PL7211

AFE Calibration

Application Note

Document number：AN-71161001

Revision：1.0

Release Date：Oct 26, 2016

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

List of Tables

1. PL7211 General Description

The PL7211 is a highly integrated power/energy monitoring Analog Front End (AFE) IC that measures electricity-related data for power usage measurement applications. It has built-in 4-channel ADC and a programmable DSP which can adapt to different applications such as metering, power protection and Master/Slave.

The PL7211 has a built-in MTP that stores the chip configuration, DSP code and calibration data. It provides SPI slave interface and can be used for calibration and programming data or DSP code. MCU can be accessed through SPI interface.

The PL7211 can diagnose the electricity-related data to identify overloading, short circuit, leakage current and arcing condition for further power protection features. It also provides a flexible architecture, low system BOM cost and programmable solution, to help manufacturers minimize development efforts and design a versatile and flexible product.

1.1 Block Diagram

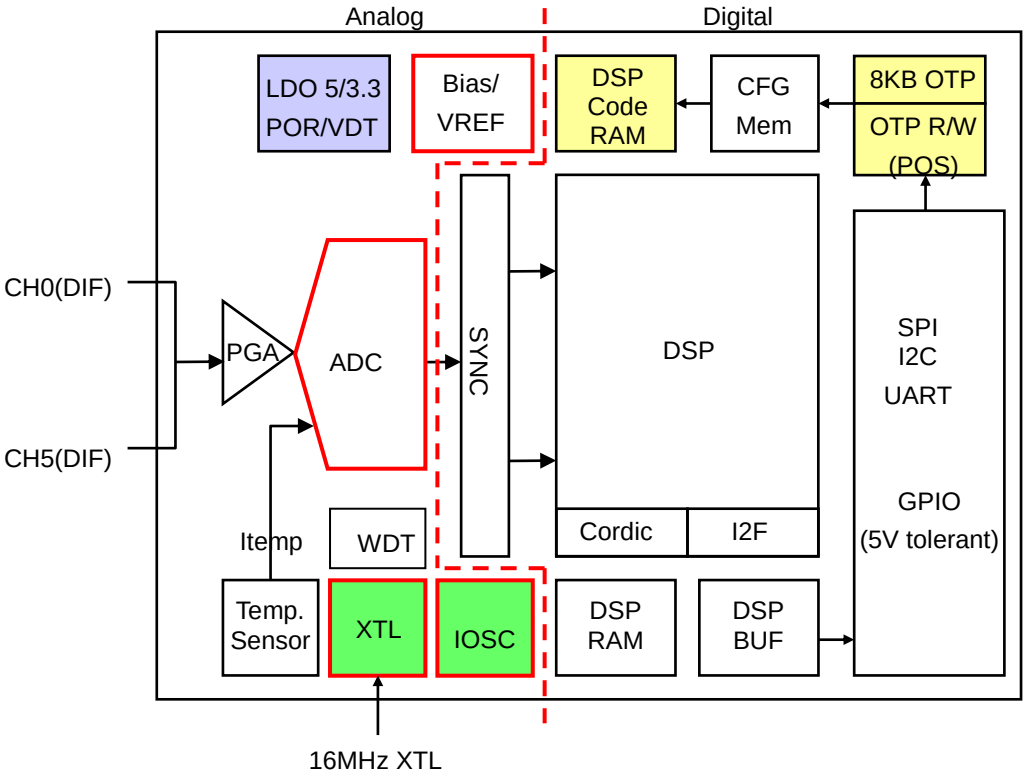


Figure 1-1 : Block diagram

1.2 RMA board and pin connect

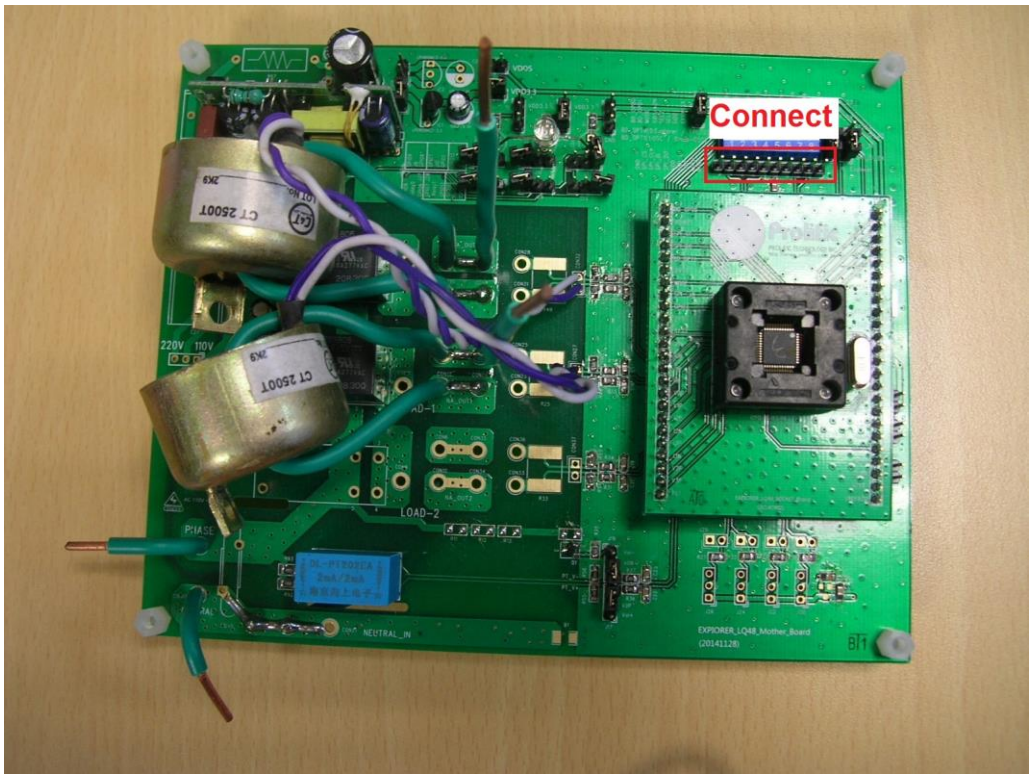


Figure 1-2 : RMA board

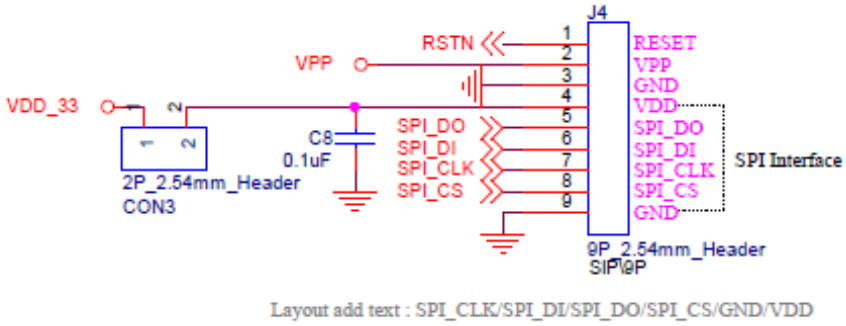


Figure 1-3 : Pin connect

1.3 Demo board and RMA board I/F setting

PL7211 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 PL7211.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**

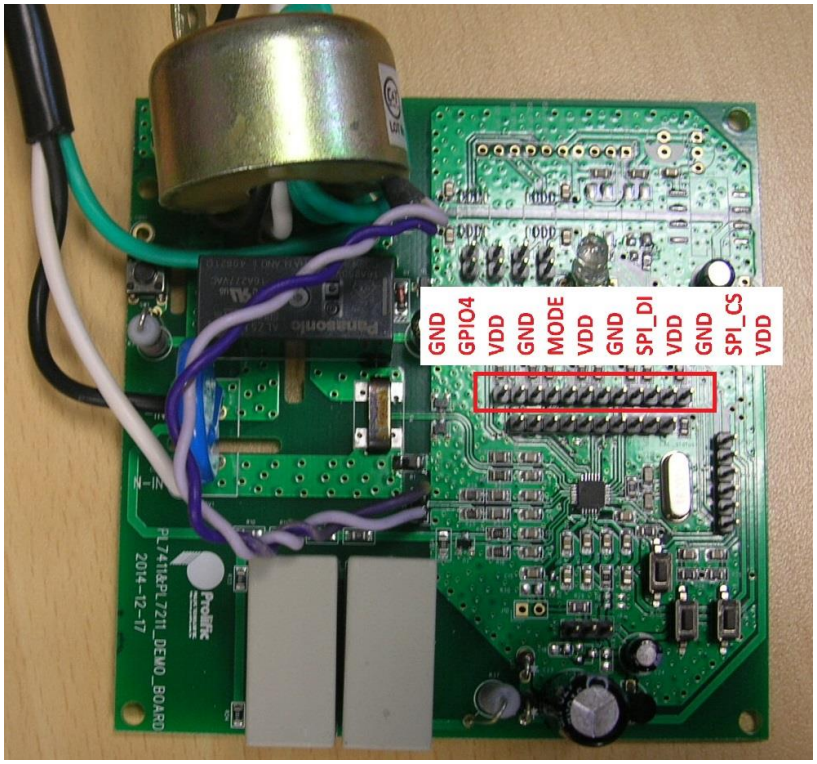


Figure 1-4 : Demo board I/F setting

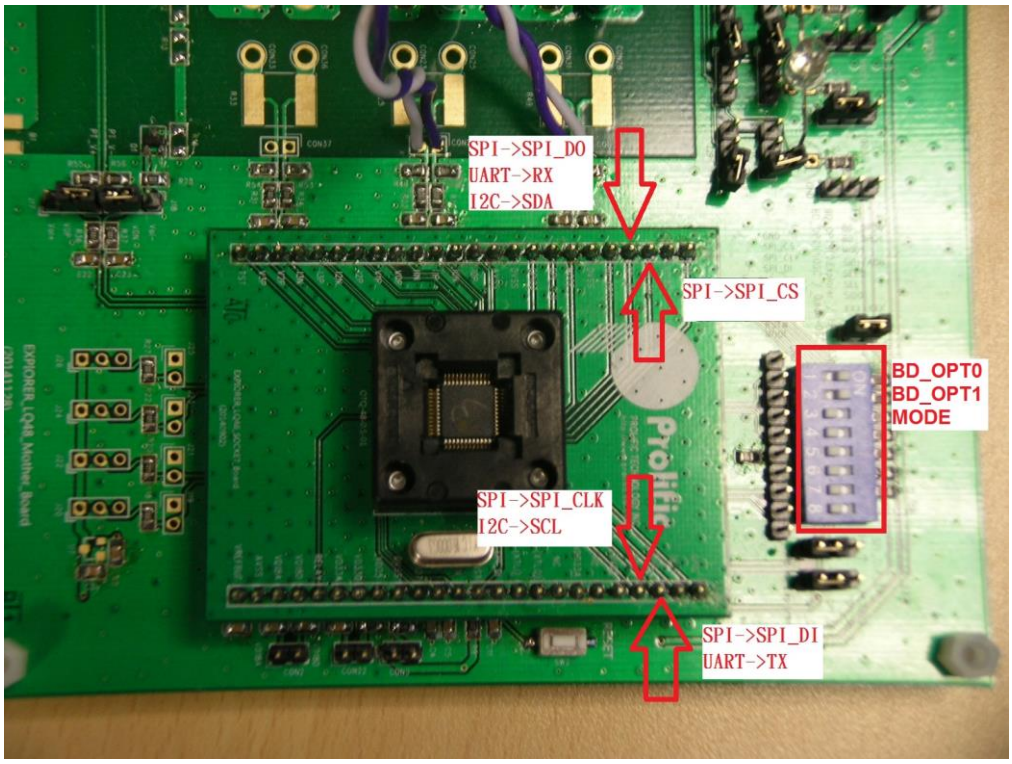


Figure 1-5 : RMA setting

2. PL7211 Interface Connection

PL7211 have three interface can communication with Host device, as below the detail spec.

2.1 SPI interface

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

1. 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

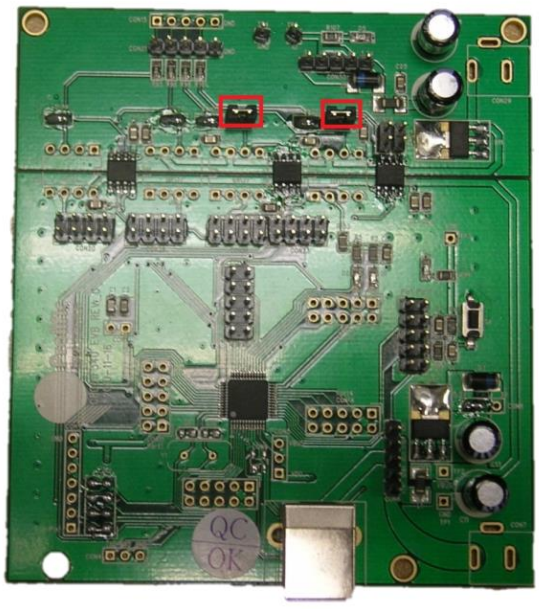


Figure 2-1 : Calibration bottom board

Top Board:

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

(reference red arrow)

Figure 2-2 : Calibration top board

•spi_en = [mode, spi_cs]= 2'b10 , Set the mode jumper short with VDD, SPI_CS jumper short with GND

Figure 2-3 :Demo board SPI I/F define

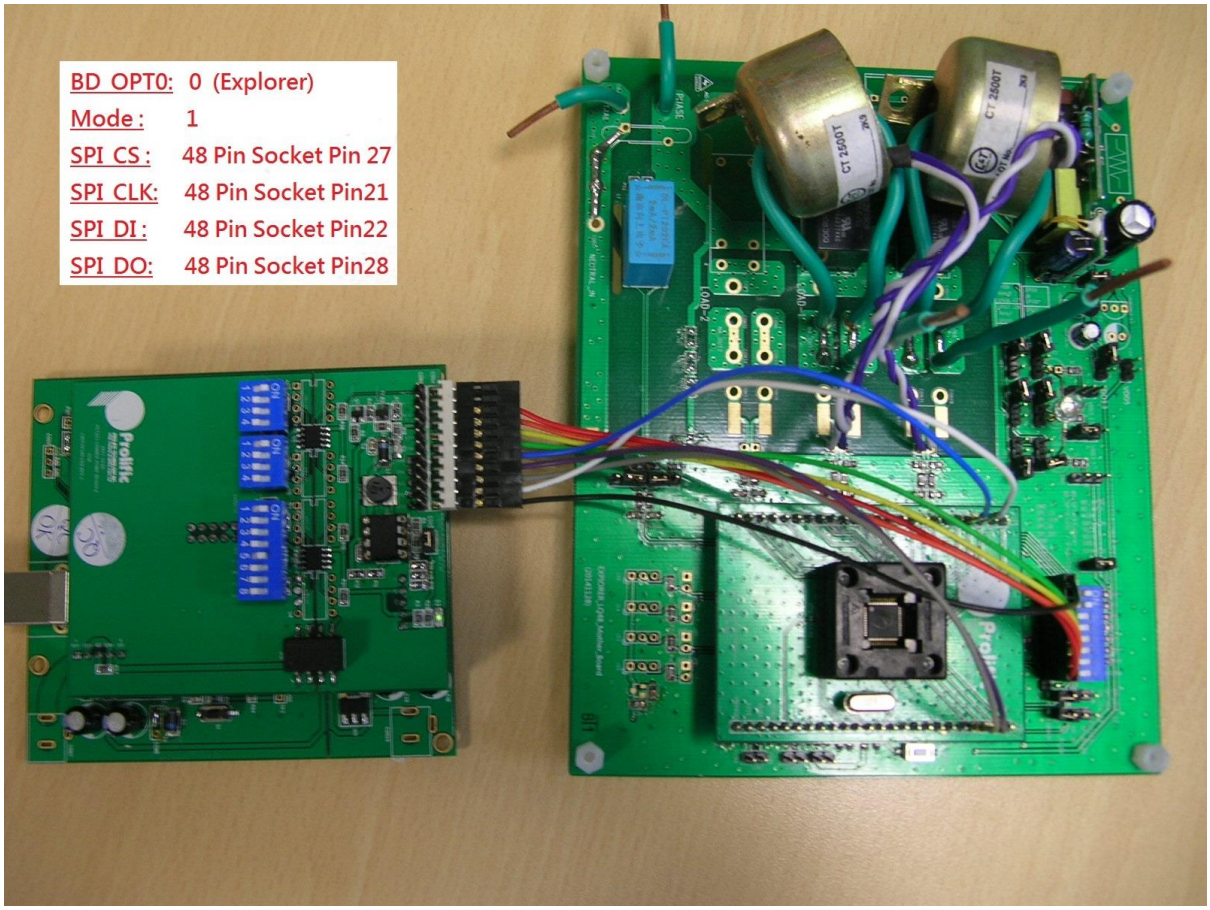


Figure 2-4 : I/F connection

2. AP connection flow

When open the PL7211 AP, you can follow below step to connect and access the raw data via SPI interface,

Step1: Select USB, and click open to connect Calibration board with PC

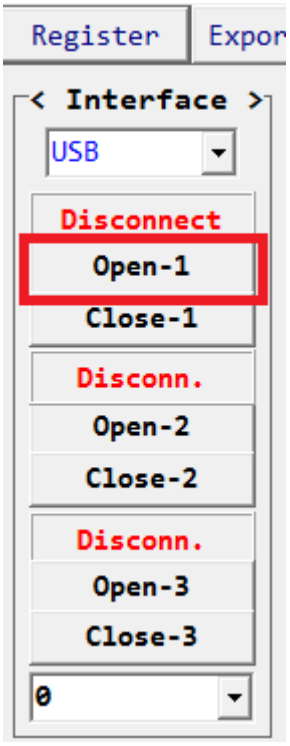


Figure 2-5 : AP connect USB 1

Step 2: Connect success

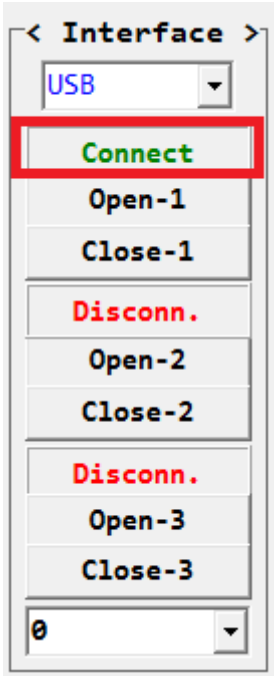


Figure 2-6 : AP connect USB 2

Step3: Select Interface Mode to SPI mode

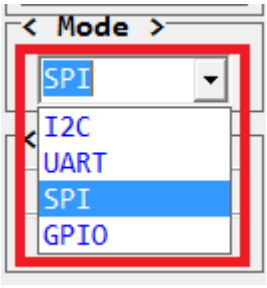


Figure 2-7 : AP I/F Selection

Setp4: Calibration board send reset command to Device

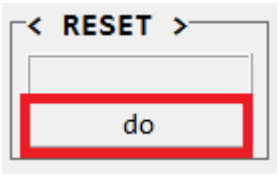


Figure 2-8 : AP reset Device 1

Step 5: Reset success

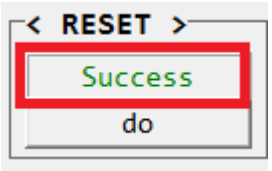


Figure 2-9 : AP reset Device 2

Step 6: Check interface link status

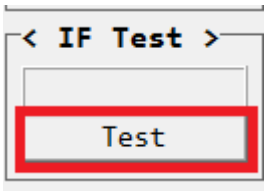


Figure 2-10 : AP I/F link test 1

Step 7: Interface link OK

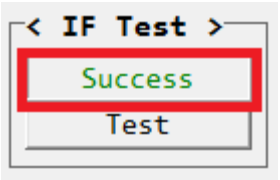


Figure 2-11 : AP I/F link test 2

2.2 UART interface

- 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

3. 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

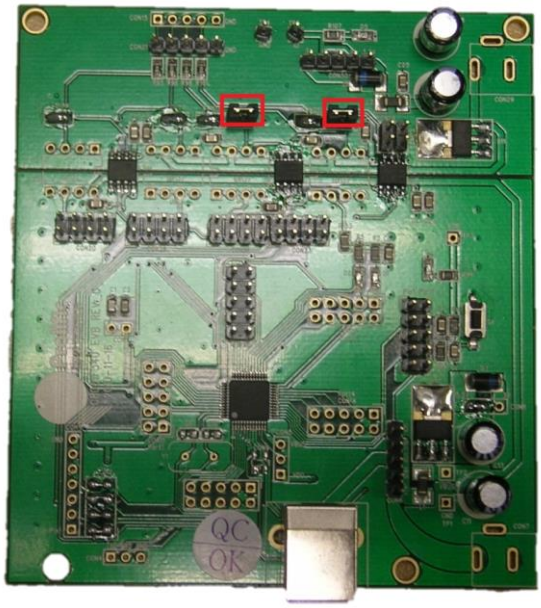


Figure 2-12 : Calibration bottom board

Top Board:

DIP1 pin4→1	DIP2 pin4 →1	DIP3 pin5→1

Figure 2-13 : Calibration top board

• uart_en = [mode, spi_cs]= 2'b01, Set the mode jumper short with GND, SPI_CS jumper short with VDD

Figure 2-14 : Demo board UART I/F define

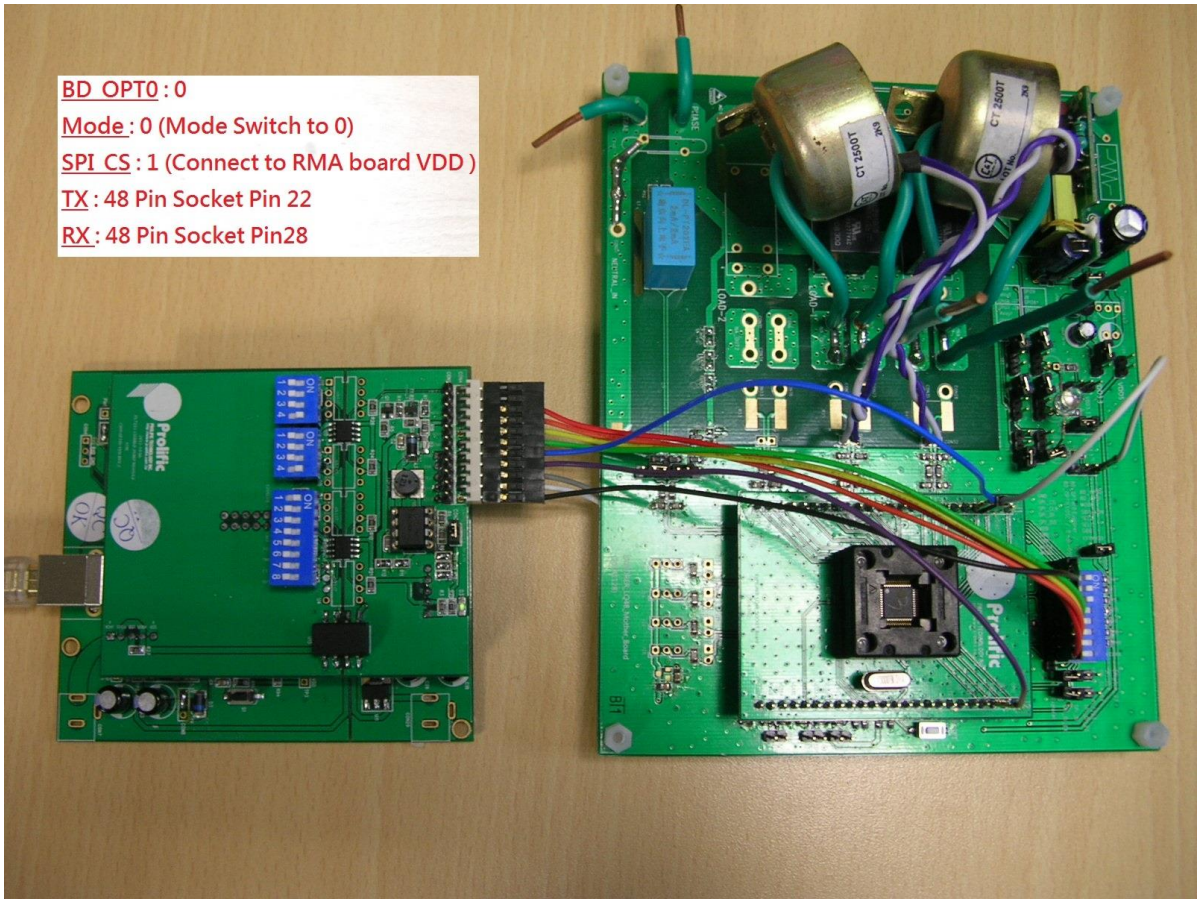


Figure 2-15 : I/F connection

4. AP connection flow

When open the PL7211 AP, you can follow below step to connect and access the raw data via UART interface,

