PL7413

(Single Phase, Power/Energy Metering IC)

ZCC + Auto Learn Demo Board Multi-Device

Auto Calibration Application Note

Document number：AN-71170501

Revision：1.1

Release Date：Oct 20, 2017

Prolific Technology Inc.
7F, No. 48, Sec. 3, Nan Kang Rd.
Nan Kang, Taipei 115, Taiwan, R.O.C.
Telephone: +886-2-2654-6363
Fax: +886-2-2654-6161
E-mail: sales@prolific.com.tw
Website: http://www.prolific.com.tw

Table of Contents

List of Figures

1. HW Setup environment and SPI communication

1.1 Setup Environment

1.1.1 PL7413 demo board + Auto Learn+ ZCC Board

Use Demo_Board Function: AC_Plug_1 Phase connect CON9(or CON8)AC_Plug_1 Neutral connect CON5/
Loading: AC_Socket Phase connect CON6(or CON7); AC_Socket Neutral connect CON4;J4 Short

Use Auto Learn Function: AC_Plug_1 Phase connect CON9(or CON8)AC_Plug_1 Neutral connect CON5/; J4 Short

Use ZCC Function: AC_Plug_1 Phase connect CON9(or CON8) AC_Plug_1 Neutral connect CON5/
AC_Plug_2 Phase connect CON10(or CON12) AC_Plug_1 Neutral connect CON3; J4 OPEN

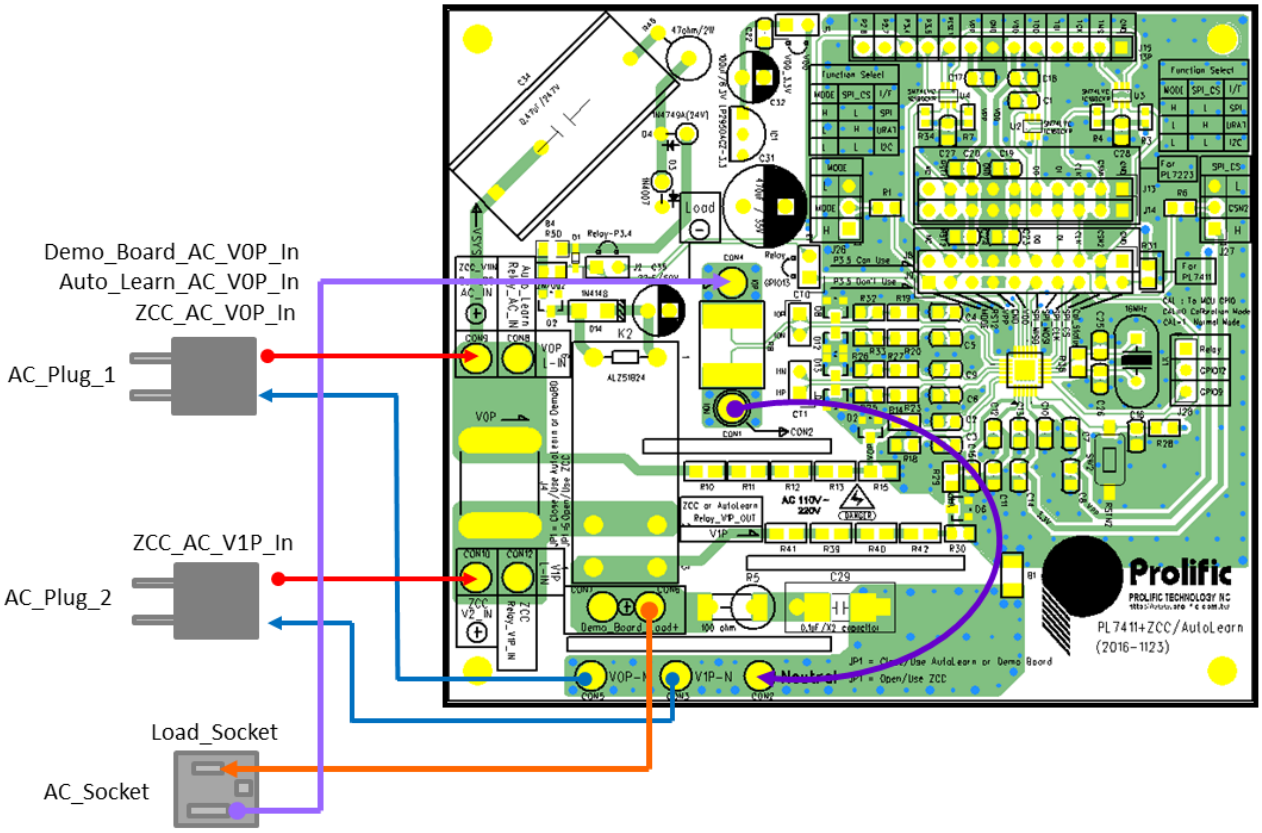


Figure 1.1：PL7413_ZCC+AL_DB demo board setup

There are three function in this board can be applied:

(1).Use Demo board function of PL7413

(2).Use auto learn function of PL7413

(3) Use ZCC function of PL7413+ PL7223 Demo board

This document is focus on using PL7413 demo board and auto learn function of PL7413.

1.1.2 Use Demo board function + Auto learn function of PL7413

If you use the PL7413's auto learn DSP codes, it will auto counting zero-crossings to help the relay switch at the zero crossing. It also has the AFE of meter functions. About auto learn ZCC, please reference Section 5.5.

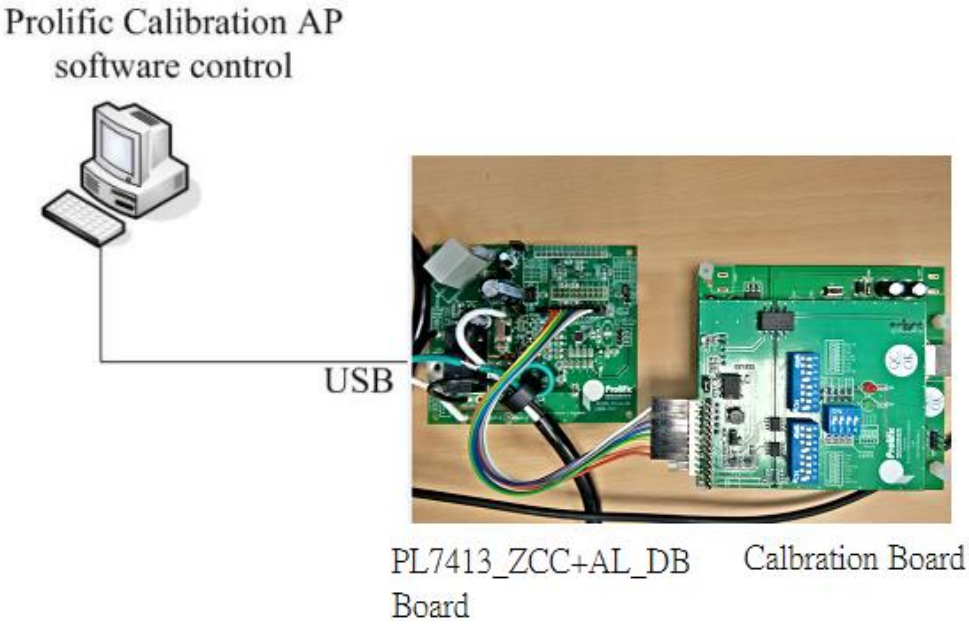


Figure 1.2：PL7413_ZCC+AL_DB demo board with auto learn ZCC

1.1.3 Use ZCC function of PL7413+PL7223 Demo board

If your calibration target is PL7223 demo board, because PL7233 DSP codes size has limit, so needs PL7413 to help count PL7223's zero crossing.

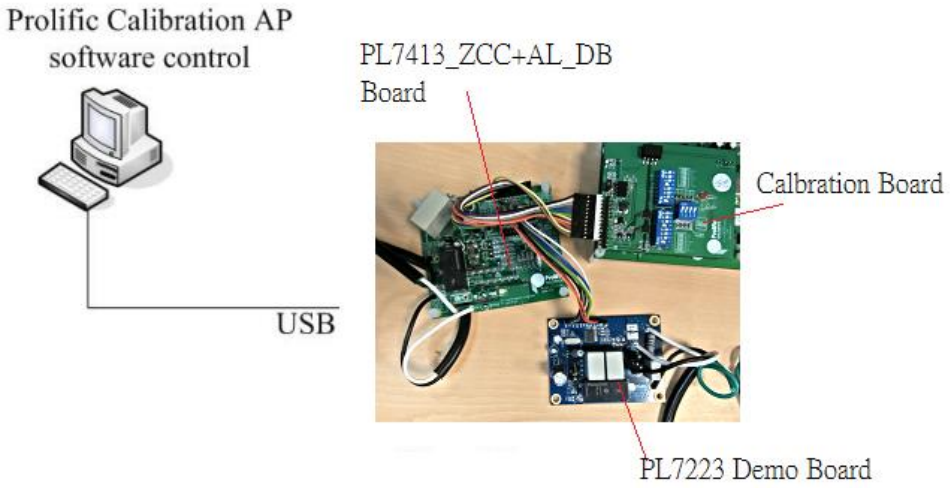


Figure 1.3：PL7413_ZCC+AL_DB board + PL223 Demo board +

1.1.4 Calibration Overview

1. Use the calibration board to supply 3.3V to VCC33 of PL7413,as below picture. (Use CON3 short for calibration board).
2. CON4 of SPI connect to SPI pin of PL7413 from calibration board.

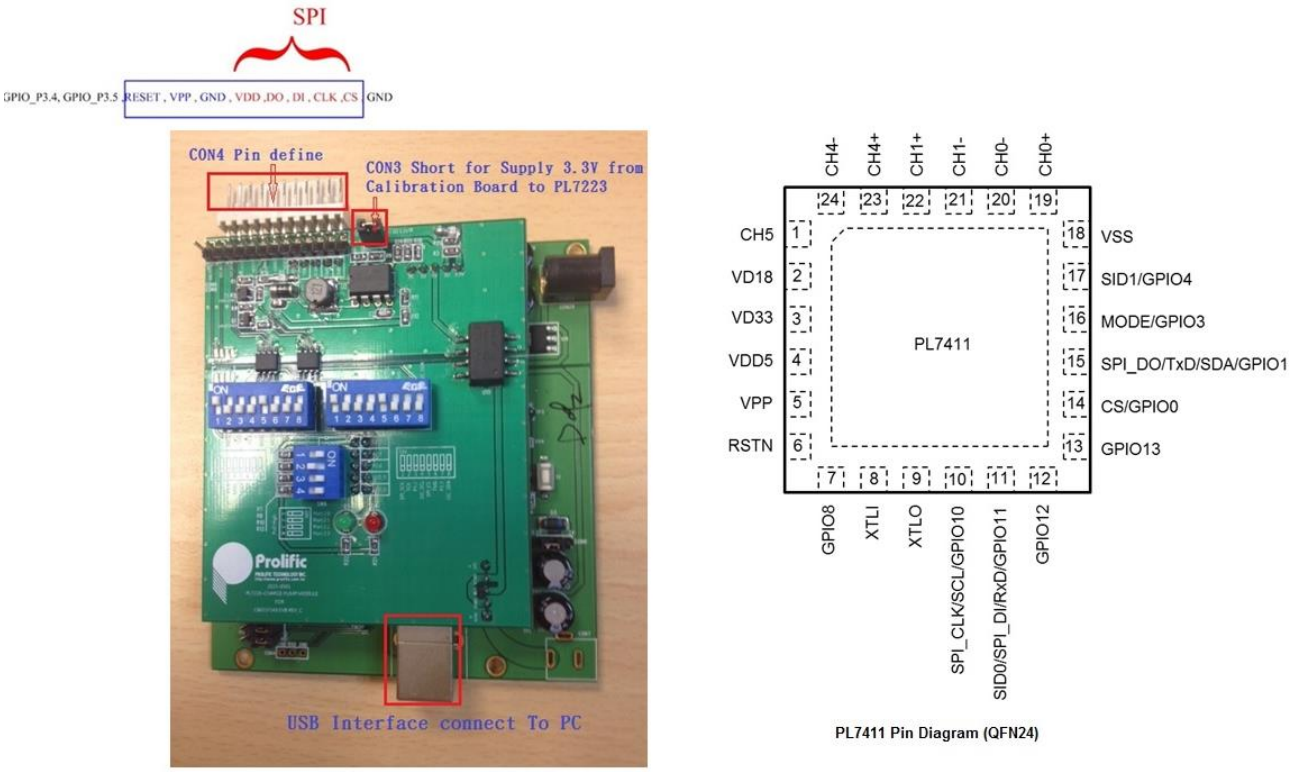


Figure 1.4：Calibration Overview

1.1.5 Calibration Board Bottom Board Setting

Below bitmap to show bottom board the interface connect via SPI, First Please set Calibration Jump as follows：

CON18 :Short	CON24:Short	CON22:Open	CON28:Open

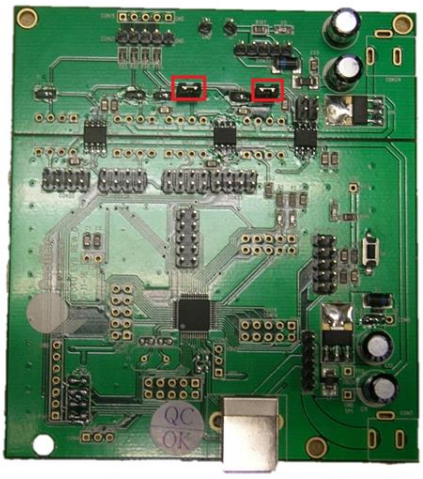


Figure 1.5：Calibration Bottom Board

Calibration board Top Board setting

Below bitmap to show Top board the interface connect via SPI, First Please set Calibration Dip Switch as follows

DIP1 pin1→1	DIP1 pin5→1	DIP2 pin1→1	DIP2 pin5→1

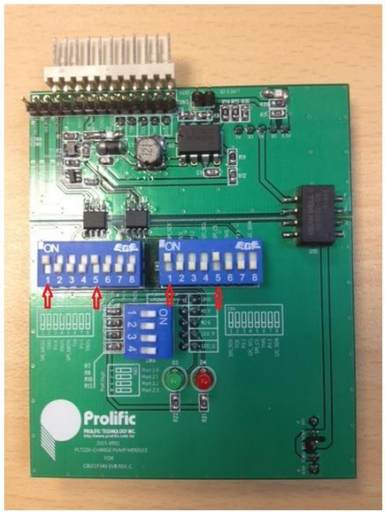


Figure 1.6：Calibration Top Board

1.1.6 New calibration board Top Board Serial Number setting

Below bitmap to show Top board can via Dip switch to change the Serial Number, This is for Multi-calibration device only

SerialNo 1:

pin1	ON
pin2	ON
pin3	ON
pin4	ON

SerialNo 2:

pin1	OFF
pin2	ON
pin3	ON
pin4	ON

SerialNo 3:

pin1	ON
pin2	OFF
pin3	ON
pin4	ON

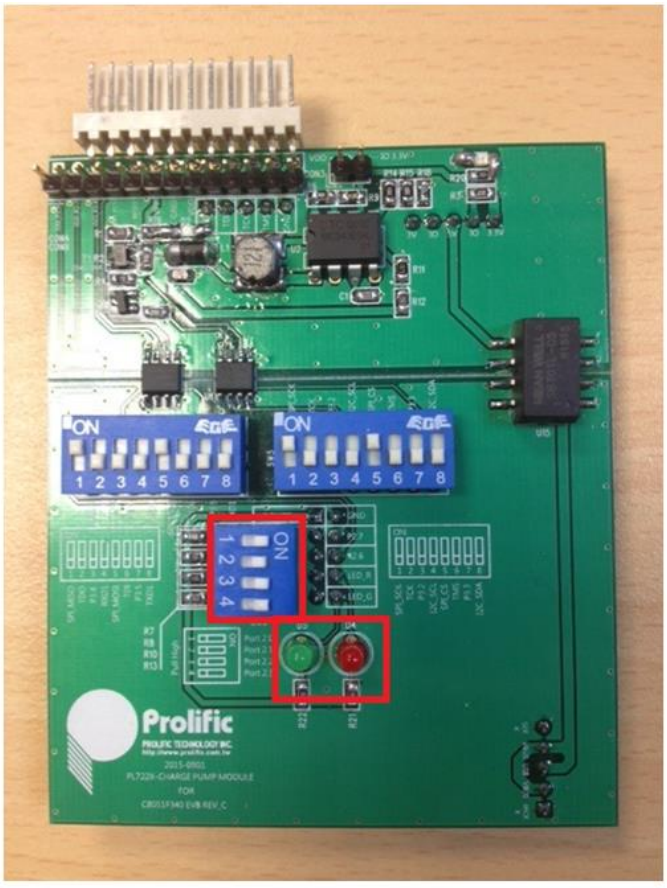


Figure 1.7：Calibration Board SN setting

1.1.7 Programming Environment Connect Overview

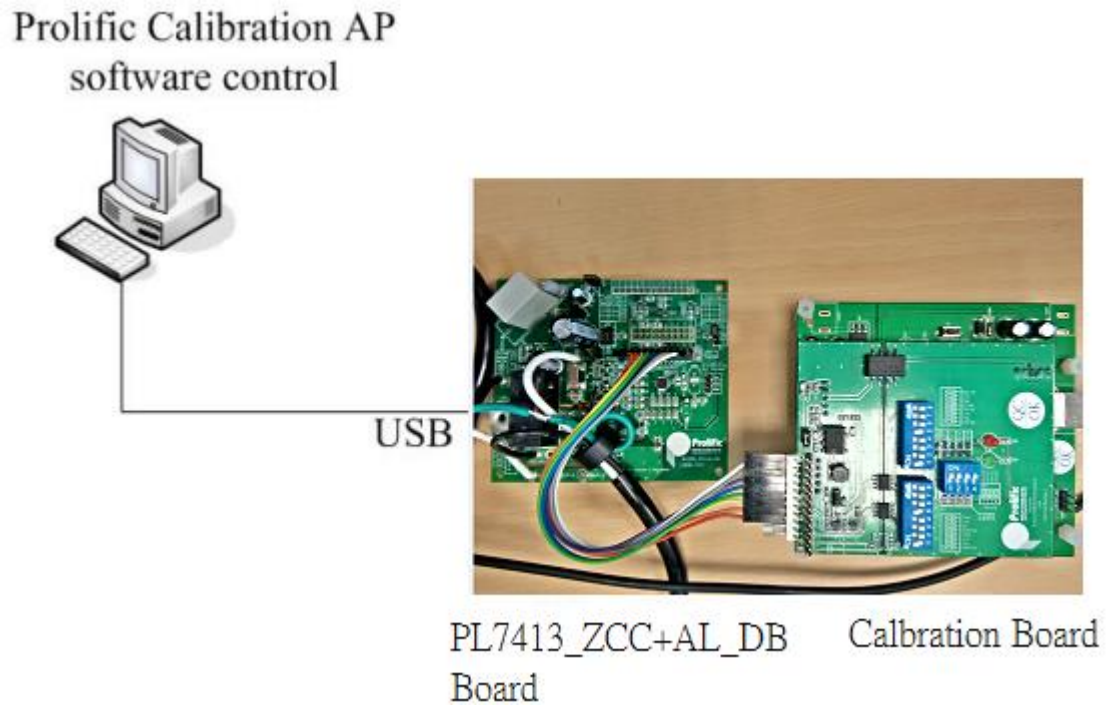


Figure 1.8：Calibration Board with PL7413 connect Overview

1. Please check the PC Device Management , calibration board driver is install and connect correct.

2.Please check “USBXpress Device” driver status is correct.

Device version 4.0.0.0 for WIN8/WIN7/WINXP.

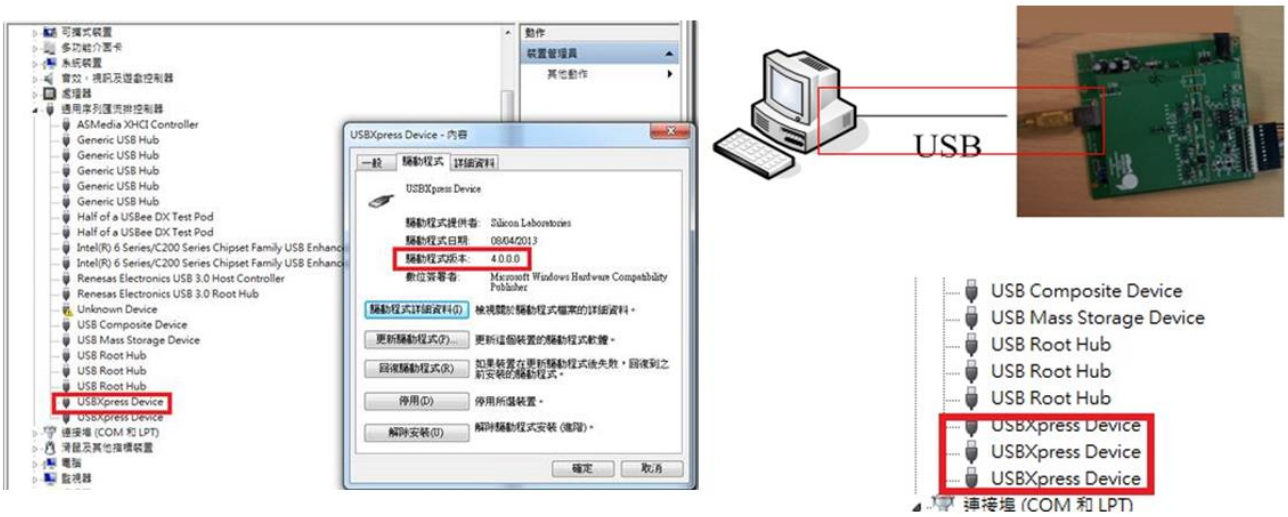


Figure 1.9：Check USB driver status

1.1.8 PL7413 Demo board I/F setting

PL7413 has three interface SPI/I2C/UART interface that operates at slave mode. It can communicate and access data with MCU. MCU should serve as the SPI/I2C/UART master and sends chip Select and clock signal to the PL7413.When Use the SPI I/F Data is written through SPI_DI and read through SPI_DO. .When Use the I2C I/F Data is written and read through SDA, .When Use the UART I/F Data is written through TX

and read through RX. Figure 1-6 to 1-8 shows the connection and pin definition:

IO Mode is latch when resetn is from low to high

- **i2c_en = [mode, spi_cs]= 2'b00**
- **uart_en = [mode, spi_cs]= 2'b01**
- **spi_en = [mode, spi_cs]= 2'b10**
- **gpio_en = [mode, spi_cs]= 2'b11**

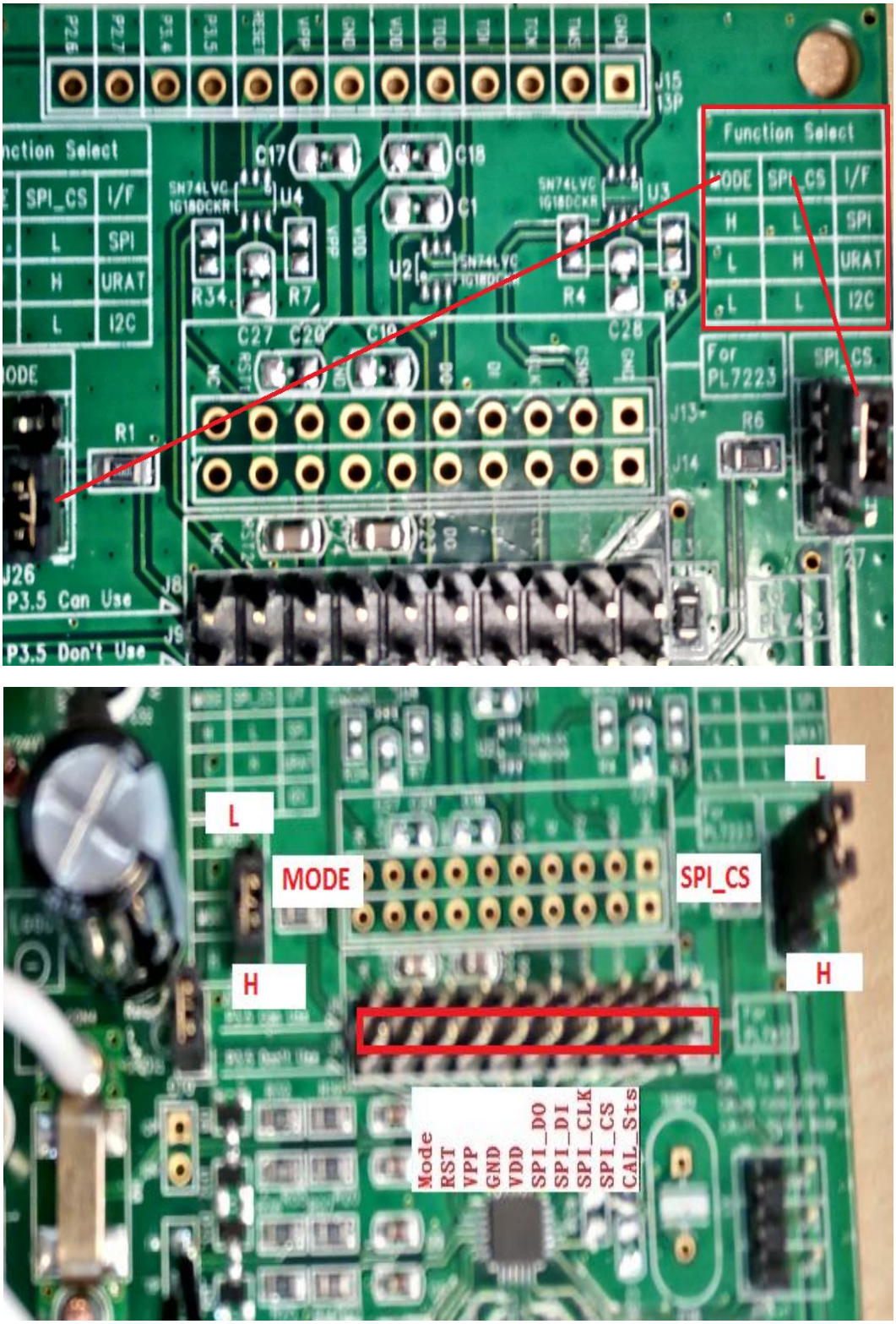


Figure 1.10 : PL7413 Demo board I/F setting

1.1.8.1 PL7413 SPI Communication

- SPI Slave mode, supports mode 0 ,mode1, mode2 and mode 3
- Supports single and multi-byte read write
- Supports CRC data check

Calibration board setting

You can reference below bitmap to change the interface connect via SPI, First Please set Calibration Jump as follows：

Bottom board: CON18 :Short

CON24:Short

CON22:Open

CON28:Open

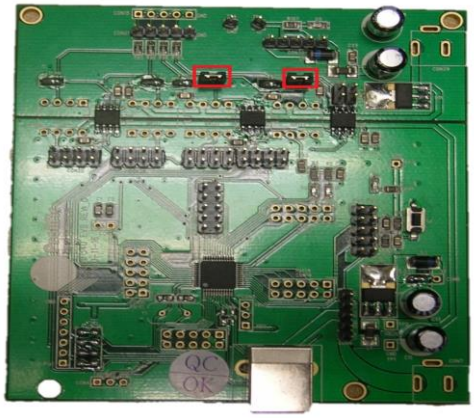


Figure 1.11 : Calibration bottom board for SPI

Top Board:

DIP1 pin1→1	DIP2 pin1 →1	DIP3 pin1→1	DIP3 pin5→1

(reference red arrow)

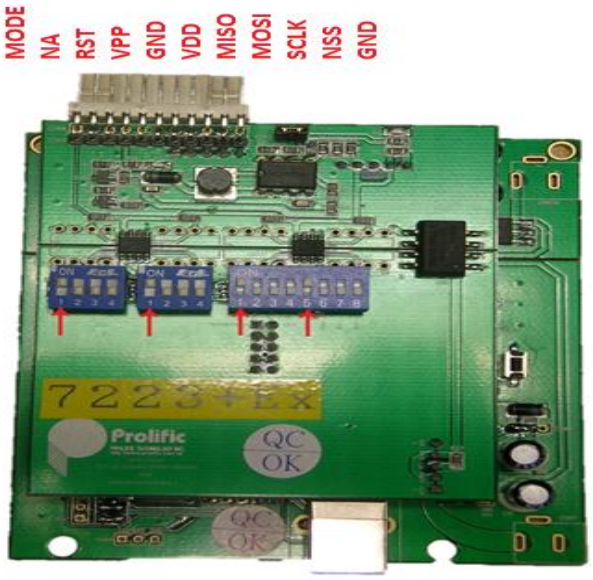


Figure 1.12 : Calibration top board for SPI

PL7413 demo board J9 through by the SPI interface connect to the calibration board.

