IS31FL3800
EVB
User Manual

Revision: A

Publication: Aug, 2020

Table of Contents

1 HARDWARE ENVIRONMENT

1.1 Appearance of Evaluation Board (EVB)

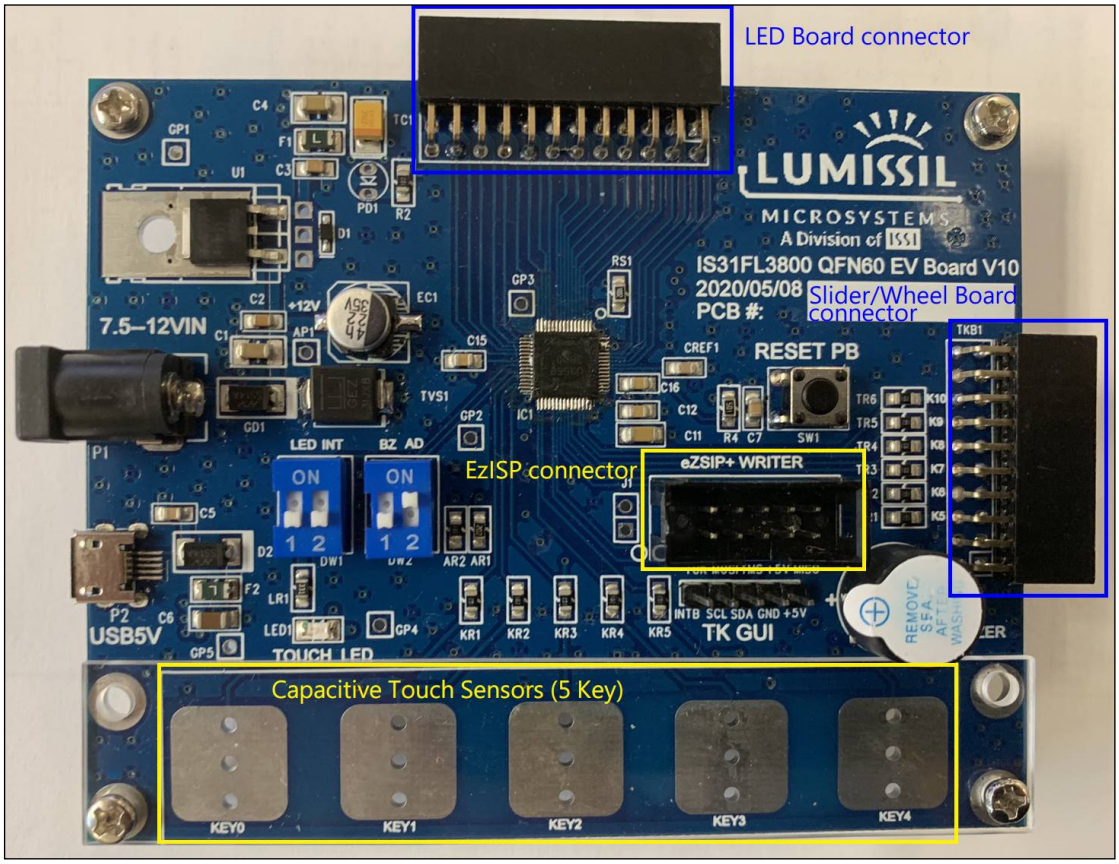

Figure 1: Photo of IS31FL3800 Evaluation Board

Figure 2: Photo of Slider Board and Wheel Board

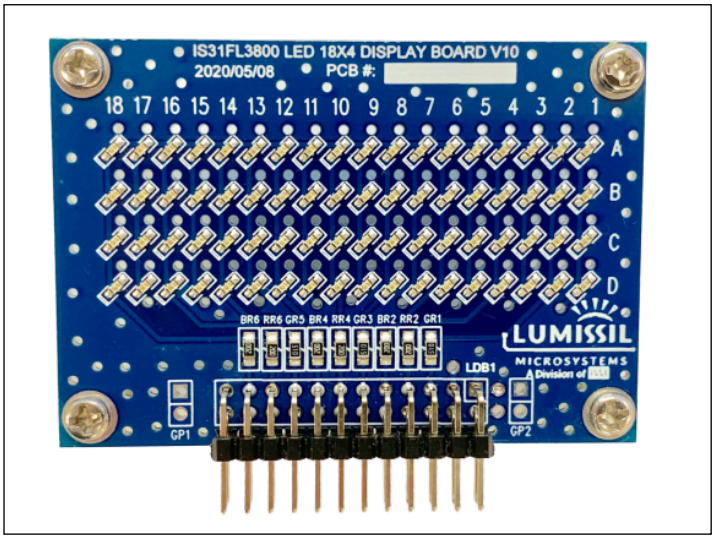

Figure 3: Photo of 18x4 LED Board

1.2 Connection Block Diagram

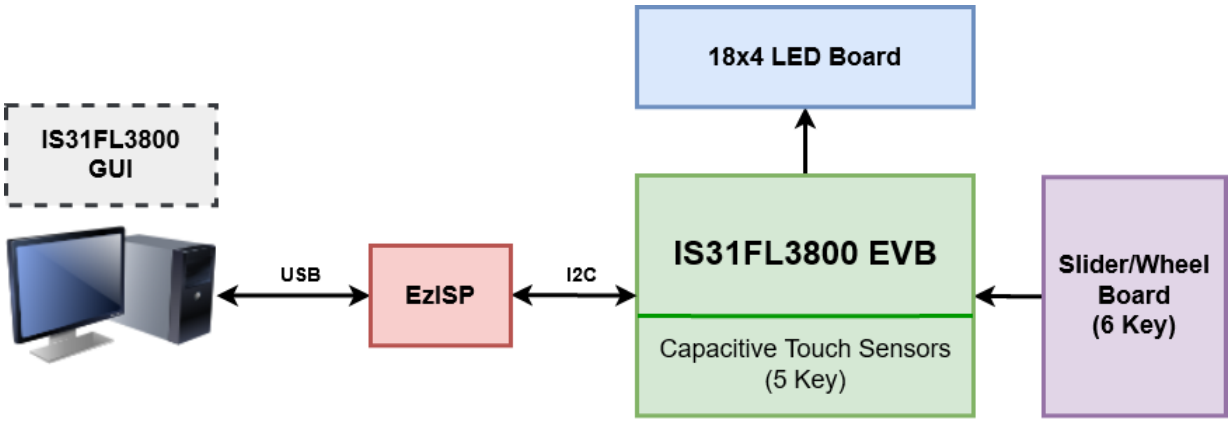

Figure 4: IS31FL3800 Evaluation Board connection block diagram

1.3 Schematic of Evaluation Board

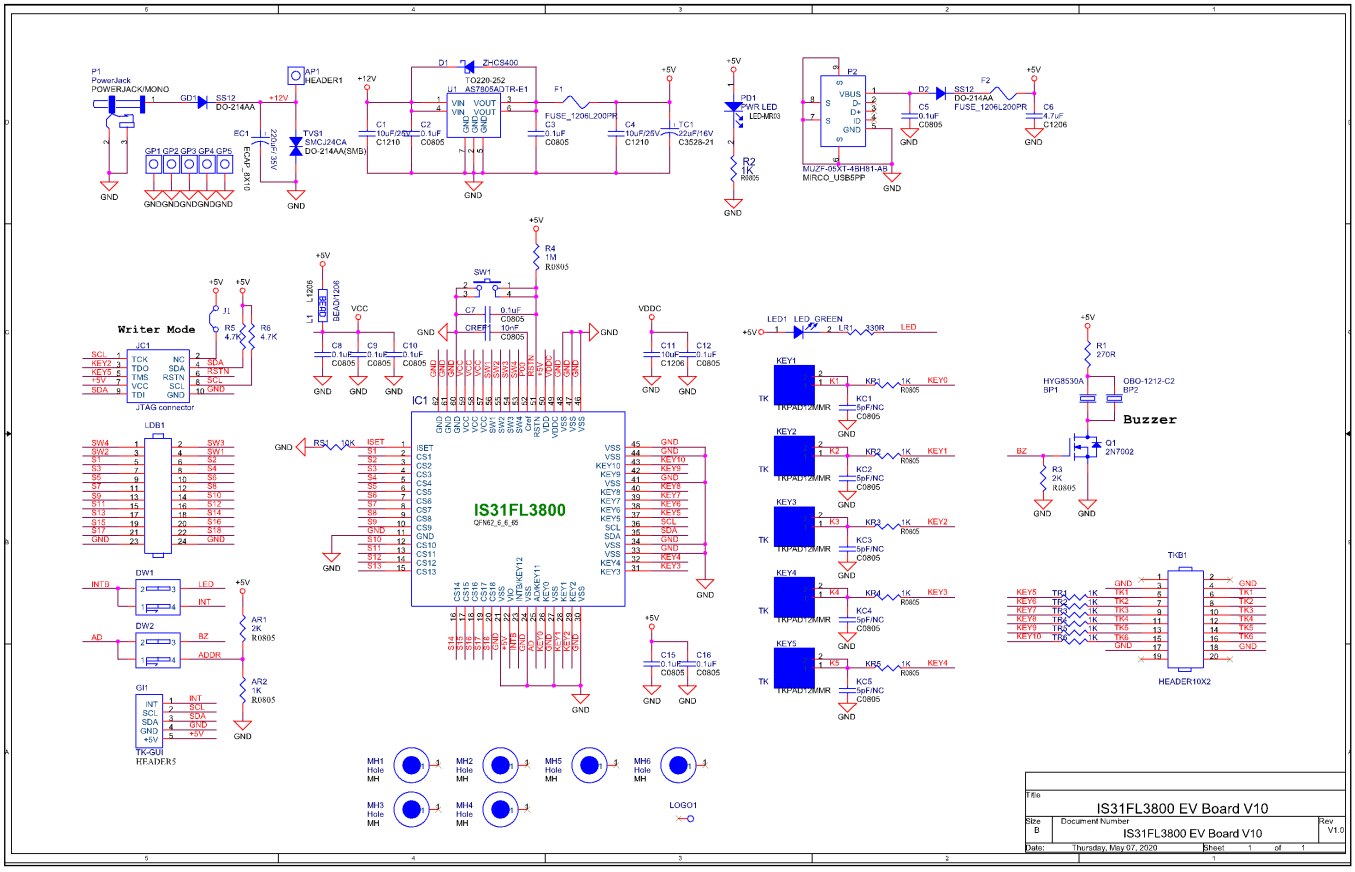

Figure 5: Schematic of IS31FL3800 Evaluation Board

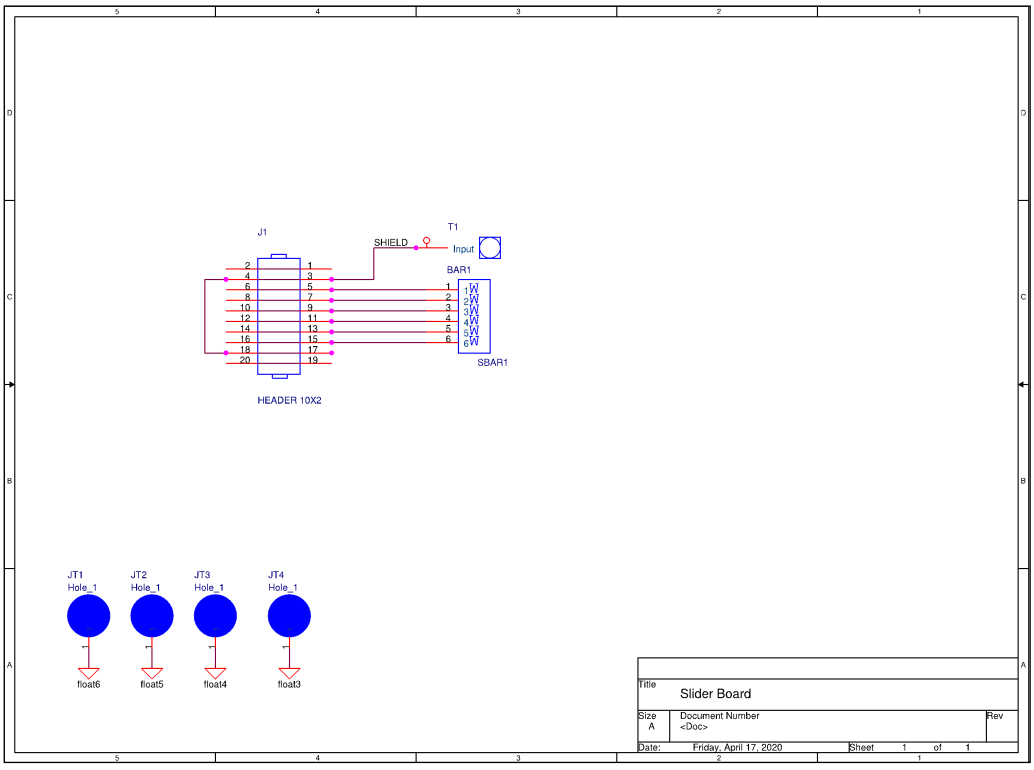

Figure 6: Schematic of Slider Board

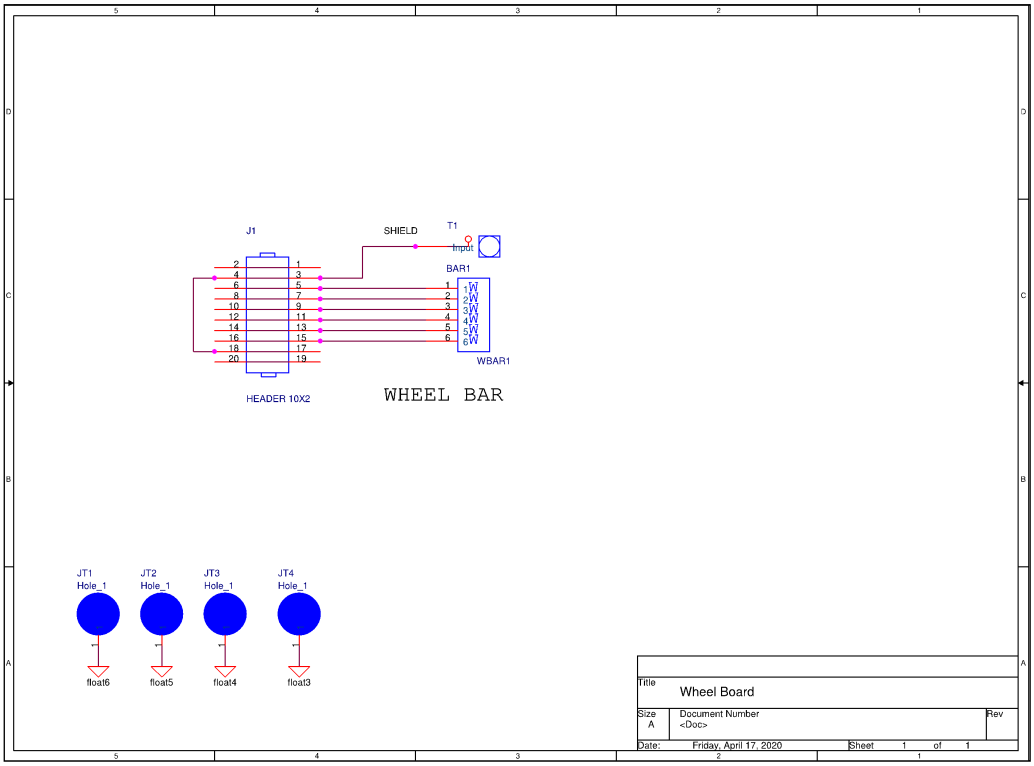

Figure 7: Schematic of Wheel Board

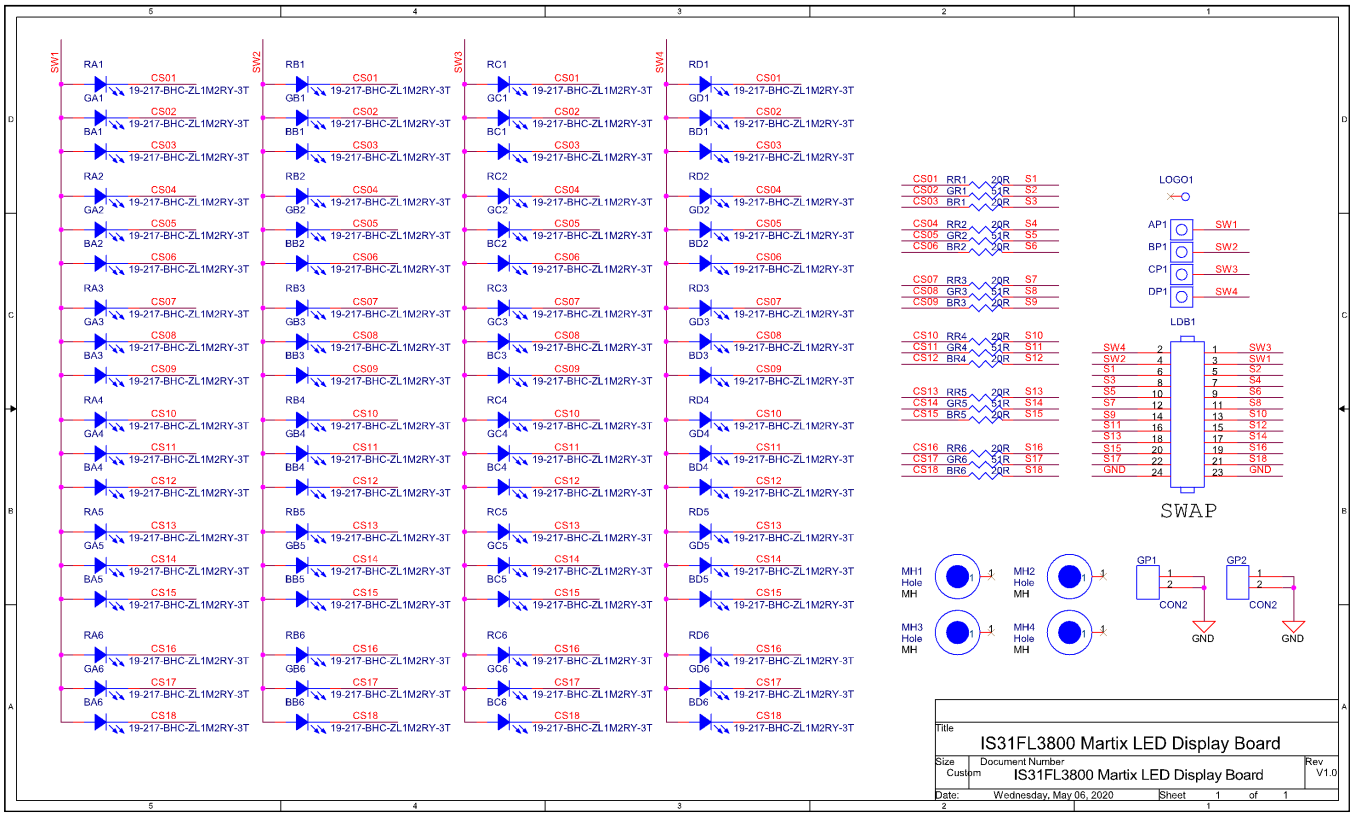

Figure 8: Schematic of 18x4 LED Board