Reference section 2.1.2 step 1~7, you should selection UART mode in step3

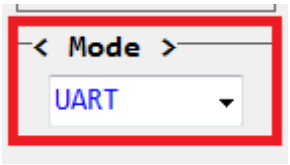


Figure 2-16 : AP I/F selection

2.3 I²C interface

5. 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

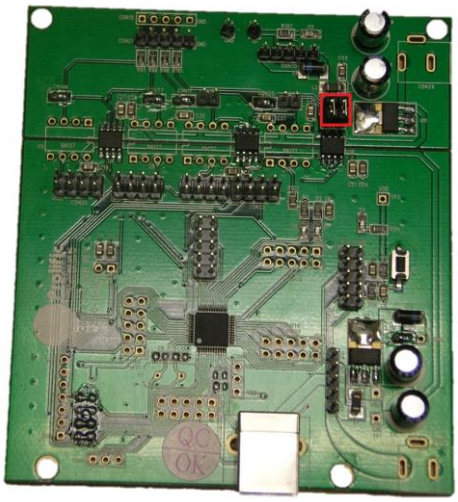


Figure 2-17 : Calibration bottom board

Top Board: DIP3 ➔ (reference red arrow)

DIP3	pin4→1	DIP3 pin8→1

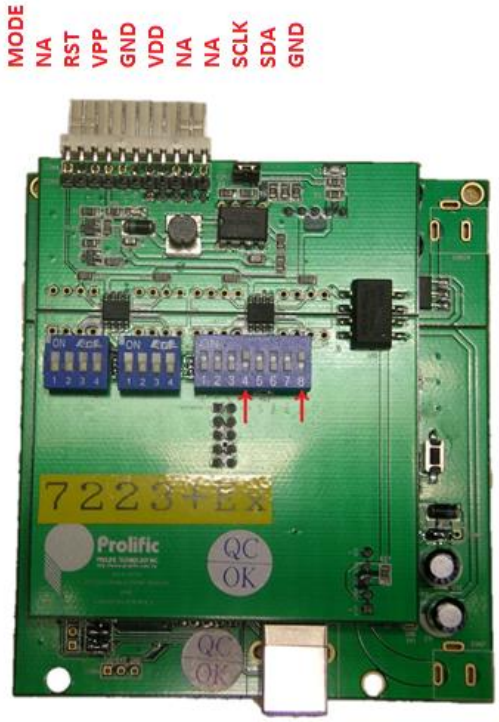


Figure 2-18 : Calibration top board

• i2c_en = [mode, spi_cs]= 2'b00, Set the mode jumper short with GND , SPI_CS jumper short with GND

Figure 2-19 : Demo board I2C I/F define

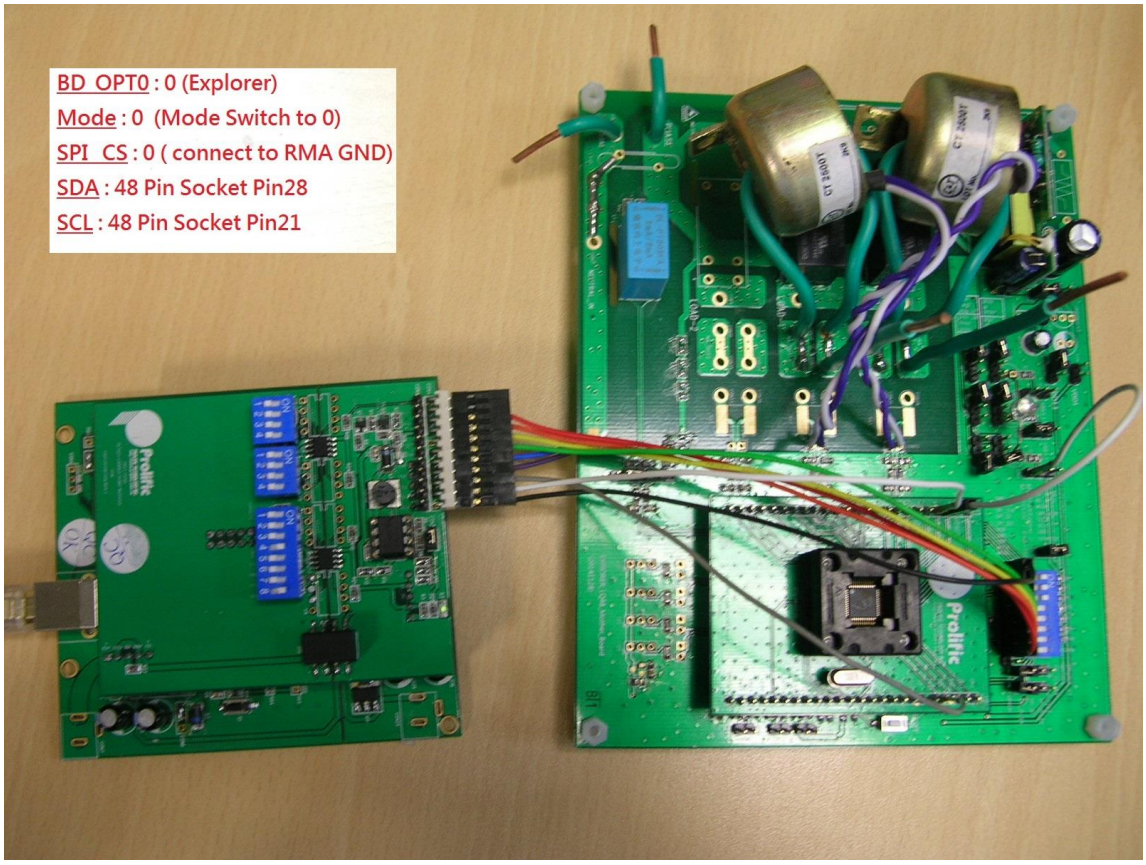


Figure 2-20 : I/F connection

6. AP connection flow

When open the PL7211 AP, you can follow below step to connect and access the raw data via I2C interface,

Reference section 2.1.2 step 1~7, you should selection I2C mode in step3

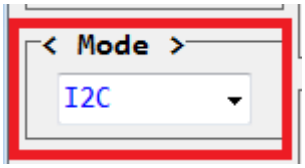


Figure 2-21 : AP I/F selection

3. Multi PL7211s Control

3.1 I²C Slave ID

PL7211 I2C interface maximum support 4 slave, If your MCU want to connect many PL7441s via I2C, you can depend on the difference IO SID to control them. The setting as below:

Slave ID = { 0x380F[4:0] SID+ slaveio }

The default value write in CFG 0x380F = 0x7F= 6b' 111111 (5-bit from 0x380f[4:0]=b' 11111, 1-bit I/O: b'1).

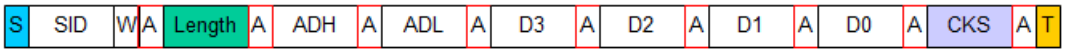
I2C Command : SID + Read or Write Command

Command[7:3] = 0x380F[4:0] SID

Command[2:1] = IO SID1(PAD_P4), IO SID0(PAD_P11)

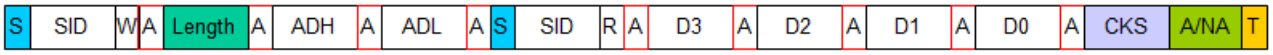
Command[0] = W or R or Sequential Current Read

Write Command

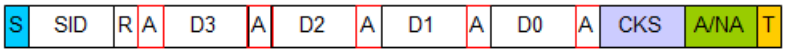


Read Command

Sequential Random address read



Sequential current address read



S : Start ; T : Stop

A : ACK

NA : No ACK

Figure 3-1：I2C Write and Read command

3.2 I2C Sequence

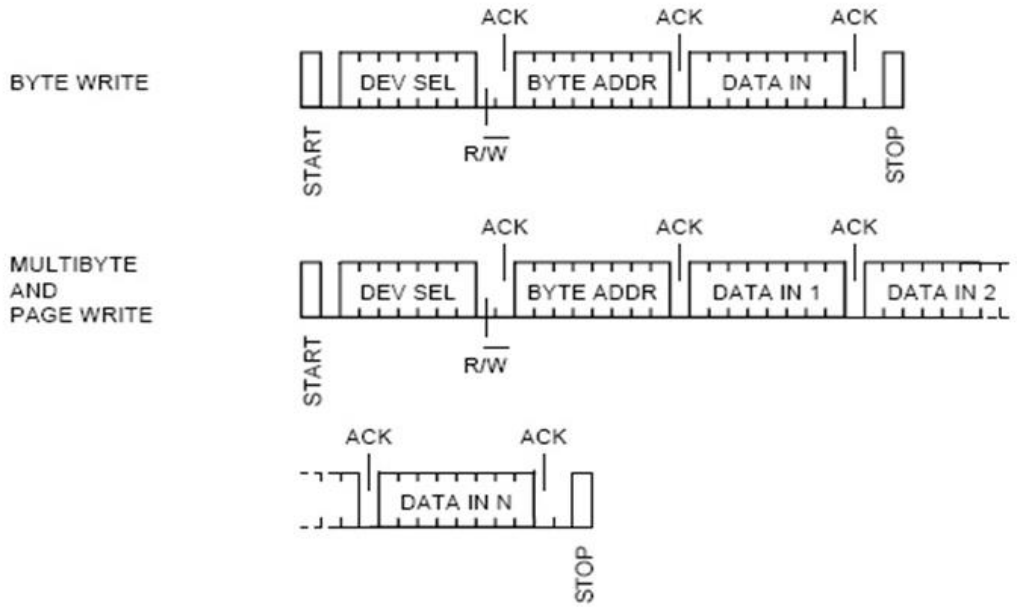


Figure 3-2：I2C Write Sequence

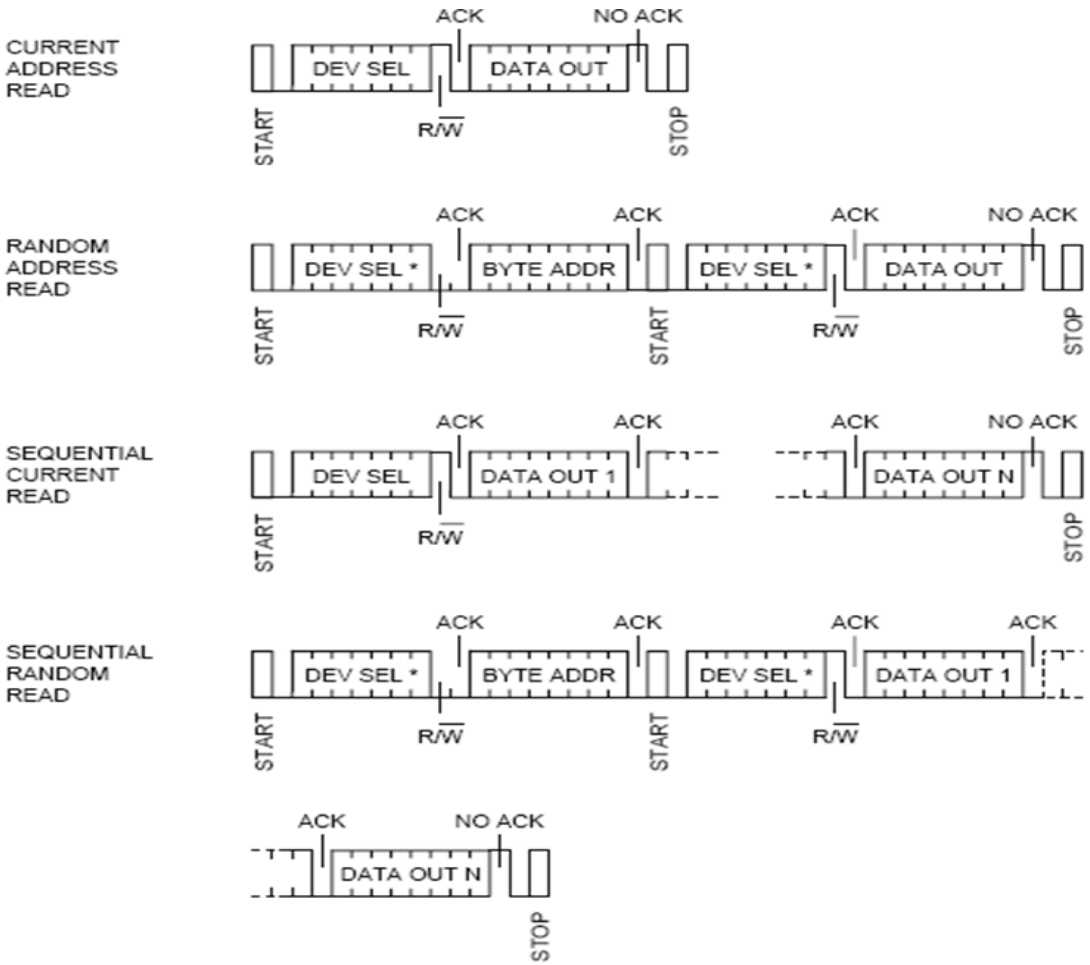


Figure 3-3：I2C Read Sequence

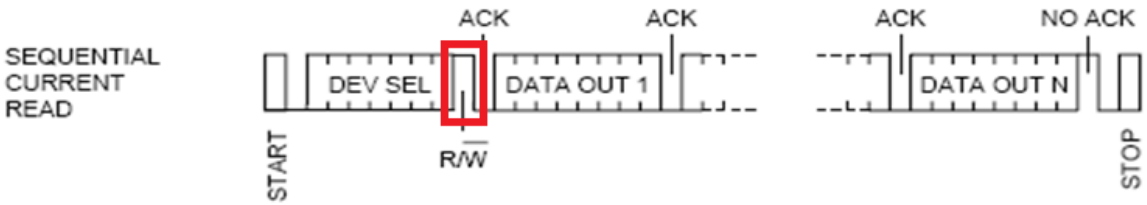


Figure 3-4：I2C Sequential Current Read

3.3 I2C-OTP

The slave ID change mechanism need to dumping from OTP , **PL7211 must write the DSP codes** (CFG.ROM, RO.ROM and DSP.ROM) **into OTP** through the Prolific's calibration board-SPI interface, you can fix 0x380F=0x7f([4:0] SID= b'11111) ,the slave ID will depend on IO SID1 and IO SID0 after you reset PL7211. If the OTP is empty(CFG has not the analog key), the slave id is the default value :bin 1111111(5bit from 0x380f[4:0]=b' 11111, 2-bit from I/O pin: SID1=1, SID0=1).

< Register Table >

Block: OTP High Byte Addr: 00 0x00 - 0x0B

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F			
0	30	C7	80	75	00	00	FF	88	0F	A1	07	40	06	F3	03	7F	Write	Clear	Read
1	00	00	90	00	01	00	00	00	90	00	01	00	00	00	06	30	Write	Clear	Read
2	06	66	60	00	22	00	30	00	00	00	00	00	00	00	00	0F	Write	Clear	Read
3	00	00	00	00	23	08	83	D0	C0	30	06	02	61	00	80	00	Write	Clear	Read
4	00	B0	02	7F	27	88	86	68	03	88	86	68	10	11	00	00	Write	Clear	Read
5	07	0C	07	00	07	00	07	00	07	00	07	00	07	00	07	00	Write	Clear	Read
6	44	45	55	42	E4	1B	8C	00	00	00	00	00	00	00	00	00	Write	Clear	Read
7	00	10	04	0F	00	56	02	40	10	03	00	BE	8F	28	09	02	Write	Clear	Read
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F			

Figure 3-5：PL7211 OTP setting for I2C SlaveID

3.4 I2C-SID IO setting

7. IO SID1=1, IO SID0=1(Default value)

■ HW Setting

PL7211 Demo board: SID1(GPIO4) connects VDD, SID0(SPI_DI) connects VDD

Mode connects GND, SPI_CS connects GND

Figure 3-6：PL7211 IO SID1, IO SID0 PIN Mapping for I2C SlaveID

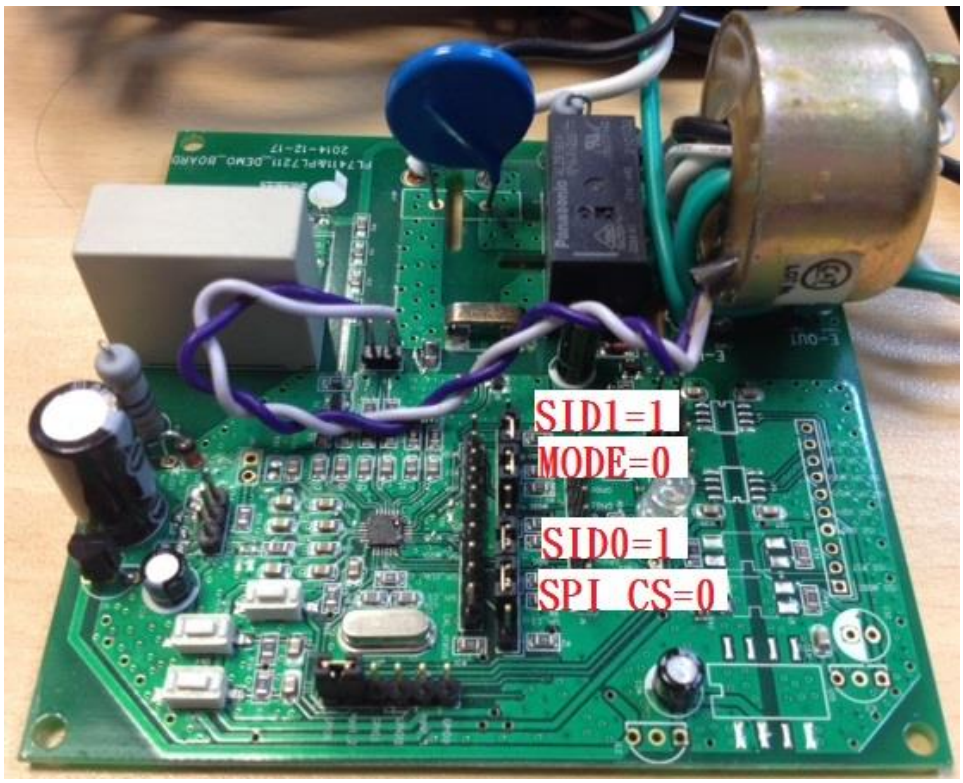


Figure 3-7：PL7211 IO SID1=1, IO SID0=1 Mode/CS setting for I2C SlaveID

■ I2C FW Protocol

Command[7:3] = 0x380F[4:0] SID = 5b’ 11111

Command[2:1] = IO SID1(PAD_P4), IO SID0(PAD_P11) = 2b’ 11

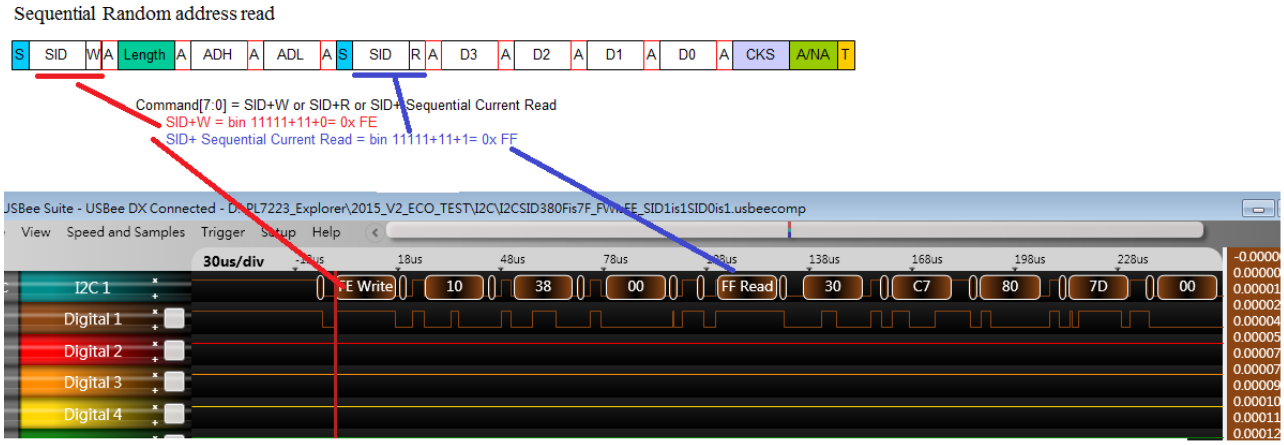
Command[0] = Write (0x0) or Read (0x1) command

