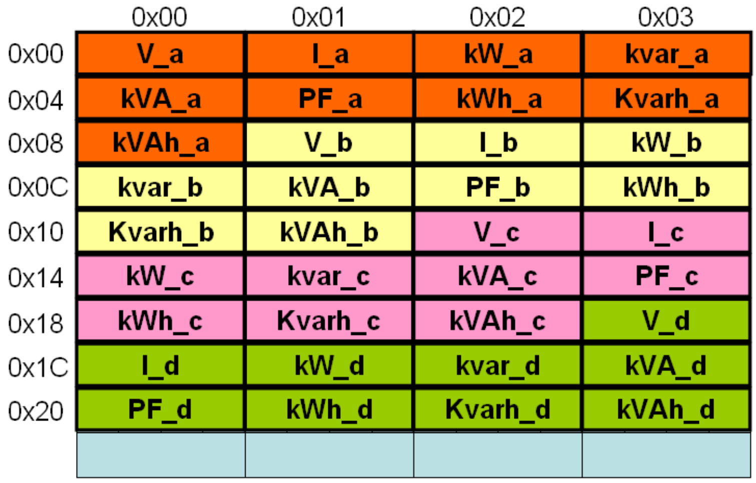
CM100PM_PMIDNotExist(1000)：The ID of the power meter does not exist in the PISO-CM100U.

CM100PM_DriverError(10001)：Driver error.

CM100PM_ActiveBoardError (10002)：Activate error.

CM100PM_BoardNumberError(10003)：Board No. error.

- **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

4.3.20 CM100PM_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 CM100PM_GetRatioValue function to read the ratio value replied from the power meter.

- **Syntax：**

```
DWORD CM100PM_GetRatioCMD (BYTE BoardNo,  BYTE PMID)
```

- **Parameter：**

BoardNo: [input] The board number of the PISO-CM100U.
PMID: [input] The ID of the power meter.

- **Return：**

CM100PM_NoError(0)：OK
CM100PM_PMIDNotExist(1000)：The ID of the power meter does not exist in the PISO-CM100U.
CM100PM_DriverError(10001)：Driver error.
CM100PM_ActiveBoardError (10002)：Activate error.
CM100PM_BoardNumberError(10003)：Board No. error.

4.3.21 CM100PM_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 CM100PM_GetRatioCMD first. Then, users can get the last ratio values correctly.

- **Syntax：**

  DWORD CM100PM_GetRatioValue (BYTE BoardNo, BYTE PMID,
  WORD *MeterRatio, WORD *PTRatio, WORD *CTRatio)

- **Parameter：**

  BoardNo: [input] The board number of the PISO-CM100U.
  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：**

  CM100PM_NoError(0)：OK
  CM100PM_PMIDNotExist(1000)：The ID of the power meter does not exist in the PISO-CM100U.
  RatioError_NoGetCMD(1010) : No “Get Ratio” command was sent.
  RatioError_NoResp(1012) : There is no response message from the power meter.
  CM100PM_DriverError(10001)：Driver error.
  CM100PM_ActiveBoardError (10002)：Activate error.
  CM100PM_BoardNumberError(10003)：Board No. error.

4.3.22 CM100PM_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 CM100PM_SetRatioResp function to read the response replied from the power meter to confirm if the setting is ok or not.

- **Syntax：**

```
DWORD CM100PM_SetRatioCMD (BYTE BoardNo,   BYTE PMID,
                    WORD MeterRatio, WORD PTRatio, WORD CTRatio)
```

- **Parameter：**

  BoardNo: [input] The board number of the PISO-CM100U.
  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：**

  CM100PM_NoError(0)：OK
  CM100PM_PMIDNotExist(1000)：The ID of the power meter does not exist in the PISO-CM100U.
  CM100PM_DriverError(10001)：Driver error.
  CM100PM_ActiveBoardError (10002)：Activate error.
  CM100PM_BoardNumberError(10003)：Board No. error.

4.3.23 CM100PM_SetRatioResp

- **Description：**

After calling CM100PM_SetRatioCMD, the function can read the response values which have been replied from the power meter. Before using this function, users need to call CM100PM_SetRatioCMD first. Then, the users can get the last response correctly.

- **Syntax：**

```
DWORD CM100PM_SetRatioResp (BYTE BoardNo,   BYTE PMID,
                            WORD *RespValue)
```

- **Parameter：**

BoardNo: [input] The board number of the PISO-CM100U.
PMID: [input] The ID of the power meter.
RespValue: [output] The meter ratio value of the power meter.

- **Return：**

CM100PM_NoError(0)：OK
CM100PM_PMIDNotExist(1000)：The ID of the power meter does not exist in the PISO-CM100U.
RatioError_NoSetCMD(1011) : No “Set Ratio” command was sent.
RatioError_NoResp(1012) : There is no response message from the power meter.
CM100PM_DriverError(10001)：Driver error.
CM100PM_ActiveBoardError (10002)：Activate error.
CM100PM_BoardNumberError(10003)：Board No. error.

4.3.24 CM100PM_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 CM100PM_GetPMFirmVerValue function to read the response firmware version replied from the power meter.

- **Syntax：**

  DWORD CM100PM_GetPMFirmVerCMD (BYTE BoardNo,　BYTE PMID)

- **Parameter：**

  BoardNo: [input] The board number of the PISO-CM100U.
  PMID: [input] The ID of the power meter.

- **Return：**

  CM100PM_NoError(0)：OK
  CM100PM_PMIDNotExist(1000)：The ID of the power meter does not exist in the PISO-CM100U.
  CM100PM_DriverError(10001)：Driver error.
  CM100PM_ActiveBoardError (10002)：Activate error.
  CM100PM_BoardNumberError(10003)：Board No. error.

4.3.25 CM100PM_GetPMFirmVerValue

- **Description：**

After calling the function CM100PM_GetPMFirmVerCMD, the specific power meter will reply the result of the processed command. This response will be kept in the PISO-CM100U. Therefore, calling the function CM100PM_GetPMFirmVerValue can read the response stored in the PISO-CM100U to confirm if the setting is ok or not. Before using this function, users need to call the function CM100PM_GetPMFirmVerCMD first. Then, users can get the firmware version of the specific power meter correctly.

- **Syntax：**

```
DWORD CM100PM_GetPMFirmVerValue (BYTE BoardNo,BYTE PMID,
                                 WORD *PMFirmVer)
```

- **Parameter：**

BoardNo: [input] The board number of the PISO-CM100U.
PMID: [input] The ID of the power meter.
PMFirmVer: [output] Return the firmware version of the power meter.

- **Return：**

CM100PM_NoError(0)：OK
CM100PM_PMIDNotExist(1000)：The ID of the power meter does not exist in the PISO-CM100U.
FirmError_NoGetCMD(1020) : No “Get Firm Ver” command was sent.
FirmError_NoResp(1021) : There is no response message from the power meter.
CM100PM_DriverError(10001)：Driver error.
CM100PM_ActiveBoardError (10002)：Activate error.
CM100PM_BoardNumberError(10003)：Board No. error.

4.3.26 CM100PM_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 CM100PM_SetNewPMIDResp function to read the response replied from the power meter.

- **Syntax：**

```
DWORD CM100PM_SetNewPMIDCMD (BYTE BoardNo,
                             BYTE OldPMID, BYTE NewPMID)
```

- **Parameter：**

BoardNo: [input] The board number of the PISO-CM100U.
OldPMID: [input] The original ID of the power meter.
NewPMID: [input] The desired new ID of the power meter.

- **Return：**

CM100PM_NoError(0)：OK
CM100PM_PMIDNotExist(1000)：The ID of the power meter does not exist in the PISO-CM100U.
CM100PM_DriverError(10001)：Driver error.
CM100PM_ActiveBoardError (10002)：Activate error.
CM100PM_BoardNumberError(10003)：Board No. error.

4.3.27 CM100PM_SetNewPMIDResp

- **Description：**

After calling CM100PM_SetNewPMIDCMD, the specific power meter will reply the result of the processed command. This response will be kept in the PISO-CM100U. Therefore, calling the function CM100PM_SetNewPMIDResp can read the response stored in the PISO-CM100U to confirm if the setting is ok or not. Before using this function, users need to call the function CM100PM_SetNewPMIDCMD first.