ORDERING INFORMATION

Part No.	Temperature Range	Package
IS31FL3800-QFLS3-EBGUI	-40°C ~ +105°C	QFN-60, Lead-free

Table 1: Ordering Information

For pricing, delivery, and ordering information, please contact LUMISSIL's marketing and sales team at http://www.lumissil.com/company/office-locations or (408) 969-6600.

2 SOFTWARE SUPPORT

2.1 Software Requirements

Before using the GUI, the PC first needs to install the EzISP USB driver and related files (for example: Microsoft Framework and C++ library).

Figure 9: Photo of EzISP Board

Note: If there is no ".NET Framework" or the version lower than revision 4.0 on Windows system, ".NET Framework" should be downloaded as below link and install it.

https://www.microsoft.com/en-us/download/confirmation.aspx?id=24872

2.2 Run GUI Program

GUI operation process is as follows:

(1) First, confirm the setting of the AD pin on the IS31FL3800 Evaluation Board. If the switch of the AD pin is set to "ON" (Voltage division through two resistors AR1 and AR2, the default value of EVB is AR1 = 2K, AR2 = 1K), the setting of 1/3VDD is used. The following table shows the detailed AD pin settings.

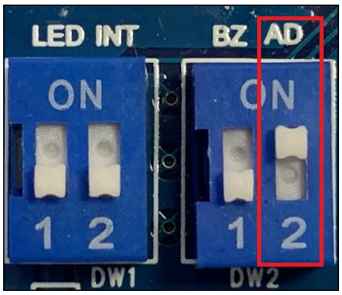

Figure 10: Photo of the AD pin switch on the IS31FL3800 Evaluation Board

I2C Slave Address* / AD Pin Select	Write Command	Read Command
AD = GND	0x78	0x79
AD = 1/3VDD	0x7A	0x7B
AD = 2/3VDD	0x7C	0x7D
AD = VDD	0x7E	0x7F

Table 2: AD pin selection and I2C slave address definition

* I2C Slave Address: 7-bit Address + 1 bit(R/W)

(2) Connect USB cable between the connector of the EzISP Board and the USB port of your PC.

(3) Use a 10-pin 2x5 Socket-Socket 1.27mm IDC cable from the connector on the EzISP Board to the connector on the IS31FL3800 Evaluation Board.