Command[7:0] = SID+W or SID+R

SID+Write = 5b’ 11111+ 2b’ 11+ 0= 0x FE

SID+Read = 5b’ 11111 +2b’11+1= 0x FF

Read Command



8. IO SID1=1, IO SID0=0

■ HW Setting

PL7211 Demo board: SID1(GPIO4) connects VDD, SID0(SPI_DI) connects GND

Mode connects GND, SPI_CS connects GND

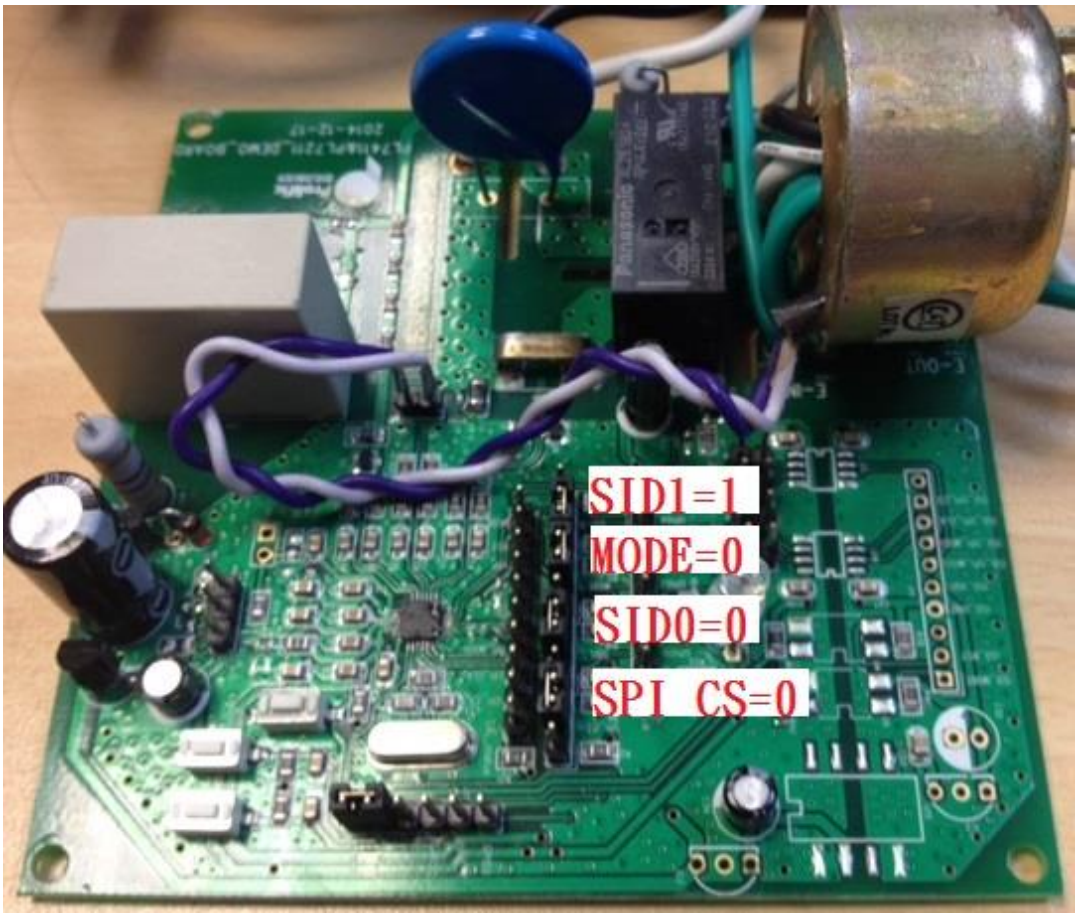


Figure 3-8：PL7211 IO SID1=1, IO SID0=0 Mode/CS setting for I2C SlaveID

■ I2C FW Protocol

Command[7:3] = 0x380F[4:0] SID = 5b’ 11111

Command[2:1] = IO SID1(PAD_P4), IO SID0(PAD_P11) = 2b’ 10

Command[0] = Write (0x0) or Read (0x1) command

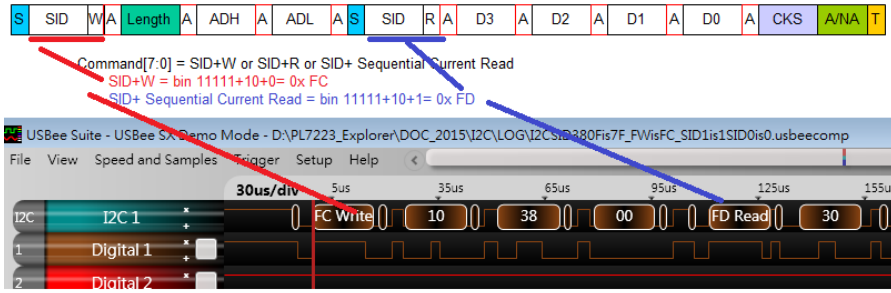
Command[7:0] = SID+W or SID+R

SID+Write = 5b’ 11111+10+0= 0x FC

SID+ Read = 5b’ 11111+10+1= 0x FD

Read Command

Sequential Random address read



9. IO SID1=0, IO SID0=1

■ HW Setting

PL7211 Demo board: SID1(GPIO4) connects GND, SID0(SPI_DI) connects VDD

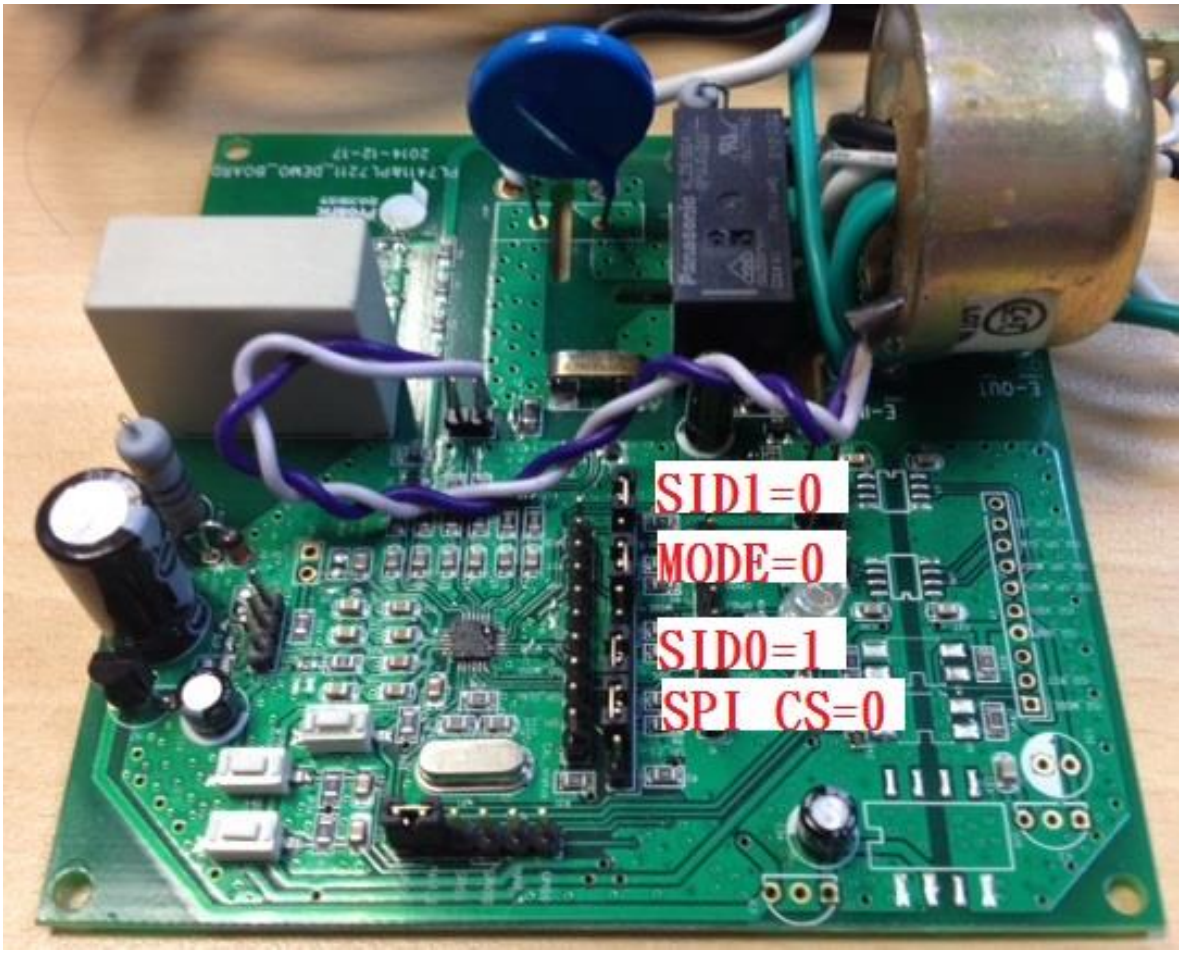


Figure 3-9：PL7211 IO SID1=0, IO SID0=1 Mode/CS setting for I2C SlaveID

- I2C FW Protocol

  Command[7:3] = 0x380F[4:0] SID = 5b' 11111

  Command[2:1] = IO SID1(PAD_P4), IO SID0(PAD_P11) = 2b' 01

  Command[0] = Write (0x0) or Read (0x1) command

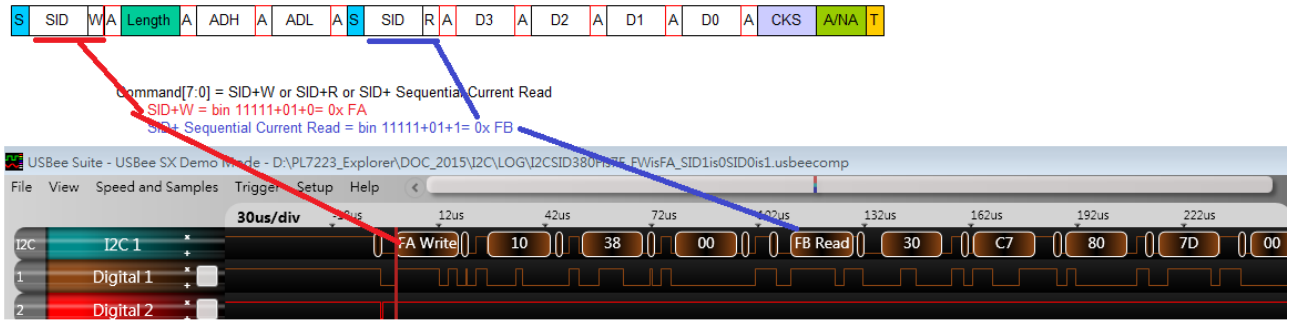
Command[7:0] = SID+W or SID+R

SID+Write = 5b' 11111+2b' 01+0= 0x FA

SID+ Read = 5b' 11111+2b' 01+1= 0x FB

Read Command

Sequential Random address read



10. IO SID1=0, IO SID0=0

- HW Setting

PL7211 Demo board: SID1(GPIO4) connects GND, SID0(SPI_DI) connects GND

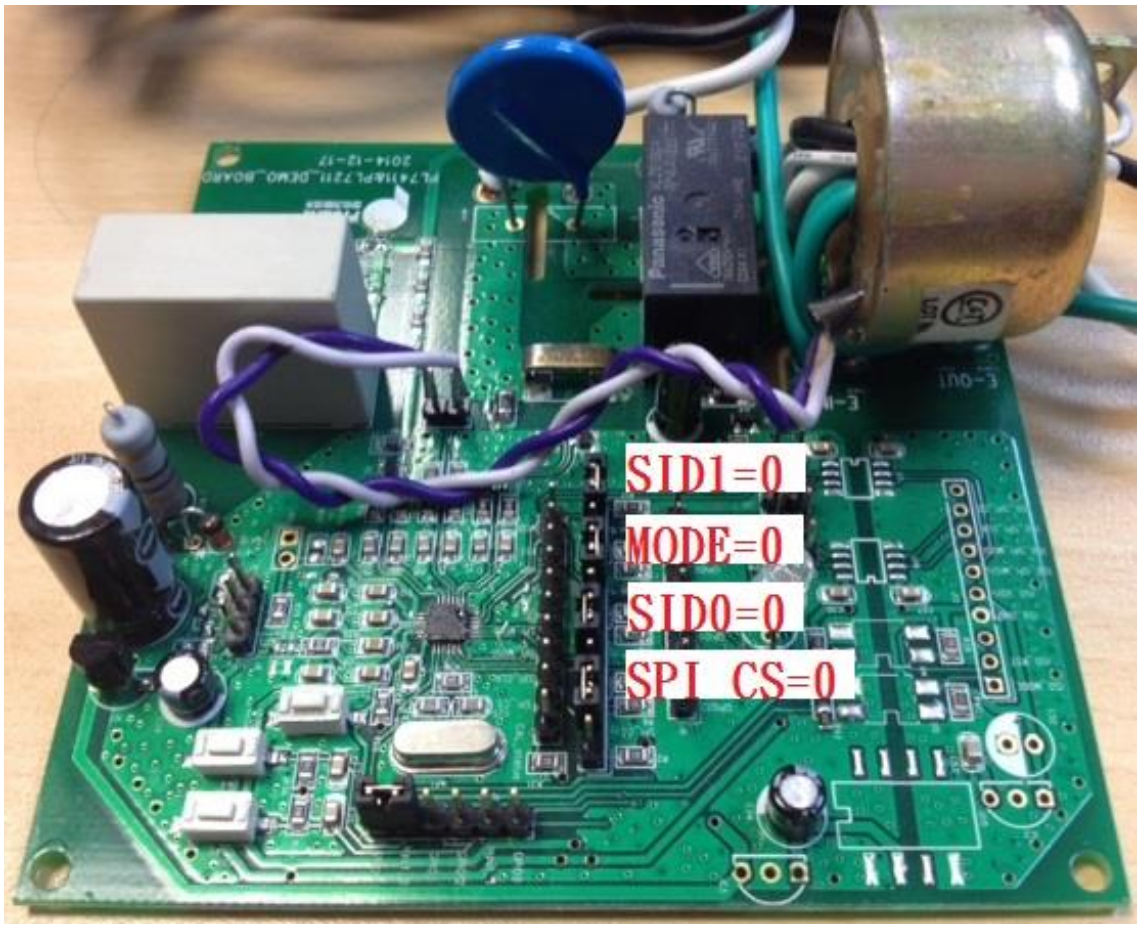


Figure 3-10：PL7211 IO SID1=0, IO SID0=0 Mode/CS setting for I2C SlaveID

- I2C FW Protocol

  Command[7:3] = 0x380F[4:0] SID = 5b’ 11111

  Command[2:1] = IO SID1(PAD_P4), IO SID0(PAD_P11) = 2b’ 00

  Command[0] = Write (0x0) or Read (0x1) command

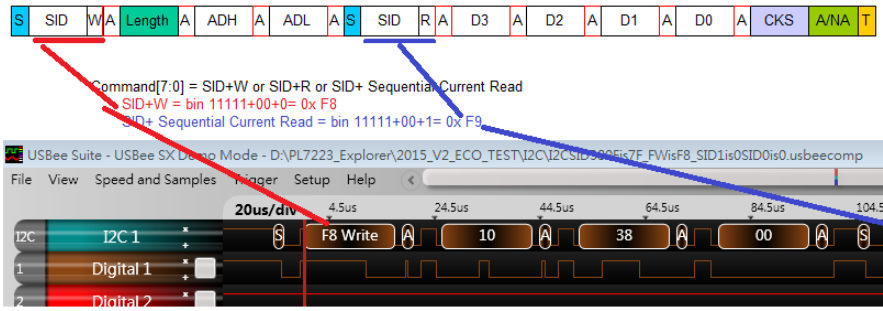
Command[7:0] = SID+W or SID+R

SID+W = 5b’ 11111+00+0= 0x F8

SID+ Read = 5b’ 11111+00+1= 0x F9

Read Command

Sequential Random address read



3.5 Multi PL7211 control via UART interface

If your MCU want to control two of PL7441s via UART, it needs the difference CS pins to control the difference PL7211.

Configure 0x380d as 0x3f to these two device at the same time, for same SlaveID of DUT1 and DUT2 (**PL7211 Must write CFG of DUT1 and DUT2 first** , that is 0x3800~0x38ff must have the codes inside).

CFG of DUT1 and DUT2 must have the codes :0x3800~0x38ff:

< Register Table >

Block: RAM CFG Regi High Byte Addr: 38 0x38 - 0x39

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F			
0	30	C7	00	7D	04	10	FF	88	0F	A1	07	40	06	F3	03	7F	Write	Clear	Read
1	00	00	90	00	01	00	00	00	90	00	01	00	00	00	06	32	Write	Clear	Read
2	06	66	60	00	22	00	30	00	00	00	00	00	00	00	00	0F	Write	Clear	Read
3	00	00	00	00	23	08	83	D0	C0	30	06	02	61	00	80	00	Write	Clear	Read
4	00	B0	02	7F	27	88	86	68	03	81	86	68	10	11	00	00	Write	Clear	Read
5	07	0F	07	00	07	00	07	00	07	00	07	00	07	00	07	00	Write	Clear	Read
6	45	45	59	4A	E4	1B	8C	00	00	00	00	00	00	00	00	00	Write	Clear	Read
7	00	50	14	0F	00	56	02	40	10	03	00	BE	8F	28	09	02	Write	Clear	Read

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F			
8	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	Write	Clear	Read
9	50	41	08	BB	22	FF	F7	02	00	FF	0E	FF	FF	FF	F0	F0	Write	Clear	Read
A	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	Write	Clear	Read
B	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	Write	Clear	Read
C	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	Write	Clear	Read
D	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	Write	Clear	Read
E	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	Write	Clear	Read
F	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	Write	Clear	Read

Test 0x00~0xFF DSP Enable Check all all all

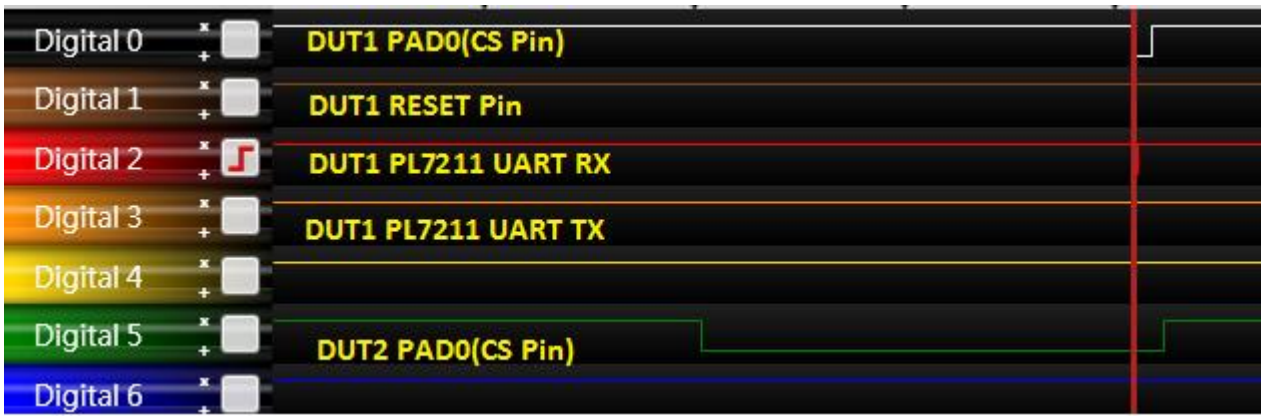
- **Configure UART multi-Slave Method:**

(a) Force low to active PAD0 (CS Pin) of DUT1 and DUT2 (CS pin controlled by your uC GPIO)

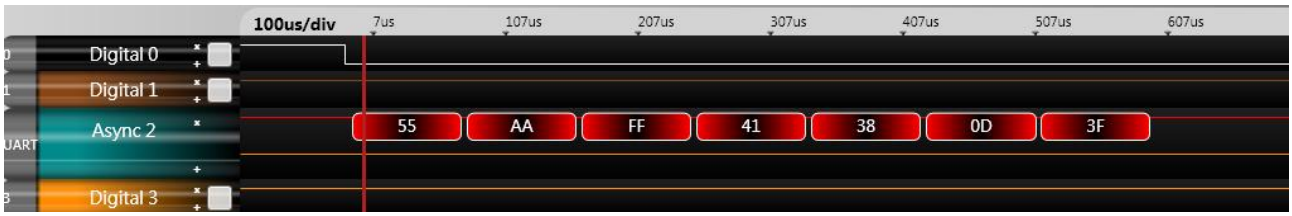
(b) Configure 0x380Das 0x3F to two PL7211 CFG at the same time.

Step as below:

Force DUT2 CS Low-> Force DUT1 CS Low->Write 0x380D to 0x3F->Resume DUT1 CS High-> Resume DUT2 CS High



ZOOM IN Write 0x380D to 0x3F:



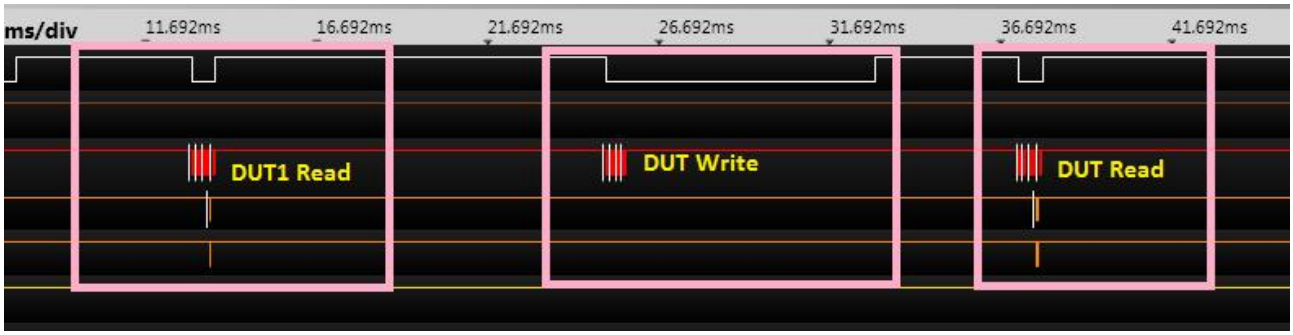
- **Read/Write DUT1 Method**

(a) Force low at PAD0(CS Pin) of DUT1 and Force high at PAD0 of DUT2 to call DUT1

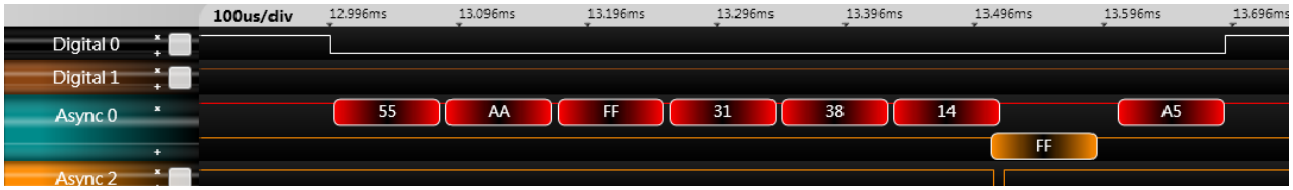
(b) Read or write data from DUT1

Step as below:

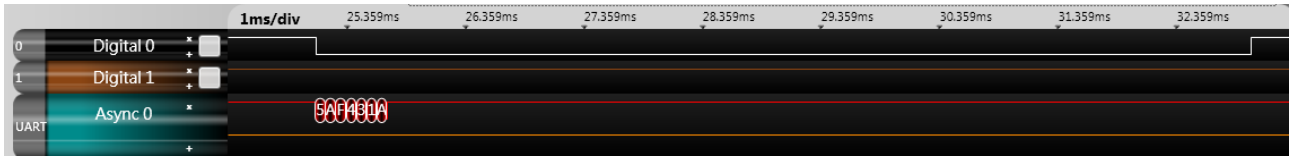
Control the CS pin (High->Low->High)of DUT1 During issue command , That is same as the CS pin control of SPI interface.



ZOOM IN: DUT1 Read



ZOOM IN: DUT1 Write

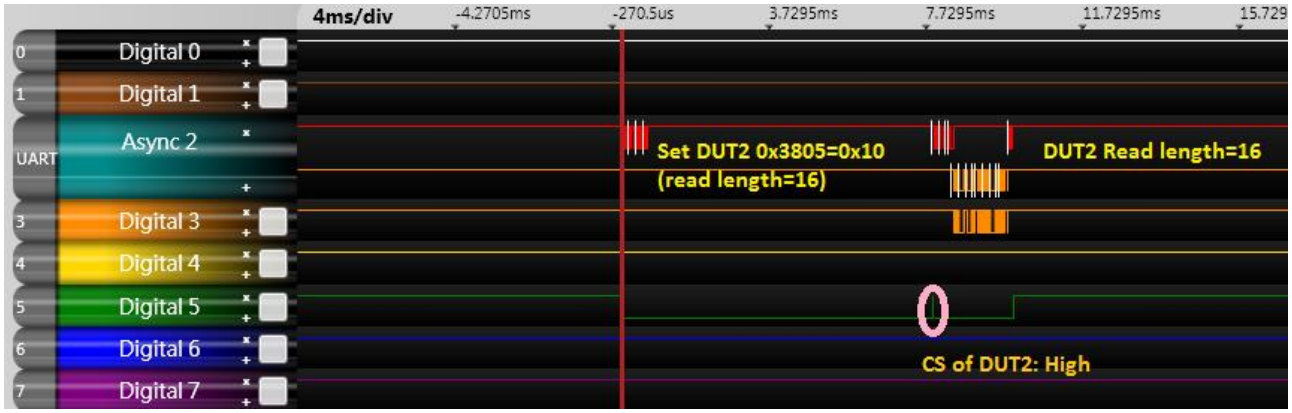


- **Read/Write DUT2 Method**

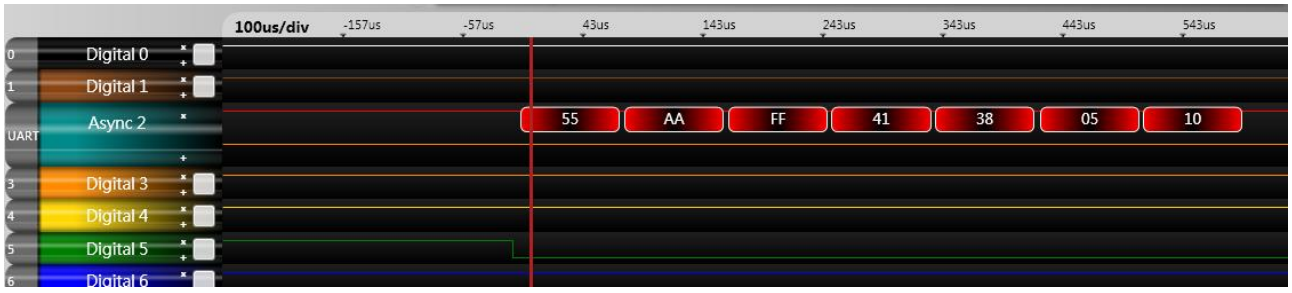
Force low at PAD0 of DUT2 and Force high at PAD0 of DUT1 to call DUT2

Read or write data from DUT2

(e) : Control the CS pin (High->Low->High)of DUT2 During issue command , That is same as the CS pin control of SPI interface.



ZOOM IN: Write DUT2 0x3805 as 0x10(read length=16)



ZOOM IN: Read DUT2 0x3800, length=16 byte

3.6 UART Slave ID

The setting as below

SID[7:0] = 0x380F[5:0] SID+SlaveIO

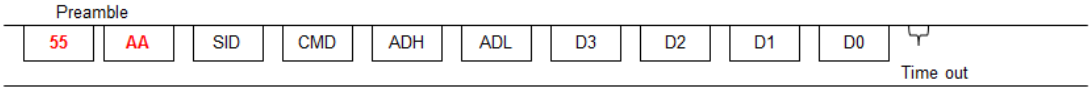
Command [7:2] = 0x380F[5:0] SID

Command [1:0] = IO SID1(PAD_P4), IO SID0(PAD_P11)

The default value : 8b' 11111111=0xFF (6-bit from 0x380f[5:0]=b' 111111, 2-bit from I/O pin: SID1=1, SID0=1).

Because UART's RxD pin is shear pin with SID0(PAD_P11), so it only has SID1=0 and SID1=1 two combinations.

Write Command



Read Command

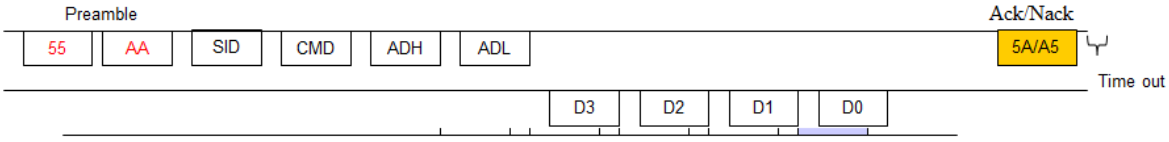


Figure 3-11：UART Write and Read Command

3.7 UART-OTP

The slave ID change mechanism need to dumping from OTP , **PL7211 must write the DSP codes** (CFG.ROM, RO.ROM and DSP.ROM) **into OTP** through the Prolific's calibration board-SPI interface, you can fix 0x380F=0x7f([5:0] SID= bin 111111), the slave ID will depend on IO SID1 and IO SID0 after you reset PL7211.