- **Syntax：**

```
DWORD CM100PM_SetNewPMIDResp (BYTE BoardNo,
                              BYTE NewPMID, DWORD *RespValue)
```

- **Parameter：**

BoardNo: [input] The board number of the PISO-CM100U.
NewPMID: [input] The new ID of the power meter.
RespValue: [output] Return the response value which has been replied from the power meter.

- **Return：**

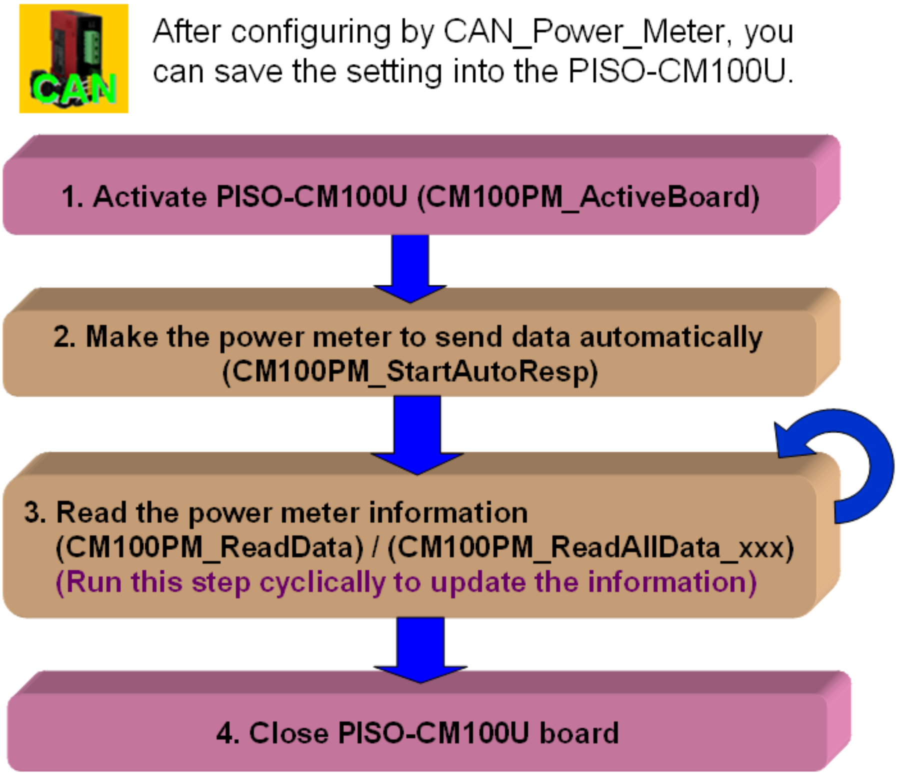
CM100PM_NoError(0)：OK
CM100PM_PMIDNotExist(1000)：The ID of the power meter does not exist in the PISO-CM100U.
PMError_NoSetCMD(1030) : No ”Set New ID” command was sent.
PMError_NoResp(1031) : There is no response message from the power meter.
CM100PM_DriverError(10001)：Driver error.
CM100PM_ActiveBoardError (10002)：Activate error.
CM100PM_BoardNumberError(10003)：Board No. error.

4.4 The Flow Chart of the Development

4.4.1 The Flow Chart of the Fast Development

The users can use CAN_Power_Meter software utility to configure the response time and other parameters. After configuring those parameters, the settings would be saved into the PISO-CM100U. 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 CAN_Power_Meter, you can save the setting into the PISO-CM100U.

1. Activate PISO-CM100U (CM100PM_ActiveBoard)

2. Make the power meter to send data automatically (CM100PM_StartAutoResp)

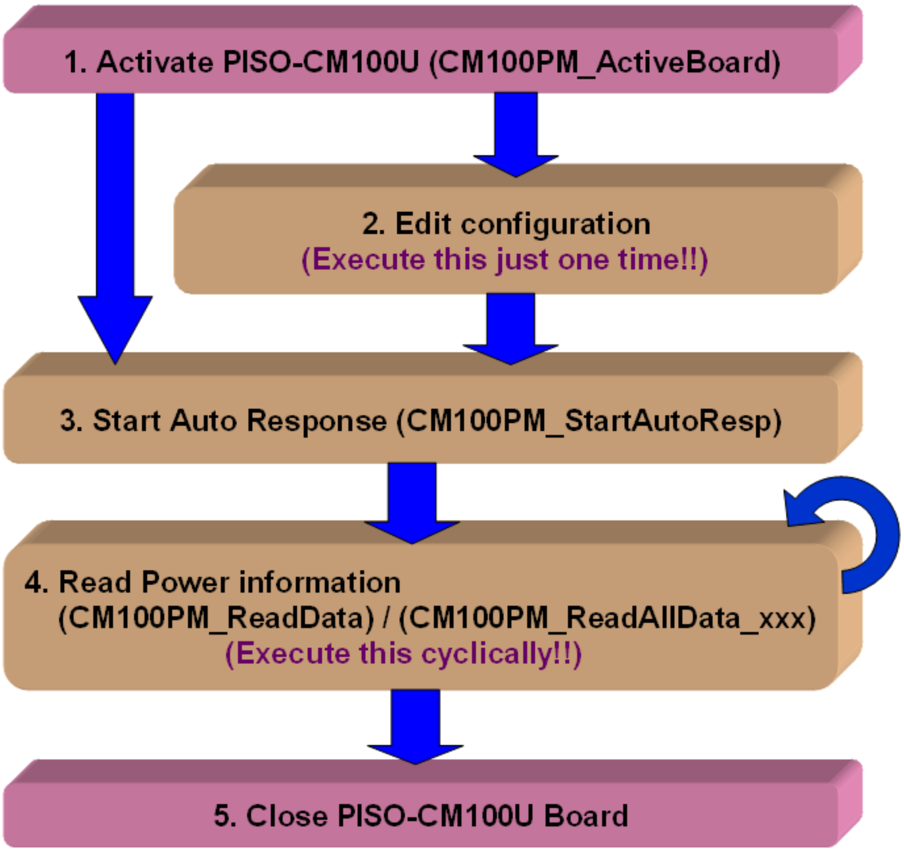
3. Read the power meter information (CM100PM_ReadData) / (CM100PM_ReadAllData_xxx) (Run this step cyclically to update the information)

4. Close PISO-CM100U board

4.4.2 The Flow Chart of the General Development

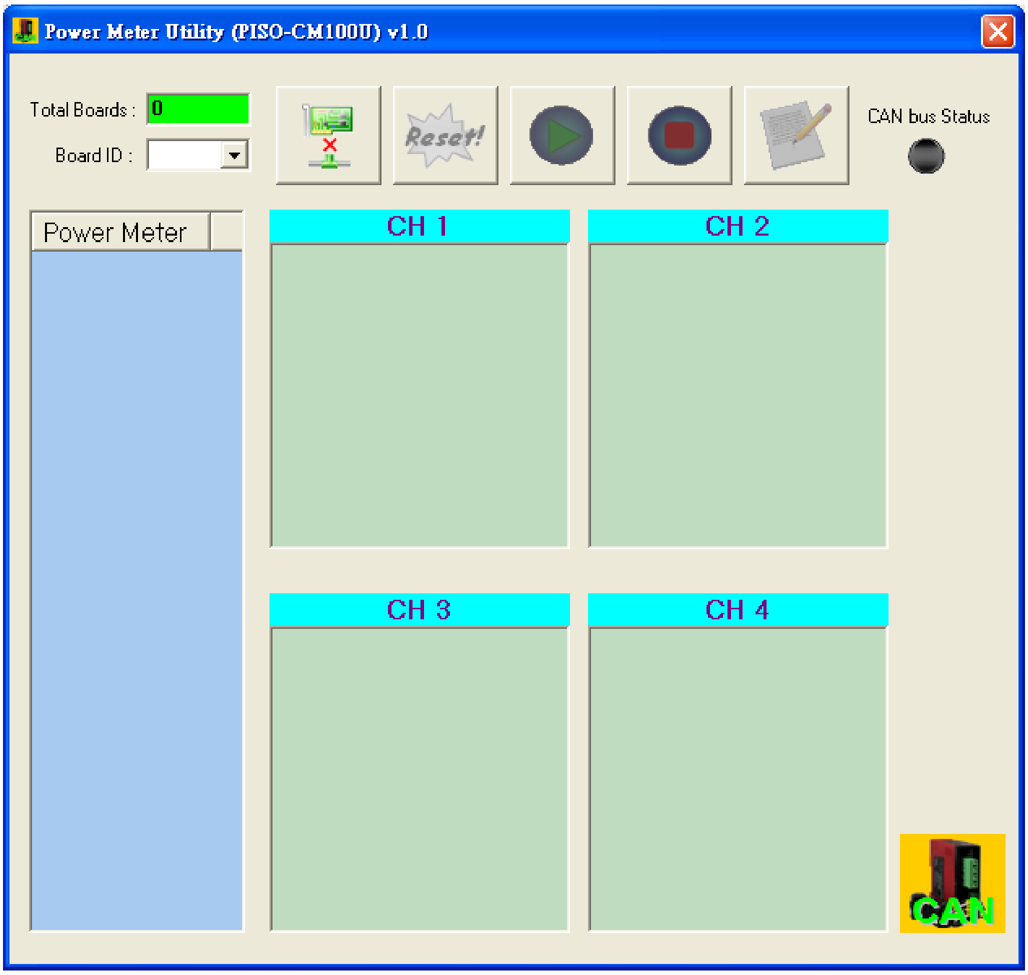
The users can design your own software including the configuration of the automatic response time and those parameters. The users can design your own interface and operation method. We show the flow chart of the general development as below.