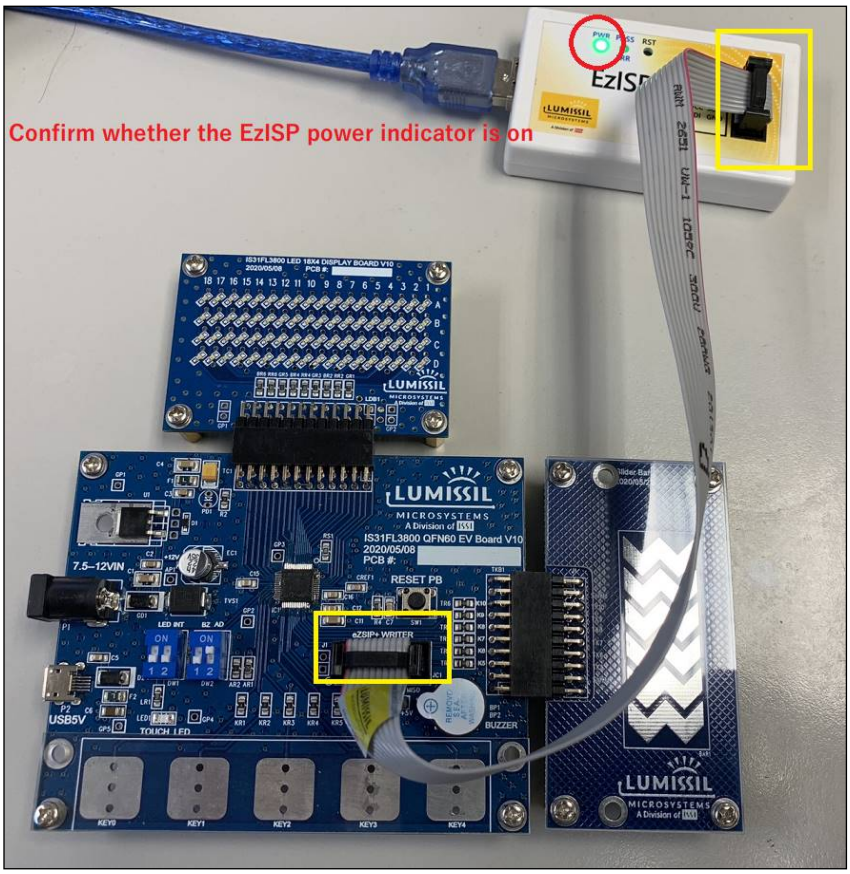

Figure 11: Connection of IS31FL3800 evaluation board and EzISP board

(4) Execute GUI program (file name: **TouchKeyGUI_3800_rls.exe**).

3 GUI INTERFACE

3.1 Connect Status

When connecting USB to IS31FL3800 EVB, first, you need to select the correct AD value. If the selected AD value is correct, the Connect Status will be displayed as "Connect" (with a green box), and the following settings can be selected. Otherwise, the Connect Status will be displayed as "Disconnect" (with a red box) and cannot select any of the following settings.

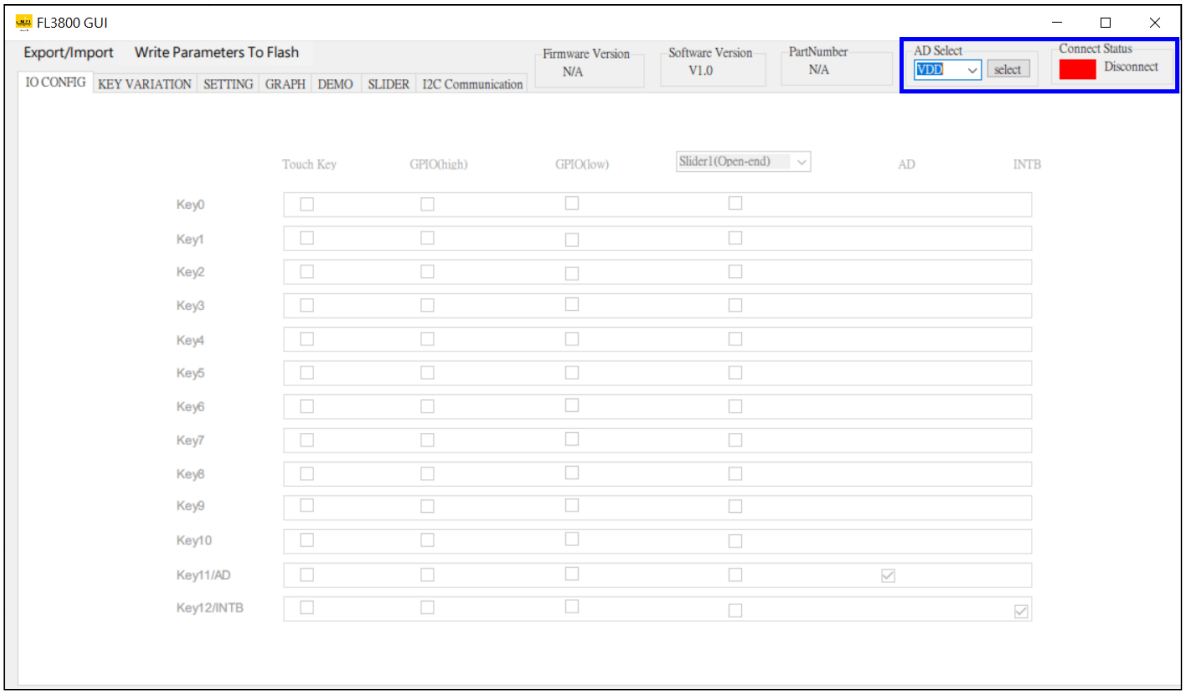

Figure 12: GUI connect status shows disconnect

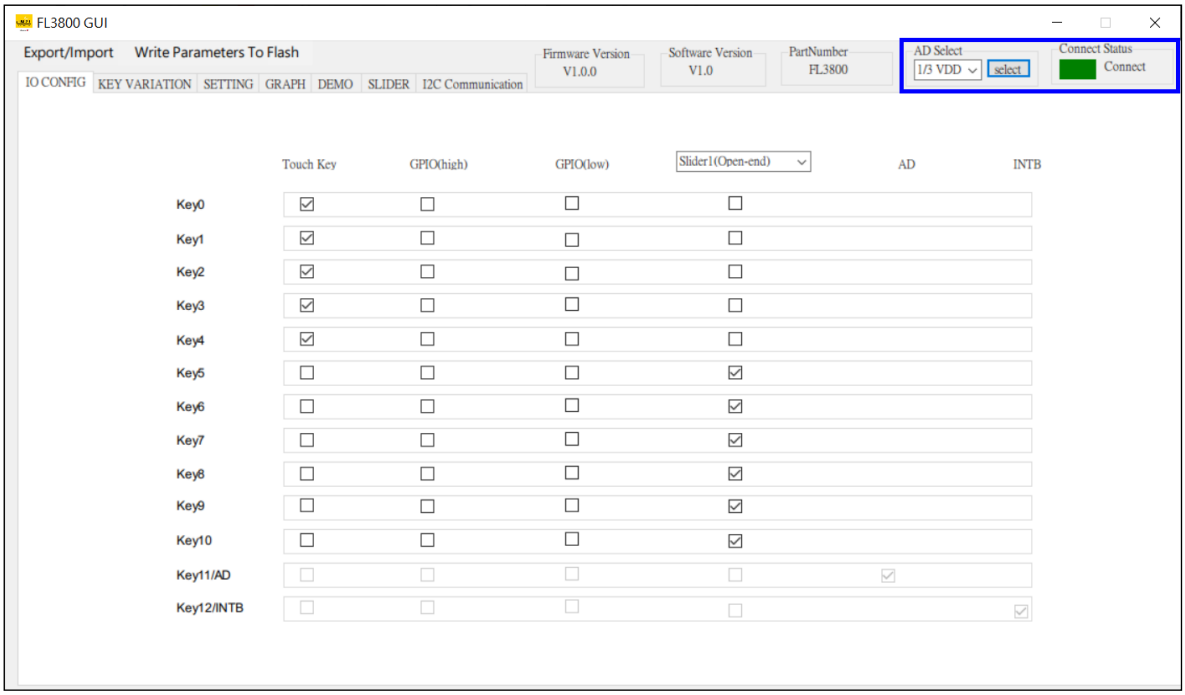

Figure 13: GUI connect status shows connect

3.2 Export/Import

As shown in Figure 14 below, "Export Register List" is used to export the register list of IS31FL3800. By exporting the register list, we can save the parameters set by the GUI to the computer. "Import Register List" is used to import the register list of IS31FL3800. By importing the register list, we can load the parameter file of the previous GUI settings into the computer.