If the OTP is empty(CFG has not the analog key), the slave id is the default value :bin 11111111(6-bit from 0x380f[5:0]=bin 111111,2-bit from I/O pin: SID1=1, SID0=1).

< Register Table >

Block: OTP High Byte Addr: 00 0x00 - 0x0B

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F			
0	30	C7	80	75	00	00	FF	88	0F	A1	07	40	06	F3	03	7F	Write	Clear	Read
1	00	00	90	00	01	00	00	00	90	00	01	00	00	00	06	30	Write	Clear	Read
2	06	66	60	00	22	00	30	00	00	00	00	00	00	00	00	0F	Write	Clear	Read
3	00	00	00	00	23	08	83	D0	C0	30	06	02	61	00	80	00	Write	Clear	Read
4	00	B0	02	7F	27	88	86	68	03	88	86	68	10	11	00	00	Write	Clear	Read
5	07	0C	07	00	07	00	07	00	07	00	07	00	07	00	07	00	Write	Clear	Read
6	44	45	55	42	E4	1B	8C	00	00	00	00	00	00	00	00	00	Write	Clear	Read
7	00	10	04	0F	00	56	02	40	10	03	00	BE	8F	28	09	02	Write	Clear	Read
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F			

Figure 3-12：PL7211 OTP setting for UART SlaveID

3.8 UART-SID IO setting

11. IO SID1=1, IO SID0=1(Default value)

■ HW Setting

PL7211 Demo board: SID1(GPIO4) connect to VDD, SID0(SPI_DI) connect to VDD, Mode connect to GND, SPI_CS connect to VDD

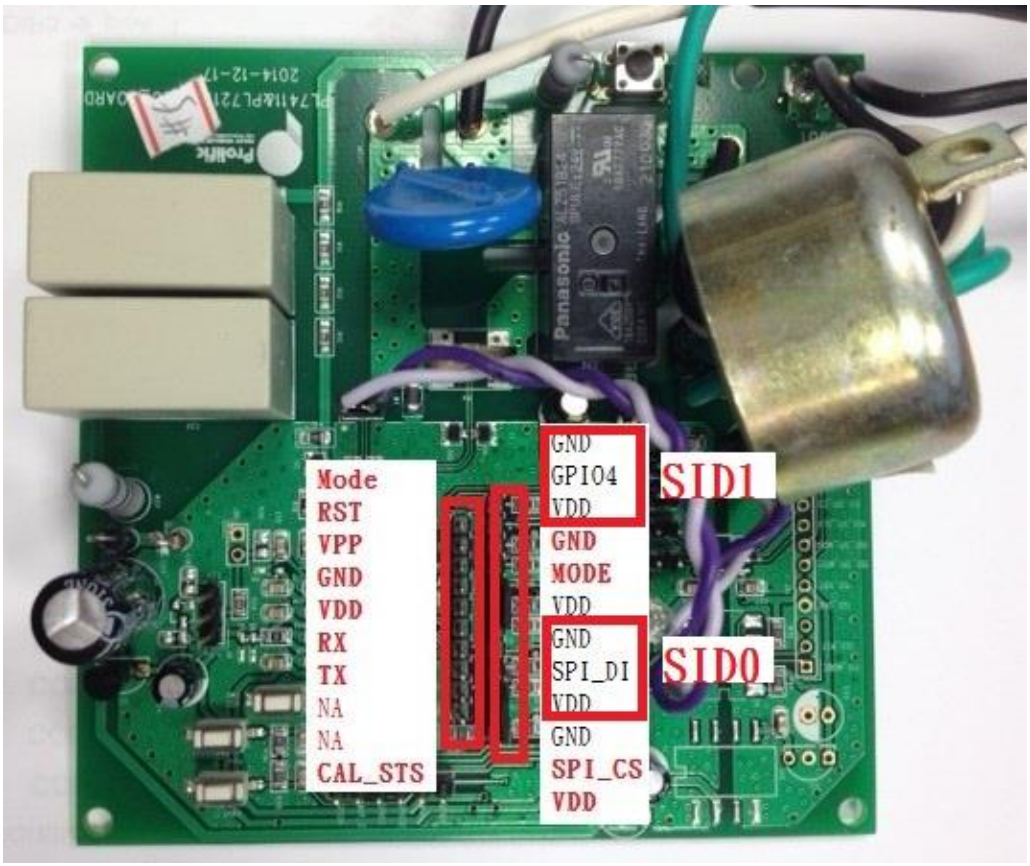


Figure 3-13：PL7211 IO SID1, IO SID0 PIN Mapping for UART SlaveID

Figure 3-14：PL7211 IO SID1=1, IO SID0=1 Mode/CS setting for UART SlaveID

■ UART FW Protocol

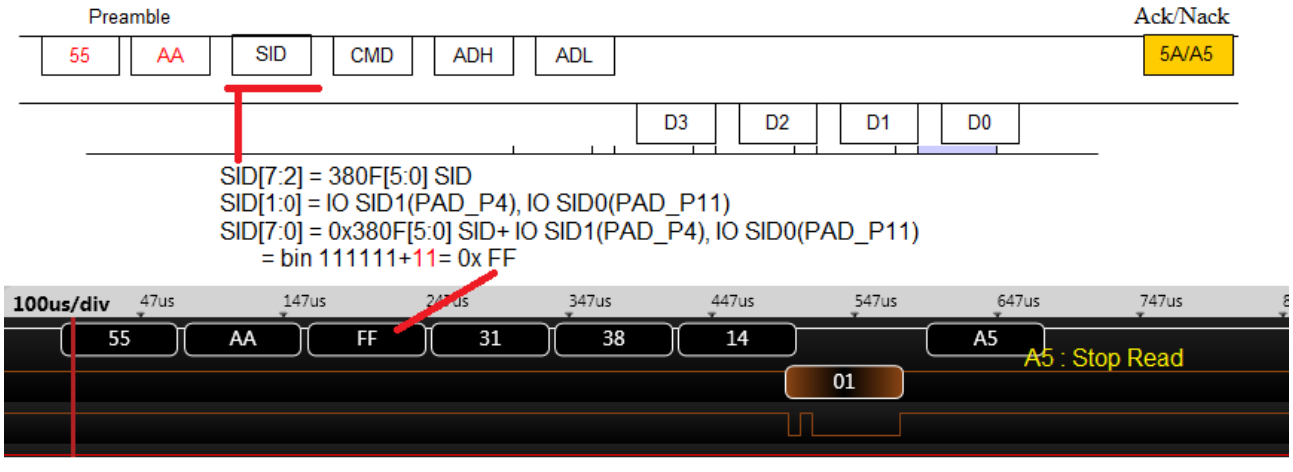
Command [7:2] = 380F[5:0] SID

Command [1:0] = IO SID1(PAD_P4), IO SID0(PAD_P11) = 2b' 11

Command [7:0] = 0x380F[5:0] SID+ IO SID1(PAD_P4), IO SID0(PAD_P11)

= 6b' 111111+11 = 0xFF

Read Command



12. IO SID1=0, IO SID0=1

■ HW Setting

PL7211 Demo board: SID1(GPIO4) connect to GND, SID0(SPI_DI) connect to VDD , Mode connect to GND, SPI_CS connect to VDD

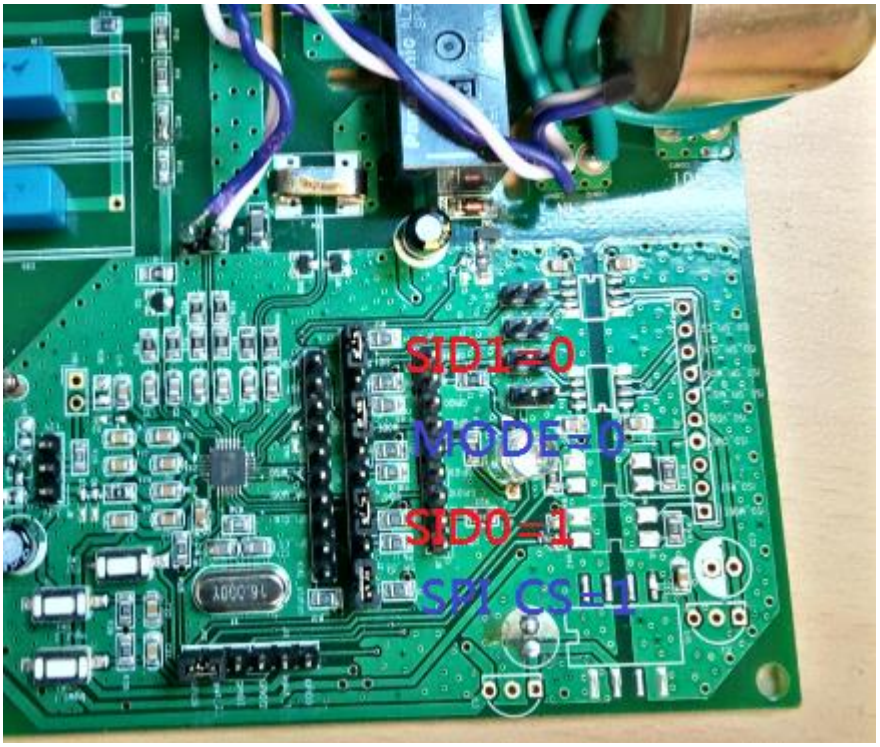


Figure 3-15：PL7211 IO SID1=0, IO SID0=1 Mode/CS setting for UART SlaveID

■ UART FW Protocol

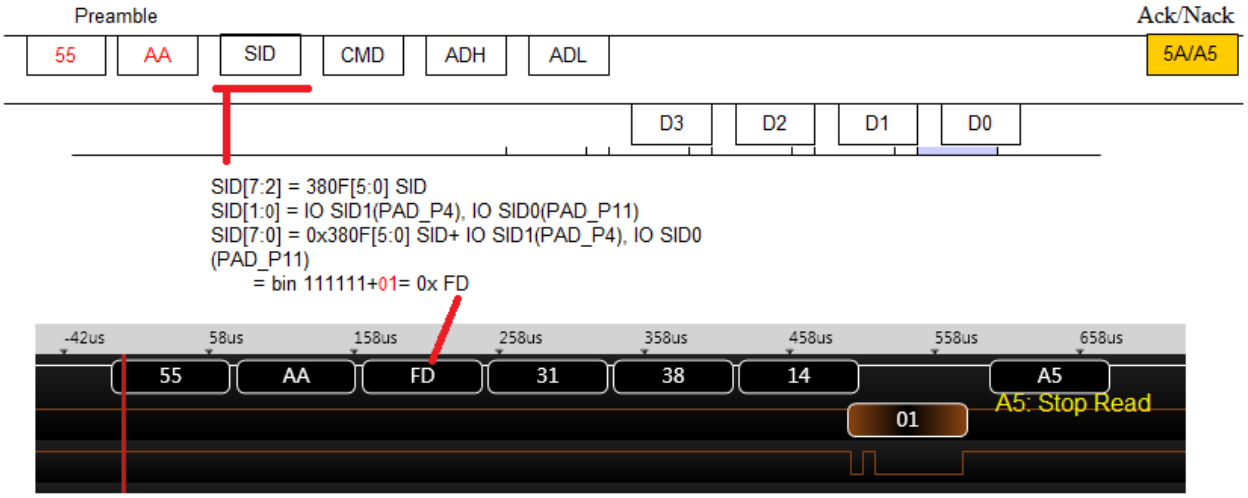
Command [7:2] = 0x380F[5:0] SID

Command [1:0] = IO SID1(PAD_P4), IO SID0(PAD_P11) = 2b' 01

Command [7:0] = 0x380F[5:0] SID+ IO SID1(PAD_P4), IO SID0(PAD_P11)

= 6b' 111111 + 01= 0xFD

Read Command



4. Power Protection Function Description

The power protection function of PL7211 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.

4.1 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.

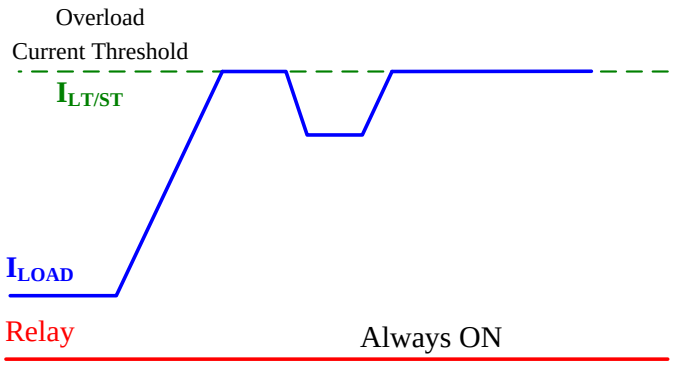


Figure 4-1：Description of long/short time protection (ILOAD = ILS/ST)

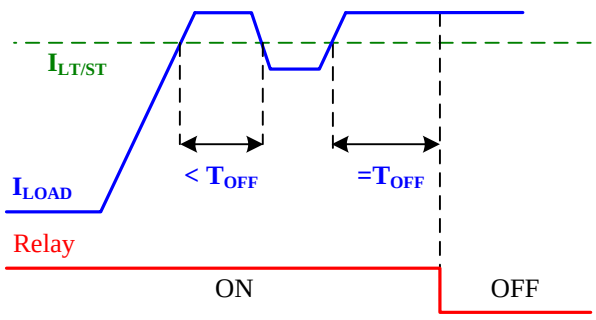


Figure 4-2：Description of long/short time protection (ILOAD > ILS/ST)

4.2 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:

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

4.3 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

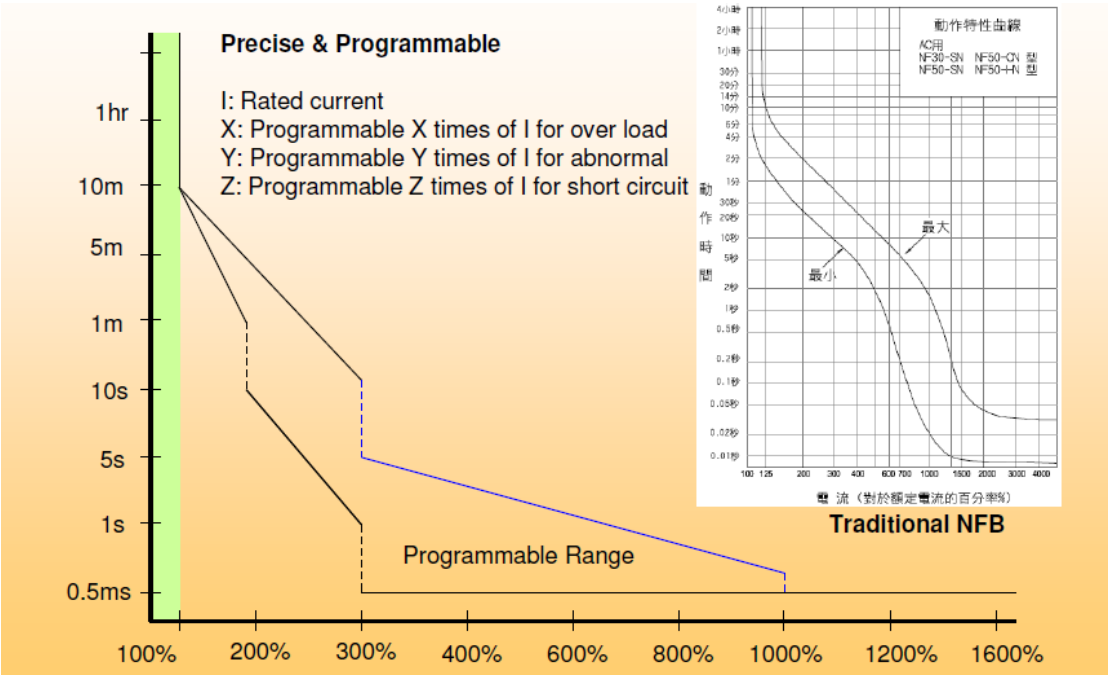


Figure 4-3：Programmable Range

4.4 OCP Threshold

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

Table 4-1：Leakage and OCP setting Table

4.5 AVM (Auto Voltage Margin)

PL7211 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.

4.6 AVM introduction

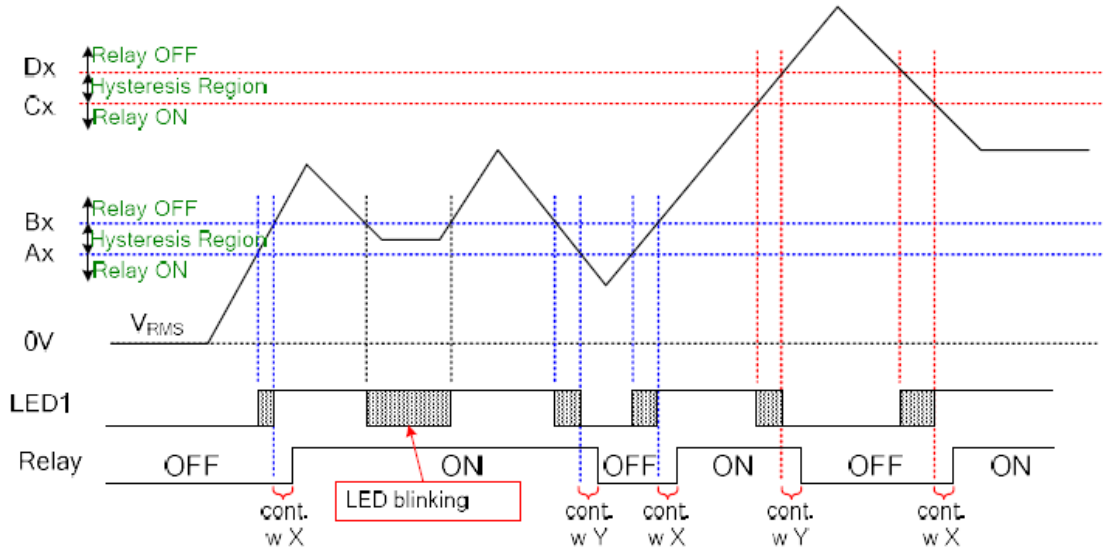
By setting AX /BX/ CX/ DX, you can use PL7211 AVM function shown as follow figure.

PL7211 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 4-2：AVM threshold



4.7 DSP AVM flow

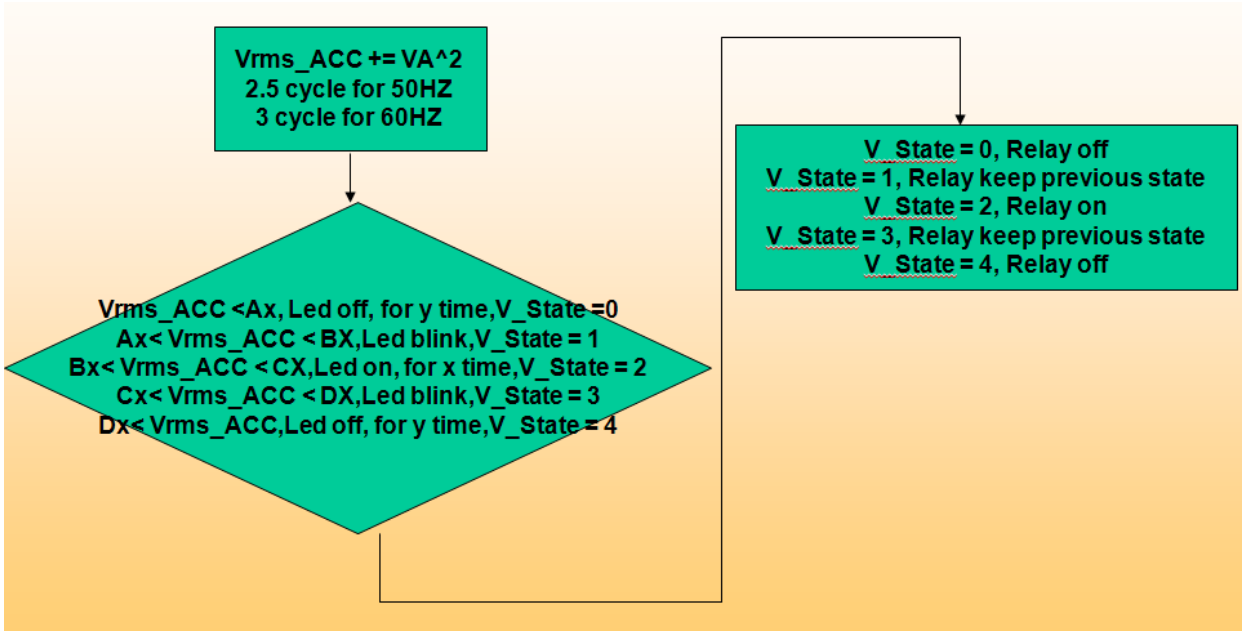


Figure 14：AVM DSP flow

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

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

5. AC Calibration Flow

PL7211 AP can suitable to calibration with HS-3103 and KP-1001 power source, after connect with device environment, you can follow the test follow to do AC calibration.