General Operation


1. Activate PISO-CM100U (CM100PM_ActiveBoard)
2. Edit configuration
(Execute this just one time!!)
3. Start Auto Response (CM100PM_StartAutoResp)
4. Read Power information
(CM100PM_ReadData) / (CM100PM_ReadAllData_xxx)
(Execute this cyclically!!)
5. Close PISO-CM100U Board

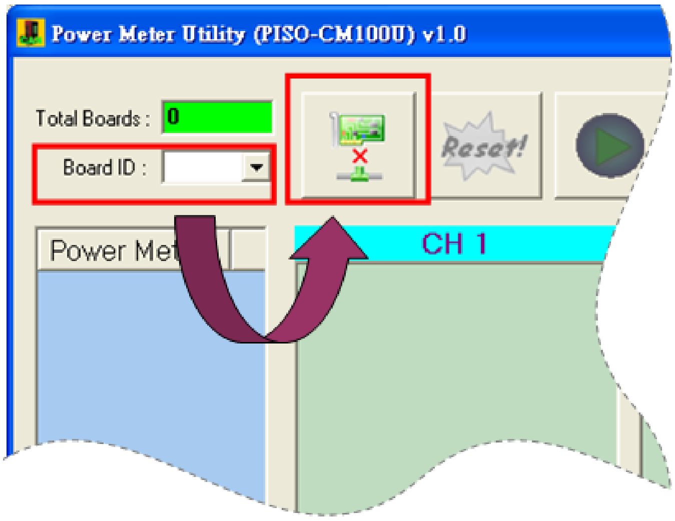
5. CAN_Power_Meter Software Utility

CAN_Power_Meter manages CAN bus power meters. The utility need to work with the PISO-CM100U which has loaded power meter management firmware. The utility interface looks like the following picture when it has started.



5.1 Activate PISO-CM100U Board

Please check the board No. of the PISO-CM100U. The users can select the correct number in the field of ”Board ID”. And then, click the “Active icon” button to active the PISO-CM100U. The following picture shows the detail steps.

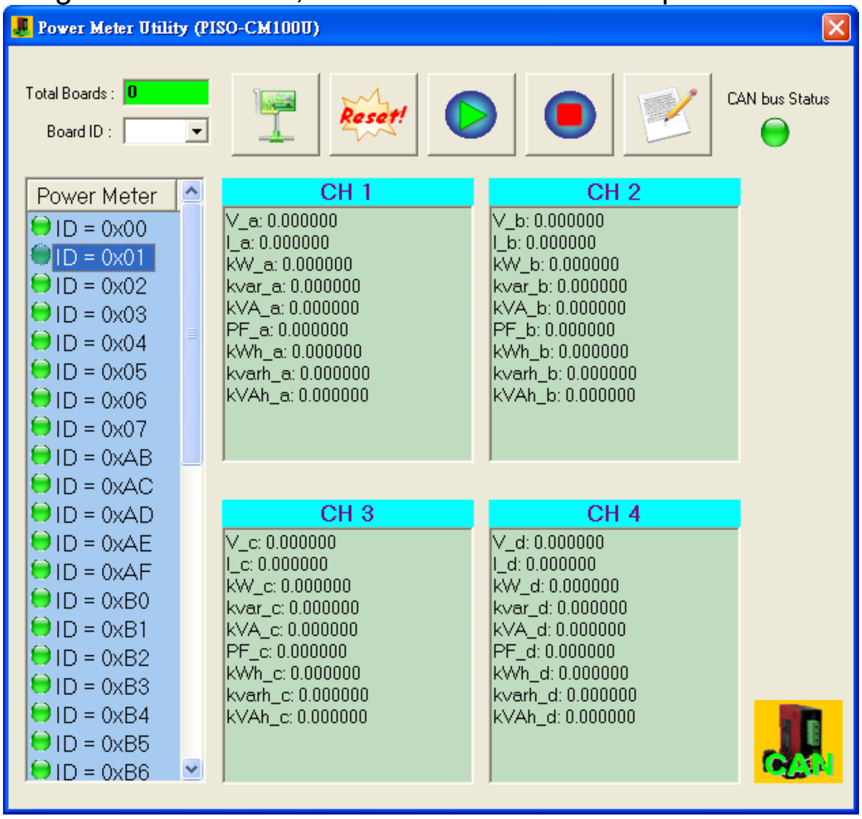


Power Meter Utility (PISO-CM100U) v1.0

Total Boards : 0

Board ID :

After activating PISO-CM100U, the screen would be the picture below.



Power Meter Utility (PISO-CM100U)

Total Boards : 0

Board ID :

CAN bus Status

Power Meter

ID = 0x00
ID = 0x01
ID = 0x02
ID = 0x03
ID = 0x04
ID = 0x05
ID = 0x06
ID = 0x07
ID = 0xAB
ID = 0xAC
ID = 0xAD
ID = 0xAE
ID = 0xAF
ID = 0xB0
ID = 0xB1
ID = 0xB2
ID = 0xB3
ID = 0xB4
ID = 0xB5
ID = 0xB6

CH 1

V_a: 0.000000
I_a: 0.000000
kW_a: 0.000000
kvar_a: 0.000000
kVA_a: 0.000000
PF_a: 0.000000
kWh_a: 0.000000
kvarh_a: 0.000000
kVAh_a: 0.000000

CH 2

V_b: 0.000000
I_b: 0.000000
kW_b: 0.000000
kvar_b: 0.000000
kVA_b: 0.000000
PF_b: 0.000000
kWh_b: 0.000000
kvarh_b: 0.000000
kVAh_b: 0.000000

CH 3

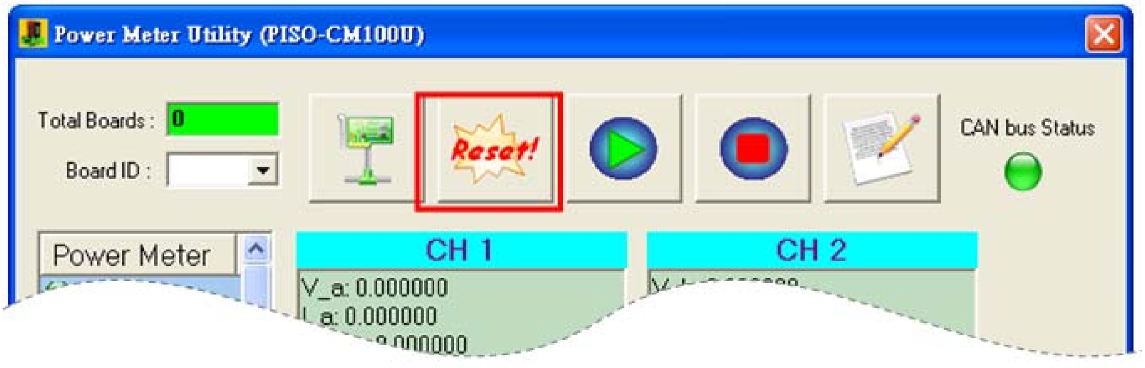
V_c: 0.000000
I_c: 0.000000
kW_c: 0.000000
kvar_c: 0.000000
kVA_c: 0.000000
PF_c: 0.000000
kWh_c: 0.000000
kvarh_c: 0.000000
kVAh_c: 0.000000