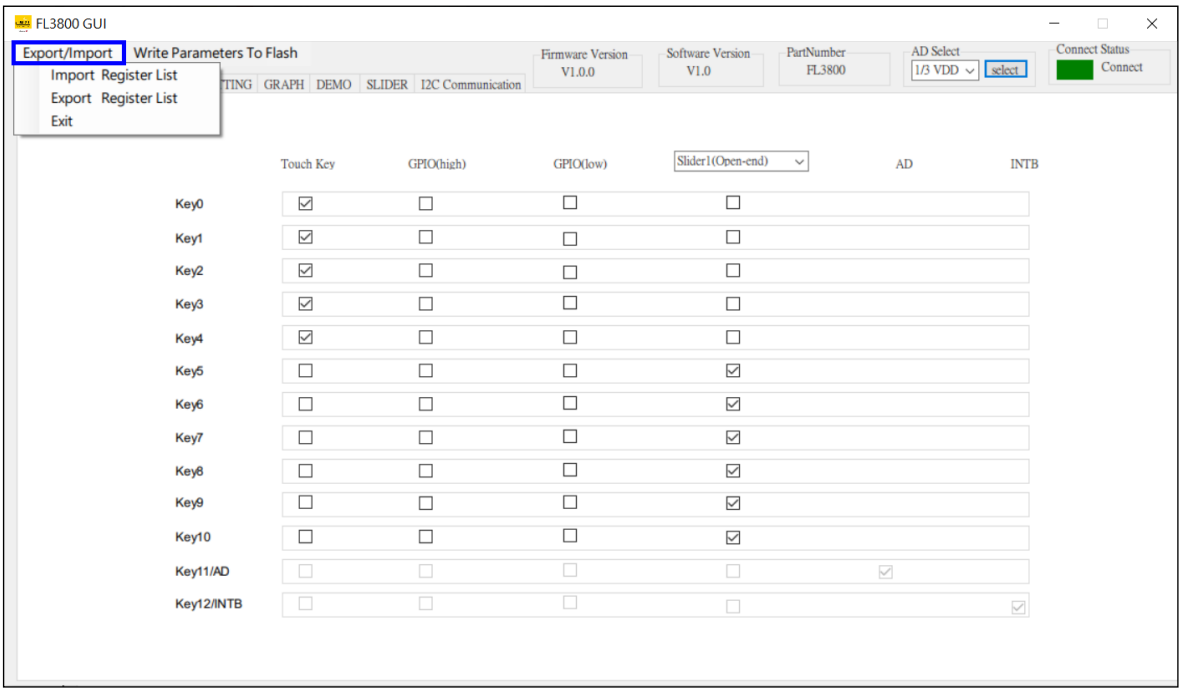

Figure 14: GUI export/import options

3.3 Write Parameters to Flash

As shown in Figure 15 below, the user can write the parameters set by the GUI into the flash memory of the IS31FL3800 chip. After clicking the "Write Parameters to Flash Memory" on the Menu Bar, the operation of writing parameters to Flash memory can be completed in about 4 seconds.

The embedded Flash Memory has the capability to hold saved data even if the power is turned off. When the chip is turned on again, the parameters previously written to the flash memory will become the default values.

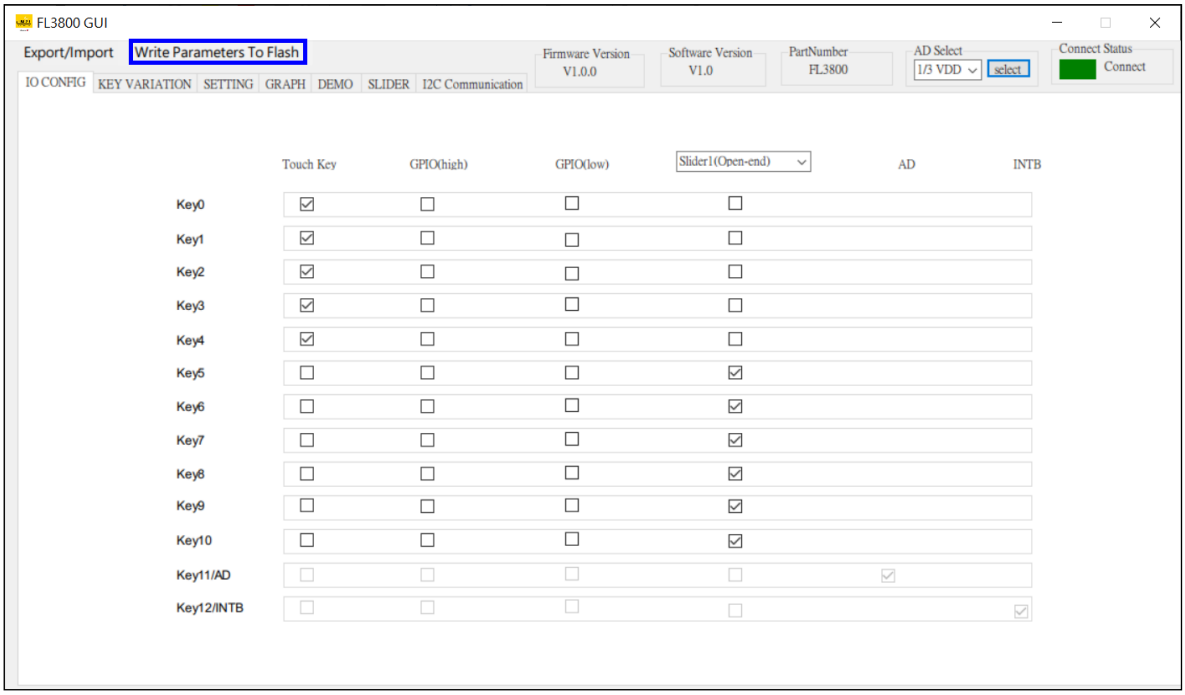

Figure 15: GUI write parameters to Flash options

4 IO CONFIG

4.1 IO Config Setting

When the correct AD value is selected and the Connect Status shows "Connect", it means the EVB has been successfully connected. As shown in the green box in Figure 16, the GUI will identify and display the correct firmware version, software version and chip part number.

As shown in the blue box in Figure 16, KEY0~KEY10 can be set as one of touch keys, GPIO (high), GPIO (low) or Slider1 functions. The functions that can be set for each IO pin are mutually exclusive. But KEY11 and KEY12 of IS31FL3800 cannot be set. Because KEY11 can only be used as AD pin, and KEY12 can only be used as INTB pin.