5.1 Setup environment

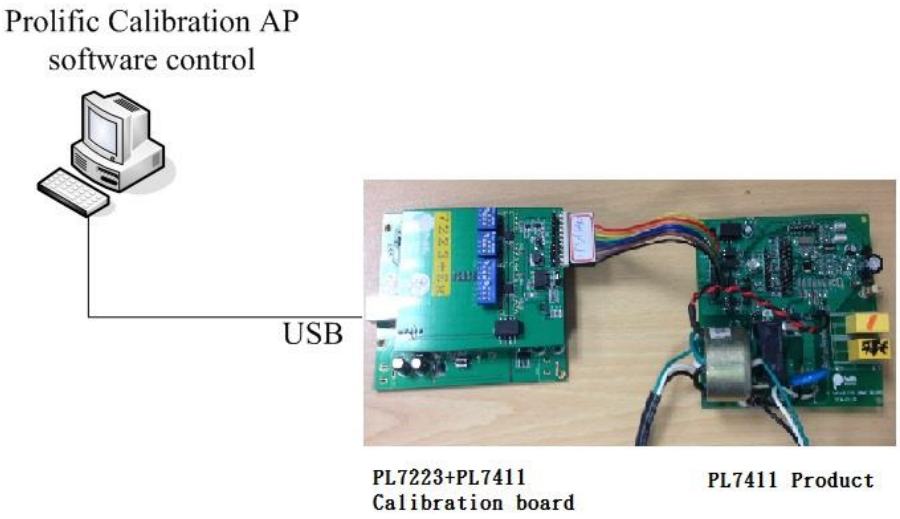


Figure 5-1 : Calibration environment setup

5.2 PL7211 Mode

Because DSP has 3 types codes for PL7211-AFE+AVM+OCP+Leakage / PL7211-1V3I /PL722X pin to pin, please make sure the mode is " AFE+AVM+OCP+Leakage" as below:

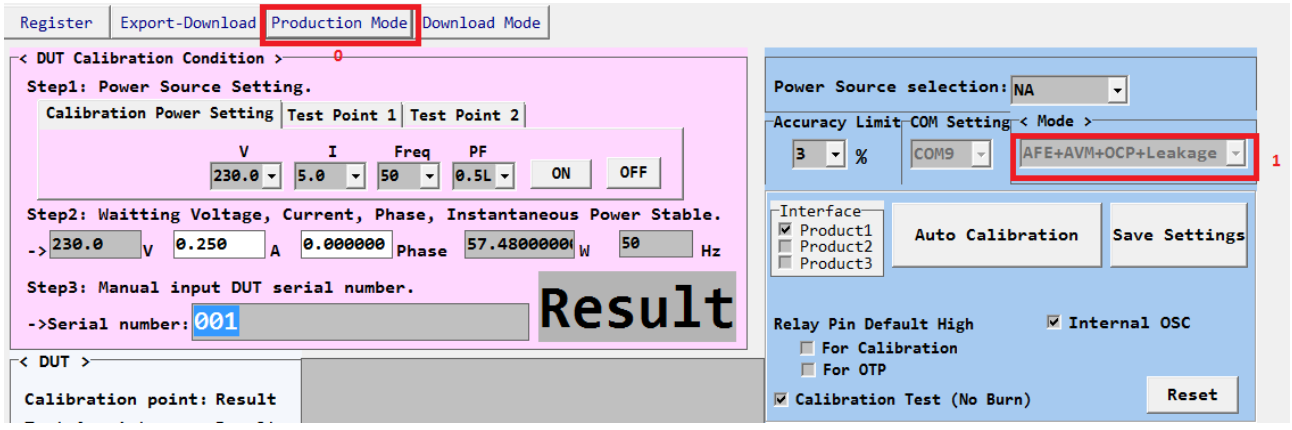


Figure 5-2 : PL7211 mode

Map with this Mode , AP will load the files from C:\Explorer\AFE\ :

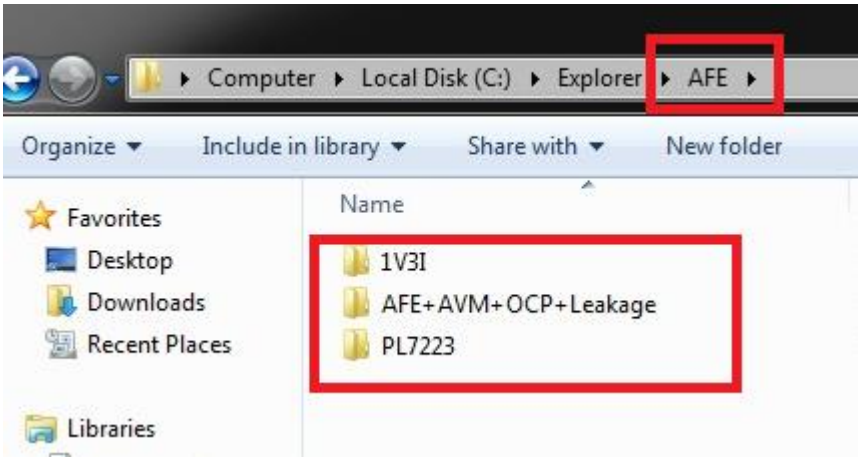


Figure 5-3 : PL7211 rom code path

5.3 AP Burn DSP

If you are first running PL7211 or you have calibration complete and export the DSP/RO/CFG/, then place the file in upper folder, You can use download mode to burn AP

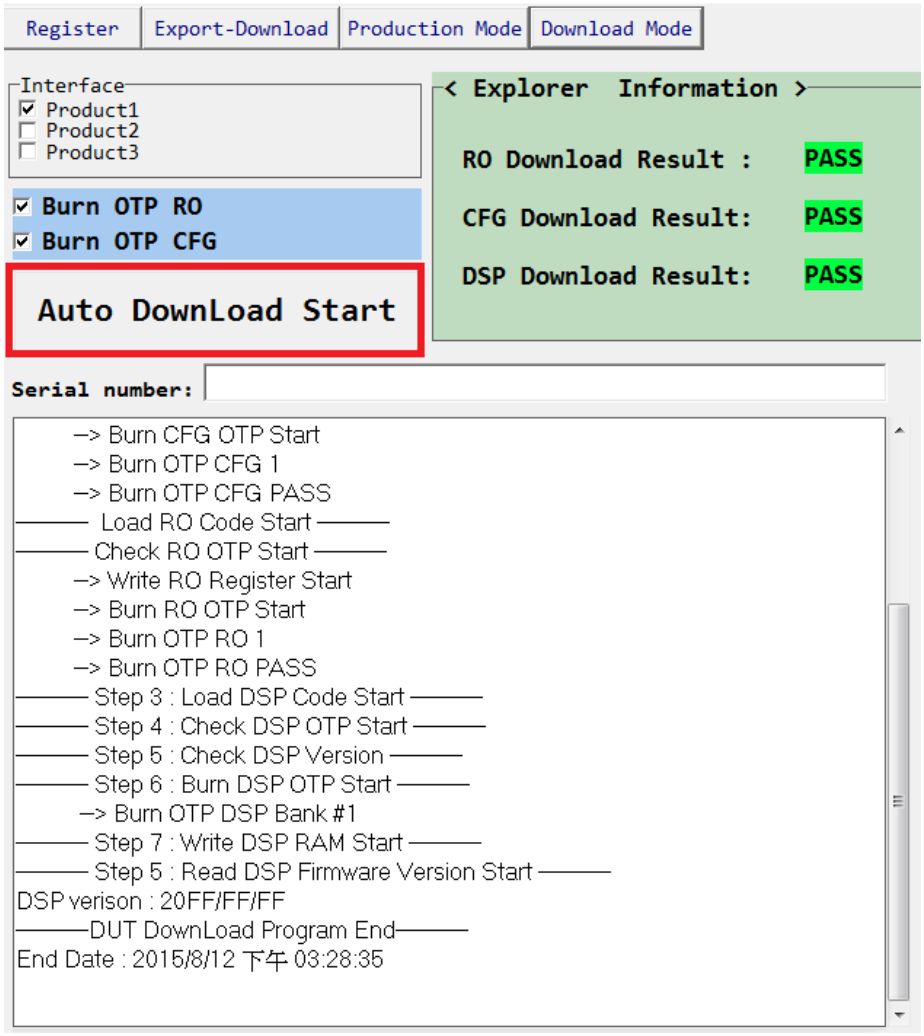


Figure 5-4 : AP burn DSP

5.4 AP enable a function before auto calibration

Example: Enable LT and Inst(OCP)

(1): If you want to enable LT, ST and Inst(OCP) functions, please select “LT Enable”, and “INST Enable”:

(2): If you want to save them, please key press the save settings button.

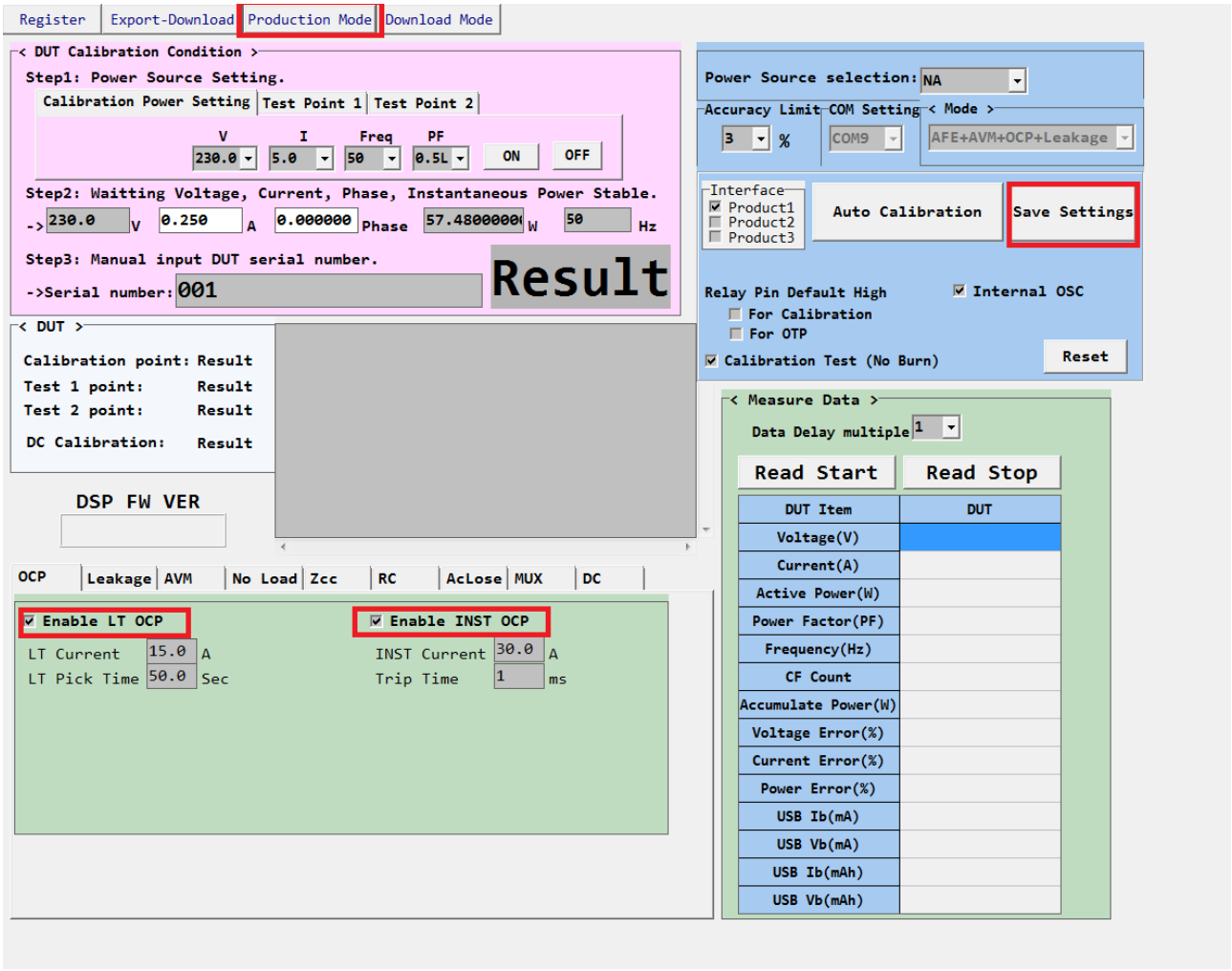


Figure 5-5: Enable LT OCP/ST OCP/INST OCP

Example: Use Internal OSC

1. If your HW don’t have the external Crystal(16MHz), and PL7221 OTP CFG BANK0 has the NT/PT trim codes, please select “Internal OSC”:

2. **OTP record address and record value: If value exist , It’s mean PL7211 have trimed code**

   0068 : VREF trim to 1.22 value

   0069 : bandgap 0x5E

   1C69~1C6B :Tsensor value

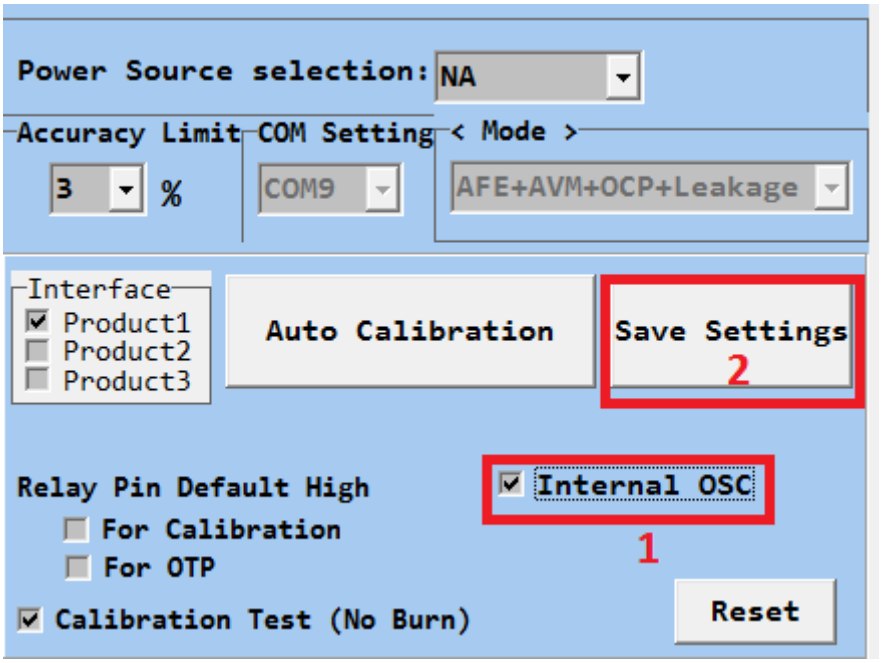


Figure 5-6: Enable Internal OCP

OTP CFG BANK1:

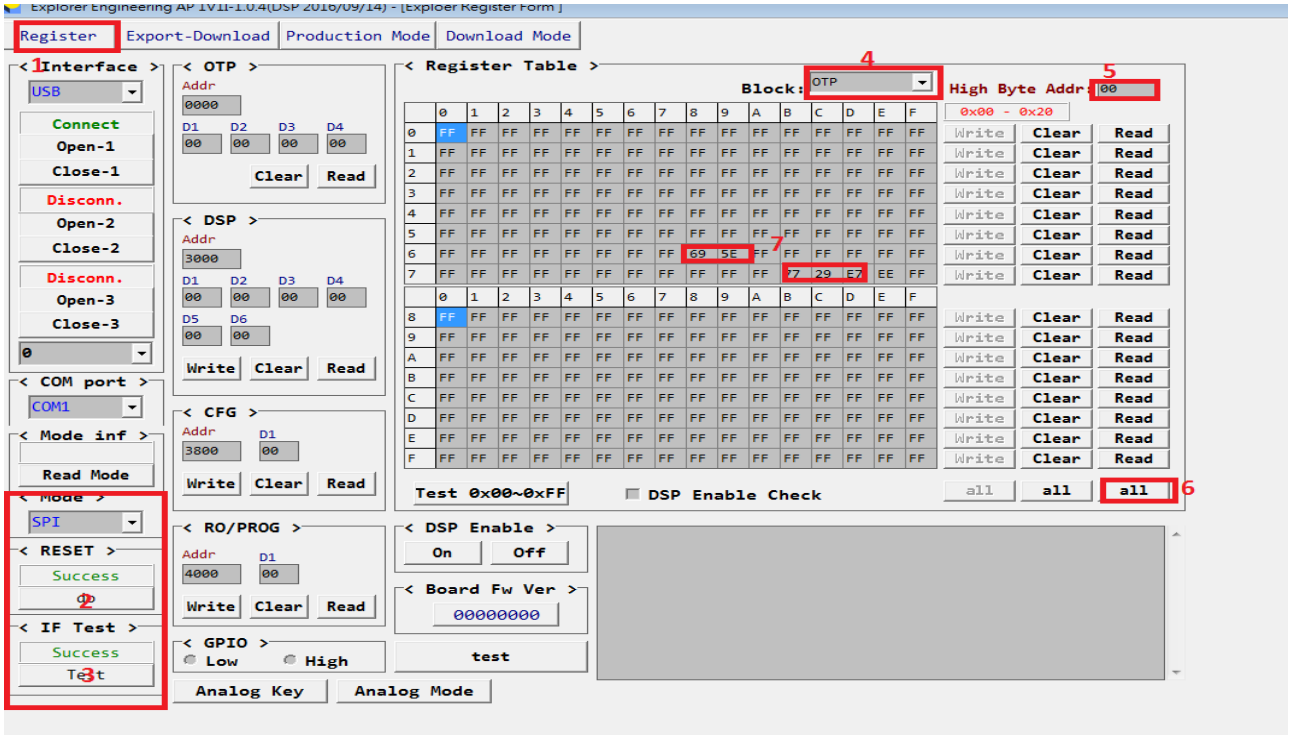


Figure 5-7: OTP CFG BANK1 value

OTP RO BANK1:

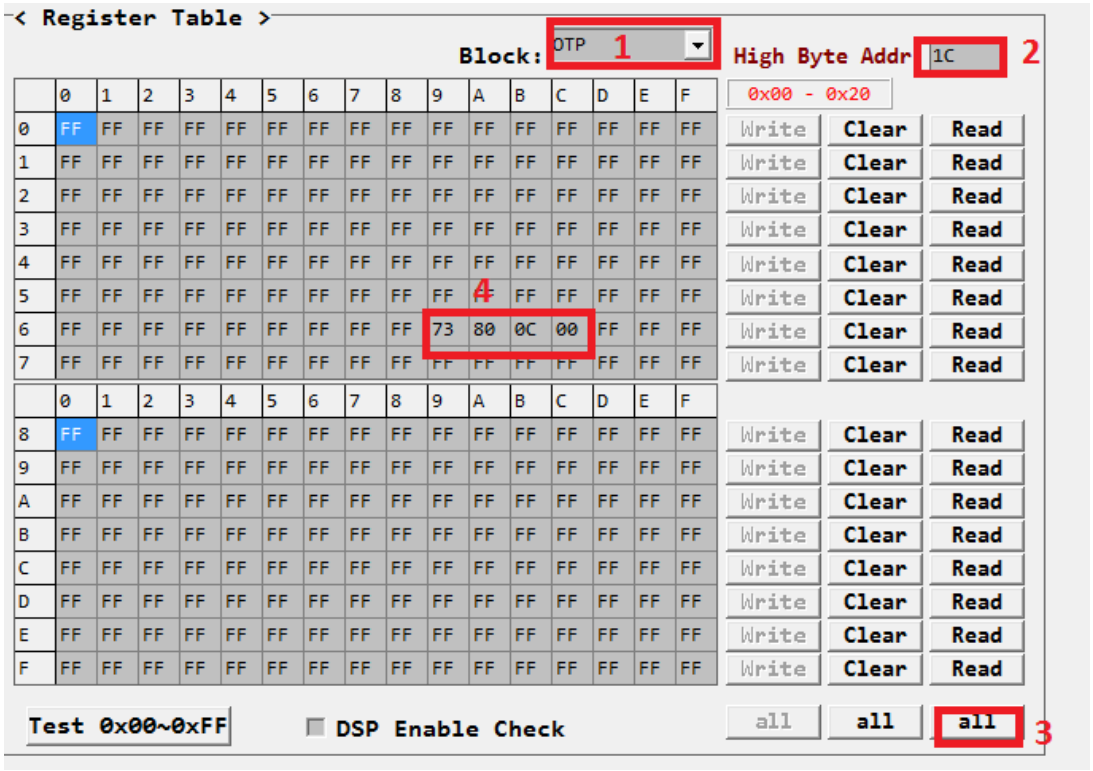


Figure 5-8: OTP RO BANK1 value

5.5 AP auto calibration flow:

User don't need download the DSP/CFG/RO, after calibration done, will burn the OTP directly

Step1: select AC Power source (KP1001 / NA / HS3103)

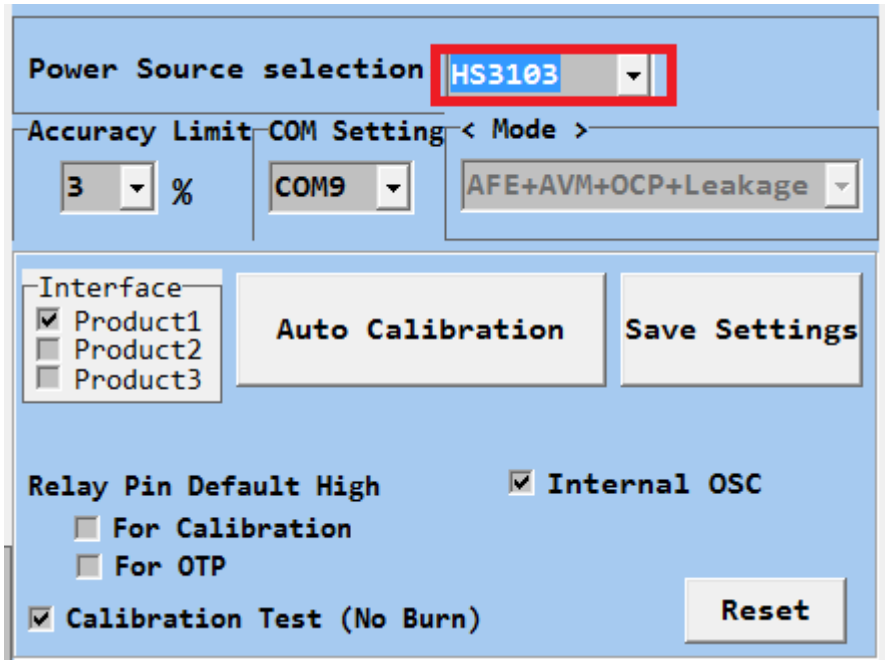


Figure 5-9: AP select power source

Step2: select power source and com port

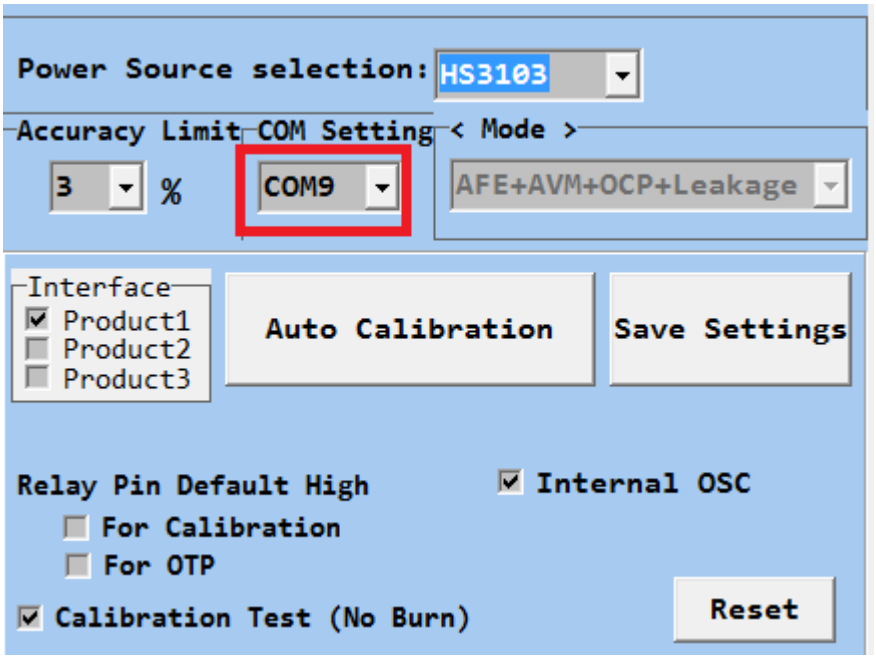


Figure 5-10 : AP setting com port

Step3: set production accuracy limit

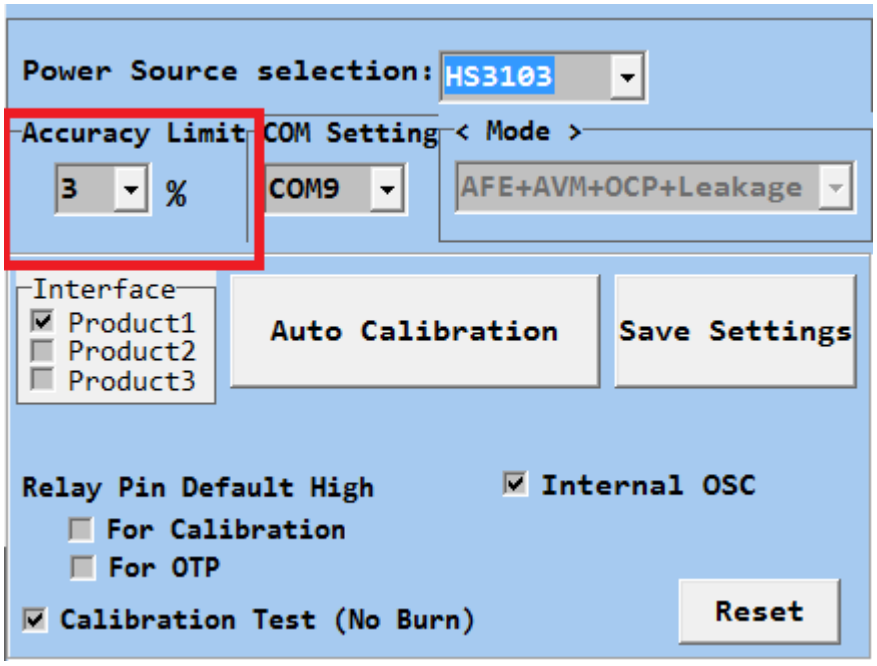


Figure 5-11: AP setting accuracy

Step4: if power source set "NA", Please manual enter Voltage/Current /Power

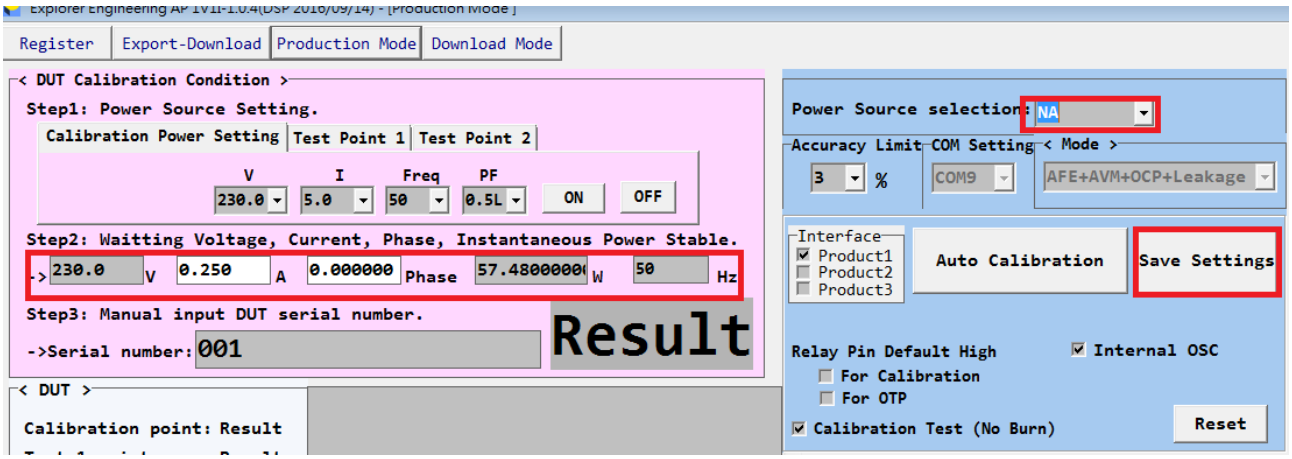


Figure 5-12: AP manual V/I/P

If select the “Single Phase Dummy Load” as power source , you need to measurement the real V and real I , Then manual input , As below, :

Example :We use PL8331 multi-meter to measure it’s real V and I

Dummy load voltage 110V -> multi-meter measure real is 124.46V.
Dummy load voltage 230V -> multi-meter measure real is 259.09V.
Dummy load current 5A -> multi-meter measure real is 4.74A.
Dummy load current 1A -> multi-meter measure real is 1.17A.