CH 4

V_d: 0.000000
I_d: 0.000000
kW_d: 0.000000
kvar_d: 0.000000
kVA_d: 0.000000
PF_d: 0.000000
kWh_d: 0.000000
kvarh_d: 0.000000
kVAh_d: 0.000000

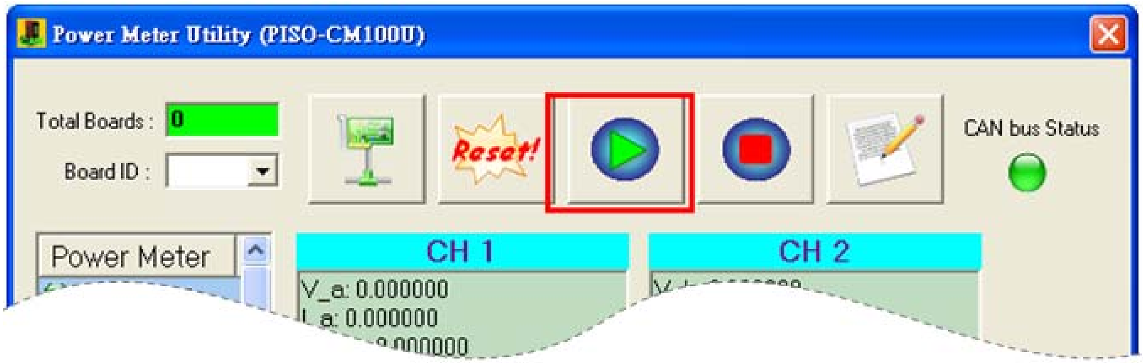
5.2 Reset the Firmware in the PISO-CM100U

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.

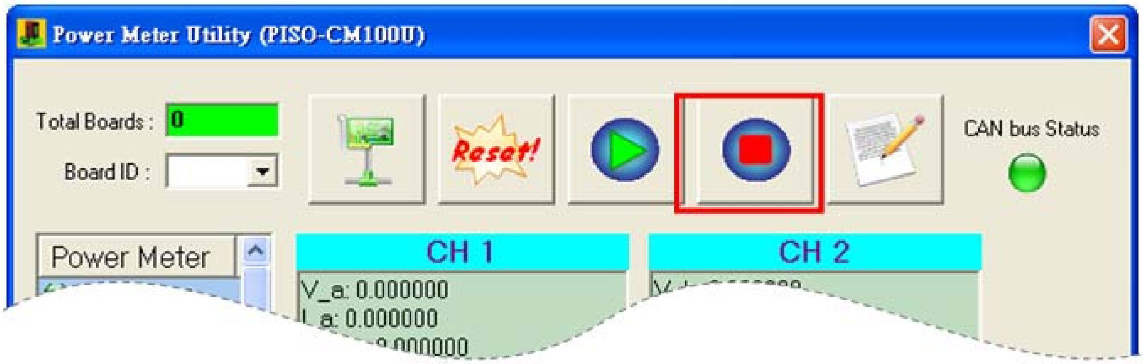


5.3 Start / Stop Automatic Response

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 icon” button to achieve this functionality. The picture below shows the detail steps.

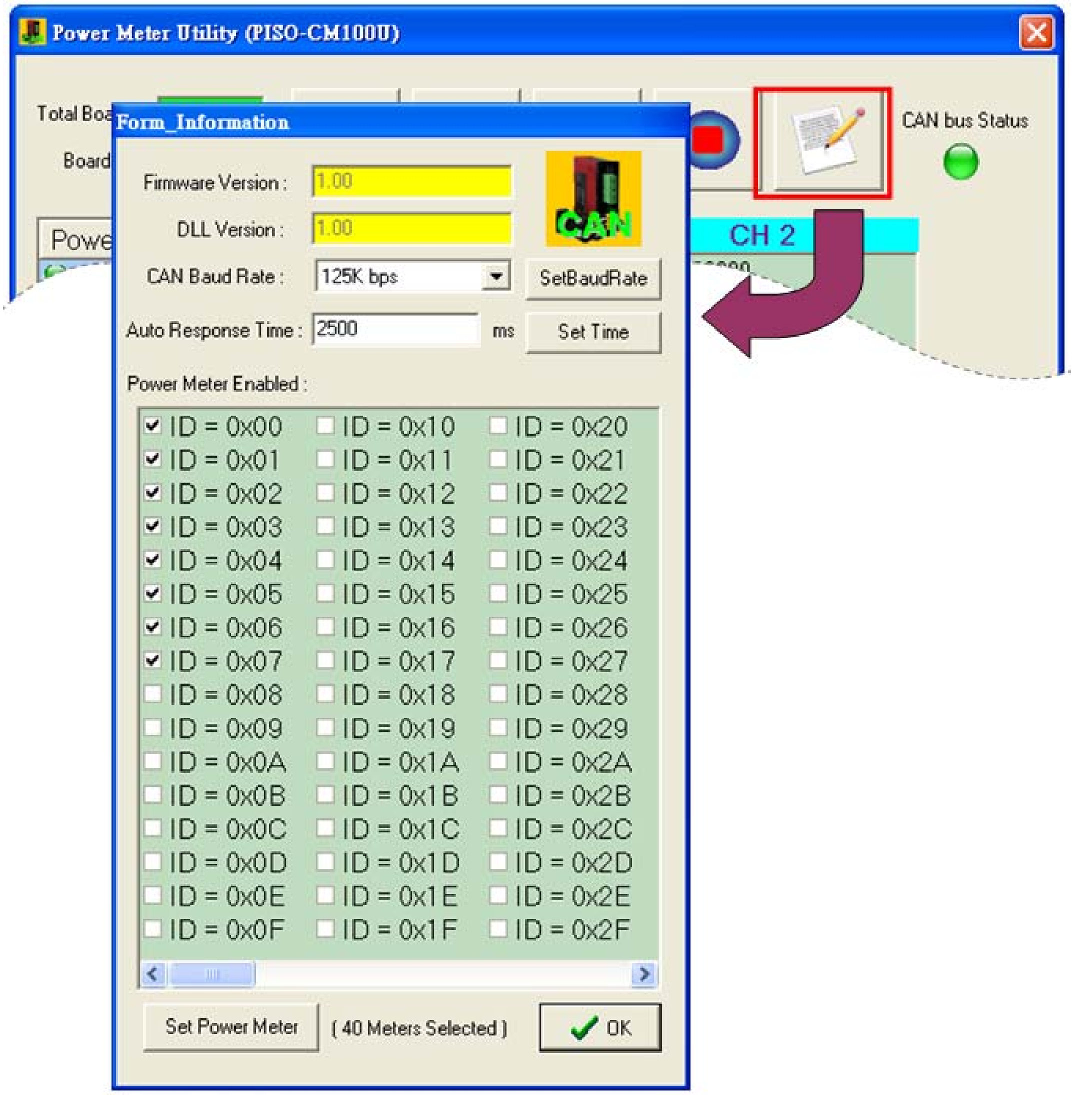


If pausing to reply is necessary, the users can click the “Stop icon” button to make all the power meters break off. The following picture shows the detail steps.



5.4 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.