J9-CAL_Sts(PIN1) does not need to connect to the GND of the calibration board.

IO Mode: MODE=H, SPI_CS=L.

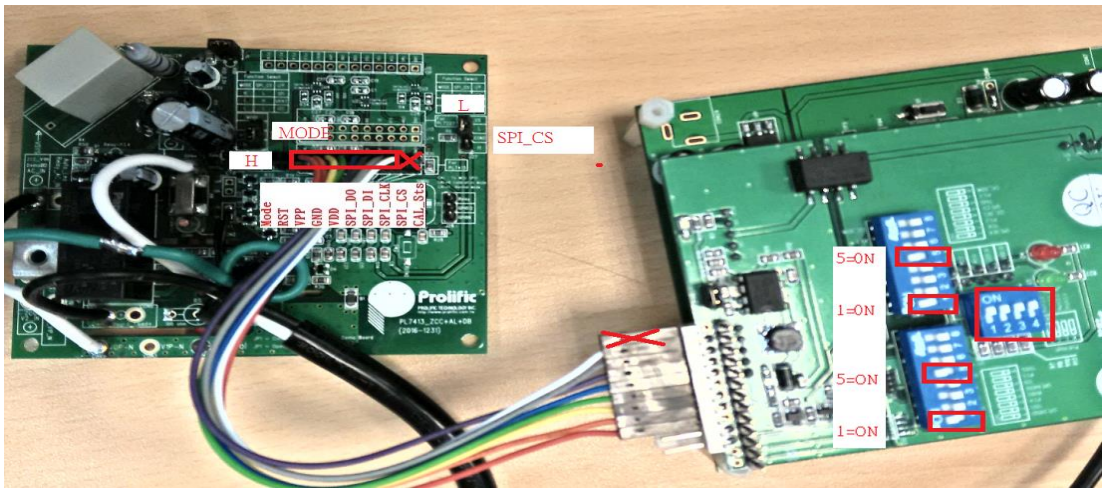


Figure 1.13 : PL7413 SPI Communication

1.1.8.2 PL7413 UART Communication

- Auto-baud rate learning
- Two hardware slave ID selection for cascade application
- Software ID programmable
- Supports single and multi-byte read write
- Supports CRC data check
- Supports UART timeout
- Supports IR38K carrier remove
- UART master mode for auto data rep

Calibration board setting

You can reference below bitmap to change the interface connect via UART,

Please set Calibration Jump as follow：

Bottom board: CON18 :Short

CON24:Short

CON22:Open

CON28:Open

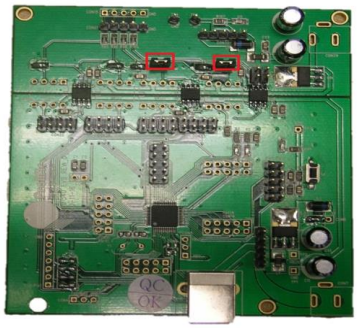


Figure 1.14 : Calibration bottom board for UART

Top Board:

DIP1 pin4→1	DIP2 pin4 →1	DIP3 pin5→1

Figure 1.15 : Calibration top board for UART

PL7413 demo board J9 through by the UART interface connect to the calibration board.

J9-CAL_Sts(PIN1) does not need to connect to the GND of the calibration board.

IO Mode: MODE=L, SPI_CS=H.

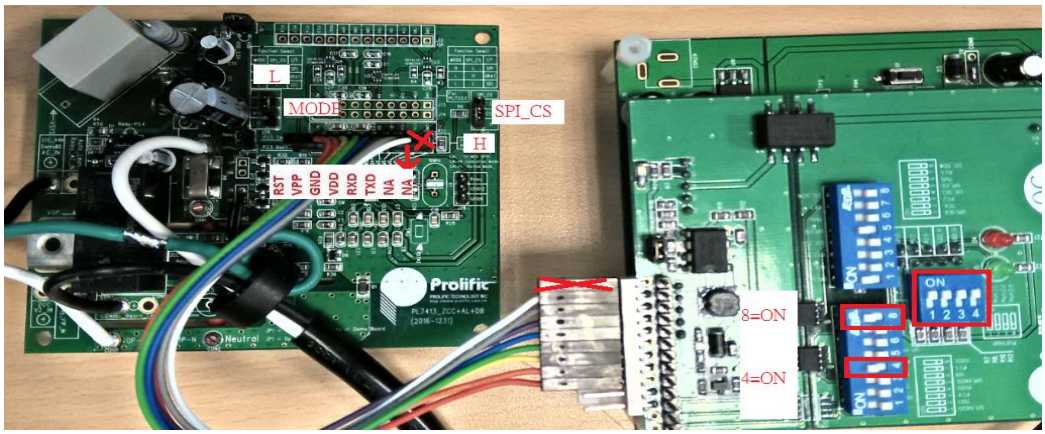


Figure 1.16 : PL7413 UART Communication

1.1.8.3 PL7413 I2C Communication

Calibration board setting

You can reference below bitmap to change the interface connect via I2C,

Please set Calibration Jump as follow：

Bottom board: CON18 : Open

CON24: Open

CON22: Short

CON28: Short

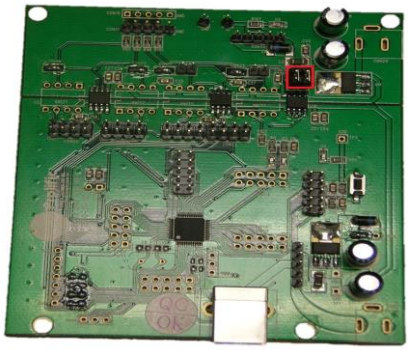


Figure 1.17 : Calibration bottom board for I2C

Top Board: DIP3 ➔ (reference red arrow)

DIP3 pin4→1	DIP3 pin8→1

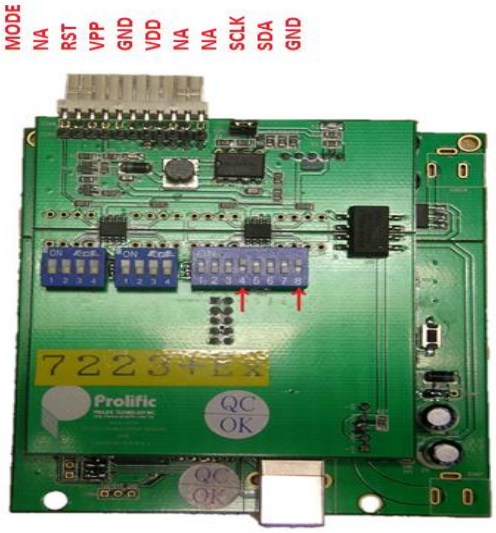


Figure 1.18 : Calibration top board for I2C

PL7413 demo board J9 through by the I2C interface connect to the calibration board.

J9-CAL_Sts(PIN1) does not need to connect to the GND of the calibration board.

IO Mode: MODE=H, SPI_CS=H.

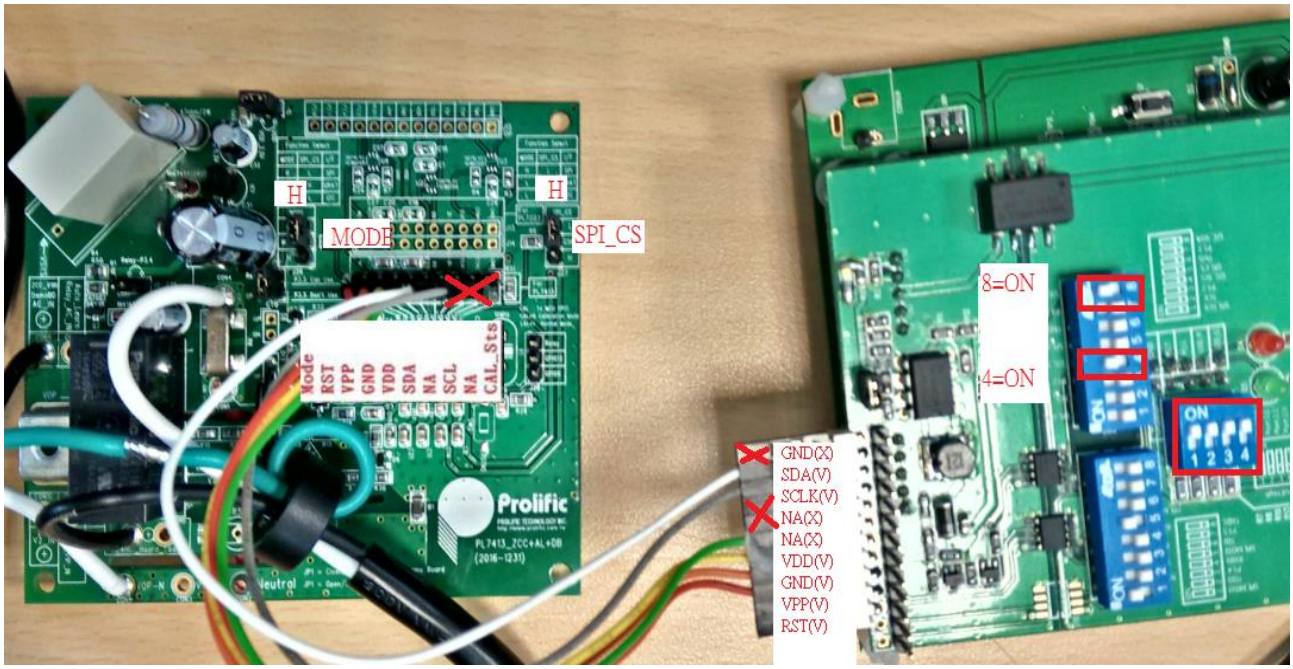


Figure 1.19 : PL7413 I2C Communication

1.2 Start SPI Communication

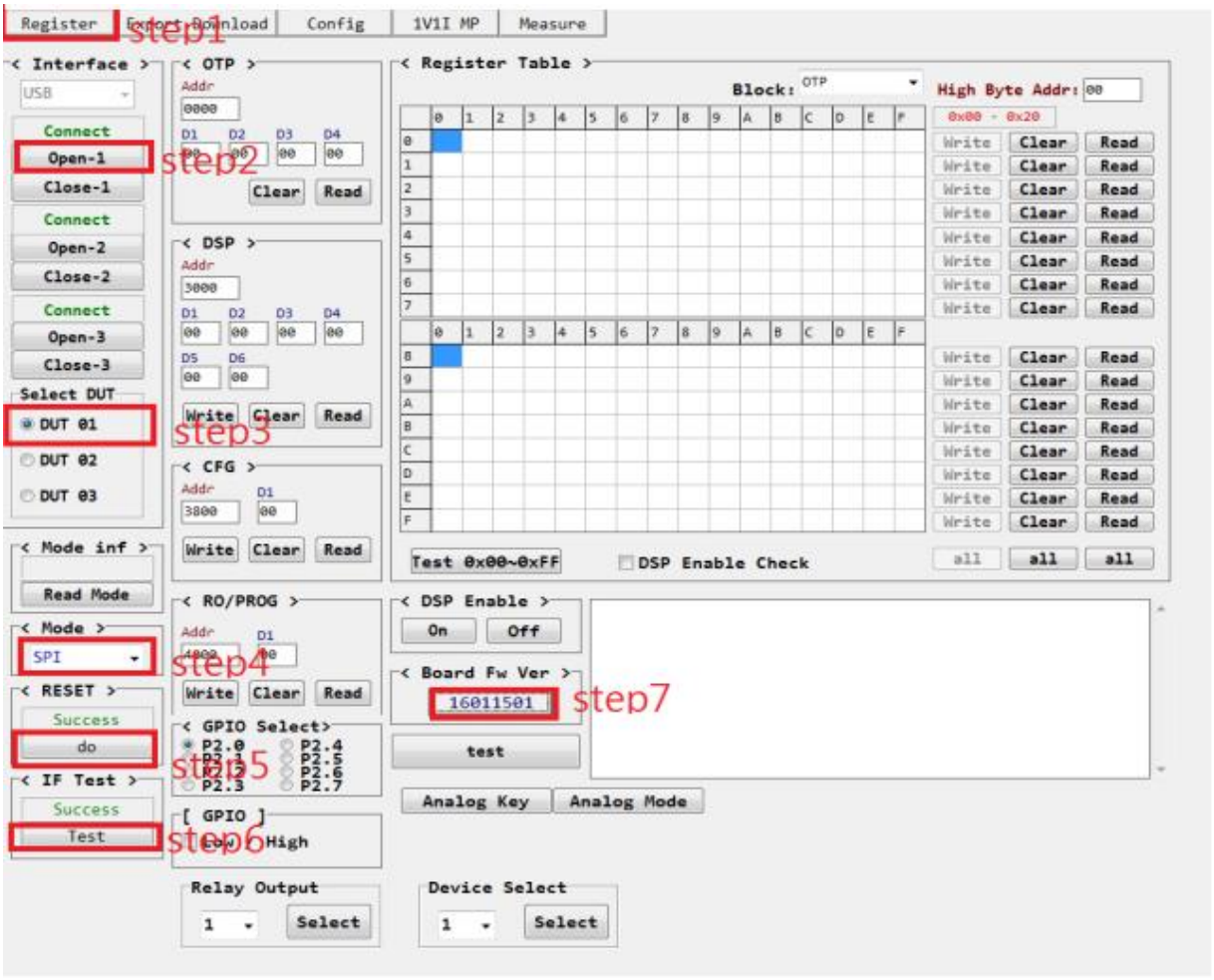


Figure 1.20：SPI communication Test

1.2.1 SPI Communication Test, Please refer to Step 0 ~ Step5,

Step 1：Select “Register” Page

Step 2：Open USB

Step 3：Select DUT

Step 4：Select “SPI” Interface

Step 5：Press “Reset” button to Reset PL7413

Step 6：Click “Test” to check the devices SPI connect Well.

It will auto test the interface (Communication) appear Pass” green word.

Note:

For Multi-device ,the Serial Number will different,

EX：DUT01 : Serial Number is 0001

DUT02 : Serial Number is 0002

DUT03 : Serial Number is 0003

1.2.2 Single or Multi-device Calibration Select , Please refer to Step 1 ~ Step3,

Step 1：Select “1V1I MP” Page

Step 2：Select “Search USB Device” ,If you device ready will show green

Step 3：Press “Connect” button to connect with PL7413, If connect OK, will show green word” Success” .

Step 4：Press “Reset” to reset PL7413

Step5：Press “Test” button . If Interface test OK will show green word “Pass”

Note1：If you connect three calibration board and test well , We will depend-on your select to do the calibration

If you select USB device “1”, AP will show DUT 01

If you select USB device “2”, AP will show DUT 01 and DUT02

If you select USB device “3”, AP will show DUT 01 and DUT02 and DUT03

Note2: If make sure DUT and calibration board connect well,

You can ignore above step, Direct press “Auto Calibration” button, AP will do above of them.

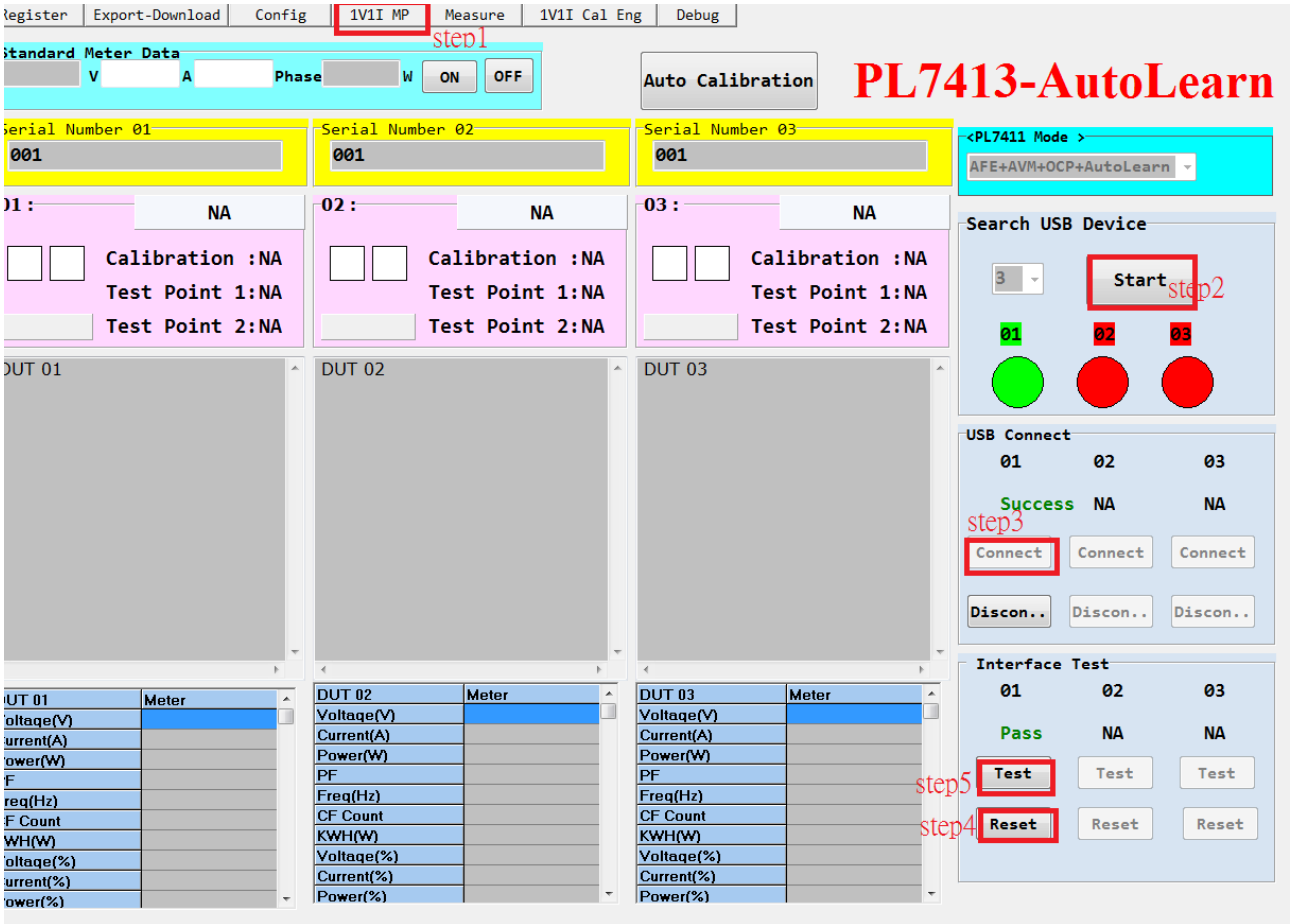


Figure 1.21：SPI communication and Device Select

1.3 Troubleshooting

1.3.1 Check “USB” CONNECT FAIL (Figure 1.9)

1. Please check the PC Device Management , calibration board driver is install and connect correct..
2. Please check “**USBXpress Device**” driver status is correct.

   Device version 4.0.0.0 for WIN8/WIN7/WINXP.

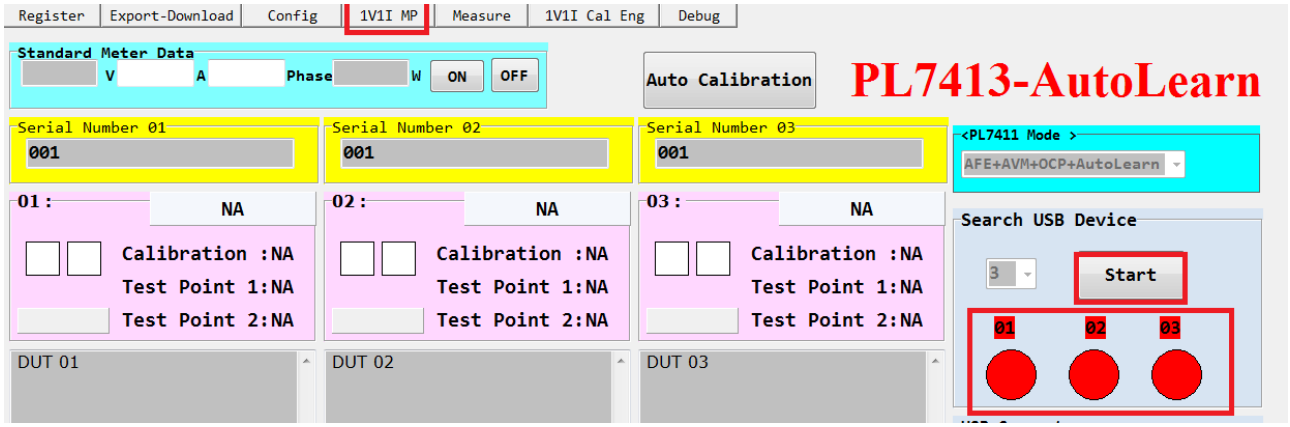


Figure 1.22：Connect fail for “USB”

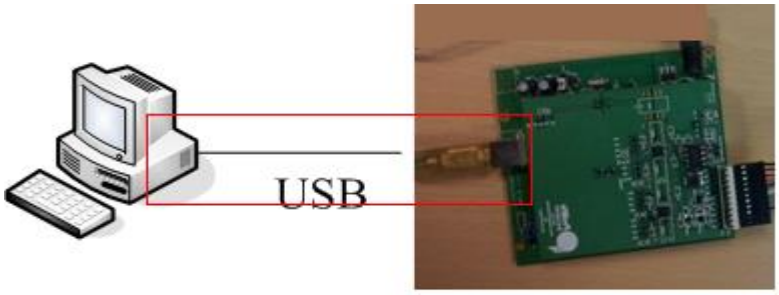


Figure 1.23：Check USB connect

Figure 1.24：Check USB driver status

1.3.2 Check process for Interface communication fail (Figure 1.12)

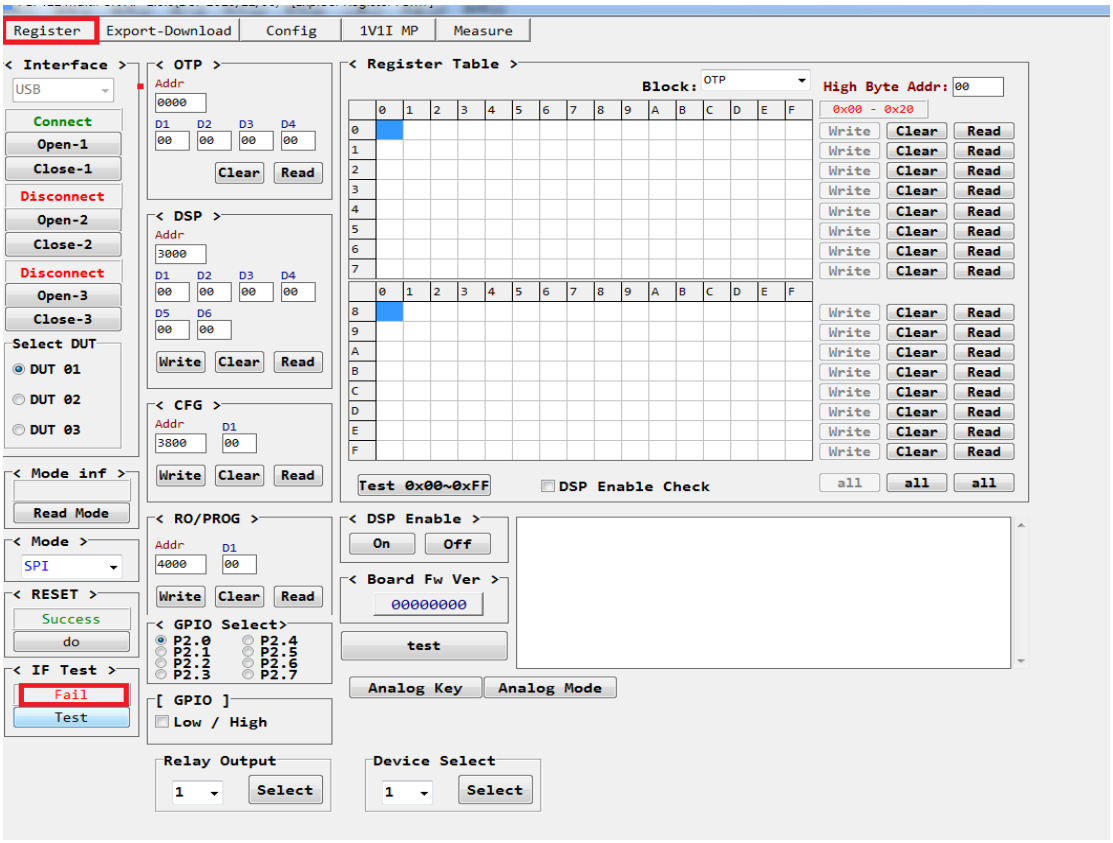


Figure 1.25：Communication fail for “Interface Test”