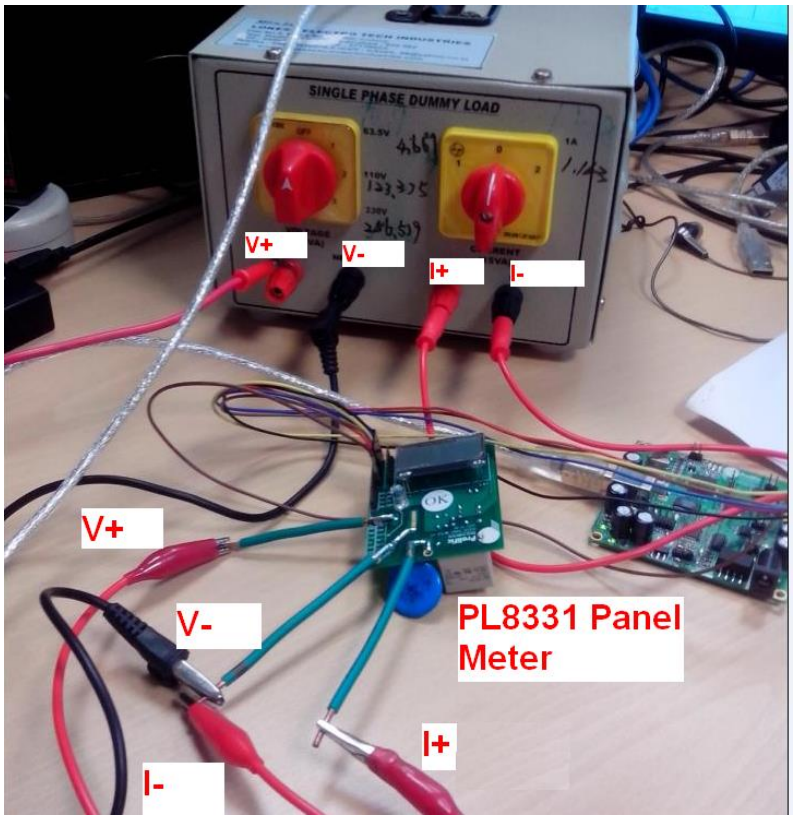


Figure 5-13: Dummy load manual V/I/P measurement

Manual enter the real value into below V/I label , We will use the real V and real I for calibration.

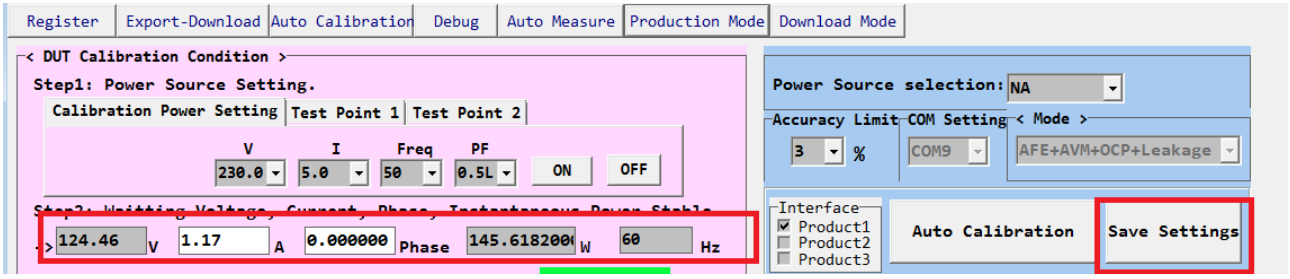


Figure 5-14: AP manual enter V/I/P

Suggest at least 1A current to calibration.

Switch single phase dummy load to 110V (real 123.375V),1A (real 1.163A) :

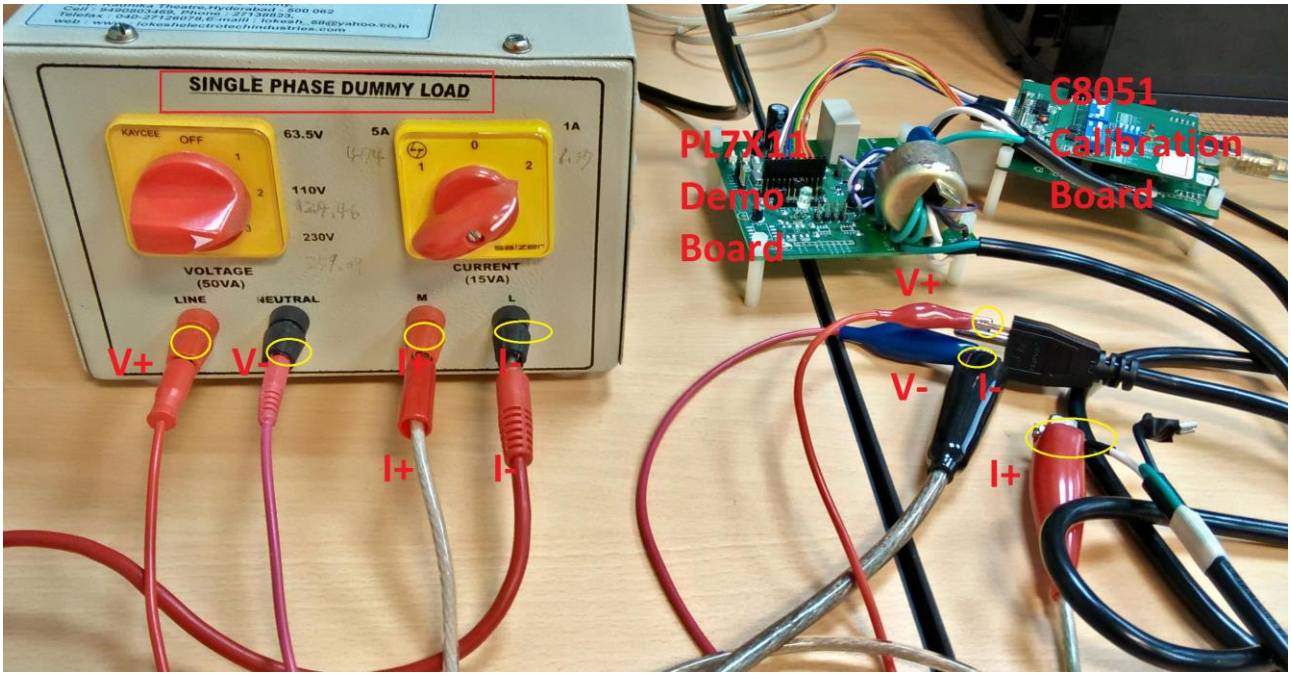


Figure 5-15: Dummy load with PL7211 Calibration

Step5: Enter production serial number

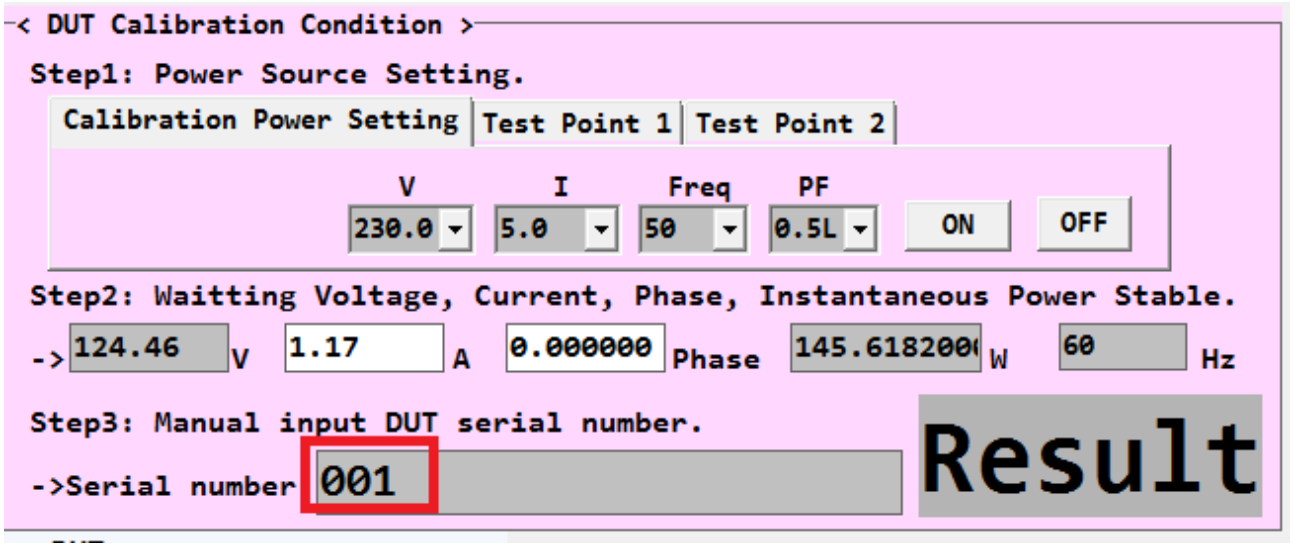


Figure 5-16: AP production serial number

Step6: click Auto calibration start button

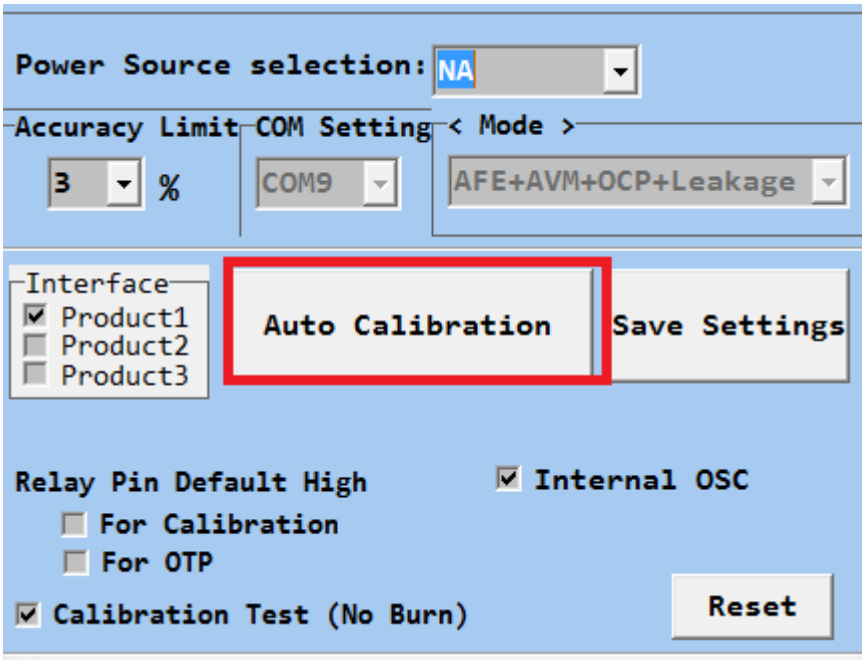


Figure 5-17 : AP calibration start button

Step 7: after Calibration done, AP will show PASS as follow picture.

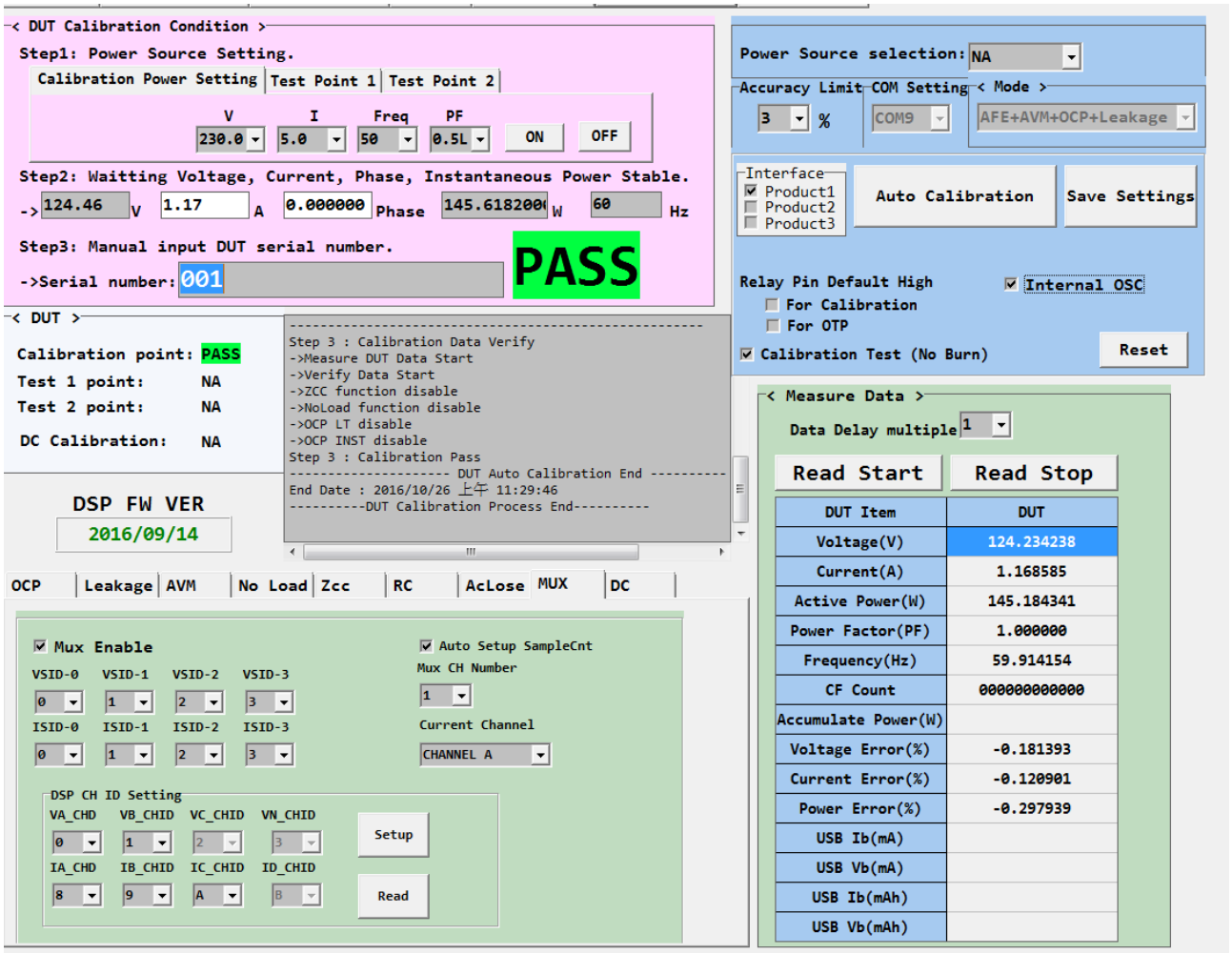


Figure 5-18: AP Calibration result

5.6 Report and Message Generator

Once the calibration is completed by the calibration AP, Report Data will be generated in the following path:

Report Path:

--> "C:\Explorer\Calibration\Report\xxxx.txt"

Report Data Example:

```
Start Date : 2015/5/13 下午 12:09:50
DUT serial number : 0001
Accuracy Limit : 3 (%)
-------------------------------------------------------------
                              V (%)        I (%)        W (%)     CH:0
120.0V,5.0A,60Hz,0.5L         0.008330(%),  0.010000(%),  0.006670(%),
120.0V,5.0A,60Hz,0.5L         0.008330(%),  0.119880(%),  0.025020(%),
120.0V,5.0A,60Hz,0.5L         0.008330(%),  0.378510(%),  0.640600(%),
-------------------------------------------------------------
PL7x11 Calibration Result : PASS
End Date : 2015/5/13 下午 12:16:31
VAGain : 0x1577
CH:0 IAGain : 0x72FC
CH:0 PAGain : 0x268E
SampleCnt : 0x07A1
sIRMS_50ms : 0x2ECFFC3
OCP SMP : 0x0062
INST SMP : 0x0001
CH:0 LTPUTH : 0x044361F3
CH:0 STPUTH : 0x0BD79E50
CH:0 STTH : 0x001280E75D00
CH:0 LTTH : 0x0063EB47C2FF
CH:0 INSTTH : 0x007D4949
CH:0 KWH_TH : 0x06A4D254F9AC
CH:0 NoLoad_TH : 0x00000000F7DB
ZCC_ON : 0x000A
ZCC_OFF : 0x000A
End Date : 2015/5/13 下午 12:16:35
```

Message Data Path

--> "C:\ Explorer \Calibration\Message\xxxx.txt"

Message data Example:

```
Start Date : 2015/5/13 下午 12:09:50
DUT serial number : 0001
Accuracy Limit : 3 (%)
----------DUT Auto Calibration Start----------
Auto Step 1 : Initial Process
        --> USB Connect Start
Test Point CalibrationPoint Start: 120V, 5A, 60Hz, 0.5L
        --> Load CFG Code Start
        --> DUT Interface Test Start
        --> Write CFG Register Start
        --> Write RO Register Start
        --> Write DSP RAM Start
------------------------------------------------------------------
Auto Step 2 : Calibration Process
    --> Calibration PF,CH:0
    --> Calibration V,CH:0
    --> Calibration I,CH:0
    --> Calibration Active Power,CH:0
------------------------------------------------------------------
Auto Step 3 : Calibration Data Verify
        --> Measure DUT Data Start
        --> Verify Data Start
    -->ZCC function enable
    -->NoLoad function enable0
    -->OCP LT Setting : 6.0 A, 300.0 Sec,CH:0
    -->OCP-LT enable,CH:0
    -->OCP ST Setting : 10.0 A, 300.0 Sec,CH:0
    -->OCP-ST enable,CH:0
    -->OCP INST Setting : 15.0 A, 1 Sec,CH:0
    -->OCP-INST enable,CH:0
    --> Calibration Active Power,CH:0
------------------------------------------------------------------
PowerControl:TestPoint_1
Test Point 1 Start : 120V, 0.1A, 60Hz, 1.0
Test Point 1 Result: PASS
------------------------------------------------------------------
PowerControl:TestPoint_2
```

Test Point 2 Start : 120V, 15A, 60Hz, 1.0

Test Point 2 Result: PASS

--> Relay Turn On Setting0

--> Relay Turn On PASS0

--> Relay Turn On Setting1

--> Relay Turn On PASS1

--> Relay Turn On Setting2

--> Relay Turn On PASS2

Auto Step 3 : Calibration Pass

KP1001 Power OFF Start

---------- DUT Auto Calibration End ----------

End Date : 2015/5/13 下午 12:16:31

5.7 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

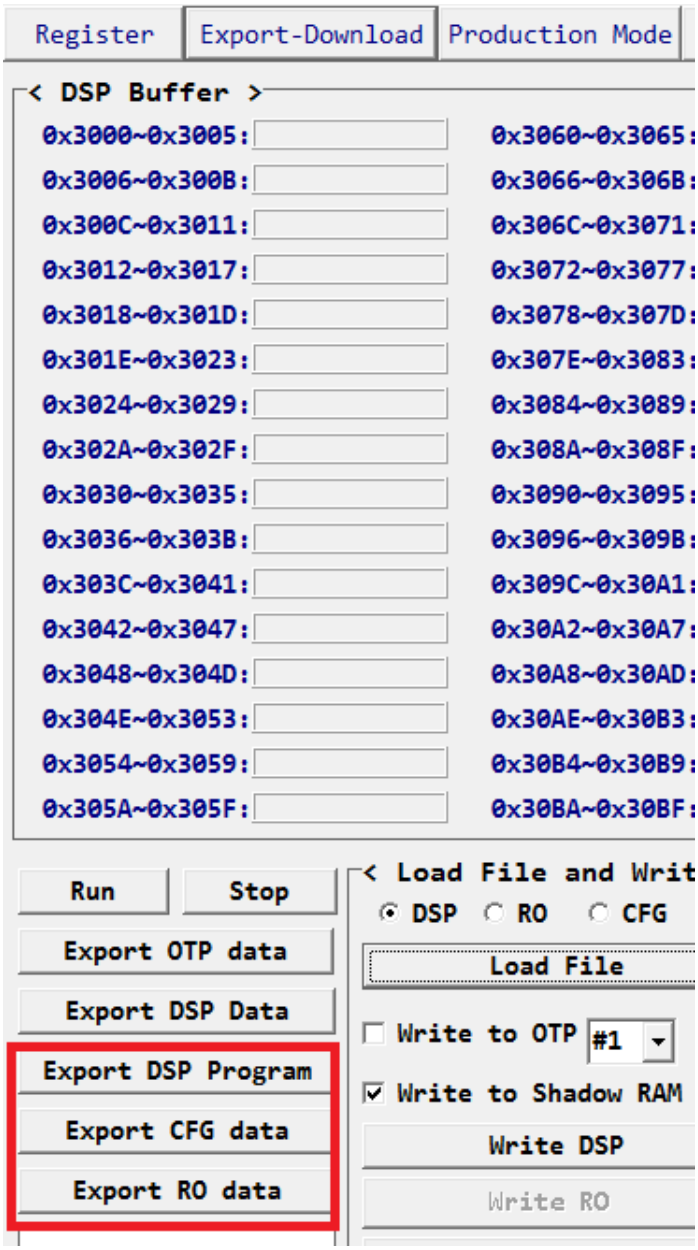


Figure 5-19: AP export code

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

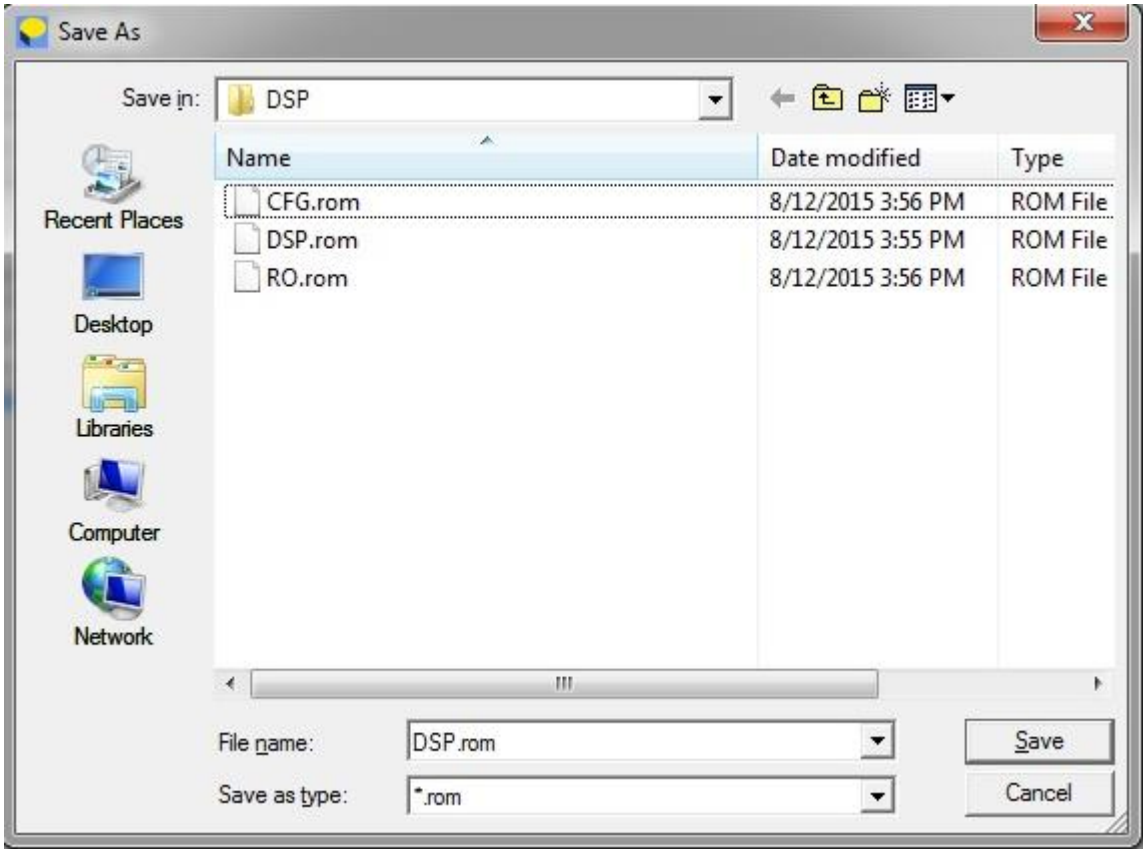


Figure 5-20: AP save code

6. AC Calculate Method

6.1 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 6-1 : DSP Buffer of 1V1I(AFE+AVM+OCP+Leakage)

6.2 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 6-2 : Calculate Vrms

6.3 Calculate Irms method

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

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 6-3 : Calculate Irms(A)

6.4 Calculate Active Power method

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

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 6-4 : Calculate Active Power(Wa)

6.5 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 6-5 : Calculate Power Factor(PF) Value and Phase Angle Value

6.6 Calculate Accumulate power Method

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

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 6-6 : Calculate Accumulate Energy (Wa)

6.7 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 : 0x3018~0x301D,

0x3018 address is Low Byte , 0x301D address is High Byte.