Form_Information

Firmware Version : 1.00

DLL Version : 1.00

CAN Baud Rate : 125K bps ▾ SetBaudRate

Auto Response Time : 2500 ms Set Time

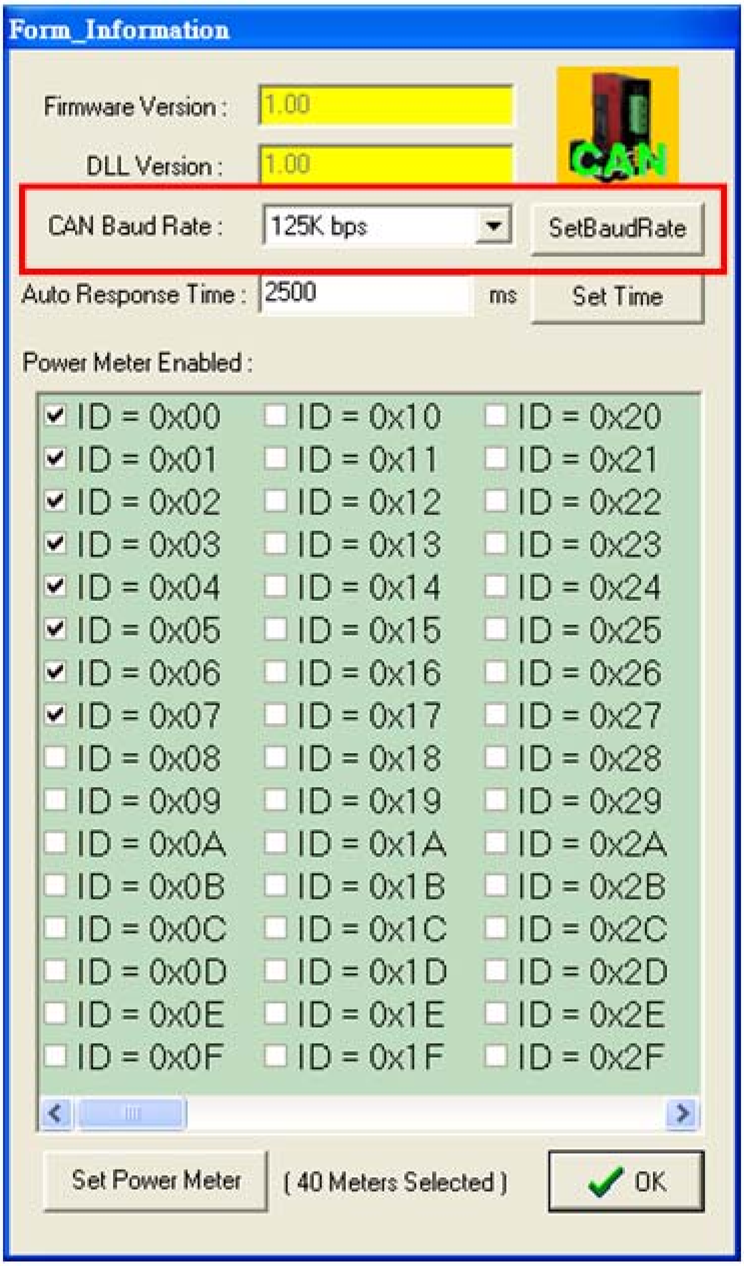
Power Meter Enabled :

☑ ID = 0x00
☑ ID = 0x01
☑ ID = 0x02
☑ ID = 0x03
☑ ID = 0x04
☑ ID = 0x05
☑ ID = 0x06
☑ ID = 0x07
□ ID = 0x08
□ ID = 0x09
□ ID = 0x0A
□ ID = 0x0B
□ ID = 0x0C
□ ID = 0x0D
□ ID = 0x0E
□ ID = 0x0F
□ ID = 0x10
□ ID = 0x11
□ ID = 0x12
□ ID = 0x13
□ ID = 0x14
□ ID = 0x15
□ ID = 0x16
□ ID = 0x17
□ ID = 0x18
□ ID = 0x19
□ ID = 0x1A
□ ID = 0x1B
□ ID = 0x1C
□ ID = 0x1D
□ ID = 0x1E
□ ID = 0x1F
□ ID = 0x20
□ ID = 0x21
□ ID = 0x22
□ ID = 0x23
□ ID = 0x24
□ ID = 0x25
□ ID = 0x26
□ ID = 0x27
□ ID = 0x28
□ ID = 0x29
□ ID = 0x2A
□ ID = 0x2B
□ ID = 0x2C
□ ID = 0x2D
□ ID = 0x2E
□ ID = 0x2F

Set Power Meter (40 Meters Selected) ✓ OK

5.4.1 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 PISO-CM100U. The following picture shows the detail steps.



Form_Information

Firmware Version : 1.00

DLL Version : 1.00

CAN Baud Rate : 125K bps

Auto Response Time : 2500 ms

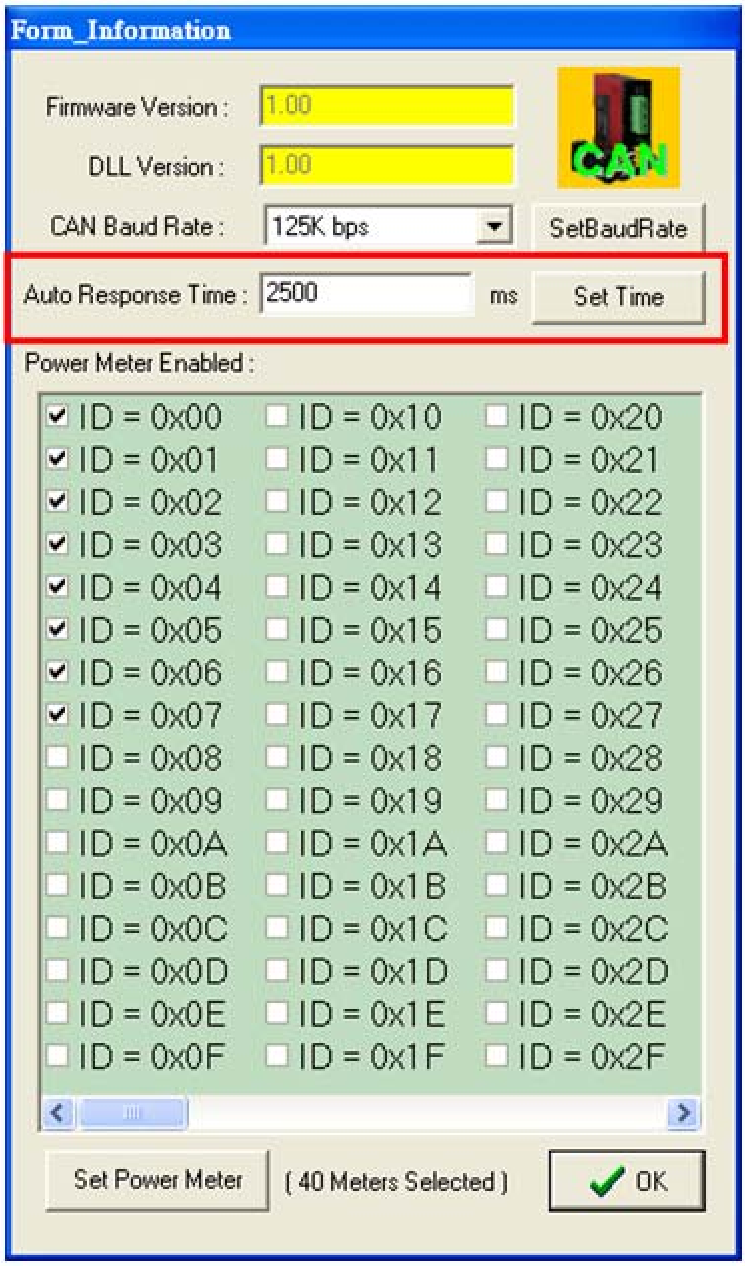
Power Meter Enabled :

☑ ID = 0x00
☑ ID = 0x01
☑ ID = 0x02
☑ ID = 0x03
☑ ID = 0x04
☑ ID = 0x05
☑ ID = 0x06
☑ ID = 0x07
□ ID = 0x08
□ ID = 0x09
□ ID = 0x0A
□ ID = 0x0B
□ ID = 0x0C
□ ID = 0x0D
□ ID = 0x0E
□ ID = 0x0F
□ ID = 0x10
□ ID = 0x11
□ ID = 0x12
□ ID = 0x13
□ ID = 0x14
□ ID = 0x15
□ ID = 0x16
□ ID = 0x17
□ ID = 0x18
□ ID = 0x19
□ ID = 0x1A
□ ID = 0x1B
□ ID = 0x1C
□ ID = 0x1D
□ ID = 0x1E
□ ID = 0x1F
□ ID = 0x20
□ ID = 0x21
□ ID = 0x22
□ ID = 0x23
□ ID = 0x24
□ ID = 0x25
□ ID = 0x26
□ ID = 0x27
□ ID = 0x28
□ ID = 0x29
□ ID = 0x2A
□ ID = 0x2B
□ ID = 0x2C
□ ID = 0x2D
□ ID = 0x2E
□ ID = 0x2F

(40 Meters Selected)

5.4.2 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 PISO-CM100U. The picture below shows the detail steps.