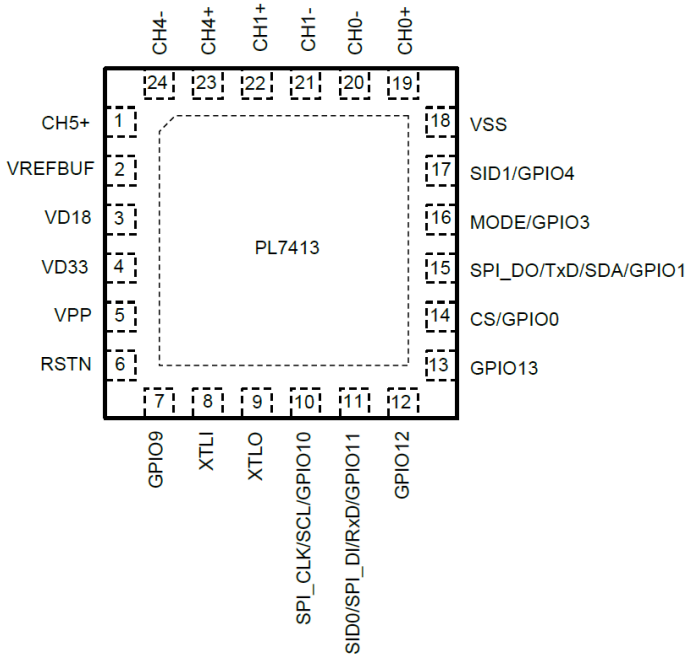


Figure 1.26：PL7413 Pin diagram(QFN24)

1. Please check calibration board connect to PL7413 demo board status. (Figure 1.14).

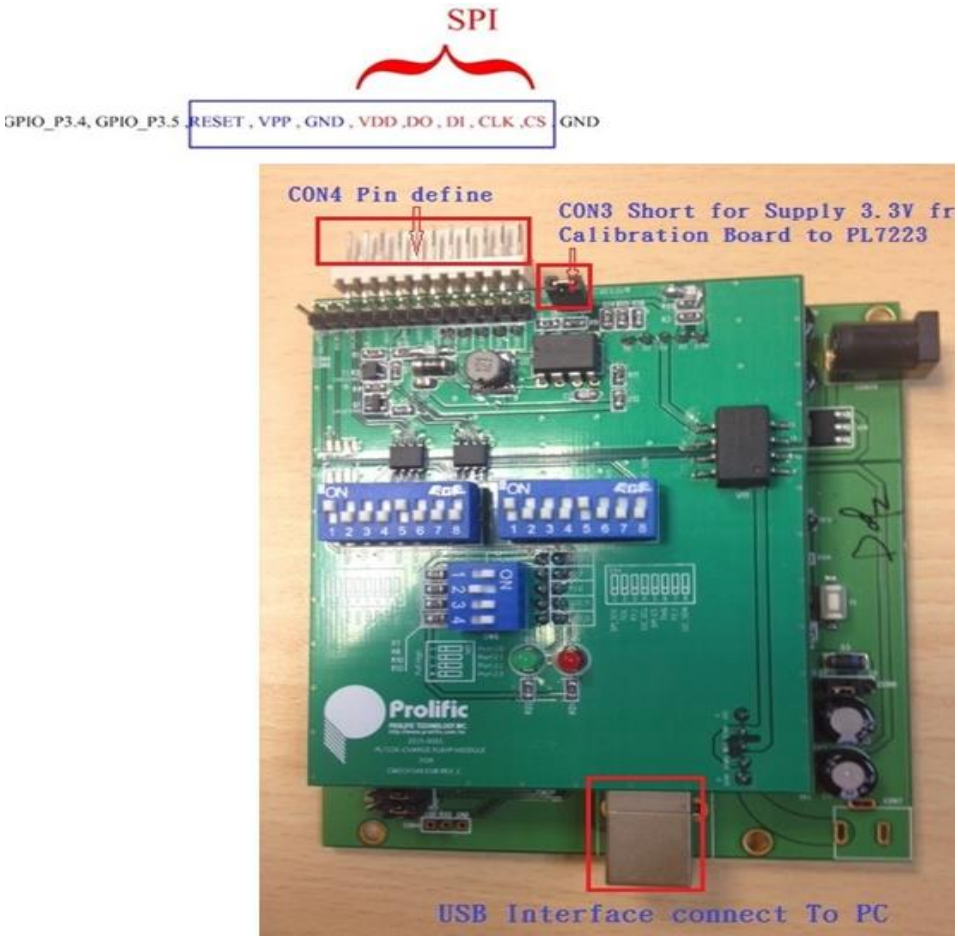


Figure 1.27：Calibration board connect Pin information

2. Check the pin (Reset, VPP, GND, VDD, DO, DI, CLK, CS) from Calibration board connect to SPI of PL7413 demo board

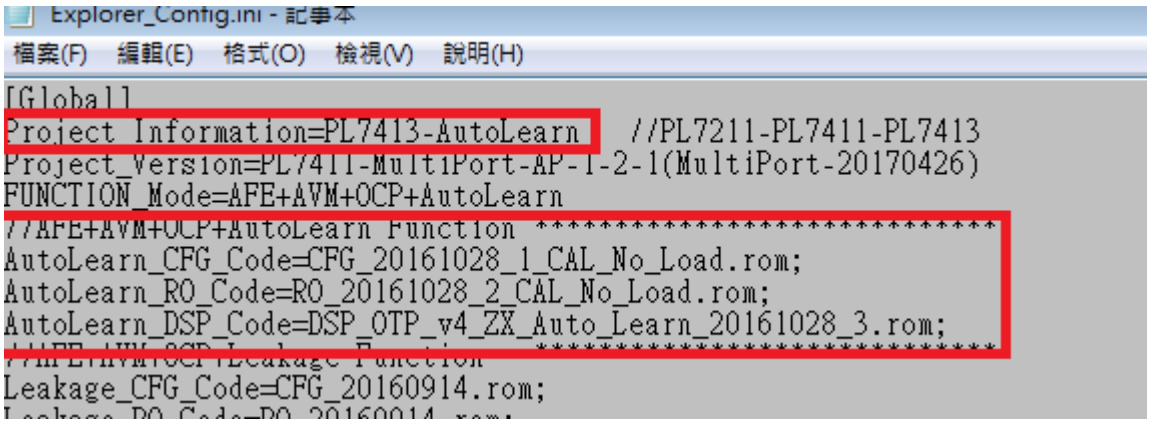
Please reference Section 1.1.8.

1.4 Verify DSP Firmware Information

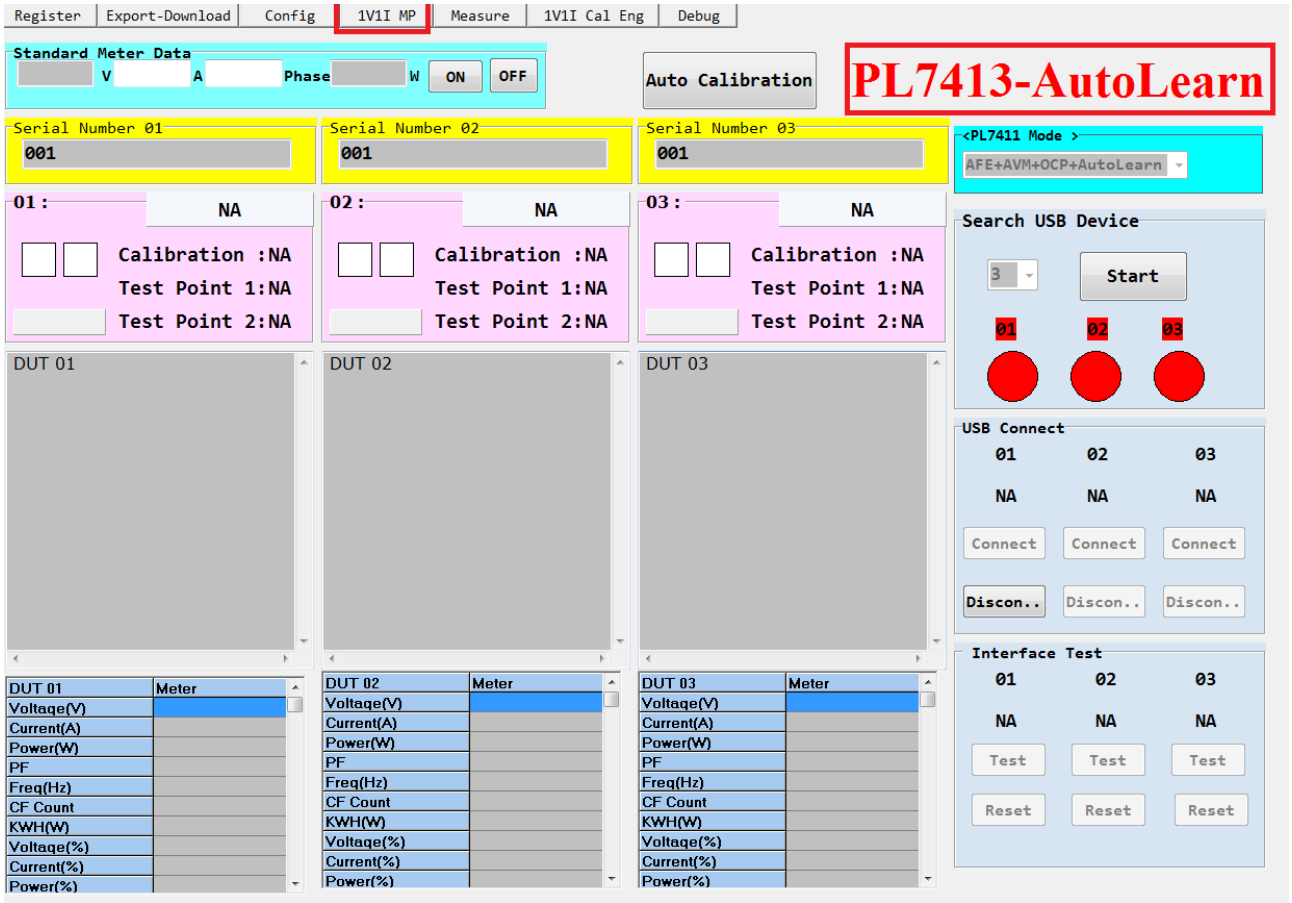
1.4.1 Project name and CFG Code/DSP Code/RO Codes names

If you have the some projects, you can modify the project name to discriminate them. Maybe the DSP codes will not the same, also you can modify CFG Code/DSP Code/RO Cods names.

Please open Explorer_Config.ini for modifying them:



Executive Exploer_Engineering_AP.exe, you can see the project name in the upper right corner:



1.4.2 CHECK CFG CODE, DSP CODE AND RO CODE PATH

please refer to Figure 1.15.

Figure 1.28：CFG/DSP/RO code path

1.4.3 Check DSP firmware code information

Open the “DSP.Rom” to check the information on last line.

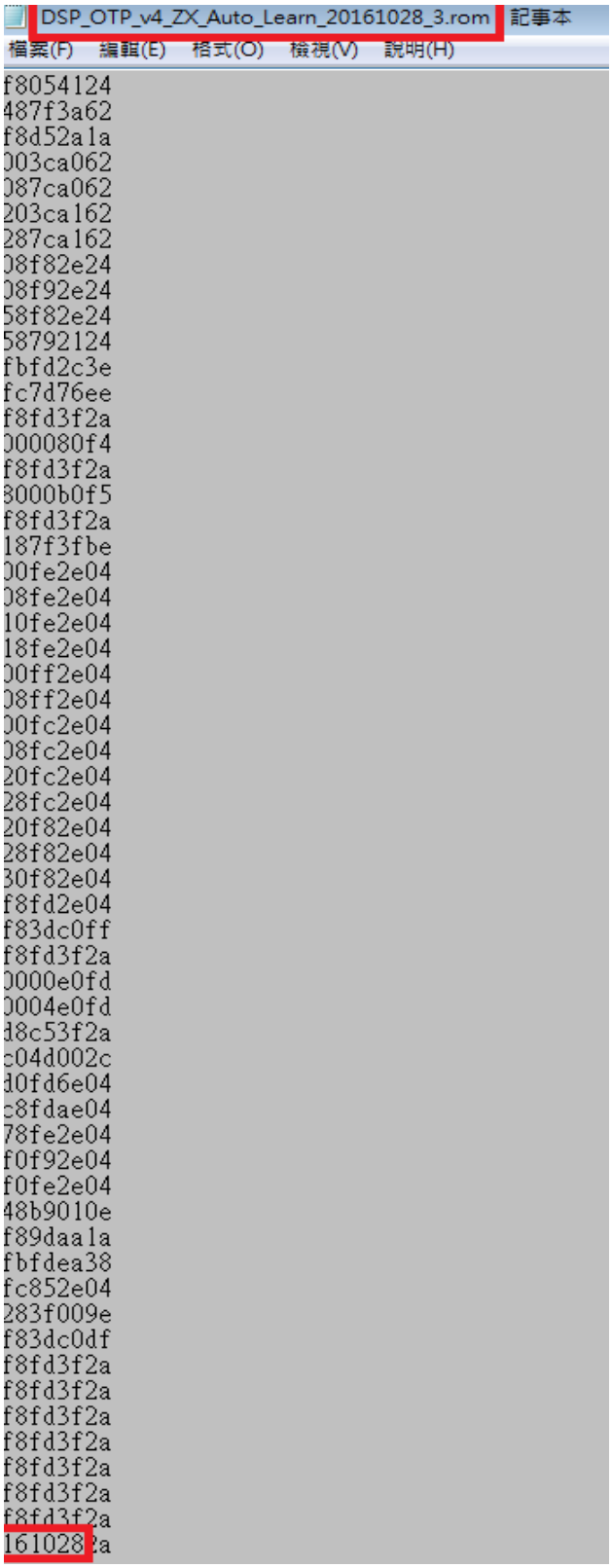


Figure 1.29：DSP.rom content

1.4.4 Read DSP Firmware information with AP

Block is “RAM” and high byte address is 0x2F, Refer to “PL7413 register” Page , Operation method at next chapter .

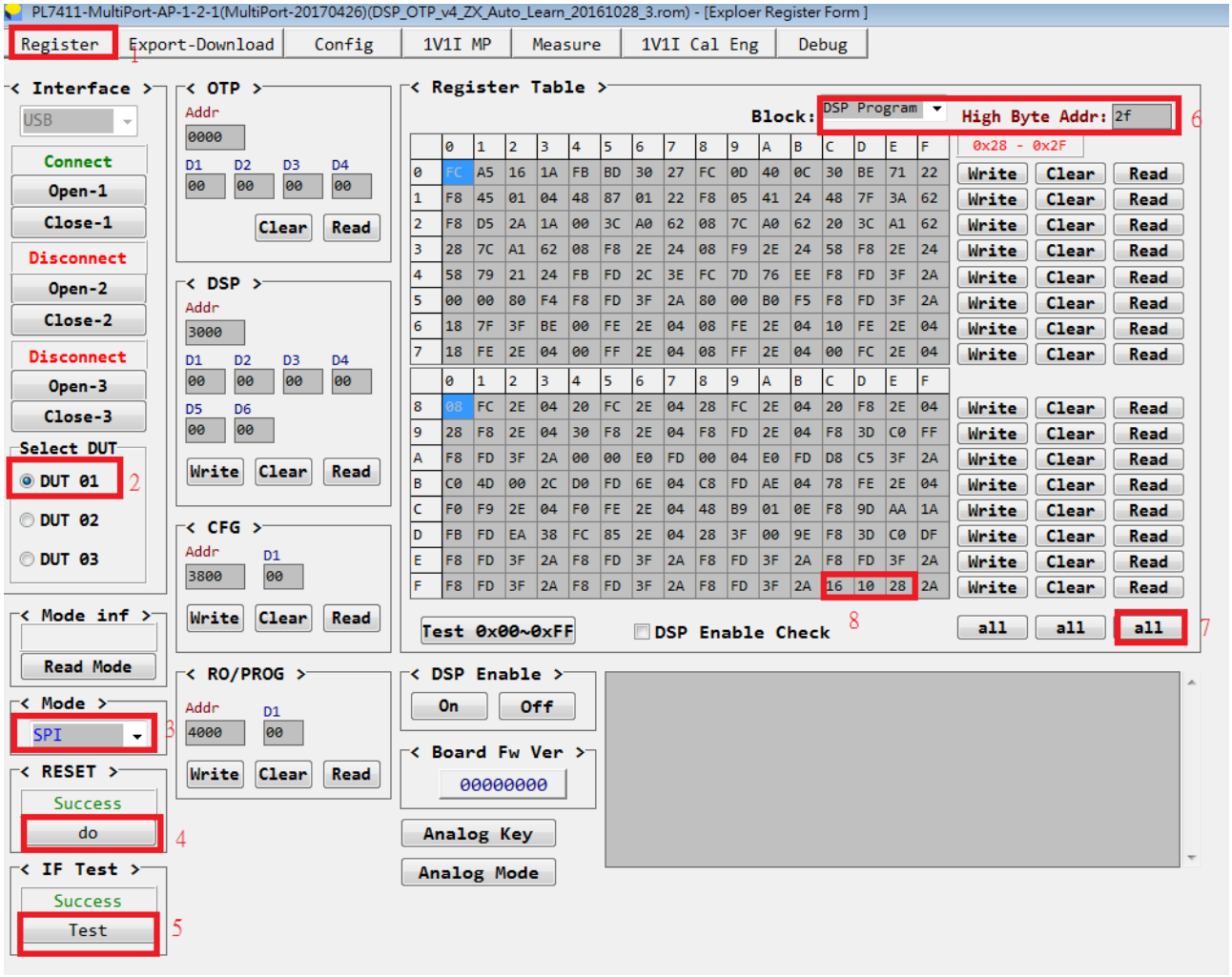


Figure 1.30：DSP firmware information in DSP RAM

Address: 0x2FFC~0x2FFE indicate HEX “16 10 28”
It ‘s mean DSP firmware Information 2016/10/28

1.5 PL7413 Register Operation Method

1.5.1 OPERATION PROCESS

PL7413 OTP and Register mapping, please refer to Figure 1.21 ~ Figure 1.23.

As below is inquire PL7413 data table process, please refer Figure 1.24 :

Step 0：Select “PL7413 Register” Page

Step 1：Select DUT

Step 2：Press “Reset” button to Reset PL7413

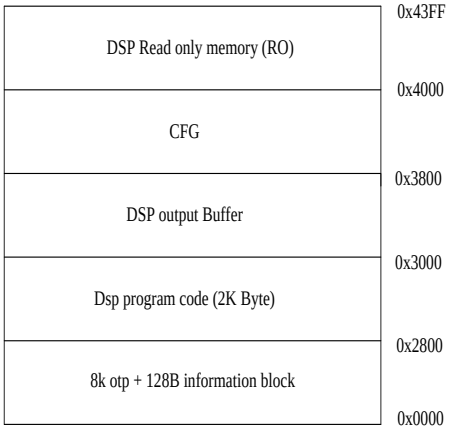
Step 3：Click “Test” to check the devices SPI connect Well.

It will auto test the interface appear “Pass” green word.

Step 4：Select inquire “Block” for OTP(OTP memory), DSP Program(DSP Code RAM),DSP(DSP Buffer), RAM CFG (Configuration Register), RAM DSP RO.

Step 5：Select high byte for inquire “Block” address.

Step 6：Click “Read All” button for “Block”, so you can know PL7413 data table information



- address decode — real address position
- 0x0000~0x27FF : 8k OTP + 128B (0x0000~0x207F)
- 0x2800~0x2FFF : 2k dsp prog (0x2800~0x2FFF)
- 0x3000~0x37FF : 2k dsp buf (0x3000~0x317f)
- 0x3800~0x3FFF : 2k cfg (0x3800~0x39FF)
- 0x4000~0x43FF : 1k dsp ro (0x4000~0x43FF)

Figure 1.31：PL7413 (Register + OTP) Mapping

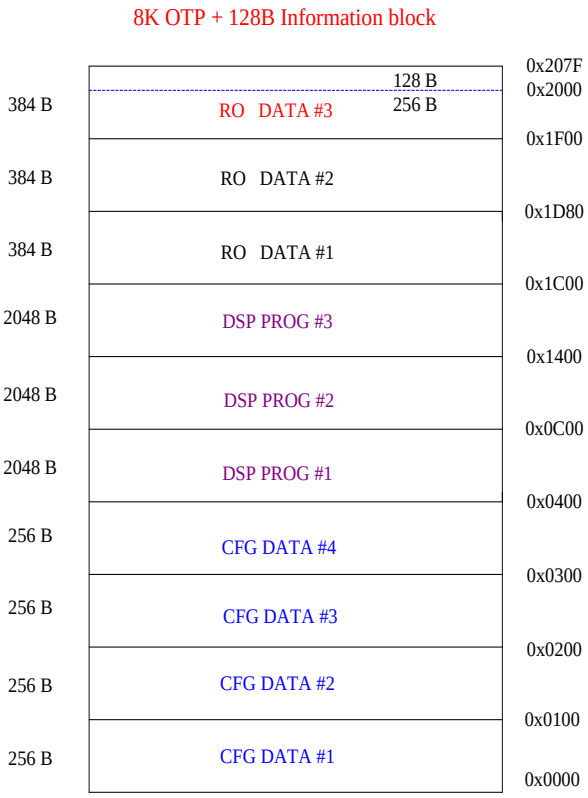


Figure 1.32：OTP memory mapping diagram

Check PL7413 OTP information –1

Step 0：Select “Measure” Page

Step 1：Press “Read” button to Read PL7413 OTP information and DSP version

Step 2：Check DSP usage Last version DSP is in Bank#2

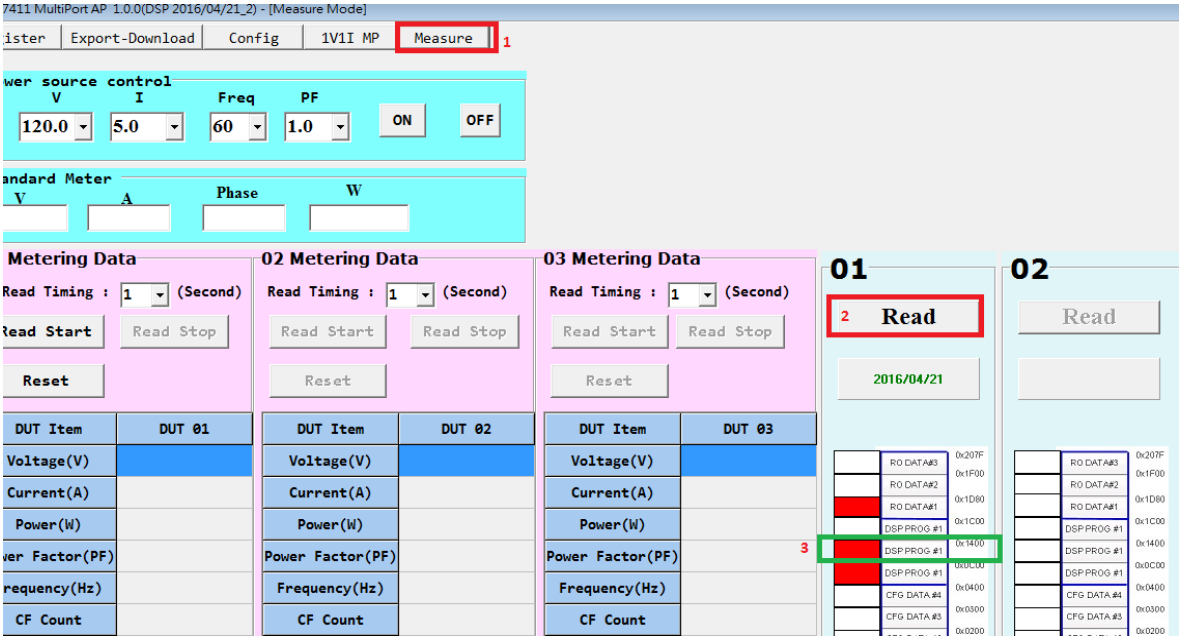


Figure 1.33：PL7413 OTP information

Check PL7413 OTP information –2

Step 0：Select “PL7413 Register” Page, as operation method

Step 1：Select Block “OTP ” And write the DSP Bank#2 address 0x13xx

Step 2：Press “Read all” button to Read PL7413 OTP information

Step 3：Check DSP version in OTP address is in 0x13FC~0x13FF

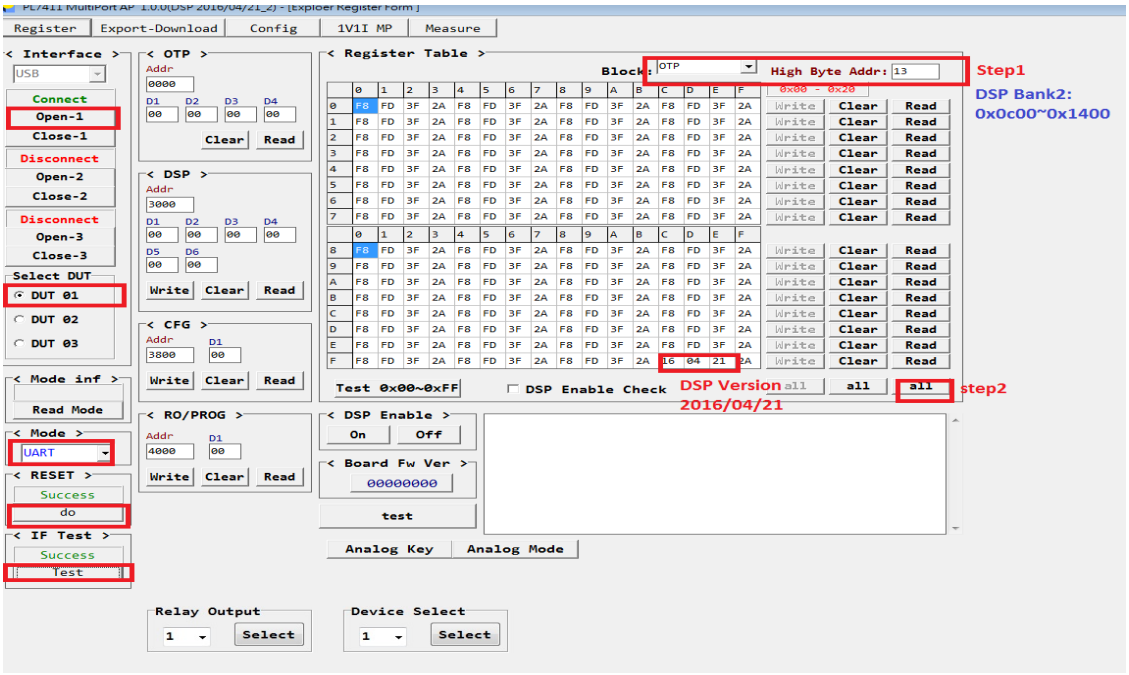


Figure 1.34：PL7413 data table information

2. Auto Calibration

2.1 Introduction

This document describes how to use the Multi-Device calibration AP to do the calibration for the application based on PL7413 chips.