If the IO pin is selected as a touch key, it can be used as a button. IO pin can be set to GPIO (high) or GPIO (low). GPIO (high) here means that the IO pin is set to output high level, and GPIO (low) means that the IO pin is set to output low level.

Slider can choose Slider Bar (Open-end) or Slider Wheel (Close-end). Slider needs at least 6 IO pins. Users can define IO pins according to the required functions.

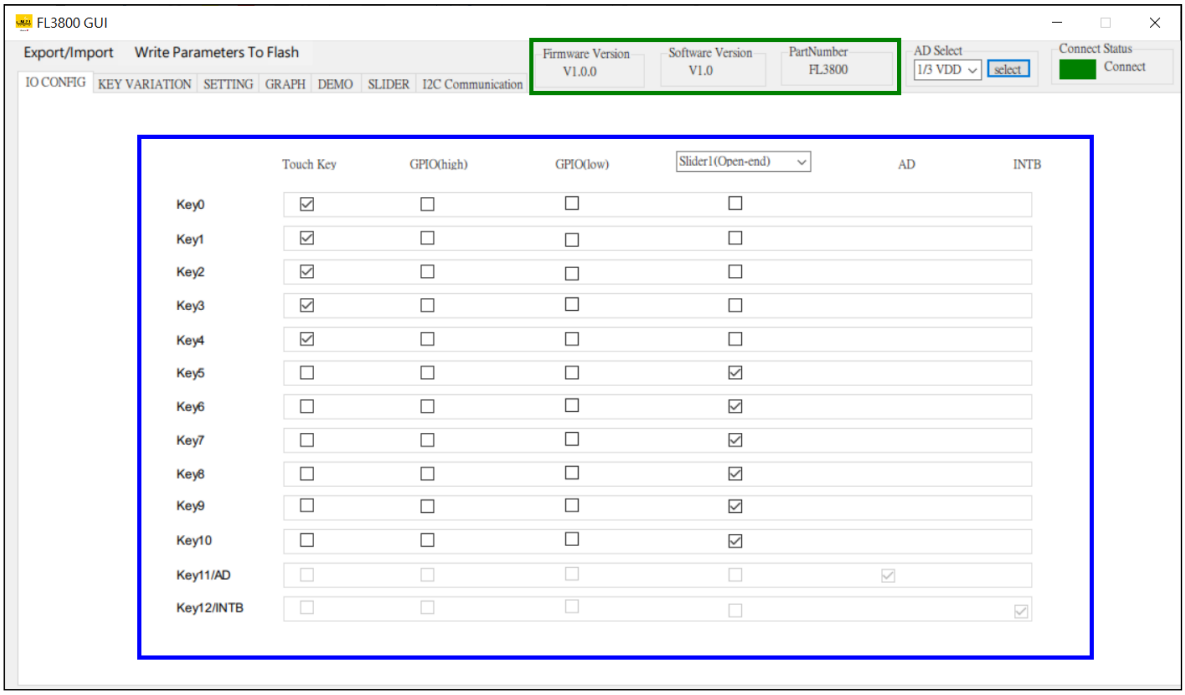

Figure 16: GUI preset IO configuration

5 KEY VARIATION

5.1 Operating Mode Switching

As shown by the red box in Figure 17, IS31FL3800 EVB is in normal mode, and the corresponding indicator light will turn green. After pressing the "Sleep" button, IS31FL3800 EVB will enter sleep mode, and the corresponding indicator light will turn red. At this time, IS31FL3800 can only be woken up by the key that has been previously set as a touch key and the wake-up function is enabled to return to Normal Mode.

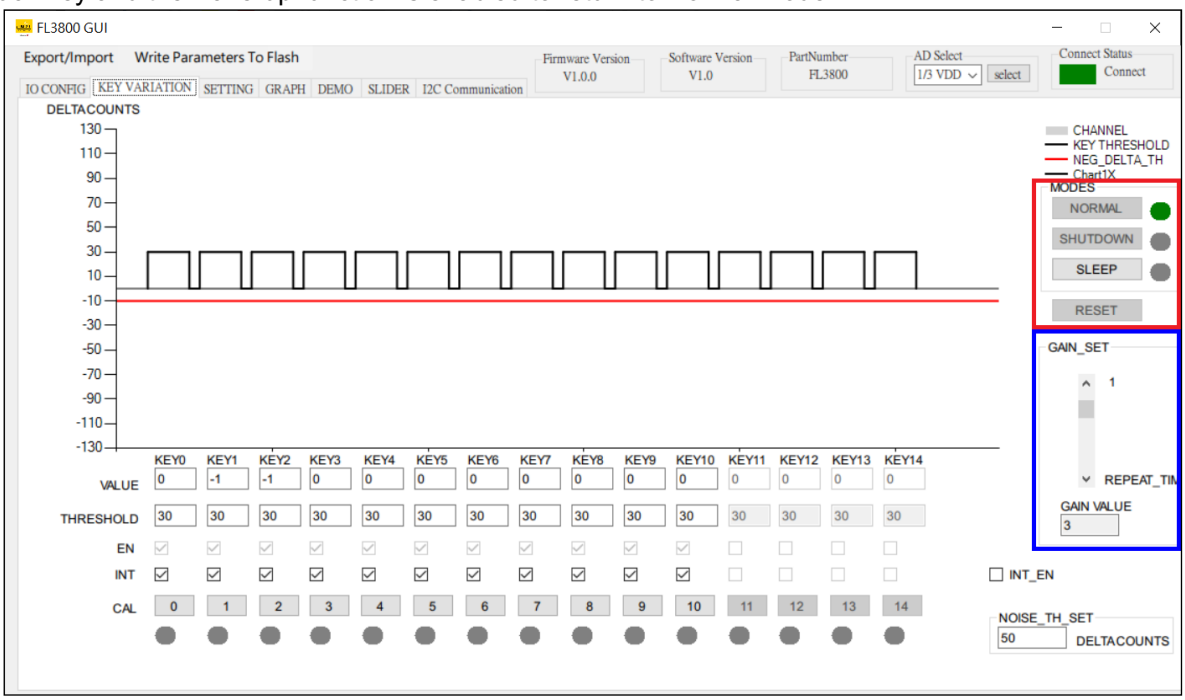

Figure 17: Key Variation page of GUI

Note: If the user sets a specific key to GPIO (high) or GPIO (low), when IS31FL3800 EVB enters sleep mode, this key will continue to maintain its previous state.

5.2 Gain Setting

The button GAIN_SET in the blue box in Figure 17 above is used to set gain for the touch keys. The gain could be set in 1~16 levels by pulling the scroll bar and the current gain value will be shown in bottom.

Setting gain will affect the sensitivity of all the touch keys. If gain value is set too large, high sensitivity of keys may cause a false trigger. Therefore, the touch key gain setting should be adjusted according to the actual touch button size and working environment.

5.3 Threshold of Key

As shown in Figure 18 below,

Figure 18: Threshold setting

THRESHOLD is used to set the threshold of keys (KEY0-KEY10). The maximum value is 127. GUI will keep 127 if input value is over 127. Input data in the corresponding box and press Enter or Tab key, or move the cursor to another location will set up the threshold.

Put mouse on corresponding box, for example in KEY0, will show a table in the red box as shown in Figure 19. The whole interface has prompted box for each programmable parameter.