Firmware Version : 1.00

DLL Version : 1.00

CAN Baud Rate : 125K bps

Auto Response Time : 2500 ms

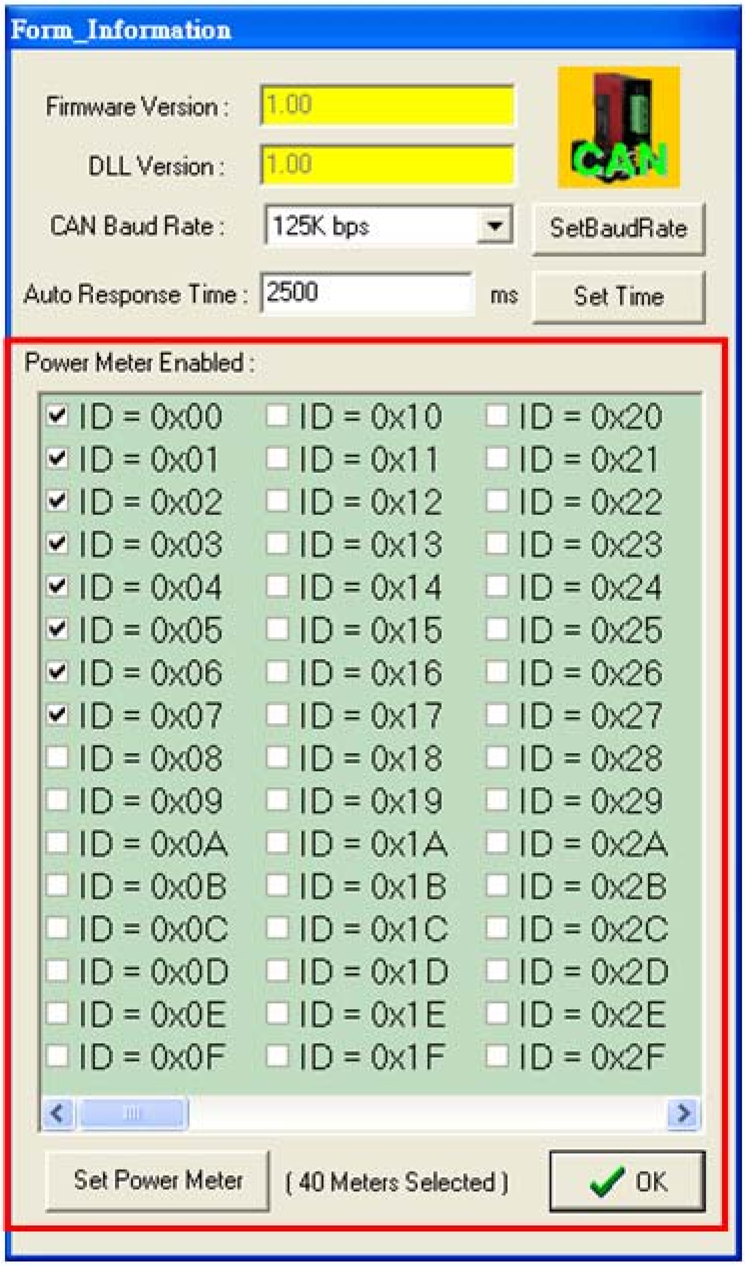
Power Meter Enabled :

☑ ID = 0x00	□ ID = 0x10	□ ID = 0x20
☑ ID = 0x01	□ ID = 0x11	□ ID = 0x21
☑ ID = 0x02	□ ID = 0x12	□ ID = 0x22
☑ ID = 0x03	□ ID = 0x13	□ ID = 0x23
☑ ID = 0x04	□ ID = 0x14	□ ID = 0x24
☑ ID = 0x05	□ ID = 0x15	□ ID = 0x25
☑ ID = 0x06	□ ID = 0x16	□ ID = 0x26
☑ ID = 0x07	□ ID = 0x17	□ ID = 0x27
□ ID = 0x08	□ ID = 0x18	□ ID = 0x28
□ ID = 0x09	□ ID = 0x19	□ ID = 0x29
□ ID = 0x0A	□ ID = 0x1A	□ ID = 0x2A
□ ID = 0x0B	□ ID = 0x1B	□ ID = 0x2B
□ ID = 0x0C	□ ID = 0x1C	□ ID = 0x2C
□ ID = 0x0D	□ ID = 0x1D	□ ID = 0x2D
□ ID = 0x0E	□ ID = 0x1E	□ ID = 0x2E
□ ID = 0x0F	□ ID = 0x1F	□ ID = 0x2F

(40 Meters Selected)

5.4.3 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 PISO-CM100U. The following picture shows the detail steps.



Form_Information

Firmware Version : 1.00

DLL Version : 1.00

CAN Baud Rate : 125K bps

Auto Response Time : 2500 ms

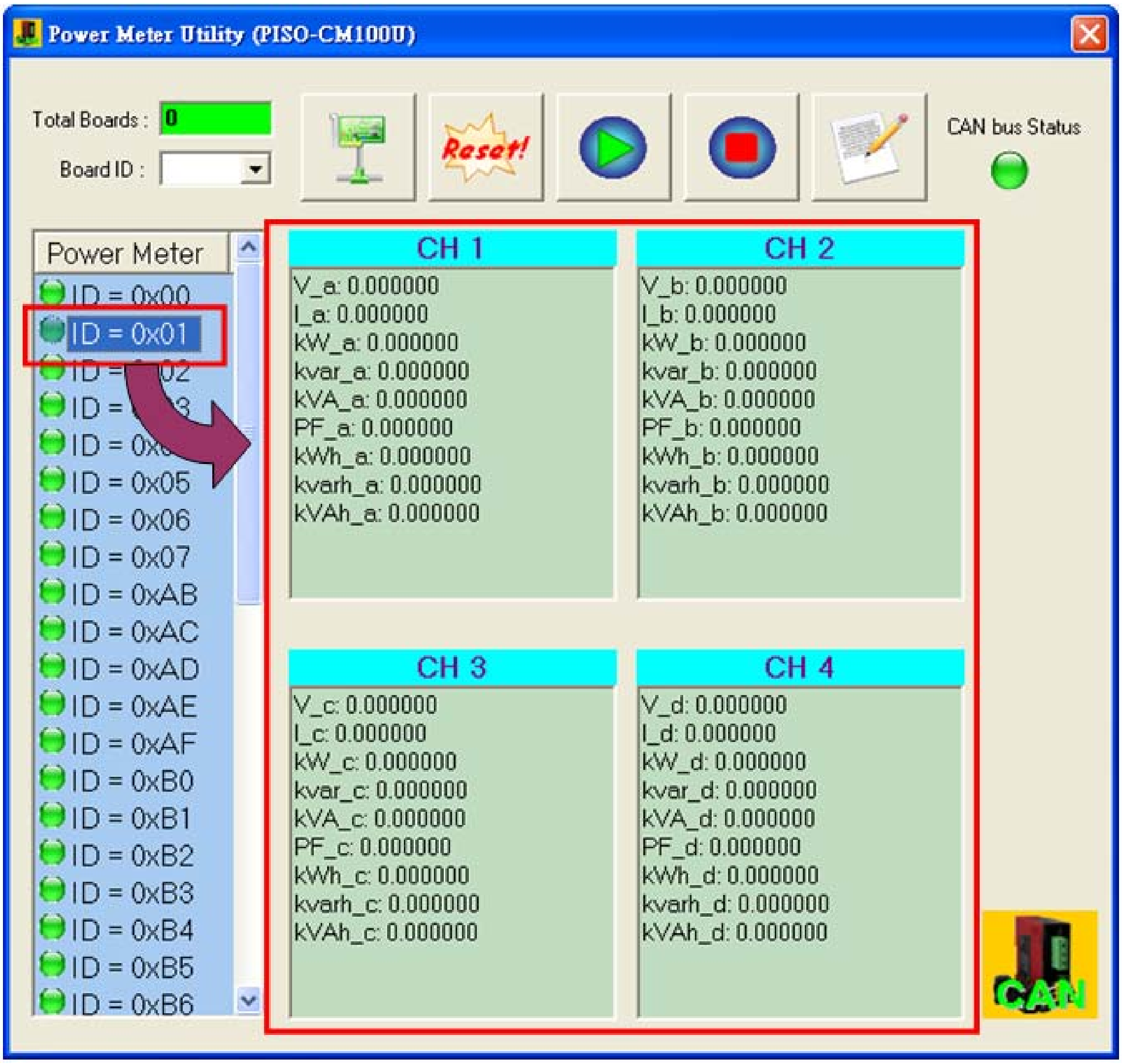
Power Meter Enabled :