Register address	0x3018	0x3019	0x301A	0x301B	0x301C	0x301D
Register Data	Data[0] = 0x64	Data[1] = 0x00	Data[2] = 0x00	Data[3] = 0x00	Data[4] = 0x00	Data[5] = 0x00

Example : ZccCnt value = 100

ZccCnt value = (Data[1]*256) +Data[0]

= (0x00*256) + 0x64

=0x0064

=100

ZccStart register address : 0x301E~0x3023,

0x301E address is Low Byte , 0x3023 address is High Byte.

Register address	0x301E	0x301F	0x3020	0x3021	0x3022	0x3023
Register Data	Data[0] = 0x20	Data[1] = 0x00	Data[2] = 0x00	Data[3] = 0x00	Data[4] = 0x00	Data[5] = 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 : 0x3024~0x3029,

0x3024 address is Low Byte , 0x3029 address is High Byte.

Register address	0x3024	0x3025	0x3026	0x3027	0x3028	0x3029
Register Data	Data[0] = 0x3B	Data[1] = 0x0F	Data[2] = 0x00	Data[3] = 0x00	Data[4] = 0x00	Data[5] = 0x00

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 6-7 : Calculate Frequency (Hz)

6.8 OCP Parameter Calculate method

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

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 6-8 : Calculate OCP_SMPA Value

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

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 6-9 : Calculate INST_SMP Value

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

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 6-10 : Calculate LTPUTHA Value

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

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 6-11 : Calculate LTTHA Value

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

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 6-12 : Calculate INSTA_TH Value

6.9 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 6-13：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 6-14：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 6-15：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 6-16：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 6-17：Calculate PUL_TH Value

6.10 Leakage Parameter Calculate method

Follow is explanation how to calculate the Leakage pickup threshold and capture as below:

Calculate ILEAK_PUTH Value

ILEAK_PUTH= ILeak_50ms / (calibration current^2) * (IEAK current^2)

ILeak_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]

ILEAK_PUTH register address : 0x4132~0x4137,

0x4137 address is Low Byte , 0x413C address is High Byte.

Register address	0x4132	0x4133	0x4134	0x4135	0x4136	0x4137
Register Data	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]

Example :

IEAK=0.006(A)

Calibration current =0.01(A)

INST =3X

ILeak_50ms = 0x415d

ILEAK_PUTH = (0x415d)/(0.01^2)*(0.006^2)

ILEAK_PUTH = 0x1787

Table 6-18：Calculate ILEAK_PUTH Value

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

Calculate ILEAK_TH Value

ILEAK_TH= ILEAK_PUTH * IEAK_Trip_time

ILeak_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]

ILEAK_TH register address : 0x4138 ~0x413D,

0x4138 address is Low Byte , 0x413D address is High Byte.

Register address	0x4138	0x4139	0x413A	0x413B	0x413C	0x413D
Register Data	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]

Example :

IEAK_Trip_time = 2 millisecond

ILEAK_PUTH = 0x1787

ILEAK_TH = (0x1787) *2

ILEAK_TH = 0x2F0F

Table 6-19：Calculate ILEAK_TH Value

Follow is explanation how to calculate the Leakage 1smp threshold and capture as below:

Calculate ILeak_1smp_TH Value

ILeak_1smp_TH= (ILeak_50ms / OCP_SMPA)*2* (ILeak_1smp^2) / (calibration current^2)* (IEAK current^2)

ILeak_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]

ILeak_1smp_TH register address : 0x413E ~0x4143,

0x413E address is Low Byte , 0x4143 address is High Byte.

Register address	0x413E	0x413F	0x4140	0x4141	0x4142	0x4143
Register Data	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]

Example :

OCP_SMPA=79

ILeak_50ms = 0x415D

ILeak_1smp=10X

IEAK=0.006(A)

Calibration current =0.01(A)

ILeak_1smp_TH = (0x415D/ 79)*2*(10^2)/ (0.01^2)*(0.006^2)

ILeak_1smp_TH = 0x3B92

Table 6-20：Calculate ILeak_1smp_TH Value

7. Register Setting and Indicate

7.1 UART Auto Baud Rate

PL7211 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 7-1: 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.

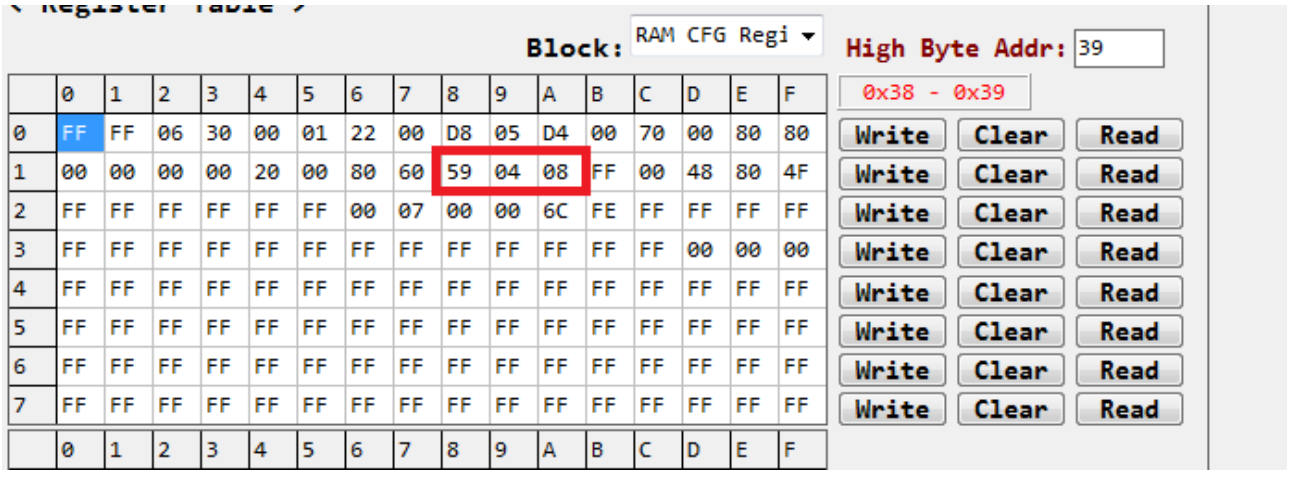


Figure 7-2: UART Baud Rate register setting

7.2 OCP and INST Protect Indicate

PL7211 have OCP and Instantaneous protect function, The function enable/disable table as below (Please reference PL7211_Leakage and OCP and AVM Demo Board User Manual.pdf):

Address	Bits	Description
0x3803~	15	
0x3804	14	OWP_EN
	13	HANDSHK_EN
	12	AC_Lose_EN
	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	
	00	AVM_EN

Figure 7-3: PL7211 1V1I(AFE+AVM+OCP+Leakage)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 7-4: PL7211 OCP and INST register

OCP long time accumulate values:

LTACC_IA	0x30D2~0X30D7

OCP Threshold (PL7211 1V1I 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.

7.2.1 OCP Long Time Protect Indicate

Please use PL7211 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,

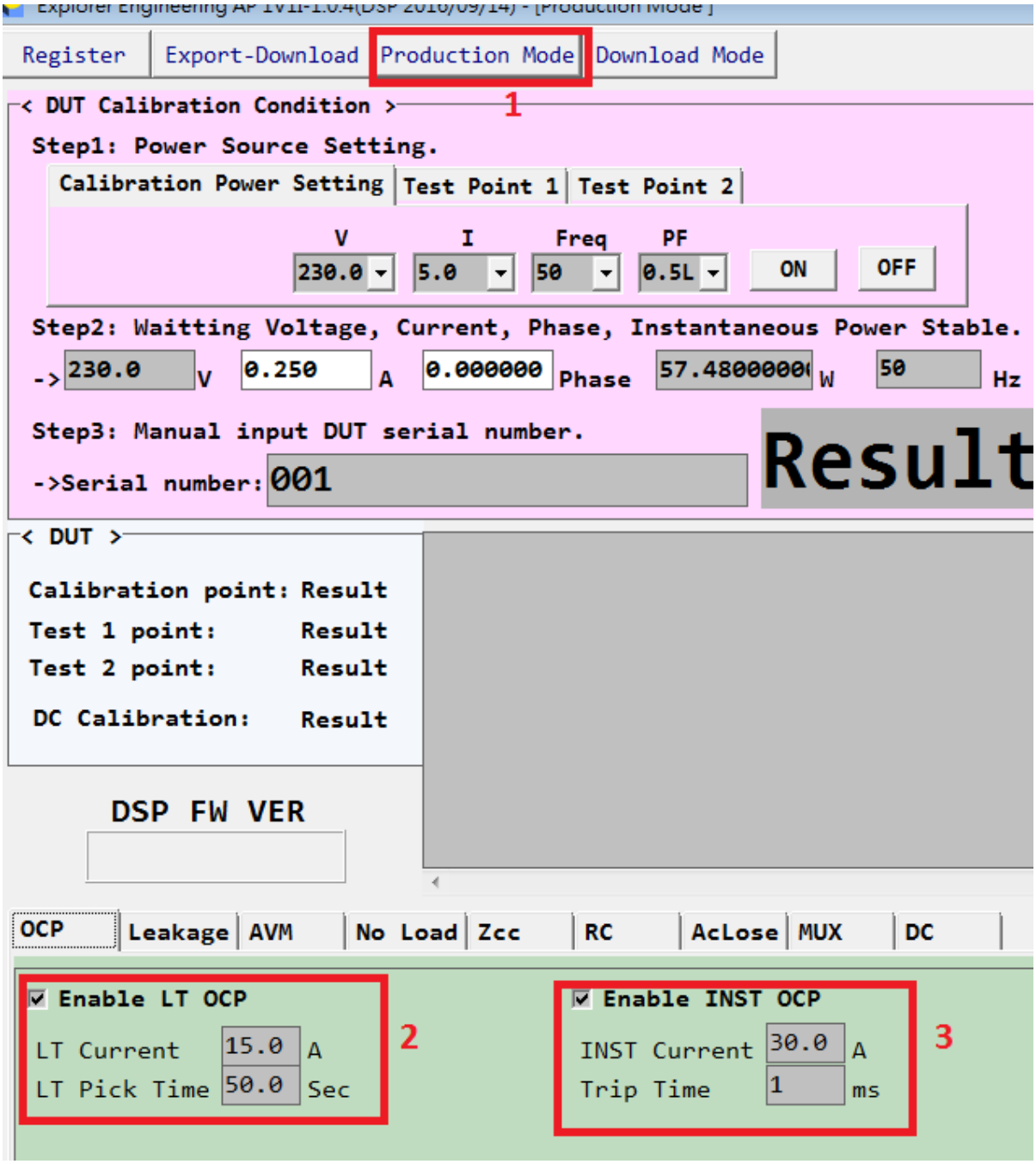


Figure 7-5: PL7211 OCP setting

- Please enter engineer mode

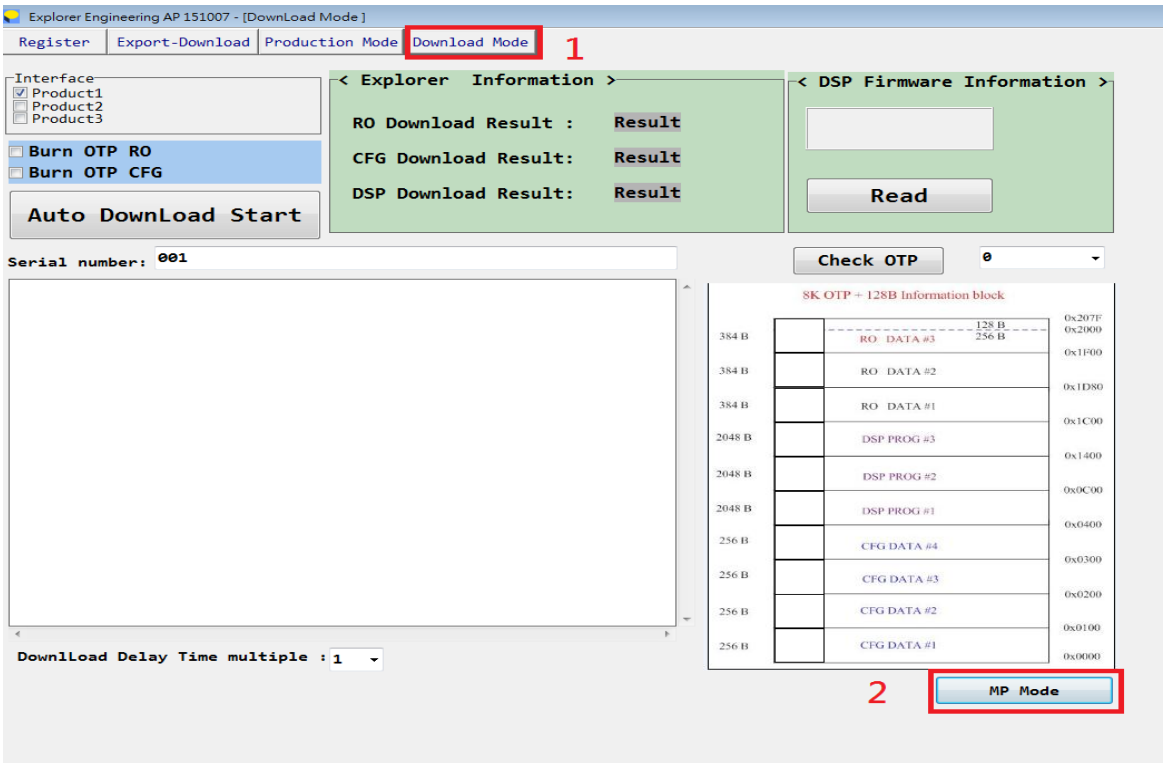


Figure 7-6: PL7211 MP mode

- Please select the debug form

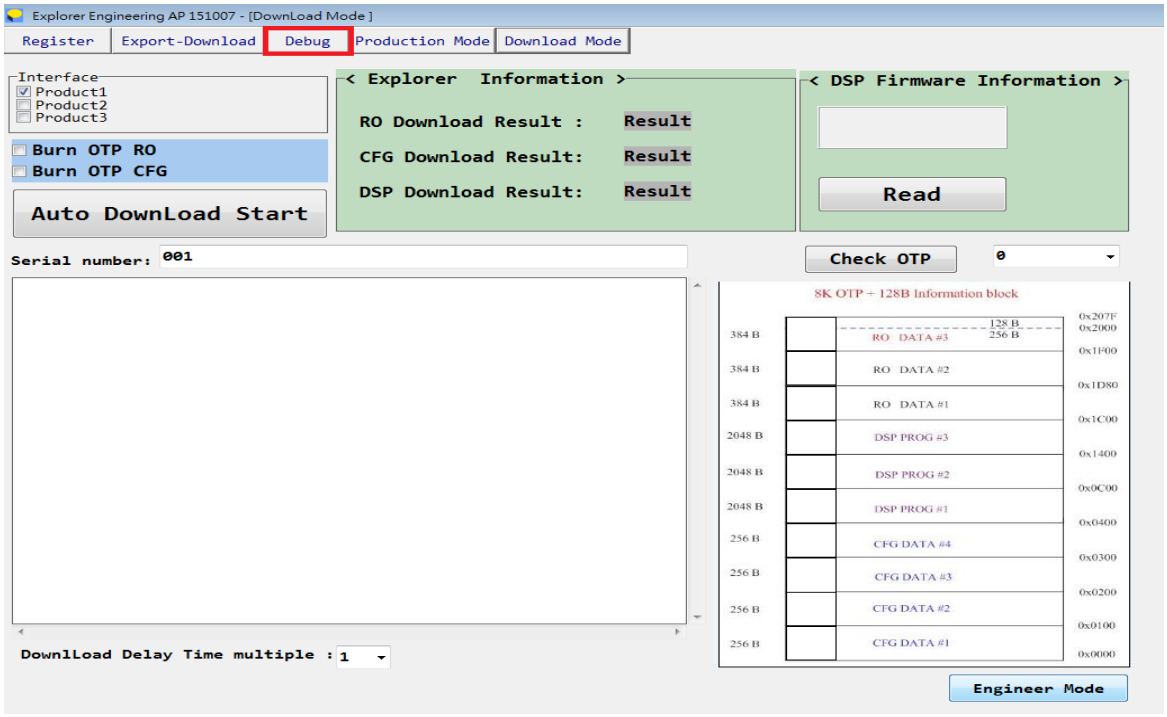


Figure 7-7: PL7211 Engineer Mode

- 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.

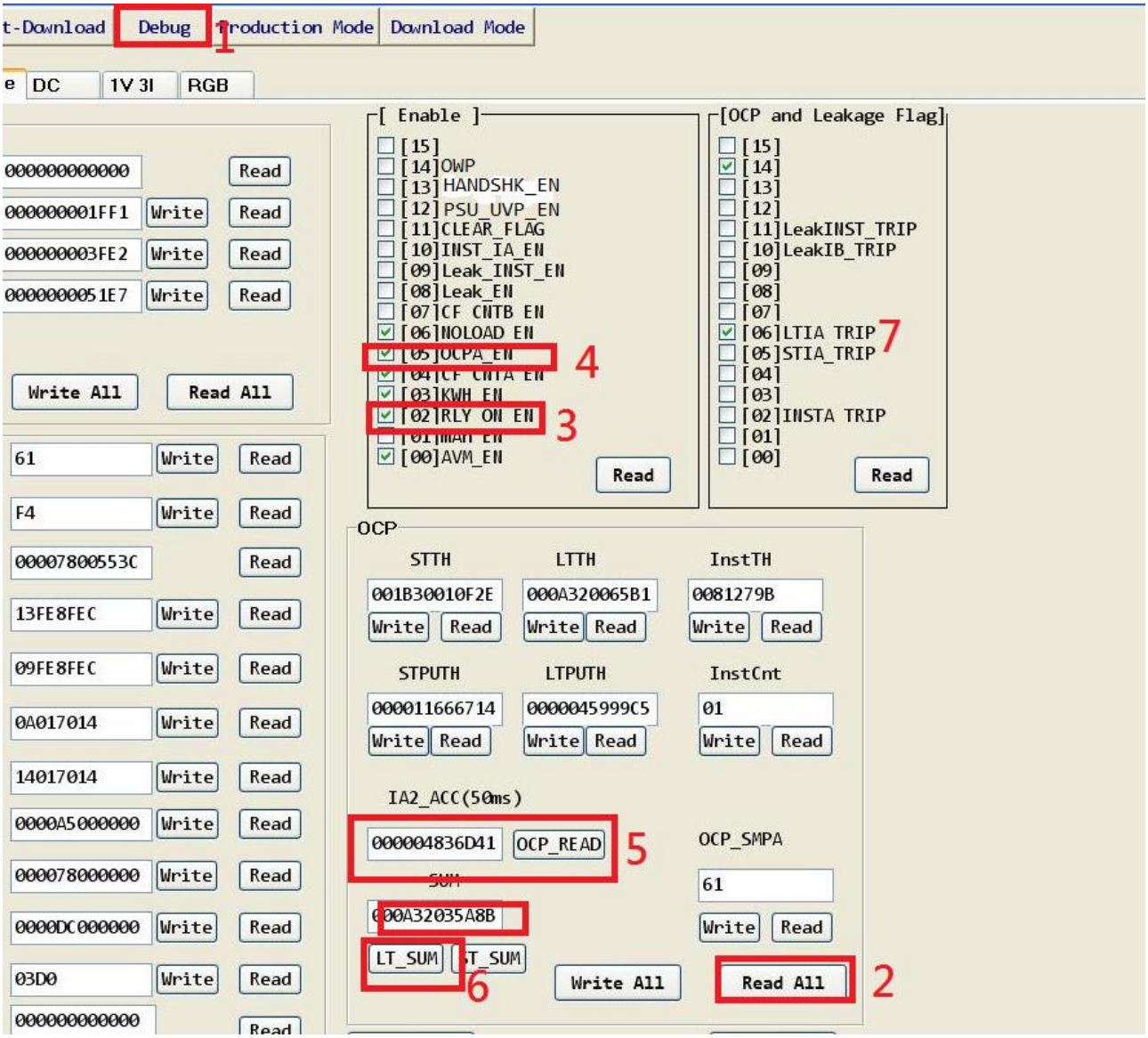


Figure 7-8: PL7211 OCP Long time Debug Flow

7.2.2 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 “INST_IA_EN” for long time and short time
- If IA current =15.1A, then check INSTA_TRIP indicate..