2.2 Environment Setup

Please executive **PL7413 Multi MP Tool.msi**

1. After Install complete , In folder(C :\Program Files\Prolific\PL7413 Multi MP Tool). Will exist 4 file --"hscom.dll" ,“Three_PL7413_Config.ini” , “PL7221AfeOcpMulti.EXE “ and “SiUSBXp.dll” together.

2. Check file "DSP.rom" and "CFG.rom" inside folder in following path:

-->C:\PL7413\Multi\AFE\CFG\CFG_20141009.rom

-->C:\PL7413\Multi\AFE\DSP\DSP_20141009.rom

2.3 Power Source Connection

2.3.1 Voltage Connection

KP1001 has 4 Voltage lines(U1~U4), HS3103 has 3 Voltage lines(U1~U3). Please connect U1(+,-) to product 1; U2(+,-) to product2; U3(+,-) to product3.

Please reference the next page.

2.3.2 Current Connection

KP1001 and HS3103 has 1 current line(I). Please connect “I-” to product1, connect “I+” to product3. And uses the series connection method to connect product 1 to product2 to product 3;

Please reference the next page.

Figure 2.1：Power Source Connect

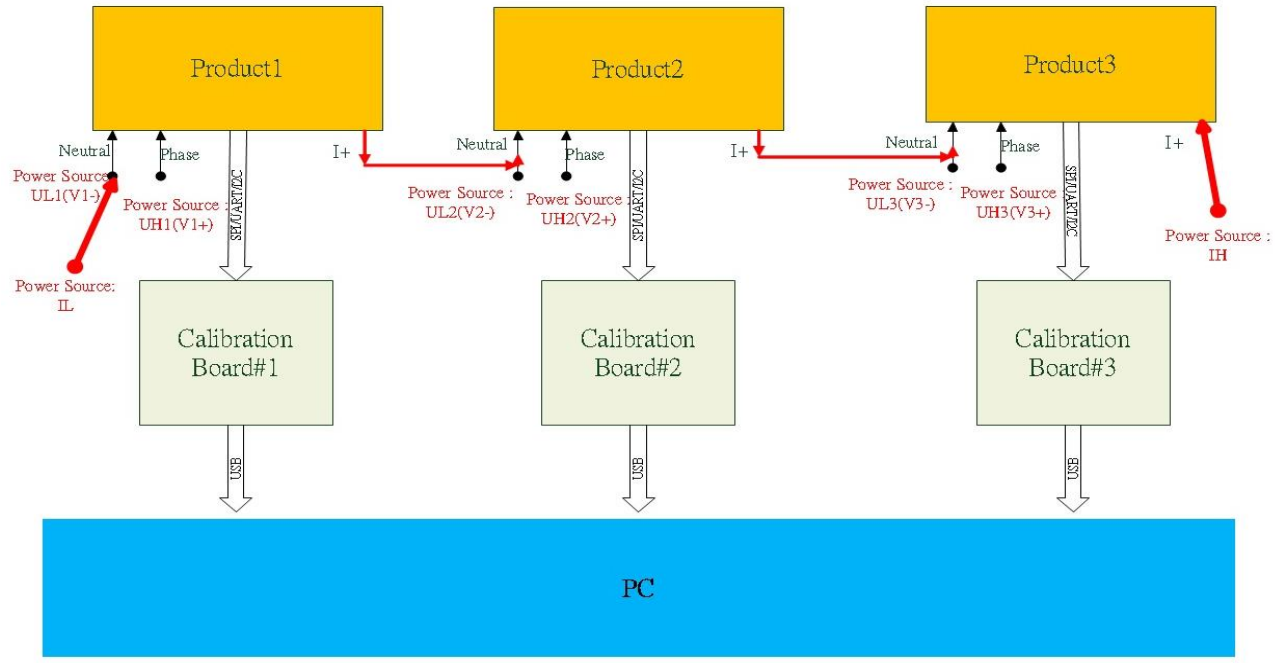


Figure 2.2：PL7413 device Connect with Power Source Block Diagram

2.4 Initialize Setting

Initialize Settings(Three_PL7413_Config.ini)

[Global]

Input_Key_Eable=true;

(1). If you don't want to input and modify the power , accuracy limit and target values setting, please set:

Input_Key_Eable=false;

(2). If your PL7413 board has not the crystal, please select Inter OSC.

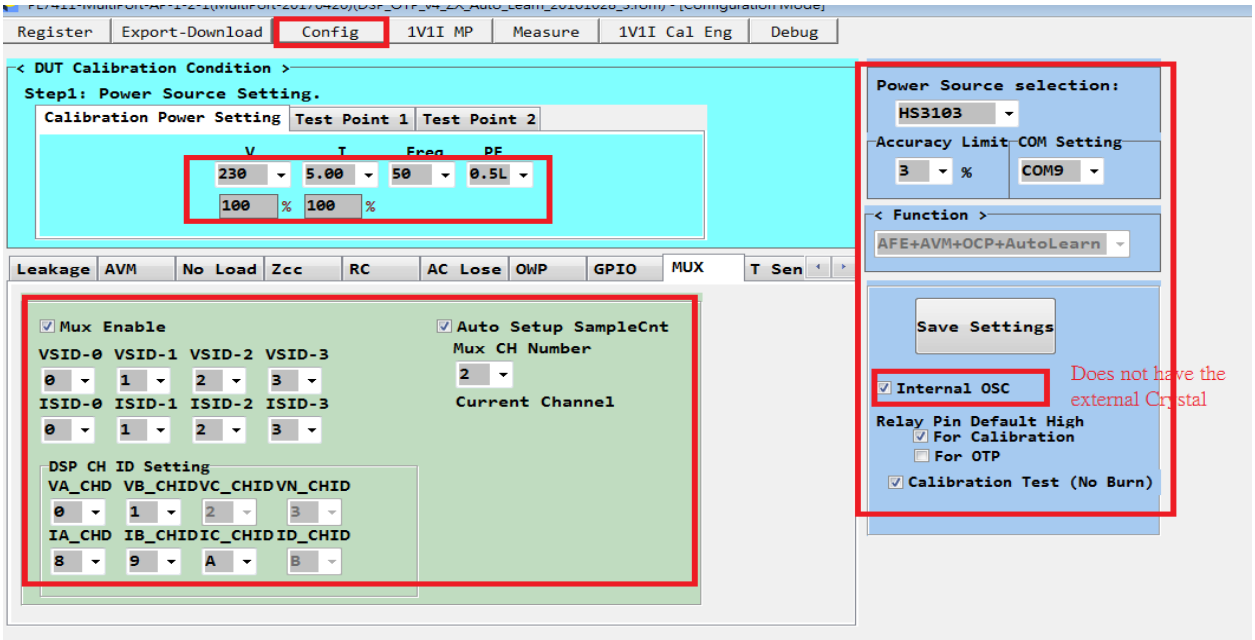


Figure 2.3：Initialize setting

2.5 Configure Setting

1. Set the accuracy limit:

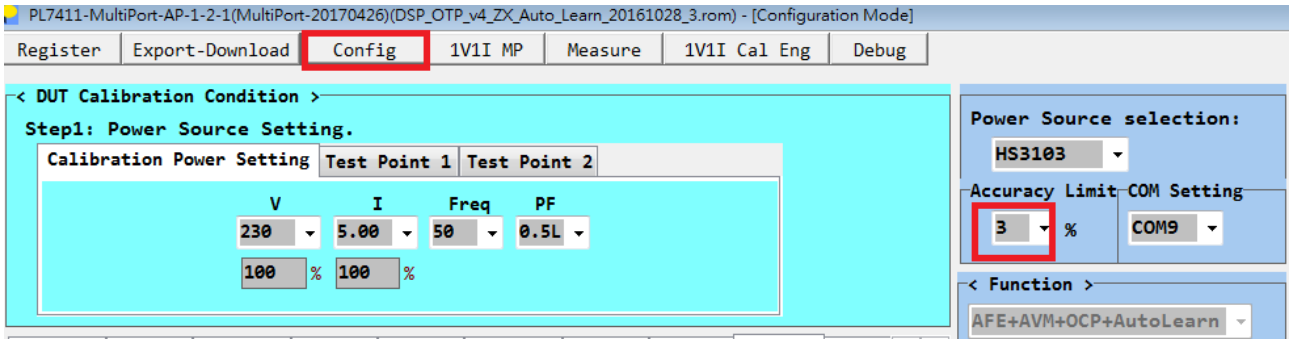


Figure 2.4：Accuracy Setting

2. Power source setting :

Use HS3103 as power source,and select the RS232 COM port:

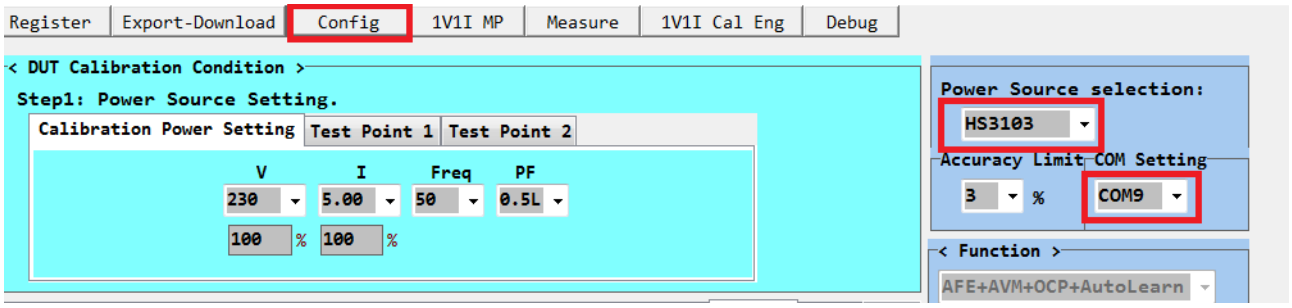
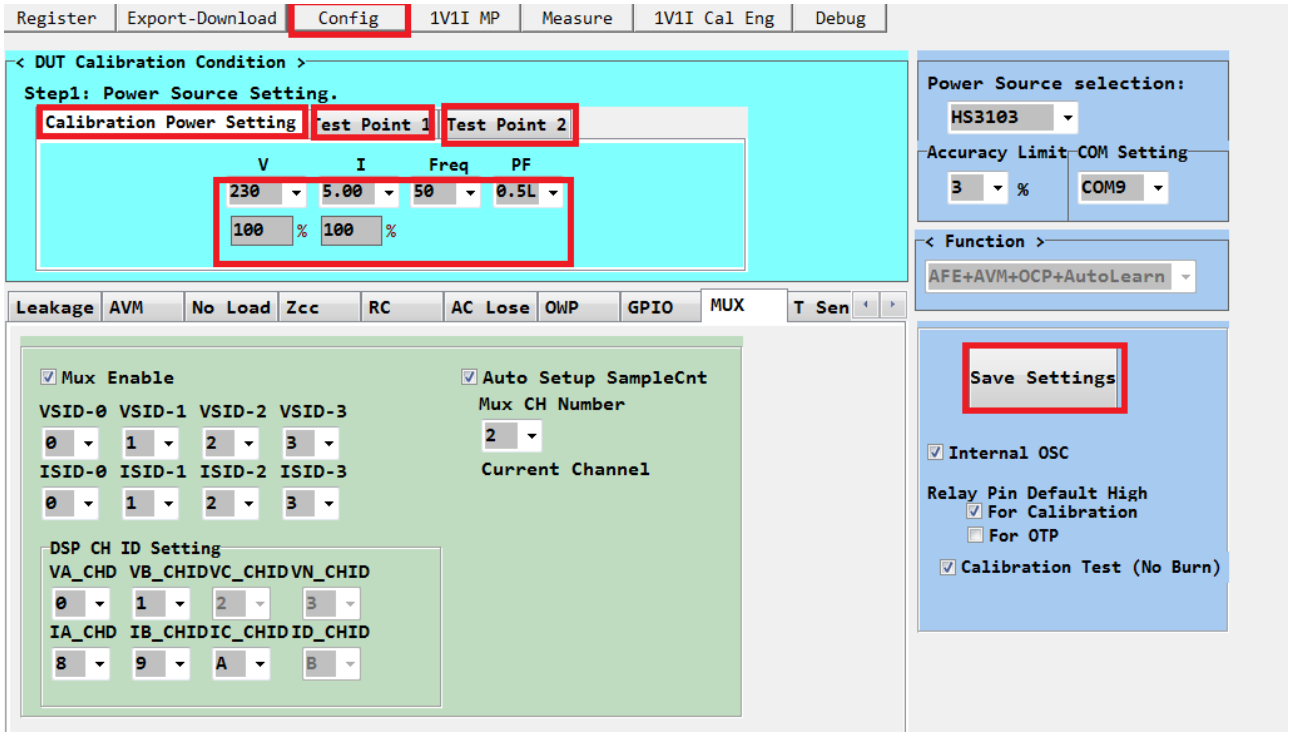


Figure 2.5：Power Source COM port setting

3. Power Source Calibration Point Setting



4. Test Point 1 & 2 setting (for HS-3103 & KP1001, “NA” will not support)

Figure 2.6：Save Setting

2.6 Calibration Process with the “NA” Power Source

1. If you have other power source, please select the “NA” power source type
2. If you just want to try it , not burn the codes into OTP, please enable “Calibration Test(No Burn)” in Function Setting. Else it will burn the codes into OTP after auto calibration success.

3. Select the “Production” form
4. Please follow the steps below.

Step1：Key-in the serial number1~3.

Step2：Wait your power source stable, then input the power information (V/I/Phase/W) in ”Stand Meter Data”

Step3：Press button “Auto Calibration” Start.

Step4：If the calibration pass, will show “PASS”

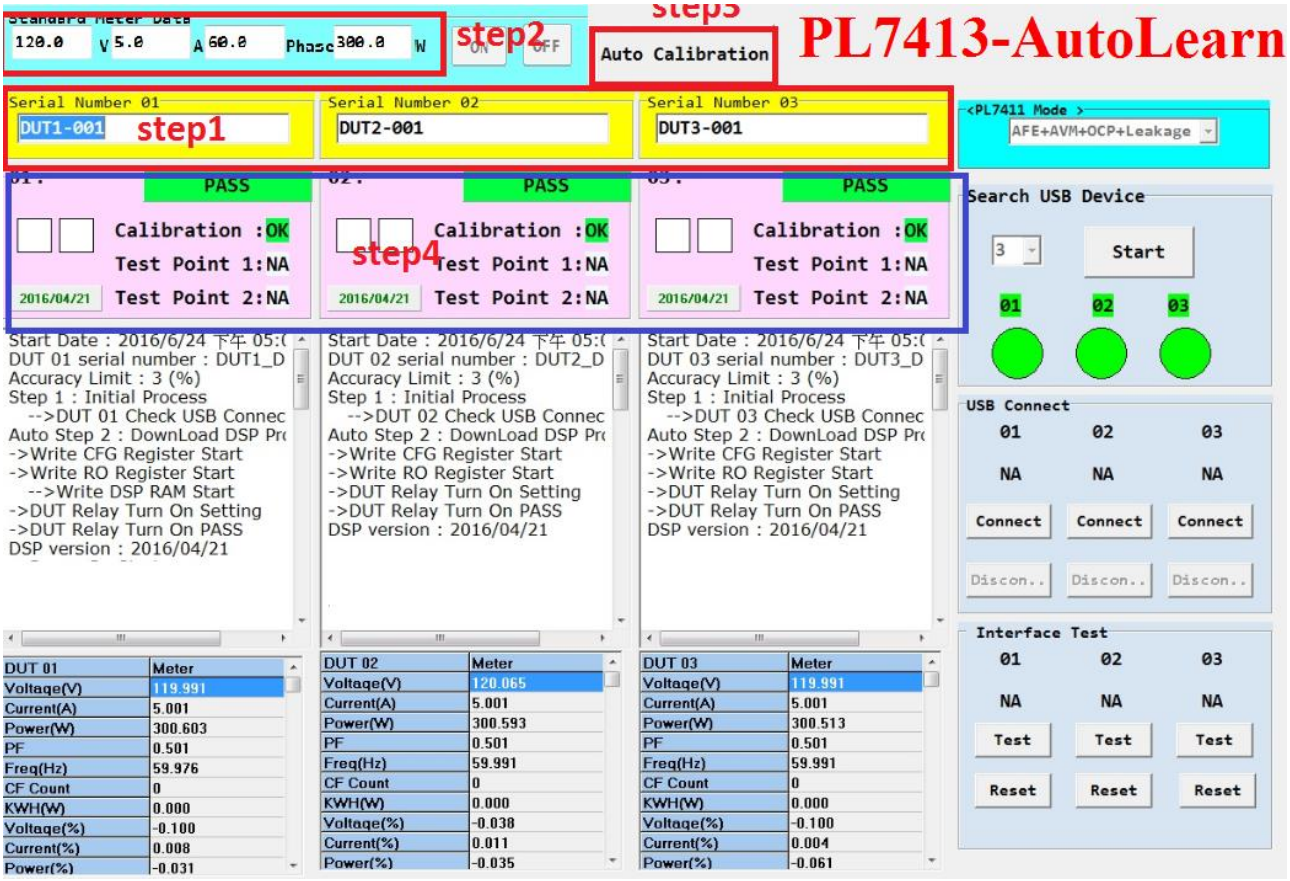


Figure 2.7：Select “NA” Meter Data setting

Note：If your power source have Phase input(Ex: PF=0.5L　, Input Phase=60°) , Please input as you need, Else input Phase=0 (PF=1.0).

2.7 Calibration Process with the “HS3103” Power Source

1. Please select the “HS3103” power source type

If you can control the “HS3103”(power source + USB to UART standard meter), please follow the steps below.

Step1：Select COM setting

Step2：Set “Accuracy Limit(%)” according to your requests.

Step3：Select Power Setting at “Power Source Calibration Point Setting” , “Test Point1” and “Test point 2”

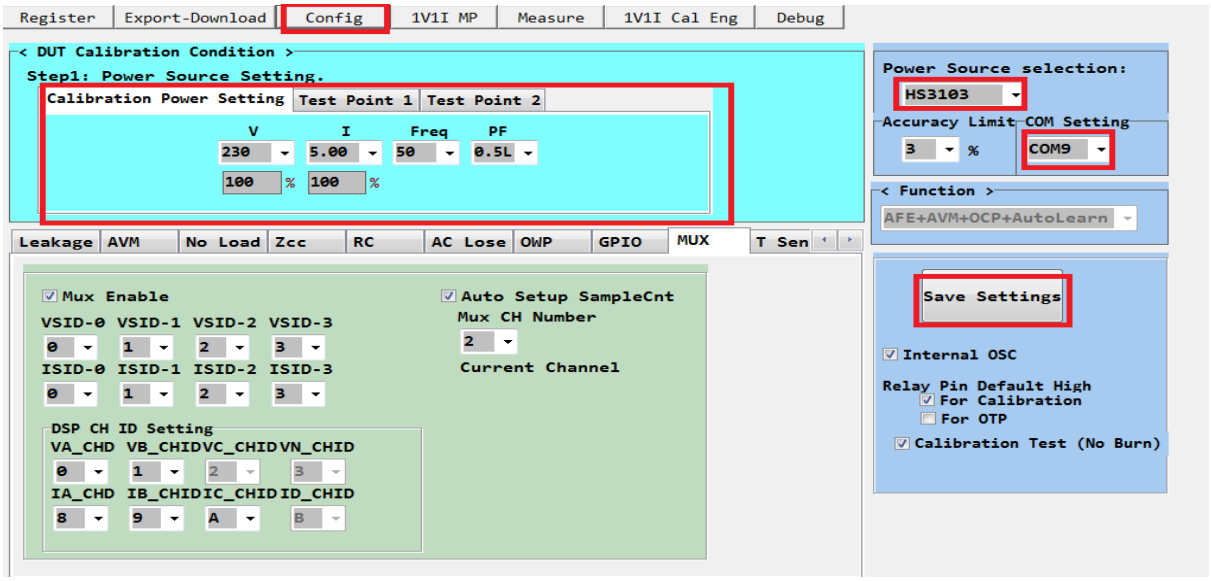


Figure 2.8：HS3103 Setting

Step4：Press button “Auto Calibration” Start

Step5：Here will show voltage, current, phase and instantaneous power that read from HS3103.

Step6：If the calibration pass, they will show “PASS”

Note: You just setting the Step1, Step2 and Step3 one time. then Press

“Save Setting” button, Next Execute the AP the parameter will the same.

Figure 2.9：HS3103 Calibration

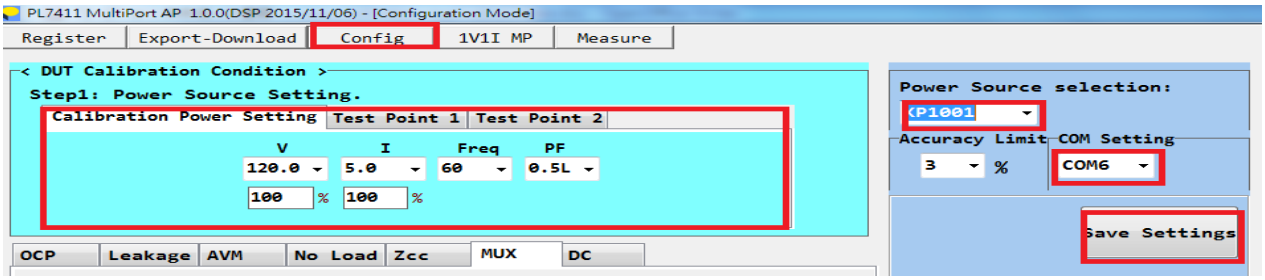
2.8 Calibration Process with the “KP1001” Power Source

1. Please select the “KP1001” power source type

If you can control the “KP1001”(power source + USB to UART standard meter), please follow the steps below.

Step1：Select COM setting

Step2：Set “Accuracy Limit(%)” according to your requests.



Step3：Select Power Setting at “Power Source Calibration Point Setting” , “Test Point1” and “Test point 2”

Figure 2.10：KP1001 Setting

Step4：Press button “Auto Calibration” Start.

Step5：Here will show voltage, current, phase and instantaneous power that read from KP1001.

Step6：If the calibration pass, they will show “PASS”

Note：You just setting the Step1, Step2 and Step3 one time. then Press

“Save Setting” button, Next Execute the AP the parameter will the same.