☑ ID = 0x00	□ ID = 0x10	□ ID = 0x20
☑ ID = 0x01	□ ID = 0x11	□ ID = 0x21
☑ ID = 0x02	□ ID = 0x12	□ ID = 0x22
☑ ID = 0x03	□ ID = 0x13	□ ID = 0x23
☑ ID = 0x04	□ ID = 0x14	□ ID = 0x24
☑ ID = 0x05	□ ID = 0x15	□ ID = 0x25
☑ ID = 0x06	□ ID = 0x16	□ ID = 0x26
☑ ID = 0x07	□ ID = 0x17	□ ID = 0x27
□ ID = 0x08	□ ID = 0x18	□ ID = 0x28
□ ID = 0x09	□ ID = 0x19	□ ID = 0x29
□ ID = 0x0A	□ ID = 0x1A	□ ID = 0x2A
□ ID = 0x0B	□ ID = 0x1B	□ ID = 0x2B
□ ID = 0x0C	□ ID = 0x1C	□ ID = 0x2C
□ ID = 0x0D	□ ID = 0x1D	□ ID = 0x2D
□ ID = 0x0E	□ ID = 0x1E	□ ID = 0x2E
□ ID = 0x0F	□ ID = 0x1F	□ ID = 0x2F

(40 Meters Selected)

5.5 Display the Real-Time Power Information

If the users have never configured any power meter within the PISO-CM100U, 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.



Total Boards : 0

Board ID :

CH 1

V_a: 0.000000
I_a: 0.000000
kW_a: 0.000000
kvar_a: 0.000000
kVA_a: 0.000000
PF_a: 0.000000
kWh_a: 0.000000
kvarh_a: 0.000000
kVAh_a: 0.000000

CH 2

V_b: 0.000000
I_b: 0.000000
kW_b: 0.000000
kvar_b: 0.000000
kVA_b: 0.000000
PF_b: 0.000000
kWh_b: 0.000000
kvarh_b: 0.000000
kVAh_b: 0.000000

CH 3

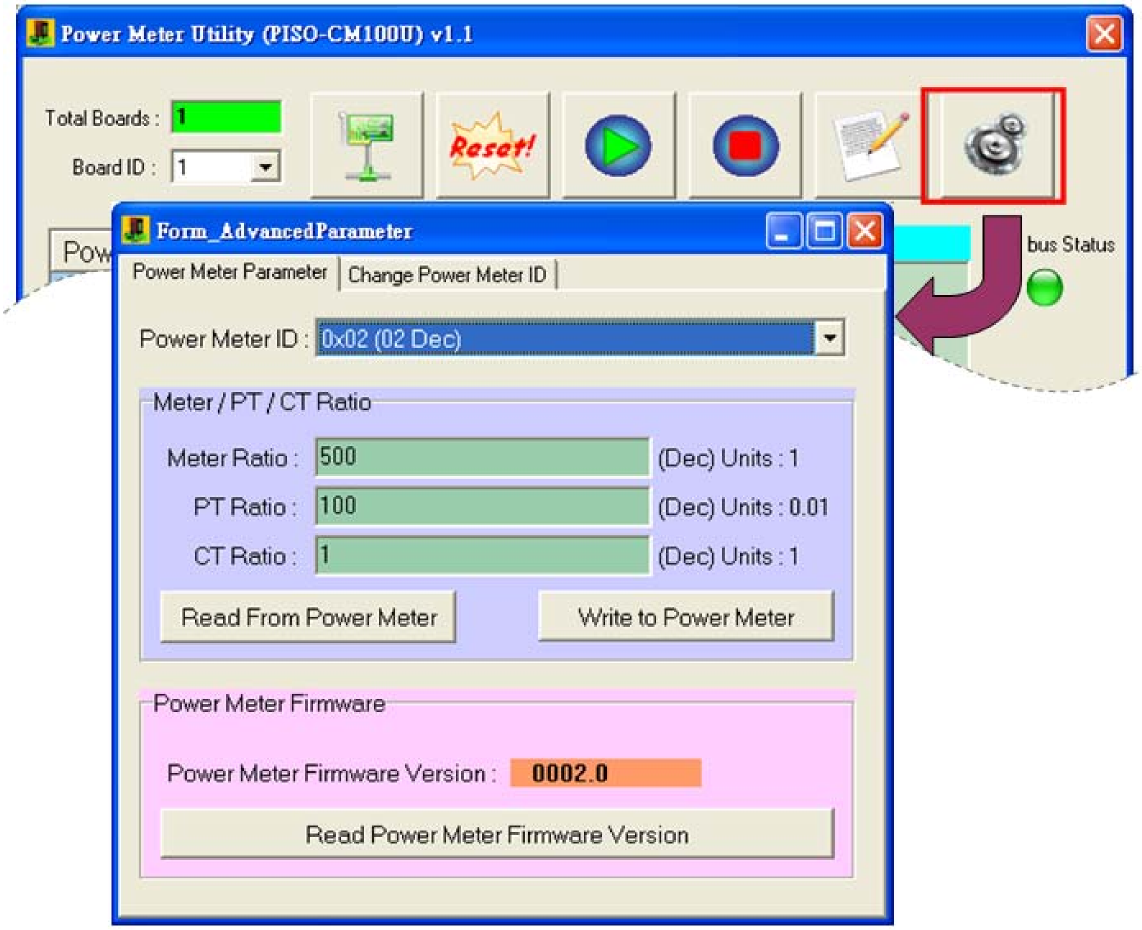
V_c: 0.000000
I_c: 0.000000
kW_c: 0.000000
kvar_c: 0.000000
kVA_c: 0.000000
PF_c: 0.000000
kWh_c: 0.000000
kvarh_c: 0.000000
kVAh_c: 0.000000

CH 4

V_d: 0.000000
I_d: 0.000000
kW_d: 0.000000
kvar_d: 0.000000
kVA_d: 0.000000
PF_d: 0.000000
kWh_d: 0.000000
kvarh_d: 0.000000
kVAh_d: 0.000000

5.6 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.



Total Boards : 1

Board ID : 1

Power Meter ID : 0x02 (02 Dec)

Meter Ratio : 500 (Dec) Units : 1

PT Ratio : 100 (Dec) Units : 0.01

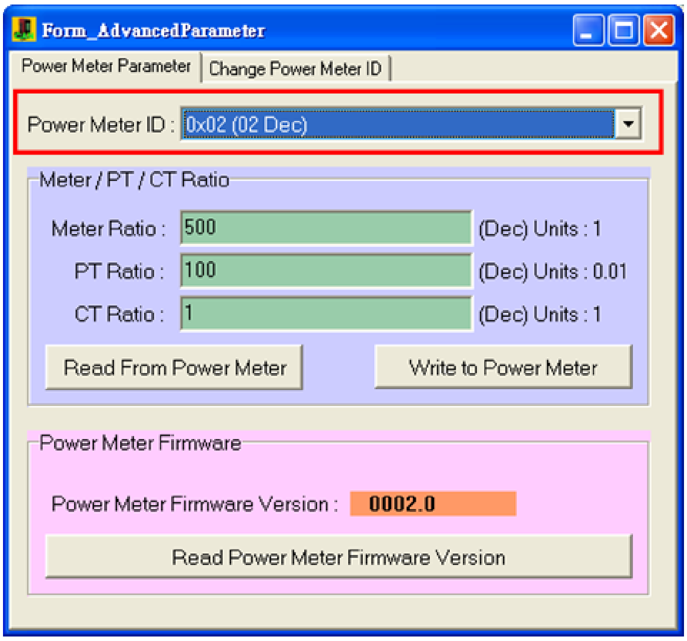
CT Ratio : 1 (Dec) Units : 1

Power Meter Firmware Version : 0002.0

5.6.1 The Ratio Parameters 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 or write the ratio parameters. Note that the ratio parameters would affect the power information. The following picture shows the detail steps.