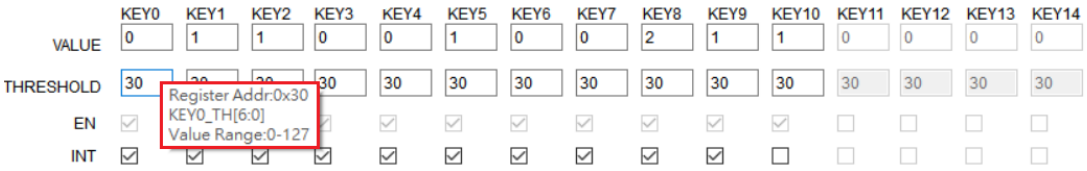

Figure 19: Set the threshold of KEY0

The address of KEY0 THRESHOLD Register is 0x30.

KEY0_TH[6:0] are the setting bits of KEY0 THRESHOLD Register (0x30).

Threshold range is from 0 to 127.

Key will be triggered when the environmental capacitance of touch key is over key threshold.

5.4 Value of KEY

As shown in Figure 20 below,

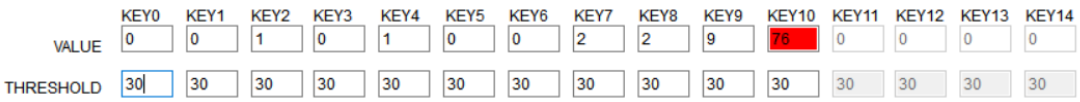

Figure 20: The VALUE of the KEY

VALUE is the current touch key value. It shows environmental capacitance if there is no object close to touch key. The corresponding value will be display in the box when pressing KEYs (KEY0~KEY10) on EVB. The box turns red if value is over key threshold and it means this key is triggered.

5.5 Key and interrupt enable

Key enabled setting, as shown in Figure 21 below,

Figure 21: KEY's EN

By default, all check boxes here are selected, which means that all keys are enabled. The user cannot modify here, but can only set it in the IO CONFIG page.

Figure 22 below is an example. If you want to disable KEY0, you cannot select the check box in the KEY0 column of the IO CONFIG page. The key enable setting in Figure 21 above will also be changed accordingly.

Key enable setting could shut down any touch key. If disable the KEY0 and KEY1~KEY10 is enabled, there is no changing by touching KEY0.

Note: If KEY is set to GPIO (High) or GPIO (low) on the IO CONFIG page, the key enable setting corresponding to KEY will also be disabled.

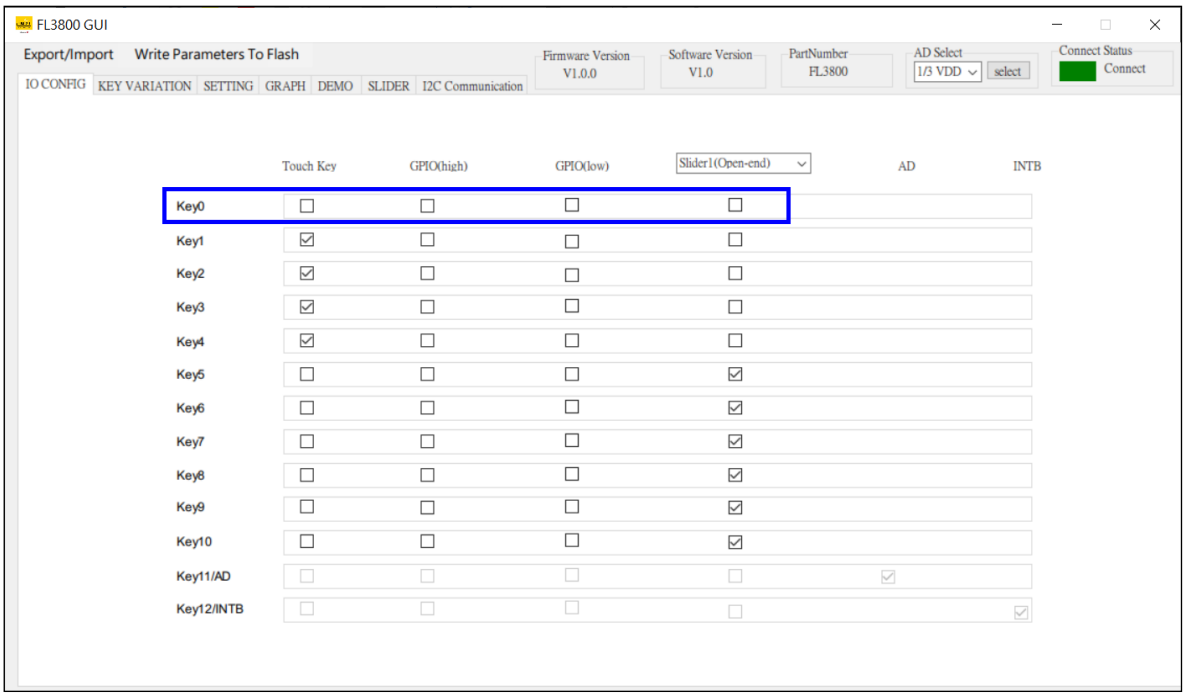

Figure 22: Set EN of KEY0 to disable

Interrupt enabled setting, as shown in Figure 23 below,

Figure 23: KEY's INT and INT_EN

Checking in the box is the action of enable interrupt function, no checking means disable. INT_EN should be set to enable first when configure key interrupt for KEY0~KEY10. INT_EN is the global interrupt setting. If it is disabled, all keys interrupt will be turned off even though key is pressed.

Note: If KEY is set to GPIO (High) or GPIO (low) on the IO CONFIG page, the INT_EN corresponding to KEY will also be disabled.

5.6 Key Calibration

As shown in Figure 24 below,

Figure 24: The calibration of the KEY

"0" in a box means KEY0 and "10" means KEY10. System will force calibrating the corresponding KEY by pressing button. Please make sure there in no action on keys during calibration, or it will cause errors.

5.7 Noise Display and Threshold Setting

As shown in Figure 25 below,

Figure 25: The noise lights of the KEY

There are 16 noise lights for KEY0~KEY15 (Currently IS31FL3800 EVB only supports KEY0~KEY10, a total of 11 buttons). Light will be red when the corresponding KEY has noise or it will be gray.

NOISE_TH_SET bit is noise threshold set from 0~127. Input data in the corresponding box and press Enter or Tab key, or move the cursor to another location will set up the threshold.

When the VALUE of the sample changes more than NOISE_TH_SET, but not exceeds the key threshold setting, the touch key will be considered to be an ambient noise disturbance. And corresponding noise display will turn red.

5.8 KEY Value Display

The key value will be shown in the red block as Figure 26.