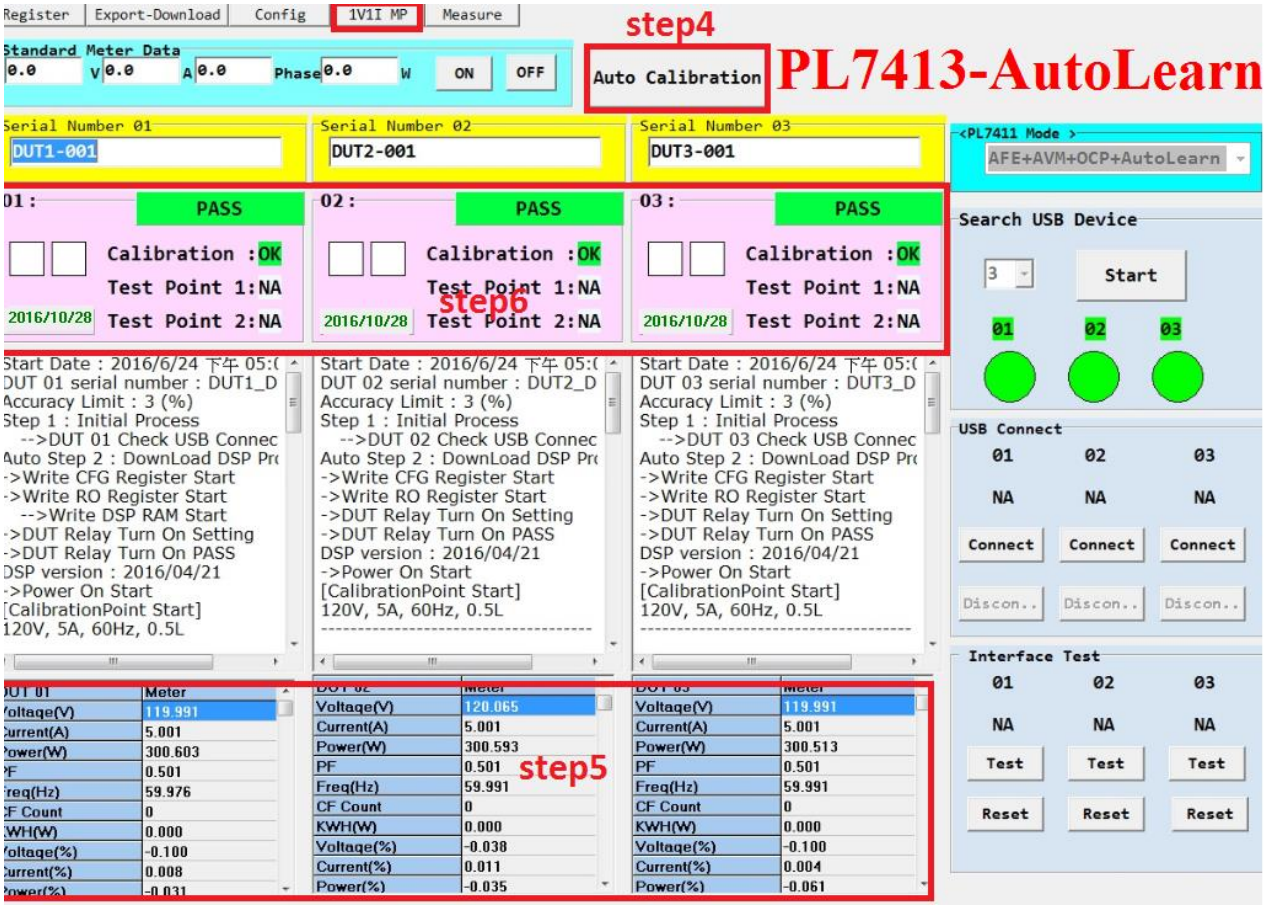


Figure 2.11：KP1001 Calibration

2.9 Report Generator

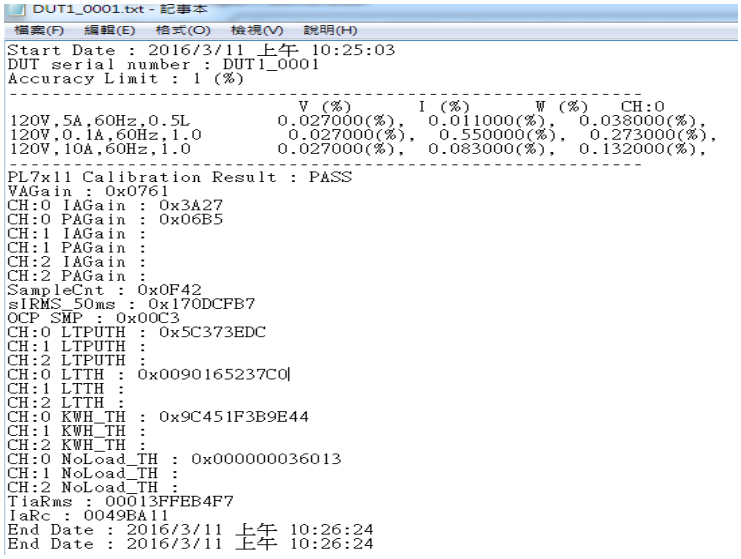
Once the calibration is completed by the calibration AP, some data will be generated in the following path:

A: Report Data Path(.txt) --> "C:\Explorer\Calibration\Report\DUT_001.txt"

C:\Explorer\Calibration\Report\DUT_002.txt

C:\Explorer\Calibration\Report\DUT_003.txt

Example:



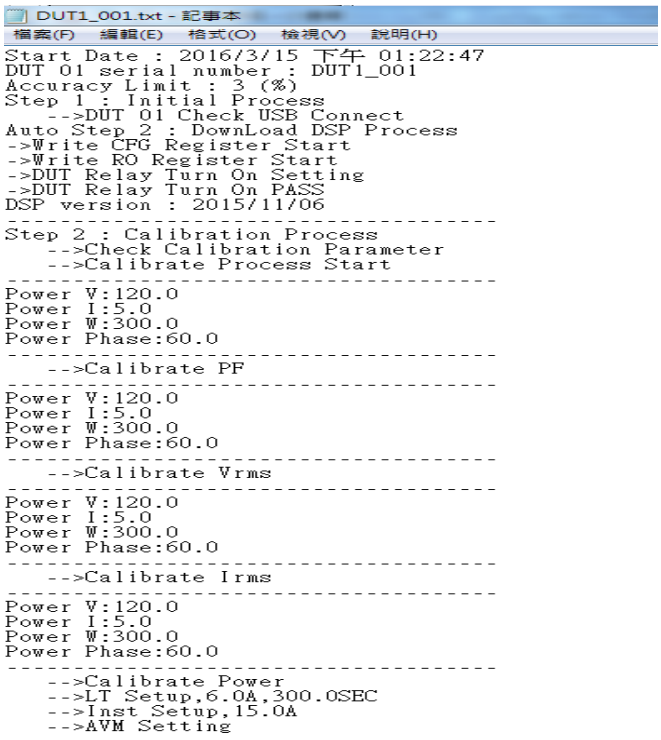
```
DUT1_0001.txt - 記事本
檔案(F)  編輯(E)  格式(O)  檢視(V)  說明(H)
Start Date : 2016/3/11 上午 10:25:03
DUT serial number : DUT1_0001
Accuracy Limit : 1 (%)
-------------------------------------------------------------
                        V (%)       I (%)       W (%)    CH:0
120V,5A,60Hz,0.5L     0.027000(%),  0.011000(%),  0.038000(%),
120V,0.1A,60Hz,1.0     0.027000(%),  0.550000(%),  0.273000(%),
120V,10A,60Hz,1.0     0.027000(%),  0.083000(%),  0.132000(%),
-------------------------------------------------------------
PL7x11 Calibration Result : PASS
VAGain : 0x0761
CH:0 IAGain : 0x3A27
CH:0 PAGain : 0x06B5
CH:1 IAGain :
CH:1 PAGain :
CH:2 IAGain :
CH:2 PAGain :
SampleCnt : 0x0F42
sIRMS_50ms : 0x170DCFB7
OCP SMP : 0x00C3
CH:0 LTPUTH : 0x5C373EDC
CH:1 LTPUTH :
CH:2 LTPUTH :
CH:0 LTTH : 0x0090165237C0
CH:1 LTTH :
CH:2 LTTH :
CH:0 KWH_TH : 0x9C451F3B9E44
CH:1 KWH_TH :
CH:2 KWH_TH :
CH:0 NoLoad_TH : 0x000000036013
CH:1 NoLoad_TH :
CH:2 NoLoad_TH :
TiaRms : 00013FFEB4F7
IaRc : 0049BA11
End Date : 2016/3/11 上午 10:26:24
End Date : 2016/3/11 上午 10:26:24
```

Figure 2.12：Report Data and Path

B: Message Data Path(.txt) -->C:\Explorer\Calibration\Message\DUT_001.txt"

C:\Explorer\Calibration\Message\DUT_002.txt

C:\Explorer\Calibration\Message\DUT_003.txt



```
DUT1_001.txt - 記事本
檔案(F)  編輯(E)  格式(O)  檢視(V)  說明(H)
Start Date : 2016/3/15 下午 01:22:47
DUT 01 serial number : DUT1_001
Accuracy Limit : 3 (%)
Step 1 : Initial Process
    -->DUT 01 Check USB Connect
Auto Step 2 : DownLoad DSP Process
->Write CFG Register Start
->Write RO Register Start
->DUT Relay Turn On Setting
->DUT Relay Turn On PASS
DSP version : 2015/11/06
---------------------------------------
Step 2 : Calibration Process
    -->Check Calibration Parameter
    -->Calibrate Process Start
---------------------------------------
Power V:120.0
Power I:5.0
Power W:300.0
Power Phase:60.0
---------------------------------------
    -->Calibrate PF
---------------------------------------
Power V:120.0
Power I:5.0
Power W:300.0
Power Phase:60.0
---------------------------------------
    -->Calibrate Vrms
---------------------------------------
Power V:120.0
Power I:5.0
Power W:300.0
Power Phase:60.0
---------------------------------------
    -->Calibrate Irms
---------------------------------------
Power V:120.0
Power I:5.0
Power W:300.0
Power Phase:60.0
---------------------------------------
    -->Calibrate Power
    -->LT Setup,6.0A,300.0SEC
    -->Inst Setup,15.0A
    -->AVM Setting
```

Figure 2.13：Message Data and Path

2.10 How to export DSP / RO / CFG to file

Please select the Export-Download Page ,

Step 1: Click Export DSP Program / Export CFG data / Export RO data button

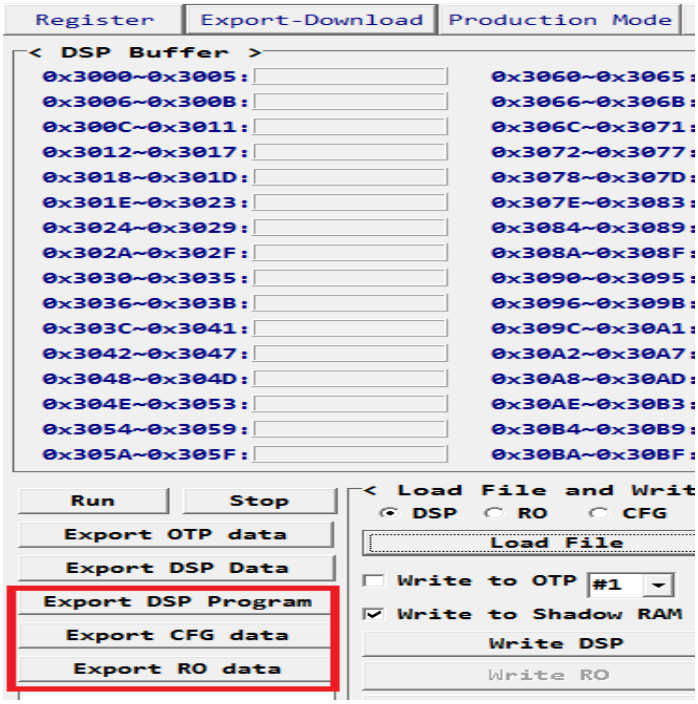


Figure 2.14: AP export code

Step 2: Save DSP.rom / CFG.rom / RO.rom to your specify path

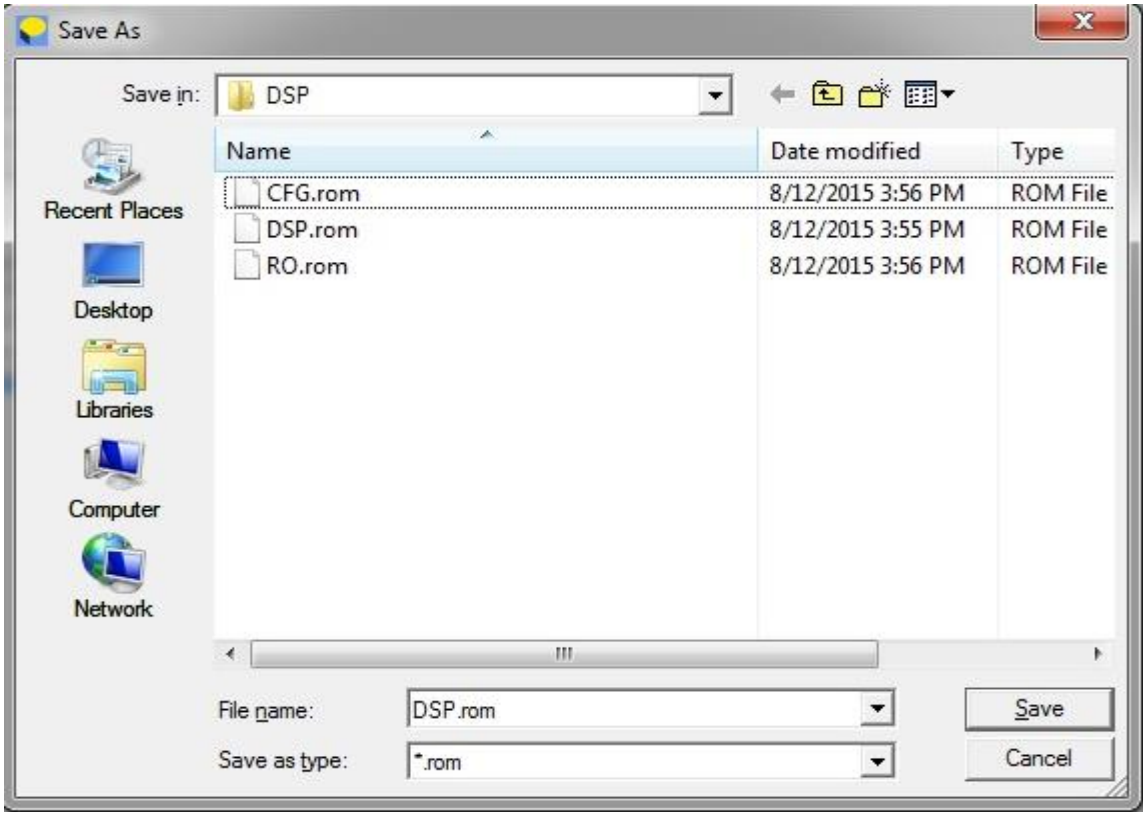


Figure 2.15: AP save code

3. Power Protection Function Description

The power protection function of PL7413 prevents overload condition which may cause equipment overheat or even catch fire.

When the load current exceeds the rated current (or pre-configured current threshold, ILT/ST) for a specified time, the relay will be switched off to prevent overheat condition. Both the overload threshold current and delay time to switch off relay can be configured through the AP provided by Prolific.

3.1 Power Protection Operation

The operation of protection is shown as below Figure.

If the load current (ILOAD) is less than or equal to ILT/ST, the relay is always ON.

If the load current (ILOAD) is larger than ILT/ST, the delay time (TOFF) to switch off relay will be shorter. Please refer to section 6.2.2 to calculate (TOFF).

Both the ILT/ST and TLT/ST can be configured by the application software provided by Prolific.

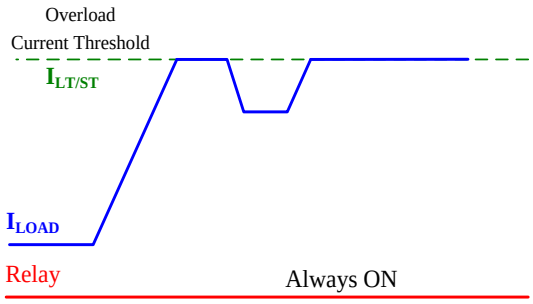


Figure 3.1：Long/Short time protection (ILOAD = ILS/ST)

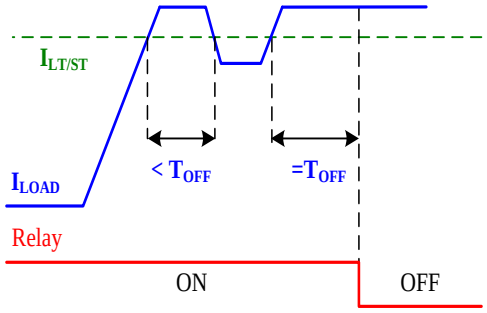


Figure 3.2：Long/Short time protection (ILOAD > ILS/ST)

3.2 Calculate the delay time to switch off relay

The delay time, T_{OFF}, can be calculated by the following equation:

$$T_{OFF} = T_{LT/ST} \times \left(\frac{I_{LT/ST}}{I_{LOAD}} \right)^2$$

For example,

- Short time threshold current (I_{ST}) is set as 30A
- T_{ST} is set as 5sec
- Exact load current is 50A.

We can obtain the delay time to switch off delay is:

$$T_{OFF} = 5 \times \left({}^{30}\!/\!_{50} \right)^2 = 1.8\,\text{sec}$$

3.3 OCP Protection

Ex: In Our Demo board, If Calibration current is 5A:

6A (1.2X) ~9.5A (1.9X) relay pick time = [T=240/ (1.2) ^2] ~ [T=240/ (1.9) ^2]

10A (2.0X) ~14.5A (2.9X) relay pick time = [T=20/ (2.0) ^2] ~ [T=20/ (2.9) ^2]

15A (3.0X) ~49.5A (9.9X) relay pick time = 1ms

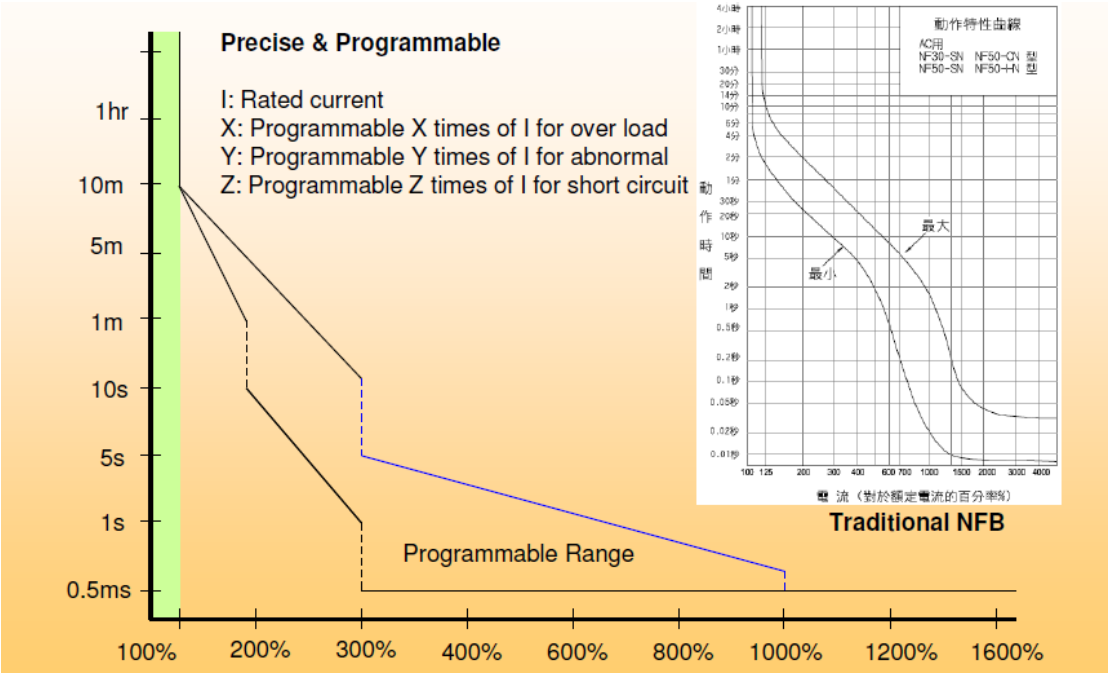


Figure 3.3：Programmable Range

3.4 OCP Threshold

LT_PU:	1.2X	PICK TIME	300	LT_PU^2*T = 432
INST:	3X	Trip time is 1		millisec
		SampleCnt		1953

Table 3.1：Leakage and OCP setting Table

3.5 AVM (Auto Voltage Margin)

PL7413 has two ADC input channels, It can be one voltage input and one current input or two current inputs. It depends on the DSP program definition. The DSP have 2K words instruction memory space and 48 words data space.

Prolific had provided some pre-defined power monitor functions, like, AVM, Power Protection.

3.6 AVM introduction

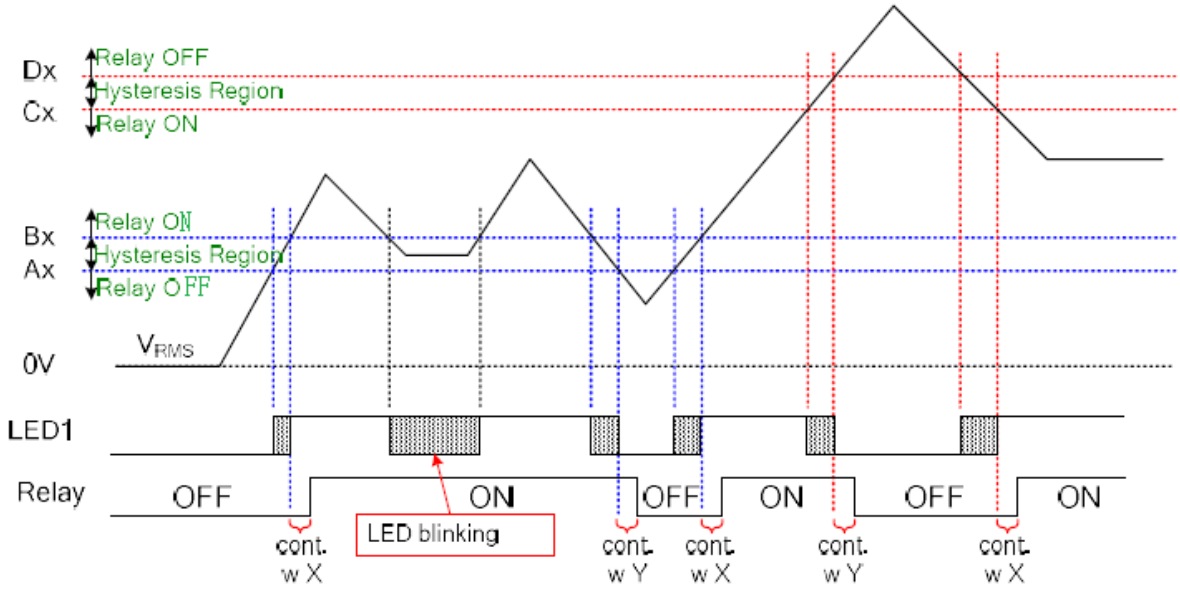
By setting AX /BX/ CX/ DX, you can use PL7413 AVM function shown as follow figure.

PL7413 will auto switch relay on/off when input voltage threshold are setting

In Hysteresis region relay will keep before status, until over /under region.

Point	Voltage	Relay	Ratio
Ax	66	relay off	0.6
Bx	88	relay on	0.8
Cx	132	relay on	1.2
Dx	154	relay off	1.4

Table 3.2：AVM threshold



3.7 DSP AVM flow

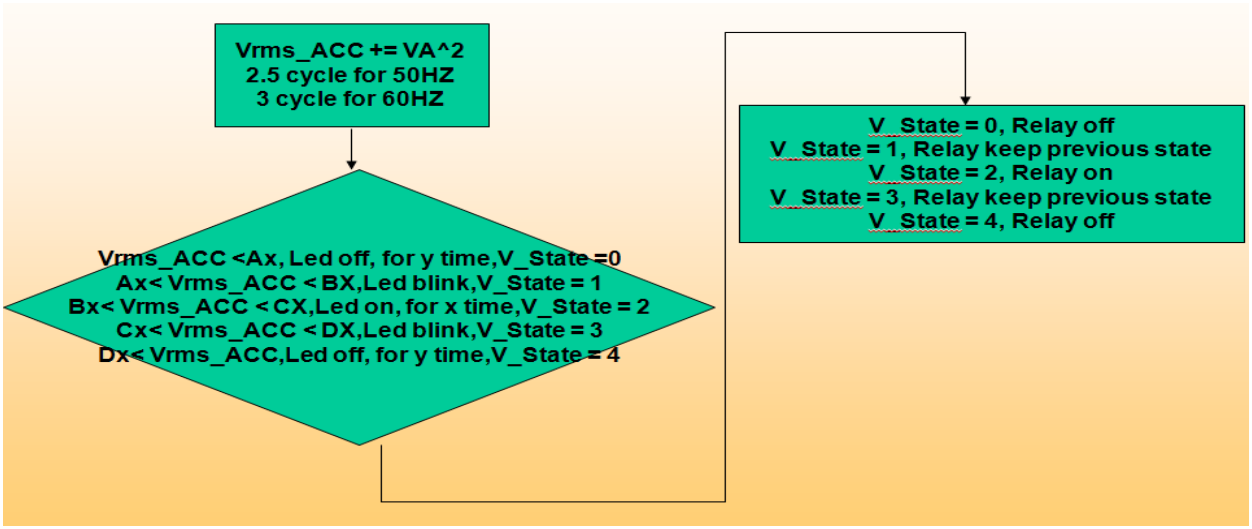


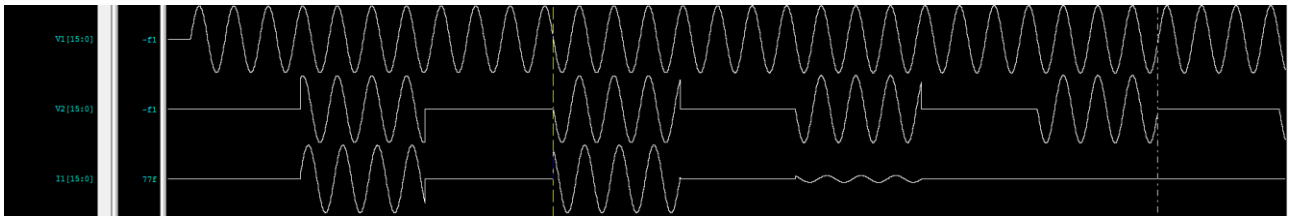
Figure 3.4：AVM DSP flow

For example , if the sample counts/second is 3906(0xF42), then the AVM sample count2 is 3/60*(0xF42) = 195(0xC3)

$$VTH = \left(\frac{V_{RMS} - V_{offset}}{V_{gain} / 2^{18}} \right) \times SC2$$

3.8 Introduce Auto Learn ZCC

PL7413 demo board have embedded the Auto learn function , DSP will auto lean when 1st relay on/off, Relay on base on V (voltage) and Relay off base on I(current), When on load, relay off will base on V. VB channel is use for load voltage detection:



If the PL7413 can do Zero crossing control, then it is also on behalf of it can do random phase control ??