Step 1 : Select the correct ID of the power meter



Power Meter ID : 0x02 (02 Dec)

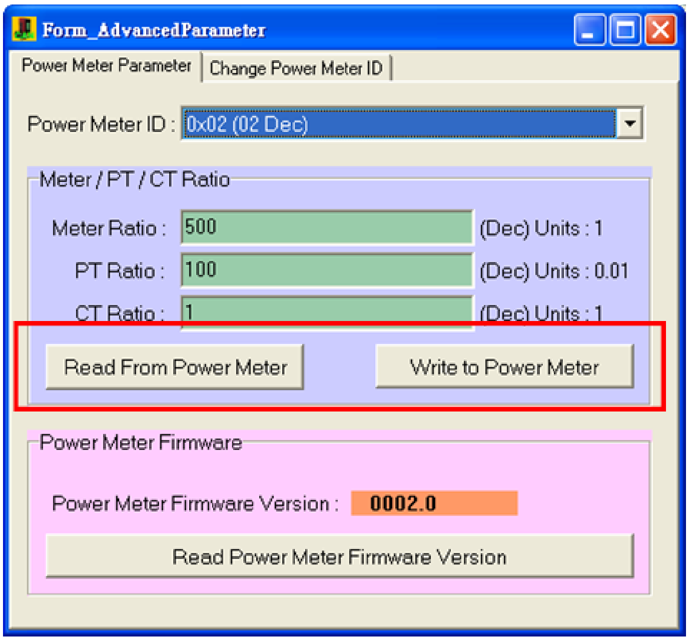
Meter Ratio : 500 (Dec) Units : 1

PT Ratio : 100 (Dec) Units : 0.01

CT Ratio : 1 (Dec) Units : 1

Power Meter Firmware Version : 0002.0

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



Power Meter ID : 0x02 (02 Dec)

Meter Ratio : 500 (Dec) Units : 1

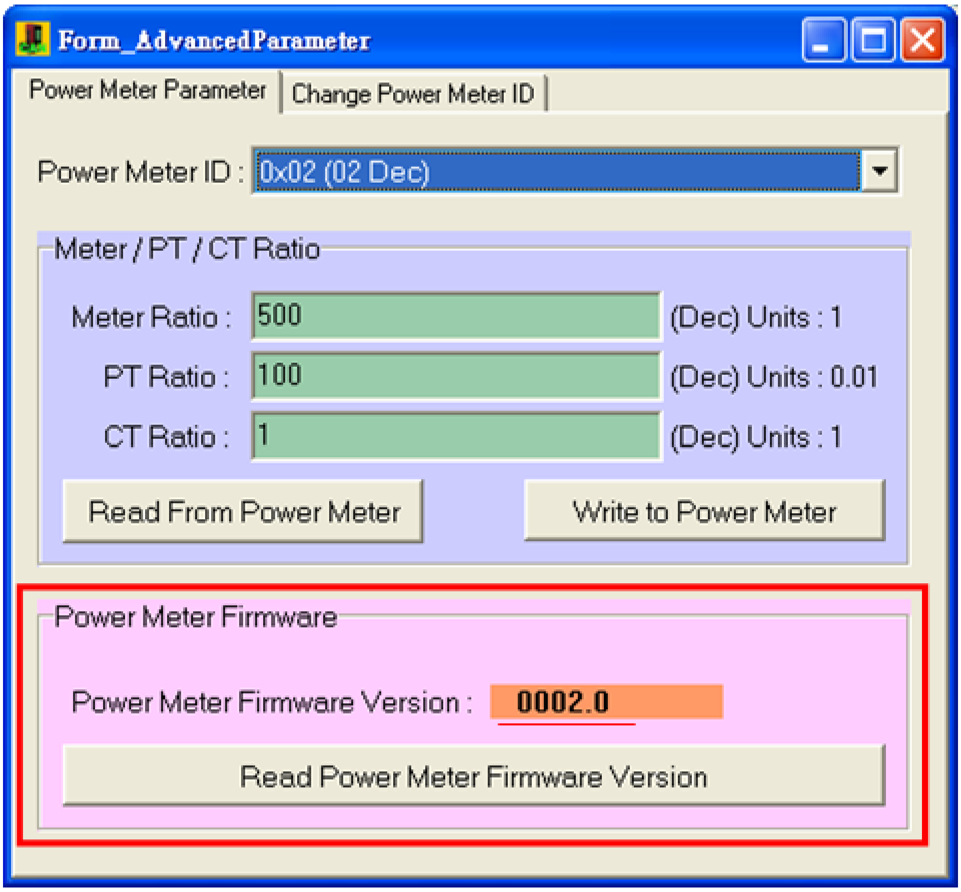
PT Ratio : 100 (Dec) Units : 0.01

CT Ratio : 1 (Dec) Units : 1

Power Meter Firmware Version : 0002.0

5.6.2 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.



Power Meter ID : 0x02 (02 Dec)

Meter / PT / CT Ratio

Meter Ratio : 500 (Dec) Units : 1

PT Ratio : 100 (Dec) Units : 0.01

CT Ratio : 1 (Dec) Units : 1

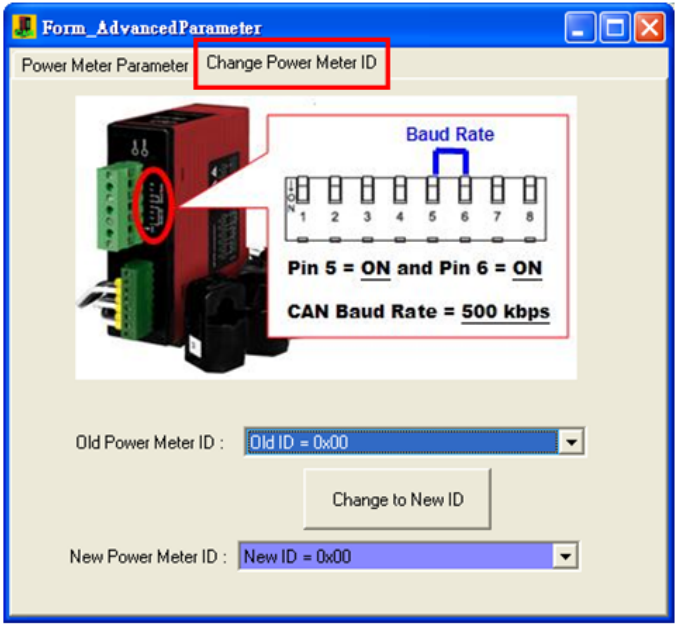
Power Meter Firmware

Power Meter Firmware Version : 0002.0

5.6.3 Change the ID of the Power Meter

After clicking the “Advanced” button and select the “Change Power Meter ID” page, the users can see the following picture. The user should follow the instruction to change the DIP switch. Then, you can select the current power meter’s ID to be changed to a new one. The following picture shows the detail steps.

Step 1 : Change to the page as below.



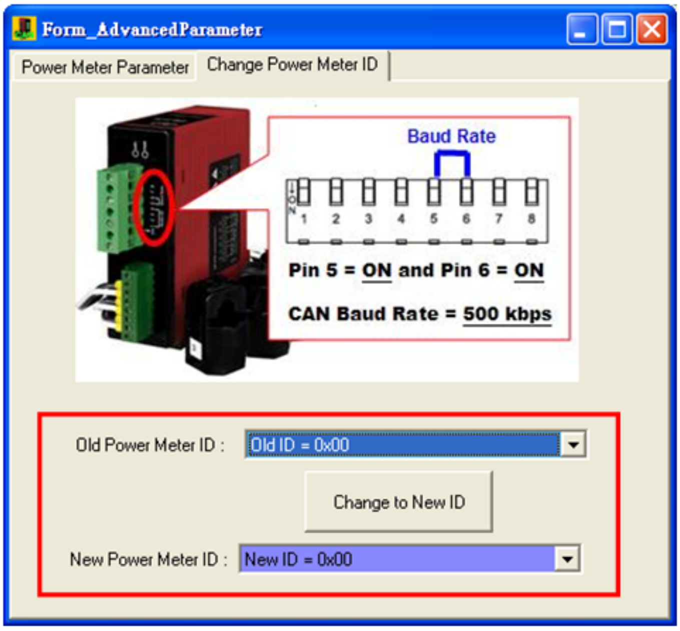
Pin 5 = ON and Pin 6 = ON

CAN Baud Rate = 500 kbps

Old Power Meter ID : Old ID = 0x00

New Power Meter ID : New ID = 0x00

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



Pin 5 = ON and Pin 6 = ON

CAN Baud Rate = 500 kbps

Old Power Meter ID : Old ID = 0x00

New Power Meter ID : New ID = 0x00