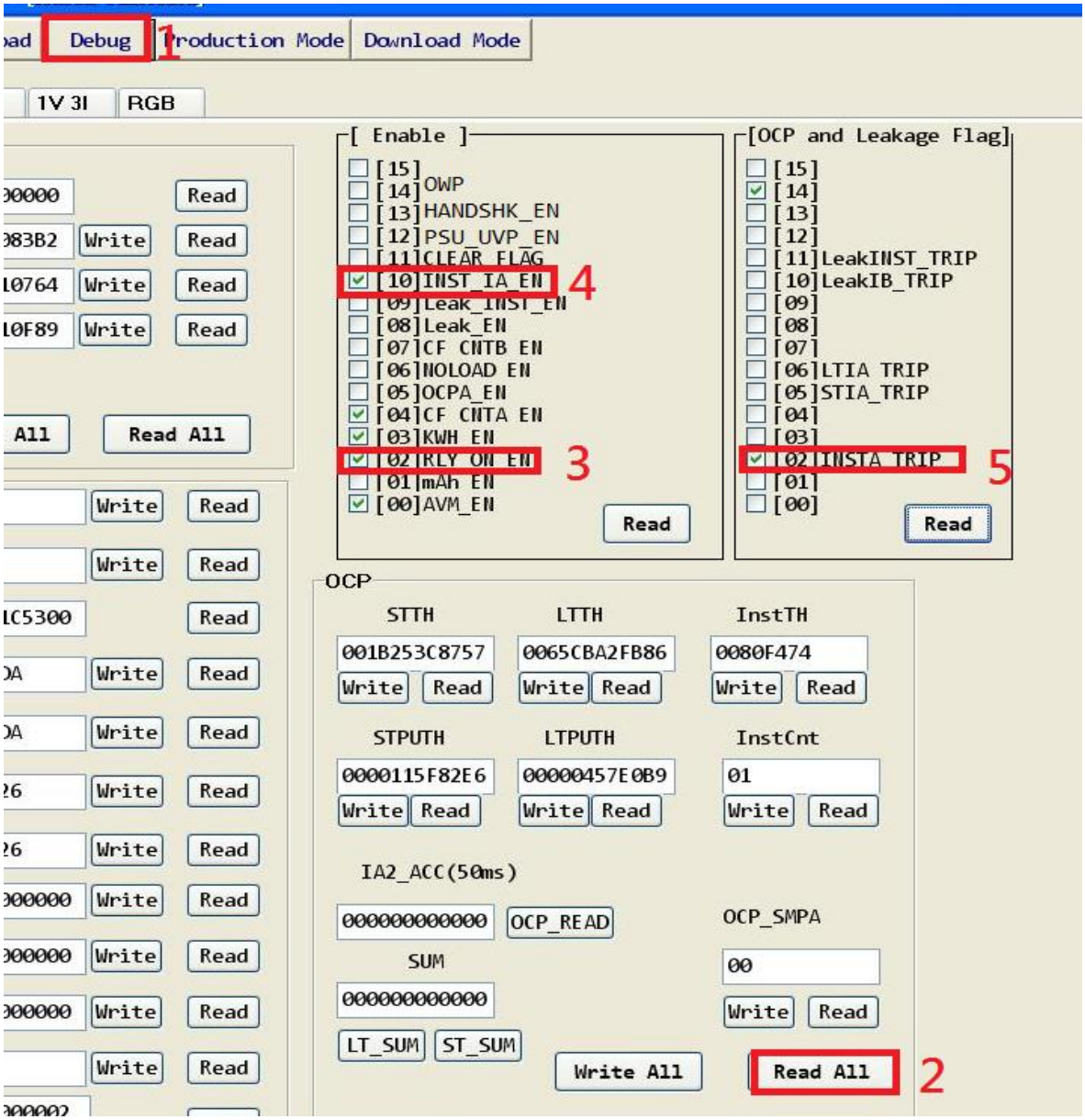


Figure 7-9: PL7211 OCP Instantaneous Debug Flow

7.3 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 7-10: PL7211 Leakage register

Leakage Threshold(Please reference PL7211_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

7.3.1 Leakage Protect Indicate

Please use PL7211 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,

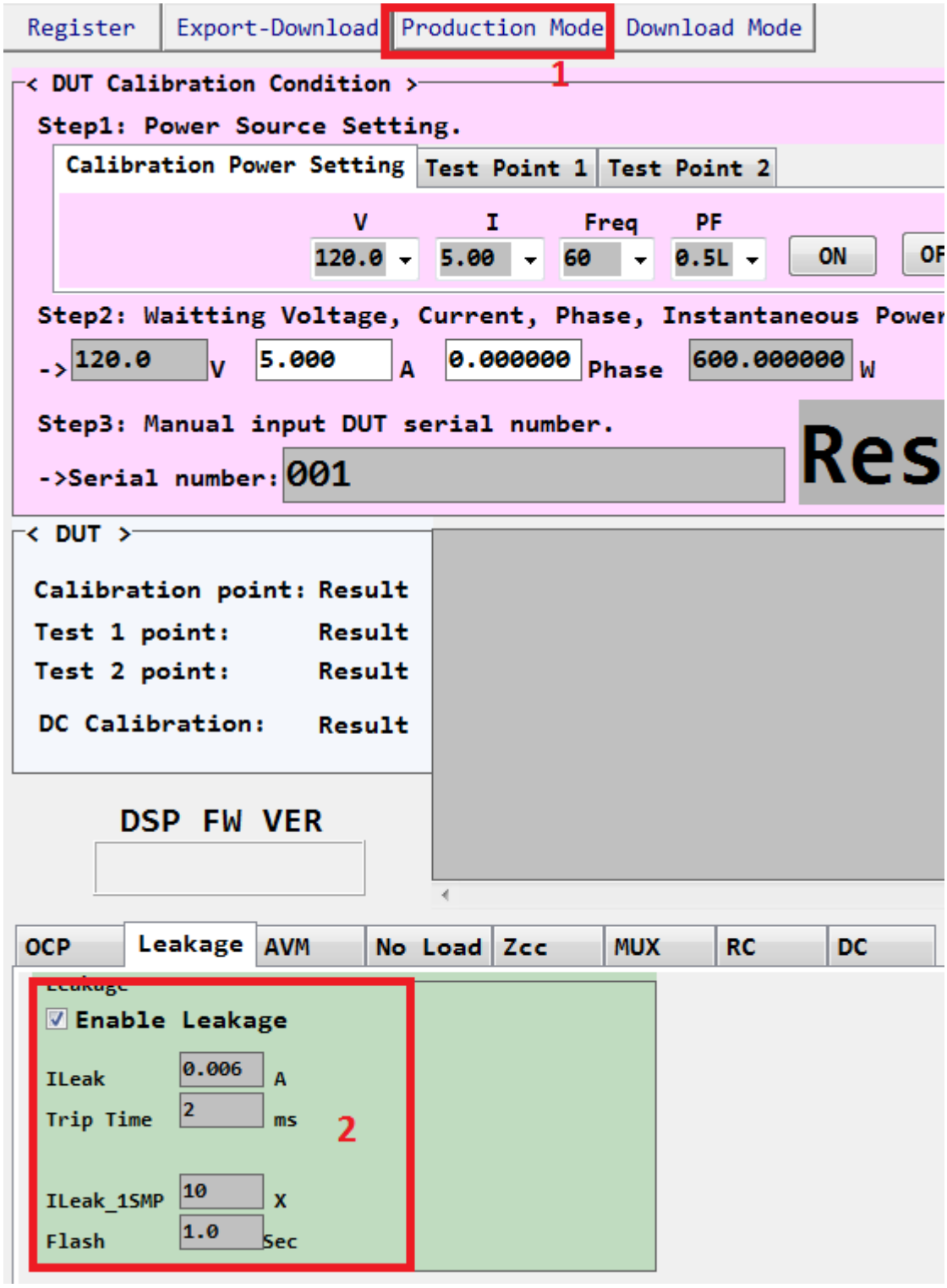


Figure 7-11: PL7211 Leakage setting

- 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.

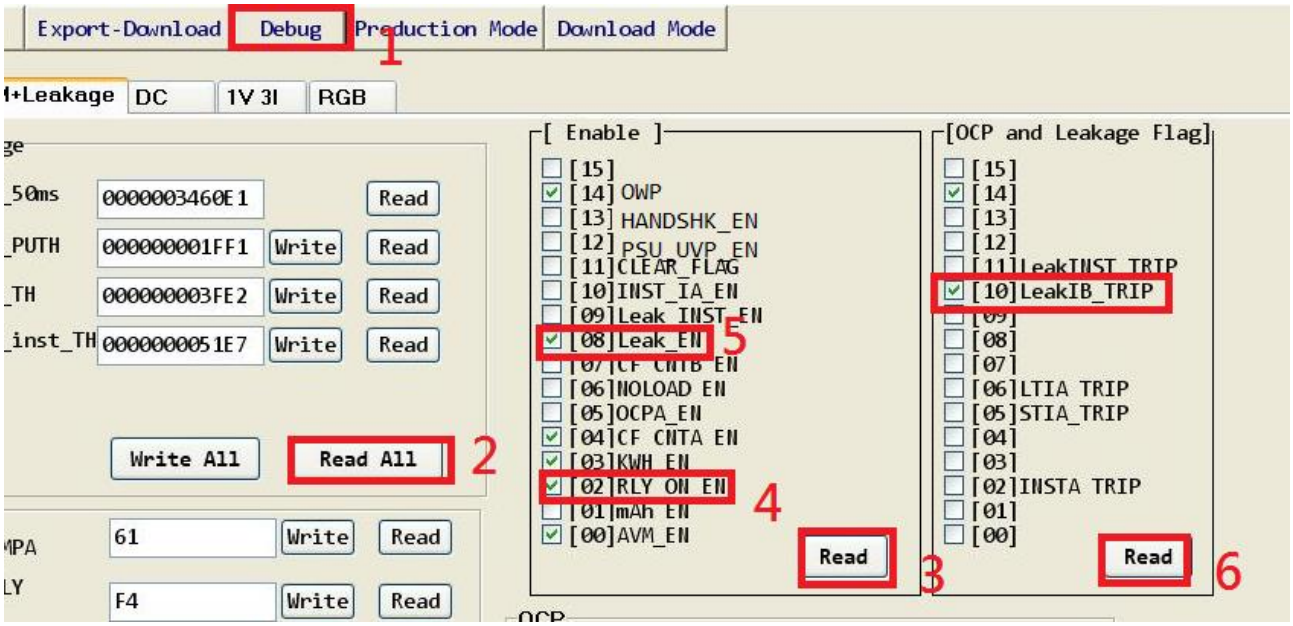


Figure 7-12: PL7211 Leakage Debug setting

7.3.2 Leakage INST Protect Indicate

Please use PL7211 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)

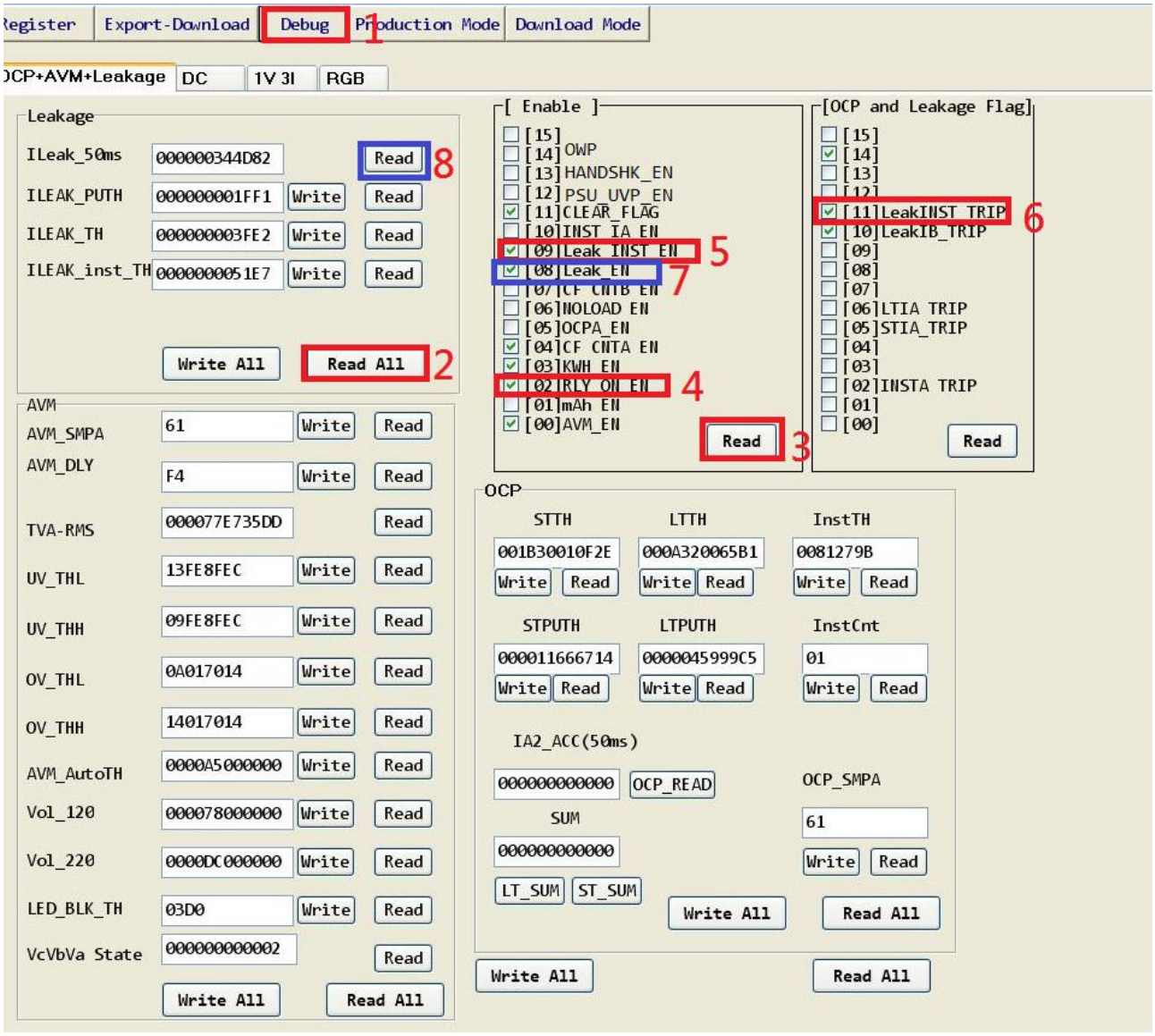


Figure 7-13: PL7211 Leakage Debug Flow

7.4 AVM Protect State

PL7211 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 PL7211 AVM function shown as follow figure.

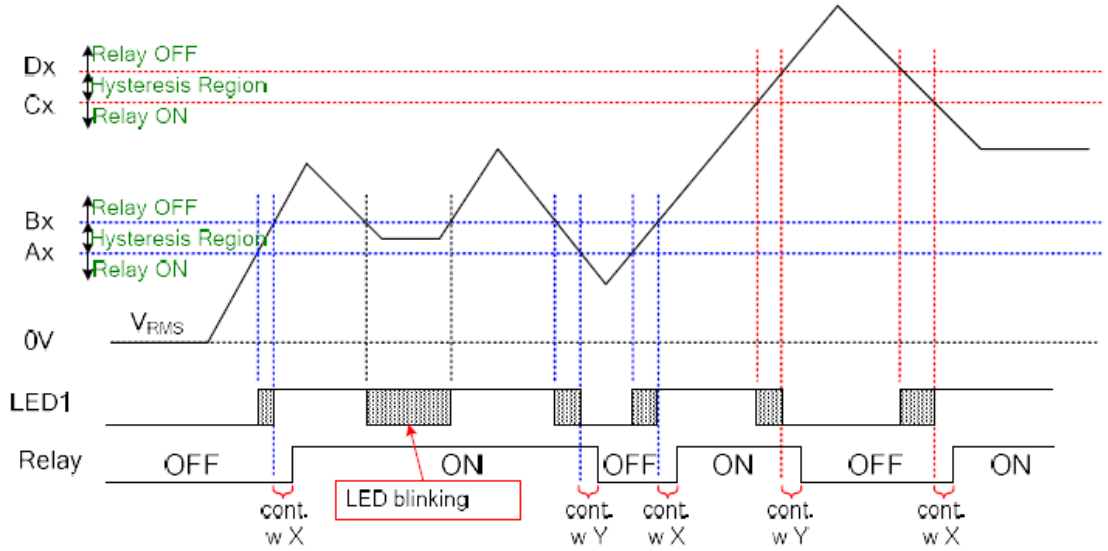
PL7211 will auto switch relay on/off when input voltage threshold are setting

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 7-1：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 7-2：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 7-3：AVM Case2 State

- You can check your setting,

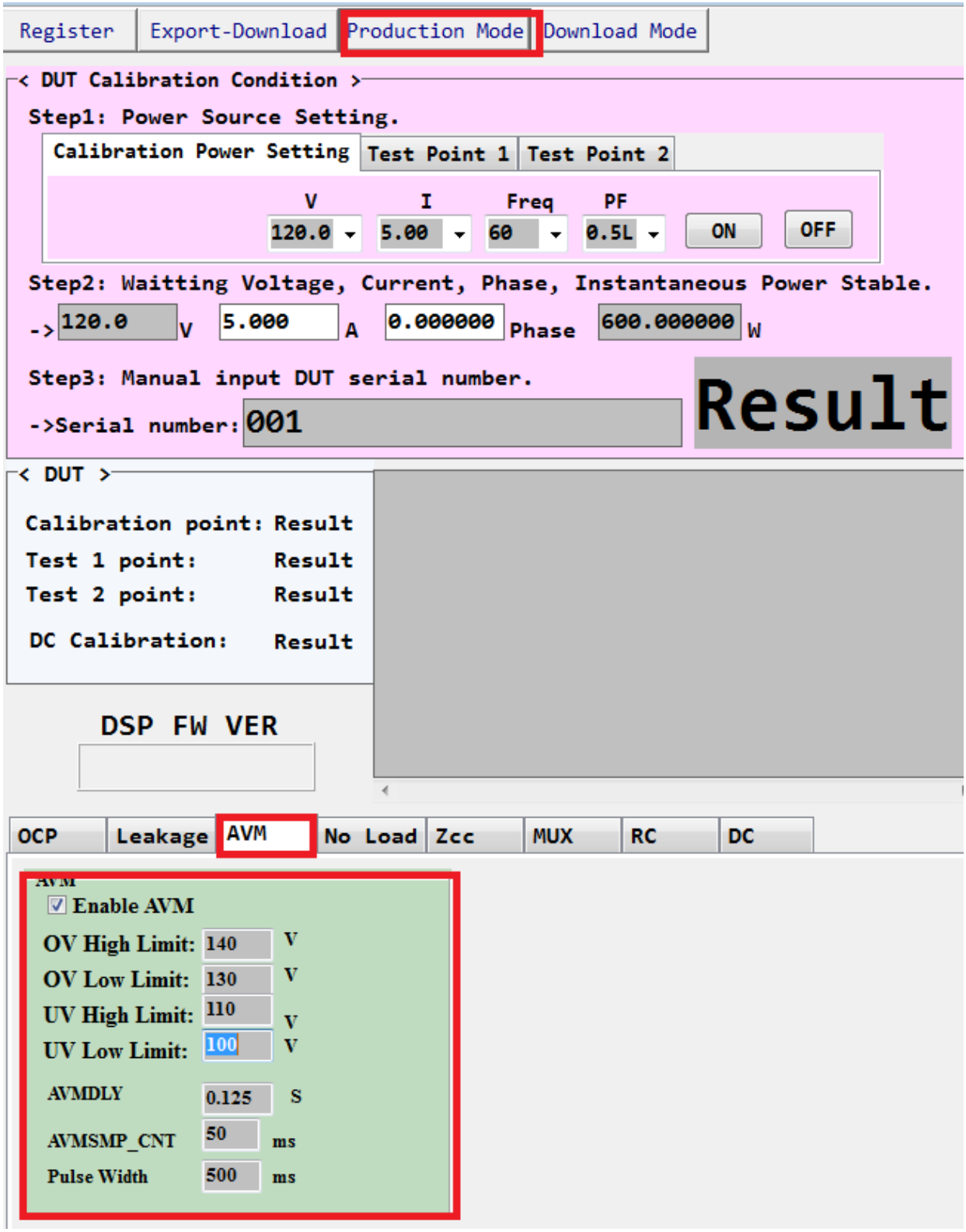


Figure 7-14: PL7211 AVM setting

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.

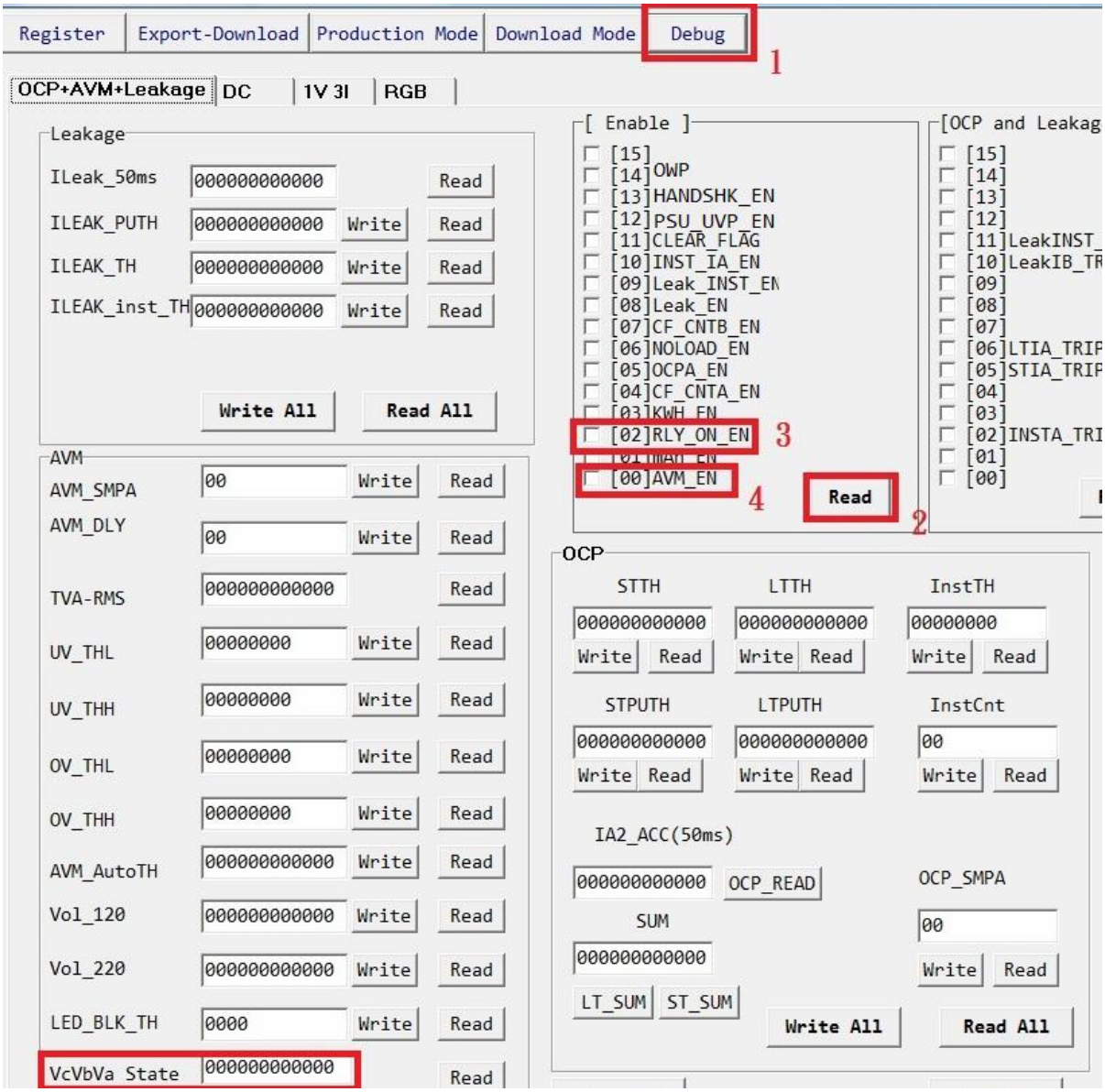


Figure 7-15: PL7211 AVM Enable

- PL7211 Demo Board -GPIO12 for AVM LED

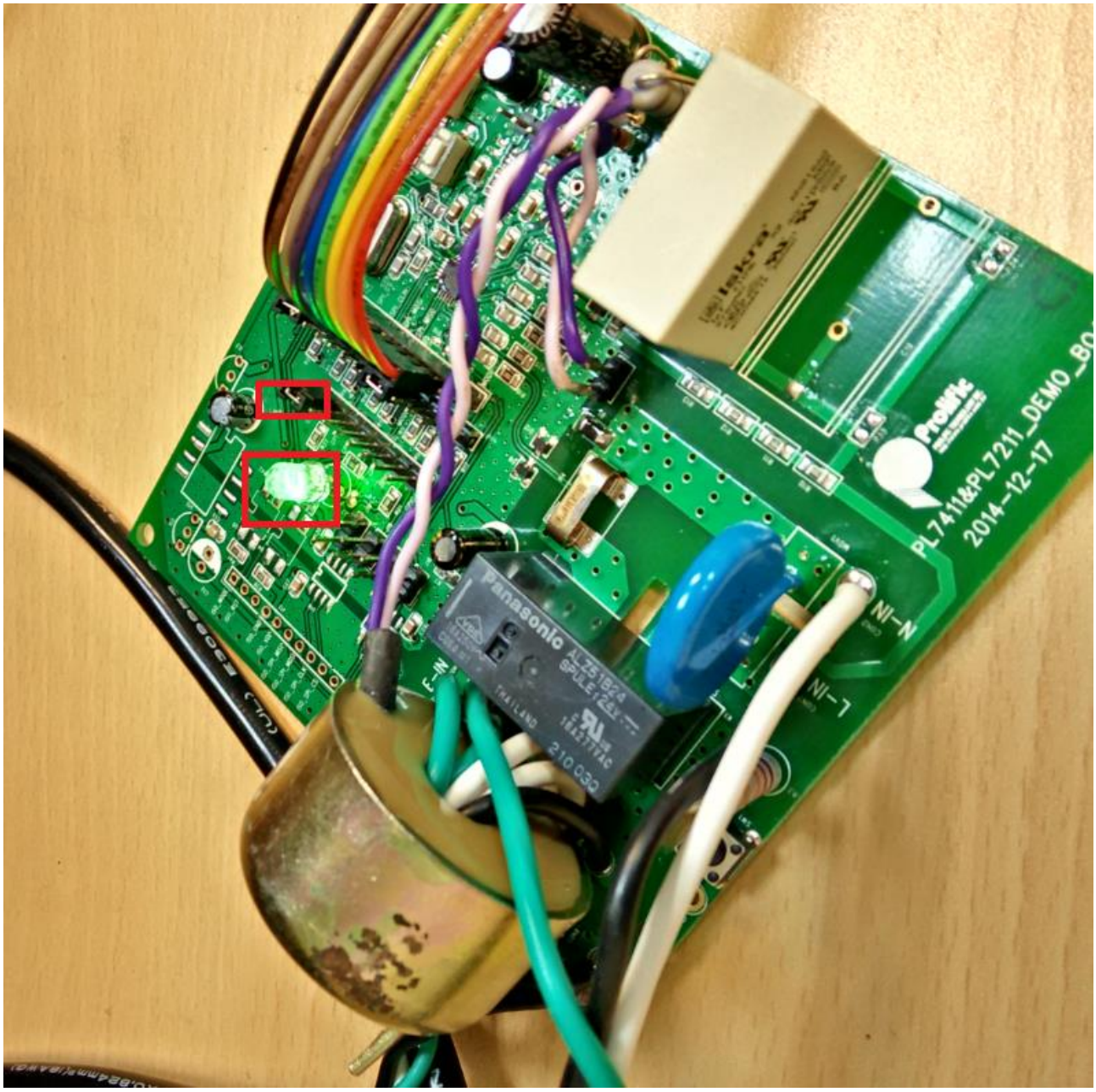


Figure 7-16: PL7211 AVM indicate

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