Answer:

(1) Yes, because the random phase control is zero, but the only difference is that zero crossing is zero and the random phase is nonzero.

(2) Part of the zero is divided into two, one voltage, the other is the current, the current auto learn ZCC, Power On after 3 (relay on / off) is learning,

(3) 3 times after the completion of learning, relay on is to see the voltage zero, relay off is to see the current zero, so you can achieve random phase control.

AC Calculate Method

3.9 Parameter Address and Mapping

Those parameter is mapping in DSP As below

0x3000~0x3005	VC	VB	VA	0x3060~0x3065	VArms	0x30C0~0x30C5	IA2_ACC	0x3120~0x3125	PA
0x3006~0X300B	VCOS	VBOS	VAOS	0x3066~0X306B	VBrms	0x30C6~0X30CB	IA2_SUM	0x3126~0X312B	PB
0x300C~0X3011	VC_LLCNT	VB_LLCNT	VA_LLCNT	0x306C~0X3071	IArms	0x30CC~0X30D1	STACC_IA	0x312C~0X3131	CF_CNTA
0x3012~0X3017	VC_LLIDX	VB_LLIDX	VA_LLIDX	0x3072~0X3077	IBrms	0x30D2~0X30D7	LTACC_IA	0x3132~0X3137	CF_CNTB
0x3018~0X301D	ZXCCnt	ZXBCnt	ZXACnt	0x3078~0X307D	TVA_rms	0x30D8~0X30DD	IB2_ACC	0x3138~0X313D	PA0
0x301E~0X3023	ZXCStart	ZXBStart	ZXAStart	0x307E~0X3083	TVB_rms	0x30DE~0X30E3	IB2_SUM	0x313E~0X3143	VARMS_ACC
0x3024~0X3029	ZXCStop	ZXBStop	ZXAStop	0x3084~0X3089	TIA_rms	0x30E4~0X30E9	LeakACC_IB	0x3144~0X3149	
0x302A~0X302F	VCZXTO	VBZXTO	VAZXTO	0x308A~0X308F	TIB_rms	0x30EA~0X30EF	TMP9	0x314A~0X314F	PB0
0x3030~0X3035	VCState	VBState	VAState	0x3090~0X3095	TPA	0x30F0~0X30F5	UV_L_Value	0x3150~0X3155	
0x3036~0X303B	Temp_Cnt		VA0	0x3096~0X309B	TPB	0x30F6~0X30FB	UV_H_Value	0x3156~0X315B	
0x303C~0X3041	IC	IB	IA	0x309C~0X30A1	KWHVAL1	0x30FC~0X3101	OV_L_Value	0x315C~0X3161	
0x3042~0X3047	ICOS	IBOS	IAOS	0x30A2~0X30A7	KWHVAL2	0x3102~0X3107	OV_H_Value	0x3162~0X3167	
0x3048~0X304D	IC_LLCNT			0x30A8~0X30AD		0x3108~0X310D	mAH_Val_VA	0x3168~0X316D	
0x304E~0X3053	IC_LLIDX			0x30AE~0X30B3		0x310E~0X3113	mAH_Val_VB	0x316E~0X3173	VARMS_AVM
0x3054~0X3059	SZX_CNT			0x30B4~0X30B9		0x3114~0X3119	mAH_Val_IA	0x3174~0X3179	MAX_IA2
0x305A~0X305F	TMP4		IA0	0x30BA~0X30BF	Relay_Trip	0x311A~0X311F	mAH_Val_IB	0x317A~0X317F	MAX_IB2

Table 4-1 : DSP Buffer of 1V1I(AFE+AVM+OCP+Leakage)

0x3005 ~ 0x3000	Relay_Trip_C	VB	VA	0x3065 ~ 0x3060	VArms	0x30C5 ~ 0x30C0	IA2_ACC	0x3125 ~ 0x3120	PA
0X300B ~ 0x3006	Relay_Trip_B	VBOS	VAOS	0X306B ~ 0x3066	**ISrms**	0X30CB ~ 0x30C6	IA2_SUM	0X312B ~ 0x3126	PB
0X3011 ~ 0x300C	Relay_Trip_A	**IS_LLCnt**	VA_LLCNT	0X3071 ~ 0x306C	IArms	0X30D1 ~ 0x30CC	STACC_IA	0X3131 ~ 0x312C	CF_CNTA
0X3017 ~ 0x3012	TRIP_CTLA	**IS_LLIdx**	VA_LLIDX	0X3077 ~ 0x3072	IBrms	0X30D7 ~ 0x30D2	LTACC_IA	0X3137 ~ 0x3132	CF_CNTB
0X301D ~ 0x3018	TRIP_CTLB	ZXBCnt	ZXACnt	0X307D ~ 0x3078	TVA_rms	0X30DD ~ 0x30D8	IB2_ACC	0X313D ~ 0x3138	PA0
0X3023 ~ 0x301E	TRIP_CTLC	ZXBStart	ZXAStart	0X3083 ~ 0x307E	**TIS_rms**	0X30E3 ~ 0x30DE	IB2_SUM	0X3143 ~ 0x313E	VARMS_ACC
0X3029 ~ 0x3024	RLY_ZXA	ZXBStop	ZXAStop	0X3089 ~ 0x3084	TIA_rms	0X30E9 ~ 0x30E4	STACC_IB	0X3149 ~ 0x3144	PC
0X302F ~ 0x302A	RLY_ZXB	VBZXTO	VAZXTO	0X308F ~ 0x308A	TIB_rms	0X30EF ~ 0x30EA	LTACC_IB	0X314F ~ 0x314A	
0X3035 ~ 0x3030	RLY_ZXC	VBState	VAState	0X3095 ~ 0x3090	TPA	0X30F5 ~ 0x30F0	IC2_ACC	0X3155 ~ 0x3150	CF_CNTC
0X303B ~ 0x3036	ISSign	HV1	VA0	0X309B ~ 0x3096	TPB	0X30FB ~ 0x30F6	IC2_SUM	0X315B ~ 0x3156	KWHVAL3
0X3041 ~ 0x303C	IC	IB	IA	0X30A1 ~ 0x309C	KWHVAL1	0X3101 ~ 0x30FC	STACC_IC	0X3161 ~ 0x315C	
0X3047 ~ 0x3042	ICOS	IBOS	IAOS	0X30A7 ~ 0x30A2	KWHVAL2	0X3107 ~ 0x3102	LTACC_IC	0X3167 ~ 0x3162	MAX_IC2
0X304D ~ 0x3048	**IC_LLCNT**	**IB_LLCNT**	**IA_LLCNT**	0X30AD ~ 0x30A8	ICrms	0X310D ~ 0x3108	ISum2	0X316D ~ 0x3168	VARMS_HALF
0X3053 ~ 0x304E	**IC_LLIDX**	**IB_LLIDX**	**IA_LLIDX**	0X30B3 ~ 0x30AE	TIC_rms	0X3113 ~ 0x310E	PB0	0X3173 ~ 0x316E	VARMS_AVM
0X3059 ~ 0x3054	PSU_UVPCNT		ISIGN	0X30B9 ~ 0x30B4	TPC	0X3119 ~ 0x3114	PC0	0X3179 ~ 0x3174	MAX_IA2
0X305F ~ 0x305A	TMP4		IA0	0X30BF ~ 0x30BA		0X311F ~ 0x311A		0X317F ~ 0x317A	MAX_IB2

Table 4.2 : DSP Buffer of 1V13

3.10 Calculate Vrms method

Below table explains how to calculate the Vrms(V) method via the mapping address:

Calculate Vrms(V) Value						
Vrms register address : 0x3078~0x307D, 0x3078 address is Low Byte, 0x307D address is High Byte.						
Register address	0x3078	0x3079	0x307A	0x307B	0x307C	0x307D
Register Data	Data[0] = 0xBA	Data[1] = 0x49	Data[2] = 0x6C	Data[3] = 0x77	Data[4] = 0x00	Data[5] = 0x00
Example : Vrms value = 119.423(V) Data[5]=0x00 Data[4]=0x00 Data[3]=0x77 Data[2]=0x6C						

Data[1]=0x49

Data[0]=0xBA

Vrms value = {(Data[5]*256^5) +(Data[4]*256^4) +(Data[3]*256^3) +
(Data[2]*256^2) + (Data[1]*256) + Data[0]} / (2^24)
= (0x0000776C49BA) / (2^24)
= 2003585466 / (2^24)
= 119.423 (V)

Table 4.3 : Calculate Vrms

3.11 Calculate Irms method

Below table explains how to calculate the Irms(A) method via the mapping address for 1V1I/1V3I:

Calculate Irms(A) Value						
Irms register address : 0x3084~0x3089, 0x3084 address is Low Byte, 0x3089 address is High Byte.						
Register address	0x3084	0x3085	0x3086	0x3087	0x3088	0x3089
Register Data	Data[0] = 0x35	Data[1] = 0x50	Data[2] = 0xFB	Data[3] = 0x00	Data[4] = 0x00	Data[5] = 0x00

Example :

Irms value = 2.405964 (A)

Data[5]=0x00

Data[4]=0x00

Data[3]=0x00

Data[2]=0xFB

Data[1]=0x50

Data[0]=0x35

Irms value ={(Data[5]*256^5) +(Data[4]*256^4) +(Data[3]*256^3) + (Data[2]*256^2) +
(Data[1]*256) + Data[0]} / (2^30)
= (0x000000FB5035) / (2^30)
= 16470069 / (2^30)
= 2.405964 (A)

Table 4.4 : Calculate Irms(A)

Below table explains how to calculate the Irms(B) method via the mapping address for **1V3I DSP only ,the algorithm is same as Irms(A):**

Calculate Irms(B) Value						
Irms register address : 0x308A~0x308F, 0x308A address is Low Byte, 0x308F address is High Byte.						
Register address	0x308A	0x308B	0x308C	0x308D	0x308E	0x308F

Table 4.5 : Calculate Irms(B)

Below table explains how to calculate the Irms(C) method via the mapping address for **1V3I DSP only ,the algorithm is same as Irms(A):**

Calculate Irms(C) Value						
Irms register address : 0x30AE~0x30B3, 0x30AE address is Low Byte, 0x30B3 address is High Byte.						
Register address	0x30AE	0x30AF	0x30B0	0x30B1	0x30B2	0x30B3

Table 4.6 : Calculate Irms(C)

3.12 Calculate Active Power method

Below table explains how to calculate the Active Power(Wa) method via the mapping address for 1V1I/1V3I::

Calculate Active Power(Wa) Value						
ActivePower register address :0x3090~0x3095 0x3090 address is Low Byte , 0x3095 address is High Byte.						
Register address	0x3090	0x3091	0x3092	0x3093	0x3094	0x3095
Register Data	Data[0]= 0x77	Data[1]= 0x9C	Data[2]= 0x22	Data[3]= 0x74	Data[4]= 0x09	Data[5]= 0x00

Example :

ActivePower value = 2420.1352(W)

Data[5]=0x00

Data[4]=0x09

Data[3]=0x74

Data[2]=0x22

Data[1]=0x9C

Data[0]=0x77

Active Power value = [(Data[5]*256^5) + (Data[4]*256^4) + (Data[3] *256^3)+
(Data[2] *256^2)+(Data[1]*256) +Data[0]] / (2^24)
= [(0x00*256^5) + (0x09*256^4) + (0x74*256^3) +
(0x22*256^2)+ (0x9C*256) + 0x77] / (2^24)
= (0x000974229C77) / (2^24)
=(40603130999) / (2^24)
= 2420.1352 (W)

Table 4.7 : Calculate Active Power(Wa)

Below table explains how to calculate the Active Power(Wb) method via the mapping address for **1V3 DSP only ,the algorithm is same as Power(Wa):**

Calculate Active Power(Wb) Value						
ActivePower register address :0x3096~0x309B 0x3096 address is Low Byte, 0x309B address is High Byte.						
Register address	0x3096	0x3097	0x3098	0x3099	0x309A	0x309B

Table 4.8 : Calculate Active Power(Wb)

Below table explains how to calculate the Active Power(Wc) method via the mapping address for **1V3 DSP only ,the algorithm is same as Power(Wa):**

Calculate Active Power(Wc) Value						
ActivePower register address :0x30B4~0x30B9 0x30B4 address is Low Byte, 0x30B9 address is High Byte.						
Register address	0x30B4	0x30B5	0x30B6	0x30B7	0x30B8	0x30B9

Table 4.9 : Calculate Active Power(Wc)

3.13 Calculate PF and Phase angle method

Below table explains how to calculate the Power Factor (PF) and phase angle method via the mapping address:

Calculate Power Factor(PF) Value and Phase Angle Value
PF value = ActivePower / (Vrms × Irms) Phase Angle value = arcCos(PF)
Active Power value and Vrms value and Irms value are known, so use rule to calculate PF and Phase Angle.
Active Power value = 275.00(W)
Irms value = 5.00 (A)
Vrms value = 110.00 (V)
Example : PF value = 0.5000 PF value = (ActivePower) / (Vrms × Irms) = (275) / (110.00 × 5.00) = 0.5 Phase Angle value = arcCos(PF) = arcCos(0.5) = 60 (Degree)

Table 4.10 : Calculate Power Factor(PF) Value and Phase Angle Value

3.14 Calculate Accumulate power Method

Below table explains how to calculate the Accumulate Energy(Wa) method via the mapping address for 1V1I/1V3I:

Calculate Accumulate Energy (Wa) Value

Accumulate Energy value = CF_Count *0.3125 WH
= 38580 (WH)
= 38.58 (KWH)

CF_Count register address： 0x312C~0x3131,

0x312C address is Low Byte, 0x3131 address is High Byte.

Register address	0x312C	0x312D	0x312E	0x312F	0x3130	0x3131
Register Data	Data[0] = 0x40	Data[1] = 0xE2	Data[2] = 0x01	Data[3] = 0x00	Data[4] = 0x00	Data[5] = 0x00

Example : CF_Count value = 123456 (imp)

CF_Count value = (Data[5]*256^5) + (Data[4]*256^4) + (Data[3] *256^3)+
(Data[2] *256^2)+(Data[1]*256) +Data[0]
= (0x00*256^5) + (0x00*256^4) + (0x00*256^3) +
(0x01*256^2)+ (0xE2*256) + 0x40
= 0x00000001E240
= 123456(imp)

Accumulate Energy Value = CF_Count *0.3125
= 123456*0.3125
= 38580 (WH)
= 38.58 (KWH)

Table 4.11 : Calculate Accumulate Energy (Wa)

Below table explains how to calculate the Accumulate Energy(Wb) method via the mapping address for **1V3I DSP only ,the algorithm is same as Accumulate Energy(Wa):**

Calculate Accumulate Energy (Wb) Value

Accumulate Energy value = CF_Count *0.3125 WH
= 38580 (WH)
= 38.58 (KWH)

CF_Count register address： 0x3132~0x3137,

0x3132 address is Low Byte,0x3137 address is High Byte.

Register address	0x3132	0x3133	0x3134	0x3135	0x3136	0x3137

Table 4.12 : Calculate Accumulate Energy (Wb)

Below table explains how to calculate the Accumulate Energy(Wc) method via the mapping address for **1V3I DSP only ,the algorithm is same as Accumulate Energy(Wa):**

Calculate Accumulate Energy (Wc) Value						
Accumulate Energy value = CF_Count *0.3125 WH = 38580 (WH) = 38.58 (KWH) CF_Count register address： 0x3150~0x3155, 0x3150 address is Low Byte ,0x315c address is High Byte.						
Register address	0x3150	0x3151	0x3152	0x3153	0x3154	0x3155

Table 4.13 : Calculate Accumulate Energy (Wc)

3.15 Calculate Frequency method

Below table explains how to calculate the Frequency (Freq) method via the mapping address:

Calculate Frequency(Hz) Value						
Frequency value = {((ZccCnt-1) / 2) / ((ZccStop - ZccStart) / SampleCnt) }= 49.9992 (Hz) ZccCnt register address : 0x3018~0x3019, 0x3018 address is Low Byte, 0x3019 address is High Byte.						
Register address	0x3018	0x3019				
Register Data	Data[0] = 0x64	Data[1] = 0x00				
Example : ZccCnt value = 100 ZccCnt value = (Data[1]*256) +Data[0] = (0x00*256) + 0x64 =0x0064 =100 ZccStart register address : 0x301E~0x301F, 0x301E address is Low Byte, 0x301F address is High Byte.						
Register address	0x301E	0x301F				
Register Data	Data[0] = 0x20	Data[1] = 0x00				
Example : ZccStart value = 32 ZccStart value = [(Data[5]*256^5) +(Data[4]*256^4)+(Data[3]*256^3)+ (Data[2]*256^2)+(Data[1]*256) +Data[0]] = (0x00*256^5) + (0x00*256^4) + (0x00*256^3)+ (0x00*256^2) + (0x00*256) + 0x20 = 0x000000000020 = 32						

ZccStop register address : 0x3024~0x3025,

0x3024 address is Low Byte, 0x3025 address is High Byte.

Register address	0x3024	0x3025				
Register Data	Data[0] = 0x3B	Data[1] = 0x0F				

Example : ZccStop value = 3899

ZccStop value = [(Data[5]*256^5) +(Data[4]*256^4)+(Data[3]*256^3)+
(Data[2]*256^2)+(Data[1]*256) +Data[0]]
= (0x00*256^5) + (0x00*256^4) +(0x00*256^3)+
(0x00*256^2) +(0x0F*256) + 0x3B
= 0x000000000F3B
= 3899

SampleCnt mapping address (from CFG Register mapping for AFE) : 0x3809~0x380A ,

0x3809 address is Low Byte, 0x380A address is High Byte.

Register address	0x3809	0x380A				
Register Data	Data[0] = 0x42	Data[1] = 0x0F				

Example : SampleCnt value = 3906

SampleCnt value = (Data[1] *256)+Data[0]
= (0x0F*256) + 0x42
= 0x0F42
= 3906

Frequency value = {((ZccCnt -1) / 2) / ((ZccStop - ZccStart) / SampleCnt)}
= {((100-1) / 2) / ((3899 - 32) / (3906))}
= (49.5) / (0.9900153)
= 49.9992 Hz

Table 4.14 : Calculate Frequency (Hz)

3.16 OCP Parameter Calculate method

Follow is explanation how to calculate the OCP sample count -OCP_SMPA method for 1V1I/1V3I:

Calculate OCP_SMPA Value

OCP_SMPA=SampleCnt / 25

SampleCnt register address : 0x3809~0x380A,

0x3809 address is Low Byte, 0x380A address is High Byte.

OCP_SMPA register address : 4036~0x4037,

0x4036 address is Low Byte, 0x4037 address is High Byte.

Example :

ADCDIV = 0x3801 bit3~0.

ADC clock = Crystal Clock/[ADCDIV+1]=16M/8=2M

SampleCnt = ADC clock/ OSR512 /Mux number

2MHz /512/2

=2000000/512/2

=1953

OCP_SMPA = DEC2HEX (1953/25)

= 0x4E

Table 4.15 : Calculate OCP_SMPA Value

Follow is explanation how to calculate the Instance sample count -INST_SMP method for 1V1I/1V3I:

Calculate INST_SMP Value

INST_SMP =SampleCnt/1000

SampleCnt register address : 0x3809~0x380A,

0x3809 address is Low Byte, 0x380A address is High Byte.

INST_SMP register address : 0x4048~0x4049,

0x4048 address is Low Byte, 0x4049 address is High Byte.

Example :

SampleCnt =1953

INST_SMP =DEC2HEX(1953/1000)

=0x01.

Table 4.16 : Calculate INST_SMP Value

Follow is explanation how to calculate the Long time pickup threshold **LTPUTHA** method, for 1V1I/1V3I:

Calculate LTPUTHA Value

LTPUTHA =IArms_50ms*(LT_PU^2)

IArms_50ms register address : 0x30C0~0x30C5,

0x30C0 address is Low Byte, 0x30C5 address is High Byte.

Register address	0x30C0	0x30C1	0x30C2	0x30C3	0x30C4	0x30C5
Register Data	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]

LT_PUTHA register address : 0x4126~0x412B,

0x4126 address is Low Byte, 0x412B address is High Byte.

Register address	0x4126	0x4127	0x4128	0x4129	0x412A	0x412B
Register Data	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]

Step1. Set OCPA_EN to 0(address=0x3804 bit2) , 0x3804 &= ~ 0x04

Step2. Set LTPUTHA = 0X7FFFFFFF

Step3. Set OCPA_EN to 1(address=0x3804 bit2) , 0x3804 |= 0x04

Step 4. Wait 2 SECS, read IArms_50ms

Step 5. LTPUTHA =IArms_50ms*(LT_PU^2)

Example :

LT_PU = 1.2X

IArms_50ms= 0x2F37809

LT_PUTHA=(0x2F37809)* (1.2^2)

= 0x43FE00C

Table 4.17 : Calculate LTPUTHA Value

Below table explains how to calculate Long time pickup threshold **LTPUTHB** method via the mapping address for **1V3I DSP only, the algorithm is same as LTPUTHA:**

Calculate LTPUTHB Value						
LTPUTHB =IBrms_50ms*(LT_PU^2)						
IBrms_50ms register address : 0x30D8~0x30DD,						
0x30D8 address is Low Byte, 0x30DD address is High Byte.						
Register address	0x30D8	0x30D9	0x30DA	0x30DB	0x30DC	0x30DD
Register Data	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]
LT_PUTHB register address : 0x405C~0x4061,						
0x405C address is Low Byte, 0x4061 address is High Byte.						
Register address	0x405C	0x405D	0x405E	0x405F	0x4060	0x4061
Register Data	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]

Table 4.18 : Calculate LTPUTHB Value

Below table explains how to calculate Long time pickup threshold **LTPUTHC** method via the mapping address for **1V3I DSP only, the algorithm is same as LTPUTHA:**

Calculate LTPUTHC Value						
LTPUTHC =ICrms_50ms*(LT_PU^2)						
ICrms_50ms register address : 0x30F0~0x30F5,						
0x30F0 address is Low Byte, 0x30F5 address is High Byte.						
Register address	0x30F0	0x30F1	0x30F2	0x30F3	0x30F4	0x30F5
Register Data	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]
LT_PUTHC register address : 0x4168~0x416D,						
0x4168 address is Low Byte, 0x416D address is High Byte.						
Register address	0x4168	0x4169	0x416A	0x416B	0x416C	0x416D
Register Data	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]

Table 4.19 : Calculate LTPUTHC Value

Follow is explanation how to calculate the Long time threshold **LTTHA** method ,for 1V1I/1V3I:

Calculate LTTHA Value						
LTTHA= (IArms_50ms*20)* (LT_PU^2)*PICK TIME IArms_50ms register address : 0x30C0~0x30C5, 0x30C0 address is Low Byte, 0x30C5 address is High Byte.						
Register address	0x30C0	0x30C1	0x30C2	0x30C3	0x30C4	0x30C5
Register Data	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]
LTTHA register address : 0x412C ~0x4131, 0x412C address is Low Byte, 0x4131 address is High Byte.						
Register address	0x412C	0x412D	0x412E	0x412F	0x4130	0x4131
Register Data	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]
Step1. Set OCPA_EN to 0(address=0x3804 bit2) , 0x3804 &= ~ 0x04 Step2. Set LTTHA = 0X7FFFFFFFFFFF Step3. Set OCPA_EN to 1(address=0x3804 bit2) , 0x3804 |= 0x04 Step 4. Wait 2 SECS, read IArms_50ms Step 5. LTTHA= (IArms_50ms*20)*(LT_PU^2)*PICK TIME **Example :** PICK TIME=300 s LT_PU = 1.2X IArms_50ms= 0x2F37809 LTTHA= [{ (0x2F37809)* 20}*(1.2^2)*300] =0x6399132FC0						

Table 4.20 : Calculate LTTHA Value

Follow is explanation how to calculate the Long time threshold **LTTHB** method via the mapping address for **1V3I DSP only , the algorithm is same as LTTHA:**

Calculate LTTHB Value						
LTTHB= (IBrms_50ms*20)* (LT_PU^2)*PICK TIME IBrms_50ms register address : 0x30D8~0x30DD, 0x30D8 address is Low Byte , 0x30DD address is High Byte.						
Register address	0x30D8	0x30D9	0x30DA	0x30DB	0x30DC	0x30DD
Register Data	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]
LTTHB register address : 0x4144 ~0x4149, 0x4144 address is Low Byte, 0x4149 address is High Byte.						
Register address	0x4144	0x4145	0x4146	0x4147	0x4148	0x4149
Register Data	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]

Table 4.21 : Calculate LTTHB Value

Follow is explanation how to calculate the Long time threshold **LTTHC** method via the mapping address for **1V3I DSP only, the algorithm is same as LTTHA:**

Calculate LTTHC Value						
LTTHC= (ICrms_50ms*20)* (LT_PU^2)*PICK TIME ICrms_50ms register address : 0x30F0~0x30F5, 0x30F0 address is Low Byte, 0x30F5 address is High Byte.						
Register address	0x30F0	0x30F1	0x30F2	0x30F3	0x30F4	0x30F5
Register Data	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]
LTTHC register address : 0x414A~0x414F, 0x414A address is Low Byte, 0x414F address is High Byte.						
Register address	0x414A	0x414B	0x414C	0x414D	0x414E	0x414F
Register Data	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]

Table 4.22 : Calculate LTTHC Value

Follow is explanation how to calculate the Instance threshold INSTA_TH method, for 1V1I/1V3I:

Calculate INSTA_TH Value						
INSTA_TH= (IArms_50ms/OCP_SMPA)*2*(INST^2)*0.9 IArms_50ms register address : 0x3066~0x306B, 0x3066 address is Low Byte, 0x0x306B address is High Byte.						
Register address	0x30C0	0x30C1	0x30C2	0x30C3	0x30C4	0x30C5
Register Data	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]
INSTA_TH register address : 0x40D2 ~0x40D7, 0x40D2 address is Low Byte, 0x40D7 address is High Byte.						
Register address	0x40D2	0x40D3	0x40D4	0x40D5	0x40D6	0x40D7
Register Data	Data[0]	Data[1]	Data[2]	Data[3]		
Step1. Set OCPA_EN to 0(address=0x3804 bit2) , 0x3804 &= ~ 0x04 Set INSTA_EN to 0(address=0x3804 bit5) , 0x3804 &= ~ 0x20 Step2. Set INSTA_TH = 0X7FFFFFFF Step3. Set OCPA_EN to 1(address=0x3804 bit2) , 0x3804 |= 0x04 Set INSTA_EN to 0(address=0x3804 bit5) , 0x3804 |= 0x20 Step 4. Wait 2 SECS,Read IArms_50ms Step 5. INSTA_TH= (IArms_50ms/OCP_SMPA)*2*(INST^2)*0.9 **Example :** INST =3X IArms_50ms= 0x2F37809 OCP_SMPA=0x4E INSTA_TH= (0x2F37809/ 0x4E)*2*(3^2)*0.9 = 0x9CE7B3						

Table 4.23 : Calculate INSTA_TH Value

Follow is explanation how to calculate the Instance threshold INSTB_TH method , via the mapping address for **1V3I DSP only , the algorithm is same as INSTA_TH:**

Calculate INSTB_TH Value						
INSTB_TH= (IBrms_50ms/OCP_SMPA)*2*(INST^2)*0.9 IBrms_50ms register address : 0x30D8~0x30DD, 0x30D8 address is Low Byte, 0x30DD address is High Byte.						
Register address	0x30D8	0x30D9	0x30DA	0x30DB	0x30DC	0x30DD
Register Data	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]
INSTB_TH register address : 0x4162 ~0x4167, 0x4086 address is Low Byte , 0x4089 address is High Byte.						
Register address	0x4162	0x4163	0x4164	0x4165	0x4166	0x4167
Register Data	Data[0]	Data[1]	Data[2]	Data[3]		

Table 4.24: Calculate INSTB_TH Value

Follow is explanation how to calculate the Instance threshold INSTC_TH method, via the mapping address for **1V3I DSP only, the algorithm is same as INSTA_TH:**

Calculate INSTC_TH Value						
INSTC_TH= (ICrms_50ms/OCP_SMPA)*2*(INST^2)*0.9 ICrms_50ms register address : 0x30F0~0x30F5, 0x30F0 address is Low Byte, 0x30F5 address is High Byte.						
Register address	0x30F0	0x30F1	0x30F2	0x30F3	0x30F4	0x30F5
Register Data	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]
INSTC_TH register address : 0x416E ~0x4173, 0x416E address is Low Byte , 0x4173 address is High Byte.						
Register address	0x416e	0x416f	0x4170	0x4171	0x4172	0x4173
Register Data	Data[0]	Data[1]	Data[2]	Data[3]		

Table 4.25 : Calculate INSTC_TH Value

3.17 AVM Calculate method

Follow is explanation how to calculate the SampleCnt and capture as below:

Calculate AVM SampleCnt Value
SampleCnt = ADC clock/OSR512/Mux number **SampleCnt** register address : 0x3809~0x380A, 0x3809 address is Low Byte, 0x0x380A address is High Byte.

Example :

ADCDIV = 0x3801 bit3~0.

ADC clock = Crystal Clock/[ADCDIV+1]=16M/8=2M

SampleCnt = ADC clock/OSR512/Mux number

=2MHz /512/2

=2000000/512/2

=1953=0x7a1

Table 4.26：Calculate AVM SampleCnt Value

Follow is explanation how to calculate the AVM_SMPA and capture as below:

Calculate AVM_SMPA Value

AVM_SMPA = SampleCnt / (1000/period)

AVM_SMPA register address : 0x403C~0x403D,

0x403C address is Low Byte, 0x403D address is High Byte.

Example :

SampleCnt= 0x07A1

Period=50ms

SampleCnt2 = {0x07A1/(1000/50)}

= 0x61

Table 4.27：Calculate AVM_SMPA Value

Follow is explanation how to calculate the AVM_DLY and capture as below:

Calculate AVM_DLY Value

AVM_DLY = SampleCnt / (1000/ Relay Delay Time)

AVM_DLY register address : 0x4042~0x4043,

0x4042 address is Low Byte , 0x4043 address is High Byte.

Example :

SampleCnt= 0x07A1

Relay Delay Time =125ms

AVM_DLY = {0x07A1/(1000/125)}

= 0xF4

Table 4.28：Calculate AVM_DLY Value

Follow is explanation how to calculate the AVM threshold value method via the mapping address

Calculate AVM Value

UV_THL(Ax) = VA_RMS_AVM * (Ax^2)

UV_THH(Bx) = VA_RMS_AVM * (Bx^2)

OV_THL(Cx) = VA_RMS_AVM * (Cx^2)

OV_THH(Dx) = VA_RMS_AVM * (Dx^2)

VA_RMS_AVM register address : 0x316E ~ 0x3173,

0x316E address is Low Byte, 0x3173 address is High Byte.

Register address	0x316E	0x316F	0x3170	0x3171	0x3172	0x3173
Register Data	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]

UV_THL(Ax) register address : 0x415C~0x4161,

0x415C address is Low Byte, 0x4161 address is High Byte.

Register address	0x415C	0x415D	0x415E	0x415F	0x4160	0x4161
Register Data	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]

UV_THH(Bx) register address : 0x4162~0x4167,

0x4162 address is Low Byte, 0x4167 address is High Byte.

Register address	0x4162	0x4163	0x4164	0x4165	0x4166	0x4167
Register Data	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]

OV_THL(Cx) register address : 0x4168~0x416D,

0x4168 address is Low Byte , 0x416D address is High Byte.

Register address	0x4168	0x4169	0x416A	0x416B	0x416C	0x416D
Register Data	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]

OV_THH(Dx) register address : 0x416E~0x4173,

0x416E address is Low Byte , 0x4173 address is High Byte.

Register address	0x416E	0x416F	0x4170	0x4177	0x4172	0x4173
Register Data	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]

Example :

TVA_RMS = 0X63EC2CE

Ax ration = 0.6 X

Bx ration = 0.8 X

Cx ration = 1.2 X

Dx ration = 1.4 X

UV_THL(Ax) = 0X63EC2CE * (0.6^2)

= 0x23F8DCF

UV_THH(Bx)= 0X63EC2CE * (0.8^2)

= 0x13FF34FE

OV_THL(Cx) = 0X63EC2CE * (1.2^2)

= 0x8FE373D

OV_THH(Dx)= 0X63EC2CE * (1.4^2)

= 0xC3D924C

Table 4.29：Calculate AVM Value

Follow is explanation how to calculate the LED_BLK_TH and capture as below:

Calculate LED_BLK_TH Value
LED_BLK_TH = SampleCnt(1s) / (1000 / Pulse_width) **LED_BLK_TH** register address : 0x4084~0x4085, 0x4084 address is Low Byte, 0x0x4085 address is High Byte. **Example :** SampleCnt = 0x07A1 Pulse_width =500 ms LED_BLK_TH = {0x07A1/(1000/500)} = 0x3D0

Table 4.30：Calculate PUL_TH Value

4. Register Setting and Indicate

4.1 UART Auto Baud Rate

PL7413 auto baud rate default is enable, UART interface will detect baud rate after Master send command, the result of UART baud rate will save in 0x3918~0x3919 address.

If you want to disable UART auto baud rate, please set 0x380d[5]=0, then 0x3918~0x3919 will been fixed.

0x380D	**iocfg**	**7:0**	**Default:0xFF**	**Access:RW**
	uart_bau_en	5	1: enable baud rate detection 0: disable	

0x3918	**BitWidthNum_B0**	**7:0**	**Default:**	**Access:R**
		7:0	BitWidthNum[7:0]	
0x3919	**BitWidthNum_B1**	**5:0**	**Default:**	**Access:R**
		5:0	BitWidthNum[13:8]	
0x391A	**BitWidthDen**	**4:0**	**Default:**	**Access:R**
		4:0	BitWidthDen[4:0]	

Figure 5.1: UART Baud Rate register

UART baud rate= system clock * BitWidthDen(0x391A[4:0]) / (BitWidthNum[13:0], 0x3919[5:0]+0x3918[7:0])

= 16M * 8/0x0459=115004.

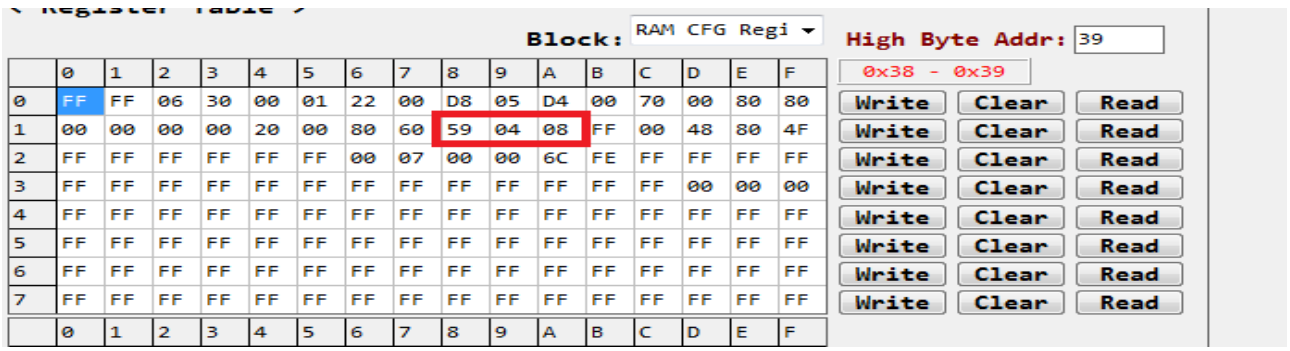


Figure 5.2: UART Baud Rate register

4.2 OCP and INST Protect Indicate

PL7413 have OCP and Instantaneous protect function, The function enable/disable table as below (Please reference PL7413_Leakage and OCP and AVM Demo Board User Manual.pdf):

Address	Bits	Description
0x3803~	15	
0x3804	14	OWP_EN
	13	HANDSHK_EN
	12	ZCC_Auto_Learn
	11	CLEAR_FLAG
	10	INST_IA_EN
	09	Leakage_INST_EN
	08	Leakage_EN
	07	CF_CNTB_EN
	06	NOLOAD_EN
	05	OCPA_EN
	04	CF_CNTA_EN
	03	KWH_EN
	02	RELY_ON_EN
	01	ZX_Half
	00	AVM_EN

Figure 5.3: PL7413 1V1I(AFE+AVM+OCP+AutoLearn)function flag register

DSP has OCP and INST protect happened indicates:

0x3916	**FlagReg_B2**	**7:0**	**Default:**	**Access:R**
		6	LTIA_TRIP: Indicate Ia current >= OCP long time protect current	
		5	STIA_TRIP: Indicate Ia current >=OCP short time protect current	
		2	INSTA_TRIP: Indicate Ia current >= INST protect current	

Figure 5.4: PL7413 OCP and INST register

OCP long time and short time accumulate values:

STACC_IA	0x30CC~0X30D1
LTACC_IA	0x30D2~0X30D7

OCP Threshold (PL7413 1V1I support LT+ST+INST , PL7413 1V3I support LT+ INST):

LTPUTHA	0x412B ~ 0x4126
LTTH_IA	0x4131 ~ 0x412C
INSTA_TH	0x40D7 ~ 0x40D2

IA-RMS 50ms value:

IA2_ACC	0x30C0~0x30C5

[OCP Long Time Protect]

If your long time current setting is 6A, 300 sec, and OCPA_EN(0x3803[5]) is 1, then you use the hair dryers to test it, IA current =8.9A,You can check IA2_ACC(IA-RMS 50ms), if IA2_ACC > LTPUTHA, LTACC_IA will been accumulated. OCP long time will happened after serval second. You can check LTACC_IA, if LTACC_IA > LTTH, then LTIA_TRP(0x3916[6]) will rise to 1.

[INST Protect]

If your INST current setting is 15A, 1ms, and INST_IA_EN(0x3804[2]) is 1, then you use three hair dryers to test it, Ia current =15.5A, INST protect will happened immediately. You can check INSTA_TRP(0x3916[2]) will rise to 1

[Clear OCP Indicate]

Set OCPA_EN(0x3803[5])=0, LTIA_TRP(0x3916[6]) and STIA_TRP(0x3916[5]) will been cleared.

[Clear INST Indicate]

Set INST_IA_EN(0x3804[2])=0 and DSP Enable(0x3802[7])=0, INSTA_TRP(0x3916[2]) will been cleared.

4.2.1.1 OCP Long Time Protect Indicate

Please use PL7413 MP tool to help you to check it. First please check your OCP long time protect settings, then enter engineer mode, and find the “debug” form. In the debug form, you can find OCP_EN and OCP_READ(read IA2_ACC), LTPUTHA, LTTH, LT_SUM(LTACC_IA), LTIA_TRIP.

- You can check your setting,

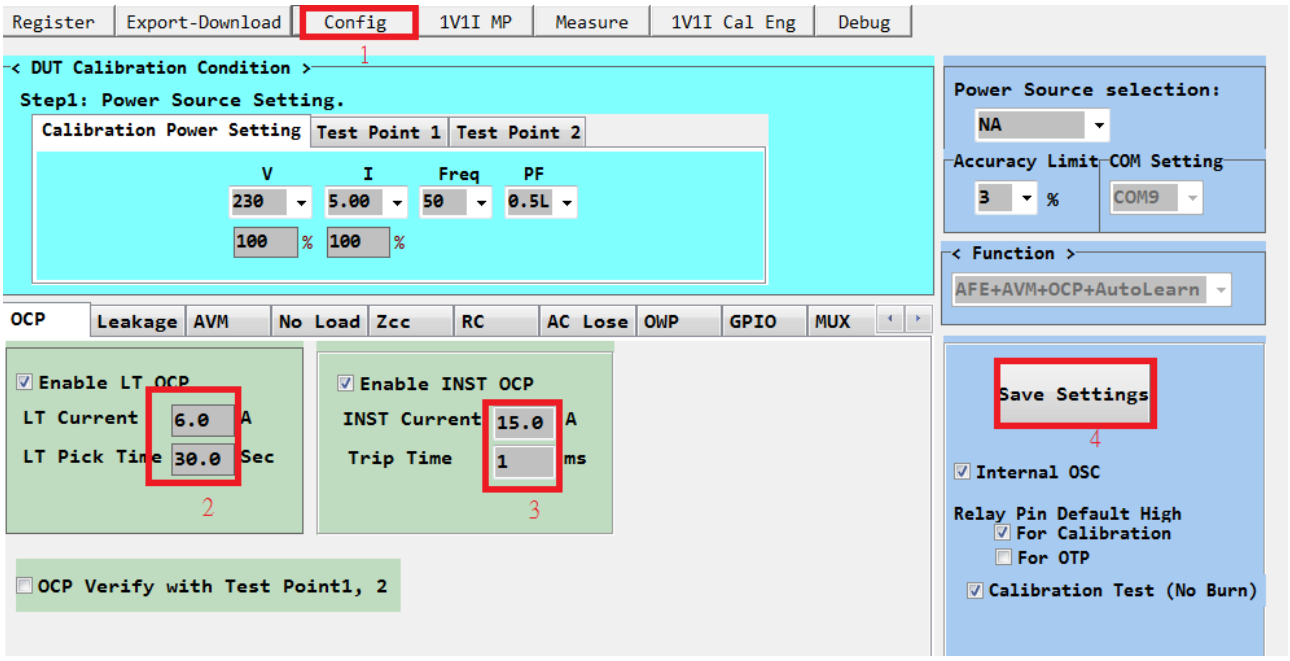


Figure 5.5: PL7413 OCP setting

- Please enter engineer mode

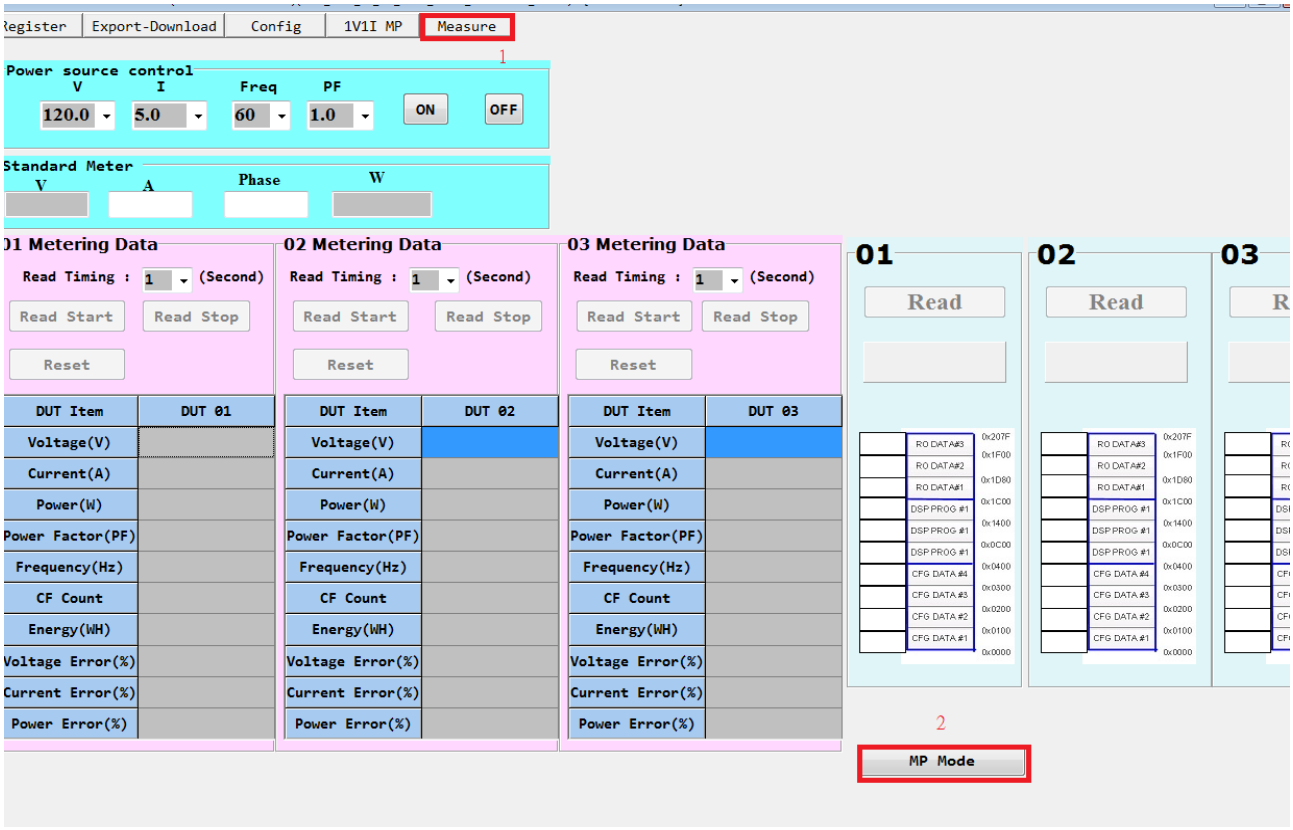


Figure 5.6: PL7413 MP mode

- Please select the debug form

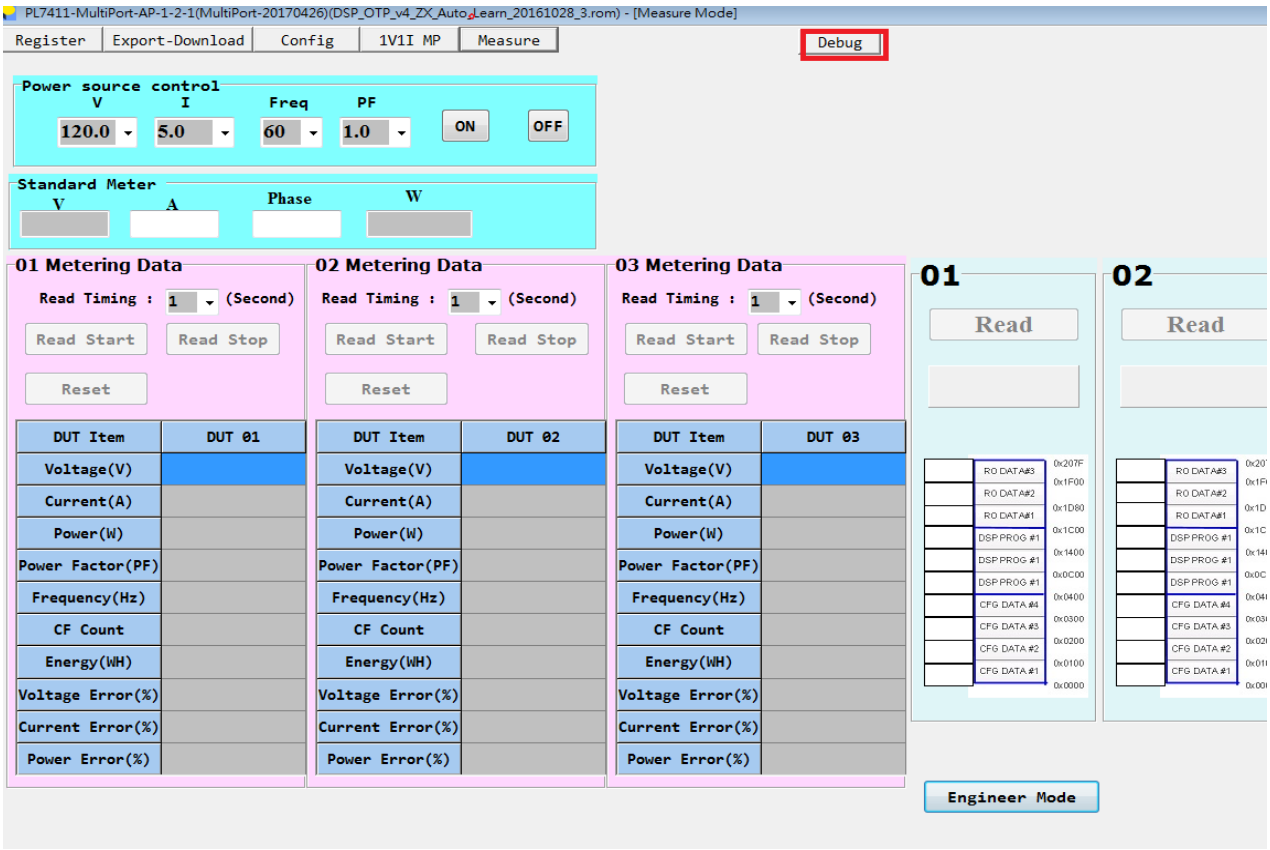


Figure 5.7: PL7413 Engineer Mode

- Press “Read All”.
- Please enable “RLY_ON_EN”
- Enable “OCPA_EN” for long time and short time
- Press “OCP_READ”, it will read IA2_ACC
- If IA2_ACC > LTPUTH, then you press “LT_SUM”, SUM will been accumulated.
- If IA current =6.1A, press “LT_SUM”, LTIA_TRIP will 1 after SUM > LTTH about 29 SECS.
- Check LTIA_TRIP indicate.

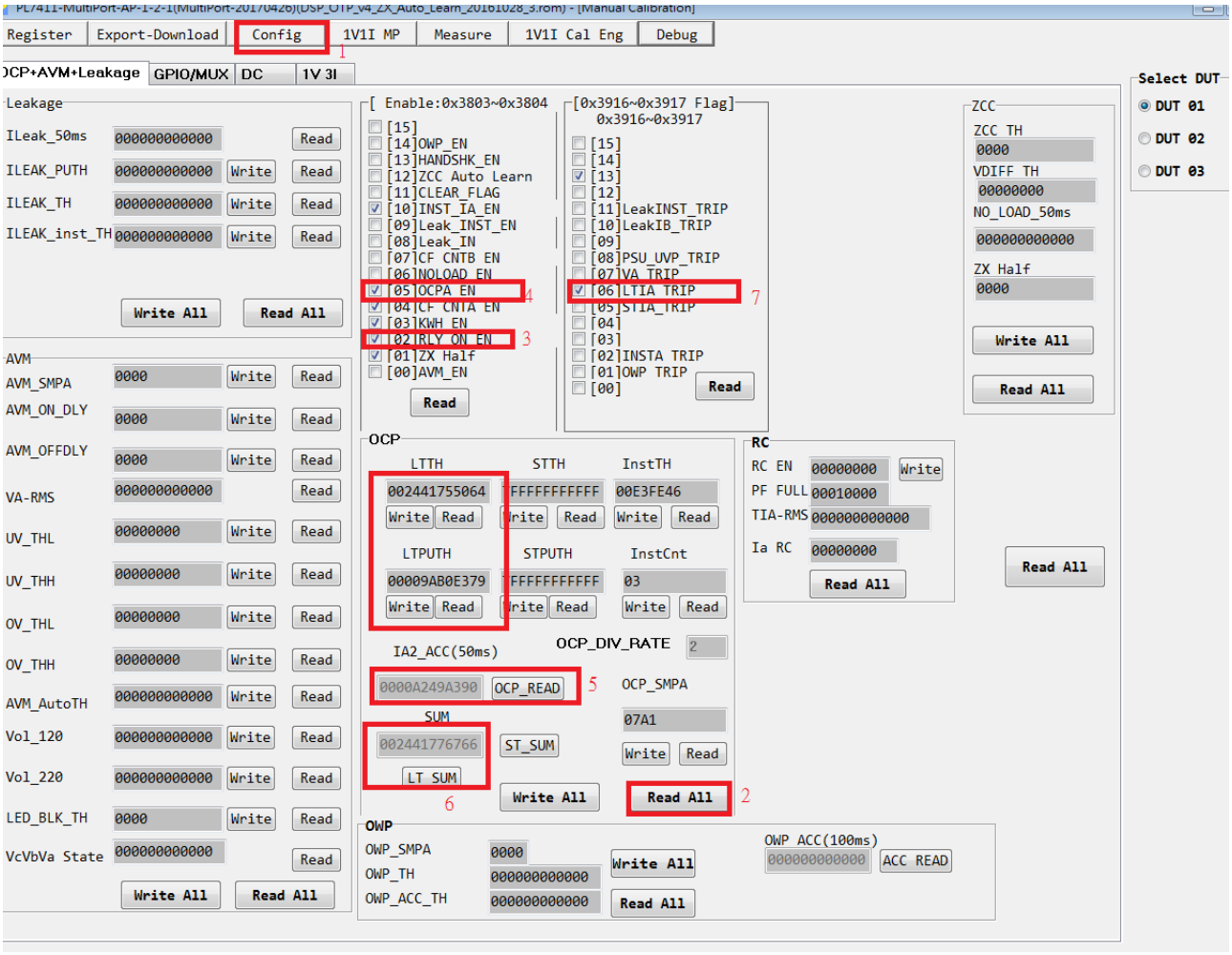


Figure 5.8: PL7413 OCP Long time Debug Flow

4.2.1.2 INST Protect Indicate

Please use our AP to help you to check it. First please check your OCP INST protect settings, then enter engineer mode, and find the “debug” form. In the debug form, you can find INST_IA_EN and InstTH, INSTA_TRIP.

- Please select the debug form
- Press “Read All”.
- Please enable “RLY_ON_EN”
- Enable “OCPA_EN” for OCP
- Enable “INST_IA_EN” for Inst
- If IA current =15.1A, then check INSTA_TRIP indicate..

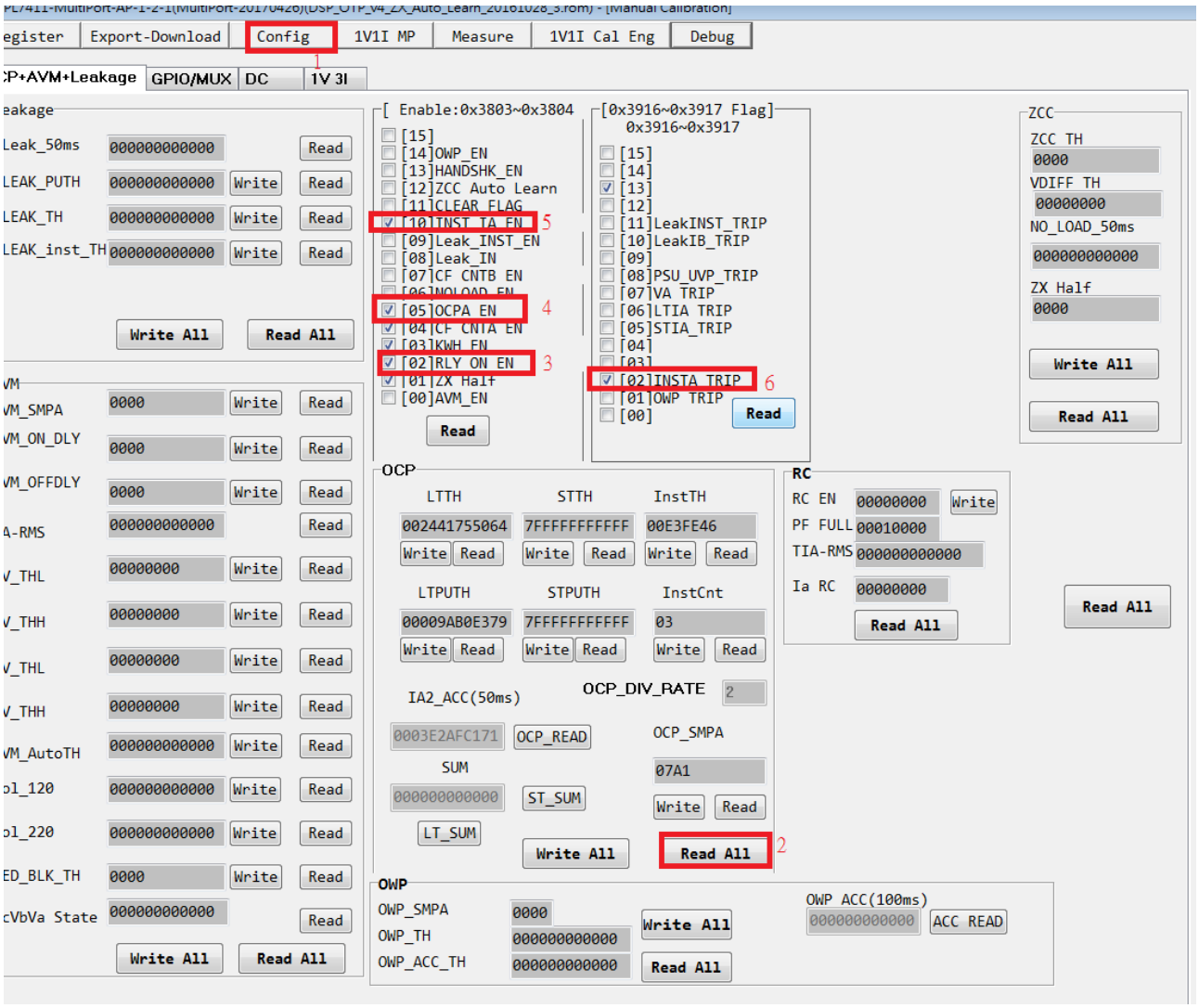


Figure 5.9: PL7413 OCP Instantaneous Debug Flow

4.3 AVM Protect State

PL7413 1V1I support this function , some of AVM State as below:

0x3030~0X3035	VCState	VBState	VAState

PS: We only use VAState

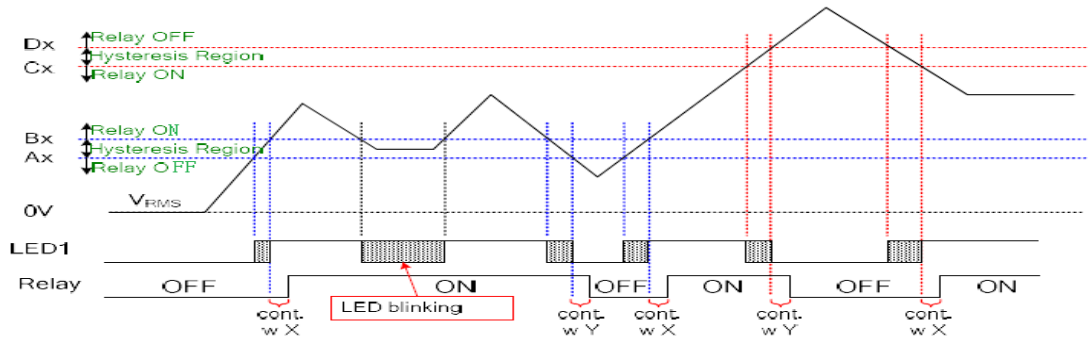
By setting AX /BX/ CX/ DX, you can use PL7413 AVM function shown as follow figure.

PL7413 will auto switch relay on/off when input voltage threshold are setting

In Hysteresis region relay will keep before status, until over /under region. Calibration voltage is 120V.

Point	Voltage	Relay	Offset Voltage
Ax	100	relay off	20
Bx	110	relay on	10
Cx	130	relay on	10
Dx	140	relay off	20

Table 5.1：AVM threshold



Test Case1:

InputVoltage	VA State	LED1
0~99	0	Off
100~109	1	Blink
110~120	2	On
130~139	3	Blink
140~164	4	Off

Table 5.2：AVM Case1 State

Test Case2:

InputVoltage	VA State	LED1
165~199	0	Off
200~209	1	Blink
210~220	2	On
230~239	3	Blink
240~	4	Off

Table 5.3：AVM Case2 State

- You can check your setting,

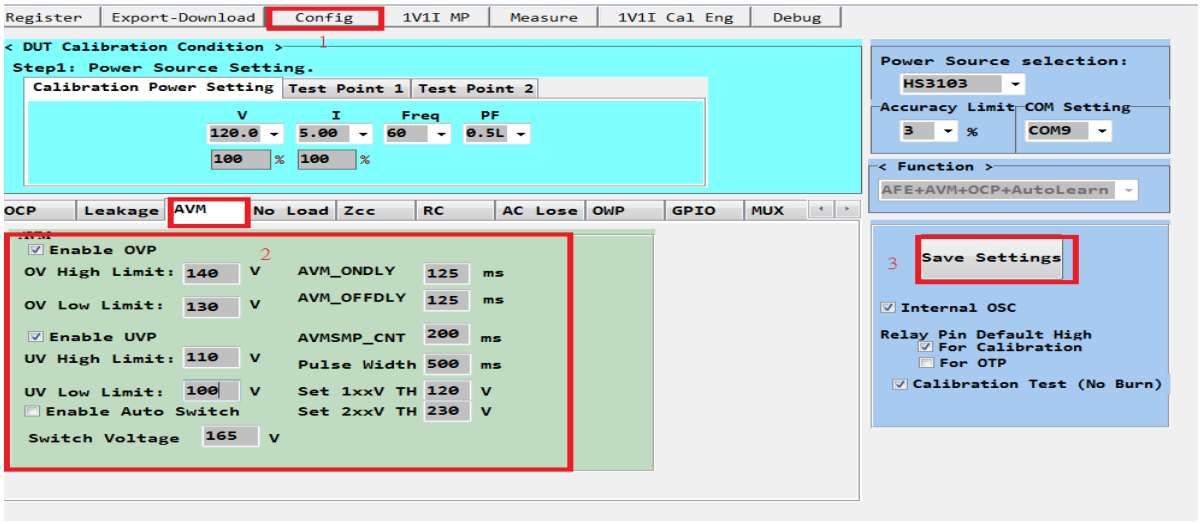


Figure 5.10: PL7413 AVM setting

Please use our AP to help you to check it. First please check your AVM protect settings, then enter engineer mode, and find the “debug” form. In the debug form, you can find AVM_EN and VcVbVa State.

- Please select the debug form
- Press “Read”
- Please enable “RLY_ON_EN”
- Enable “AVM_EN”
- If VA =120V,press “READ_ALL”, VcVbVa State will 2.
- Read 0x3916~0x3917: [07]VA_TRIP=0.
  If VA=141V, VcVbVa State will 4. 0x3916~0x3917: [07]VA_TRIP=1

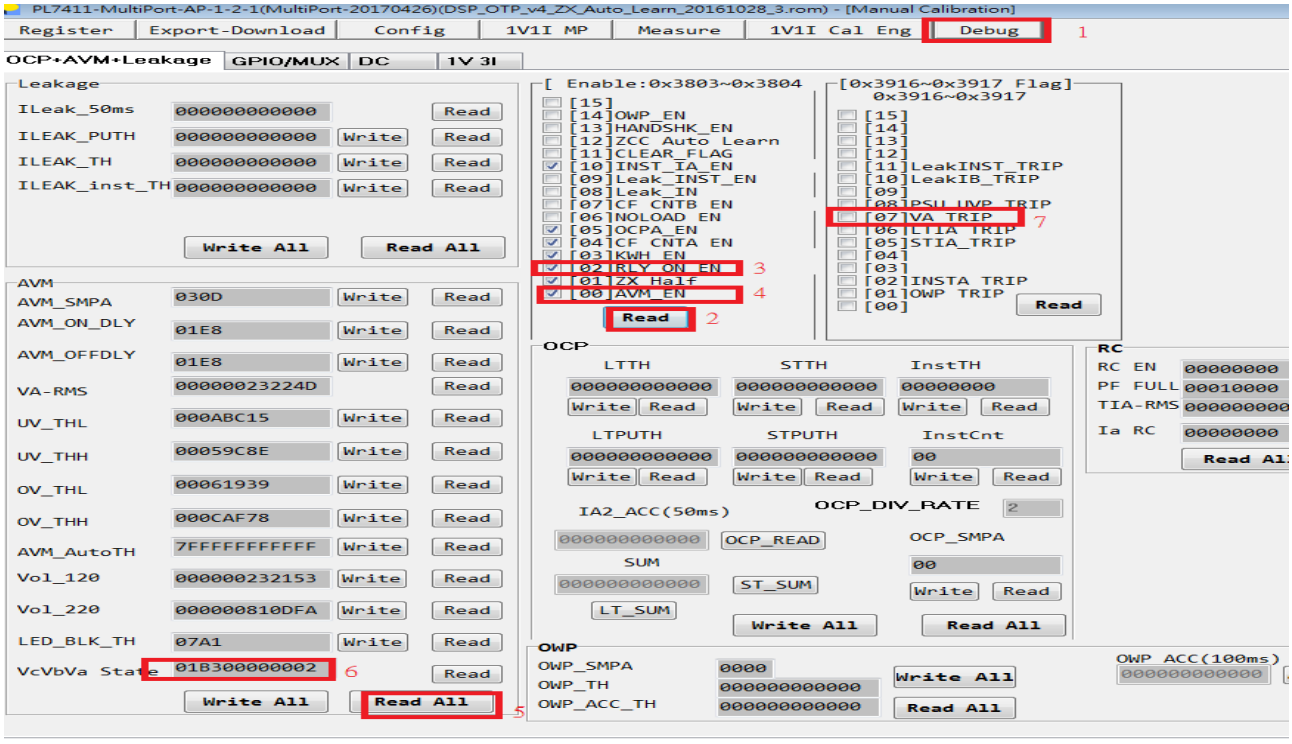


Figure 5.11: PL7413 AVM Enable

4.4 Leakage Protect Indicate

DSP has Leakage protect happened indicates:

0x3917	FlagReg_B2	7:0	Default:	Access:R
		3	LeakINST_TRIP: Indicate Ib current >= Leakage INST protect current(ILeak_1SMP=10xILeak)	
		2	LeakIB_TRIP: Indicate Ib current >=Leakage protect current(ILeak)	

Figure 5.12: PL7413 Leakage register

Leakage Threshold(Please reference PL7413_Leakage and OCP and AVM Demo Board User Manual.pdf):

LeakPUTH_IB	0x4137 ~ 0x4132
LeakTH_IB	0x413D ~ 0x4138
Leak_inst_TH	0x4143 ~ 0x413E

IbRMS 50ms(ILeak 50ms) value:

IB2_ACC	0x30D8~0X30DD

[Clear Leak Protect Indicate]

Set Leak_EN(0x3804[0])=0, LeakIB_TRP(0x3917[2]) will been cleared.

[Clear Leak Inst Protect Indicate]

Set Leak_INST_EN(0x3804[1])=0, LeakINST_TRP(0x3917[3]) will been cleared.

4.4.1 Leakage Protect Indicate

Please use PL7413 MP tool to help you to check it. First please check your Leakage protect settings, then enter engineer mode, and find the “debug” form. In the debug form, you can find Leak_EN and ILeak_50ms(read IB2_ACC), ILEAK_PUTH, ILEAK_TH, , LeakIB_TRIP.

- You can check your setting,

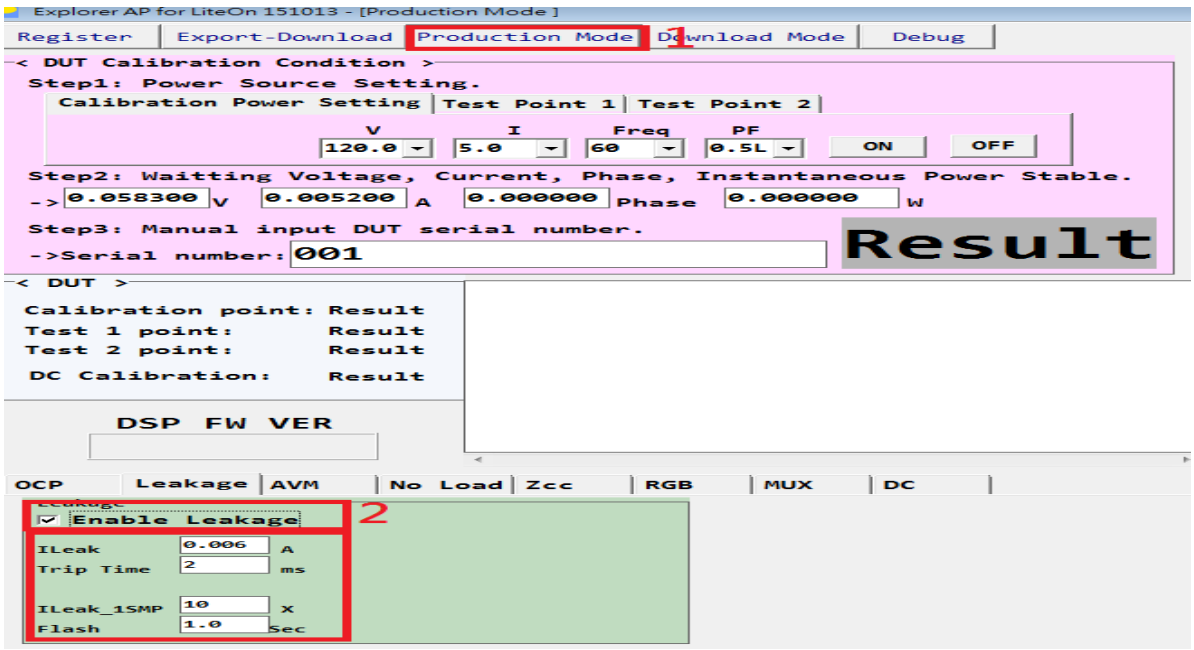


Figure 5.13: PL7413 Leakage setting

Please select the debug form

- Press “Read All”.
- Press “Read”
- Please enable “RLY_ON_EN”
- Enable “Leakage_EN”
- Press “READ”, it will read ILeak_50ms(IB2_ACC)
- If IB current =0.06A, ILeak_50ms > ILEAK_TH, LeakIB_TRIP will 1.

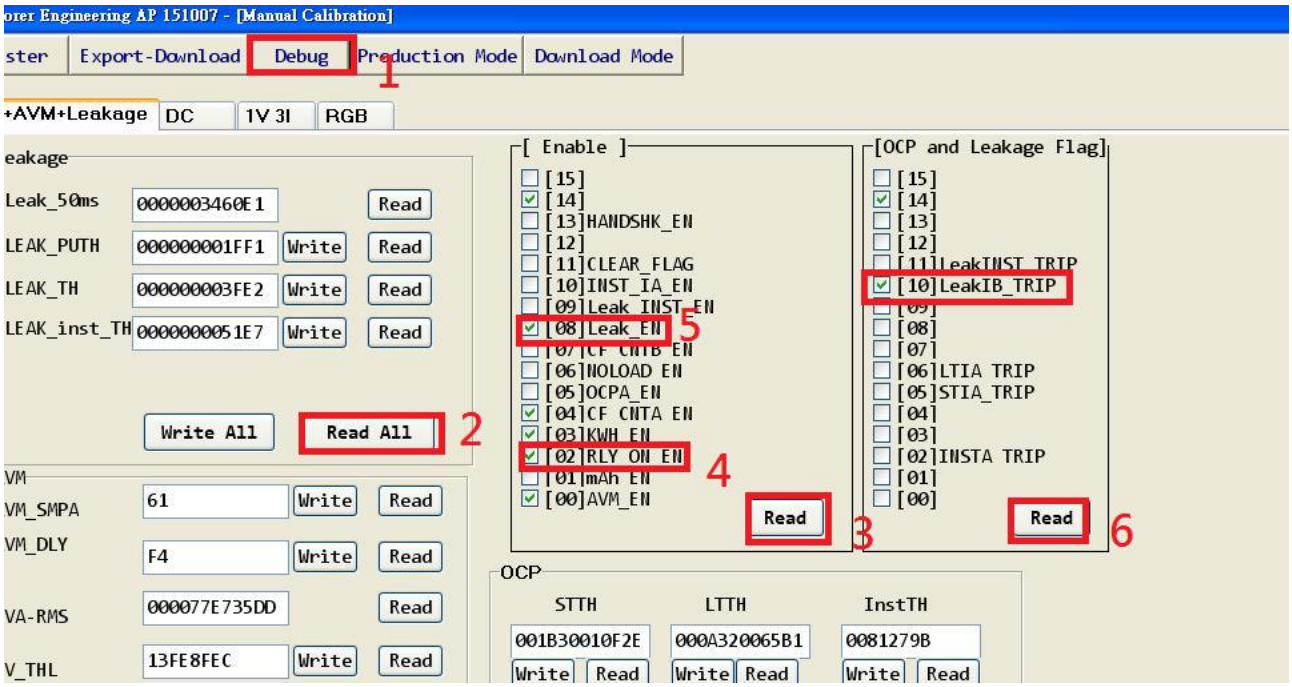


Figure 5.14: PL7413 Leakage Debug setting

4.4.2 Leakage INST Protect Indicate

Please use PL7413 MP tool to help you to check it. First please check your Leakage INST protect settings, then enter engineer mode, and find the “debug” form. In the debug form, you can find Leak_INST_EN and ILeak_50ms(read IB2_ACC), ILEAK_inst_TH, , LeakINST_TRIP.

- Please select the debug form
- Press “Read All”.
- Press “Read”
- Please enable “RLY_ON_EN”
- Enable “Leak_INST_EN”
- If IB current =0.07A, LeakINST_TRIP will 1.
- If you want to check ILeak_50ms(IB2_ACC), please enable Leak_EN
- Press “READ”, it will read ILeak_50ms(IB2_ACC)

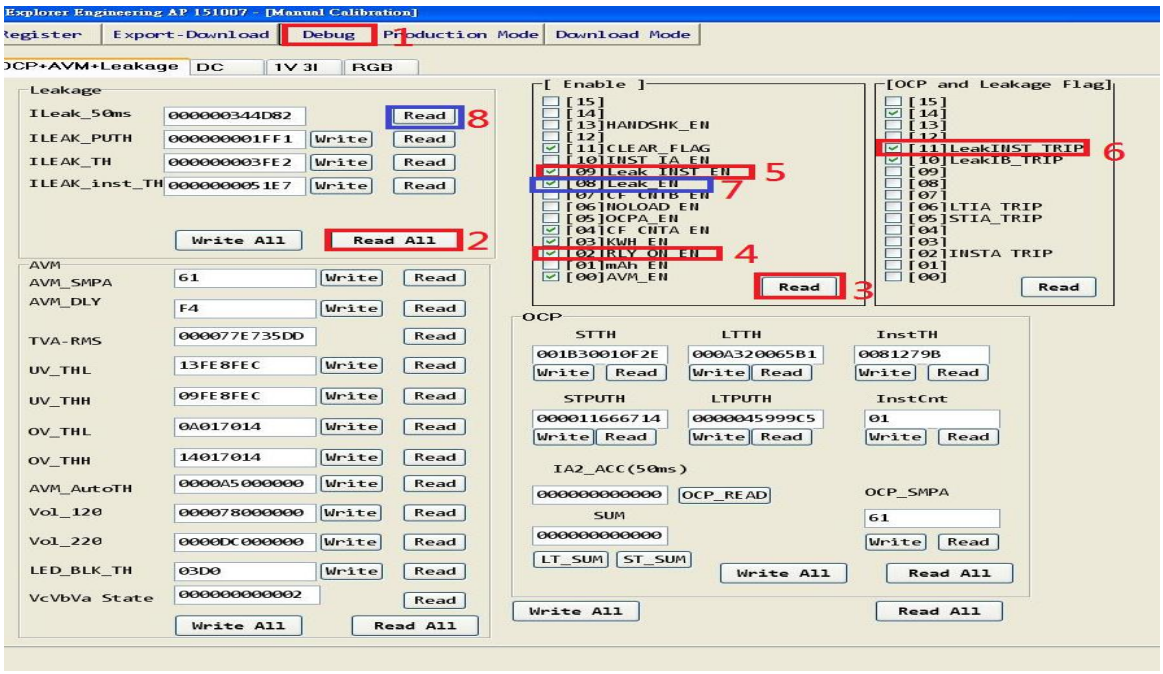


Figure 5.15: PL7413 Leakage Debug Flow

4.5 Auto Learn ZCC

DSP will auto lean when 1st relay on/off

RLY on base on V and off base on I

When on load, relay off will base on V.

Control the functions enable in CFG register (0x3803~0x3804):

Address	Bits	Description
0x3803~ 0x3804	15	
	14	OWP_EN
	13	HANDSHK_EN
	12	ZCC_Auto_Learn
	11	CLEAR_FLAG
	10	INST_IA_EN
	09	Leakage_INST_EN
	08	Leakage_EN
	07	CF_CNTB_EN
	06	NOLOAD_EN
	05	OCPA_EN
	04	CF_CNTA_EN
	03	KWH_EN
	02	RELY_ON_EN
	01	ZX_Half
	00	AVM_EN

Table 5.4：Auto Lean setting

CFG_12 Force_Learn =1

0 ZCC -- when on off

1 ZCC_Lean as "6" (3/On, 3/OFF)

0x3804 bit4(CFG_12) is 1, DSP will force to auto counting the zero-crossing.3 times after the completion of learning,

CFG_1 ZX_Auto =1

0 ZX_Htime set by ZX_Half

1 ZX_Htime auto calculate

Disclaimer

All the information in this document is subject to change without prior notice. Prolific Technology Inc. does not make any representations or any warranties (implied or otherwise) regarding the accuracy and completeness of this document and shall in no event be liable for any loss of profit or any other commercial damage, including but not limited to special, incidental, consequential, or other damages.

Trademarks

The Prolific logo is a registered trademark of Prolific Technology Inc. All brand names and product names used in this document are trademarks or registered trademarks of their respective holders.

Copyrights

Prolific Technology Inc.

7F, No. 48, Sec. 3, Nan Kang Rd.

Nan Kang, Taipei 115, Taiwan, R.O.C.

Telephone: +886-2-2654-6363

Fax: +886-2-2654-6161

E-mail: sales@prolific.com.tw

Website: http://www.prolific.com.tw