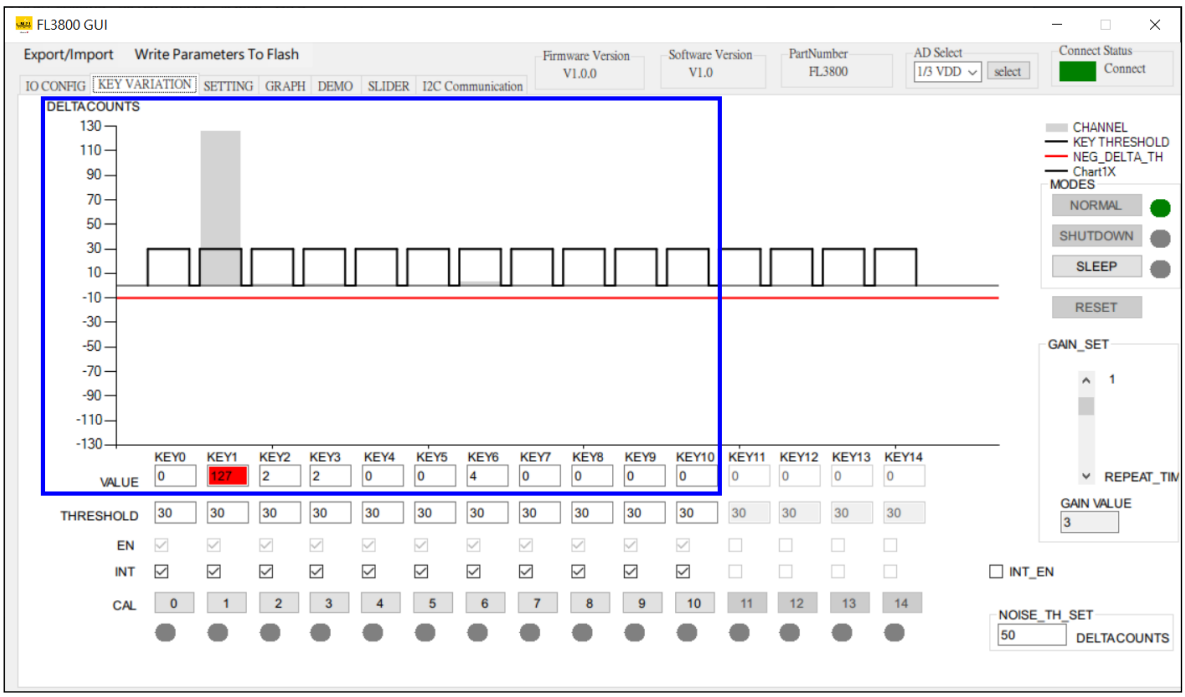

Figure 26: The value of the KEY

As shown in the blue box above, black line is key threshold value, gray line is key value and red line is negative threshold value.

Key value will update the current capacitance of key. The value of KEY2 is 127 and threshold is 30. The value is over threshold, so KEY2 value display red meaning pressed.

6 SETTING

As shown in Figure 27 below, the touch key related parameters are set as follows.

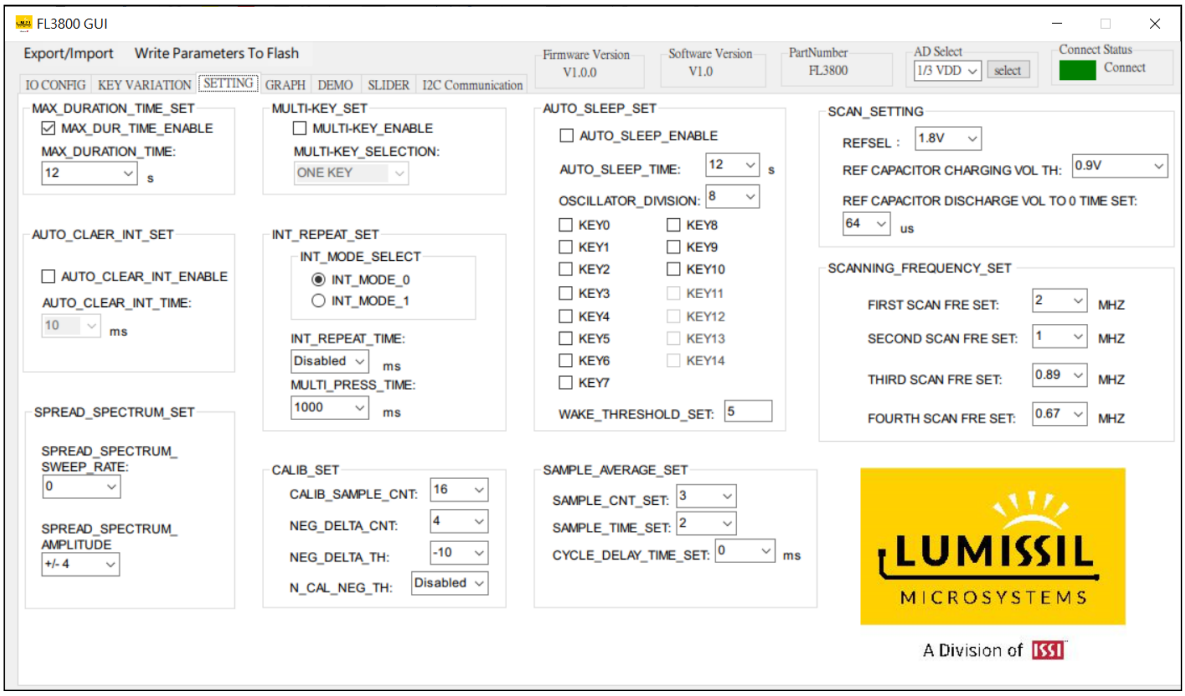

Figure 27: Setting page of GUI

6.1 MAX_DURATION_TIME_SET：Maximum Pressing Duration Time Setting

As shown in Figure 28 below,

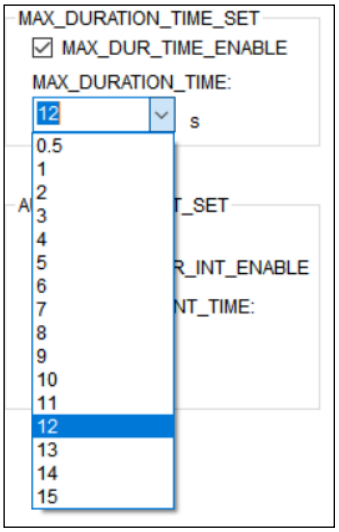

Figure 28: MAX_DURATION_TIME_SET option

MAX_DUR_ENABLE is the maximum pressing duration time function enable. Checking is enable, no checking is disable.

MAX_DURATION_TIME is the maximum pressing duration time setting. Unit is second. When pressing time is over MAX_DUR_TIME, system will force calibrating the pressed key.

This function is mainly used to prevent a touch key from the environment factor. For example, the water drop is on the touch button, which causes the touch button keep pressing status and cannot be used again.

If the maximum pressing time is set, when the touch button is pressed over programmed time, it will be force calibration. Then the touch button can be used after being affected by the water drop.

6.2 AUTO_CLEAR_INT_SET：Auto-Clean Interrupt

As shown in Figure 29 below,

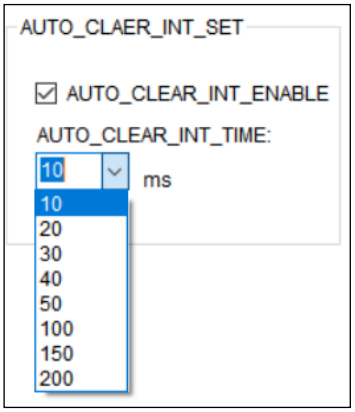

Figure 29: AUTO_CLEAR_INT_SET option

AUTO_CLEAR_INT_ENABLE is auto-clean interrupt function. Checking is enable.

AUTO_CLEAR_INT_TIME is auto-clean interrupt time to choose. Unit is milisecond.

When AUTO_CLEAR_INT_ENABLE is disabled, only reading 02h and 03h registers will releae the INTB pin, otherwise it will keep low.

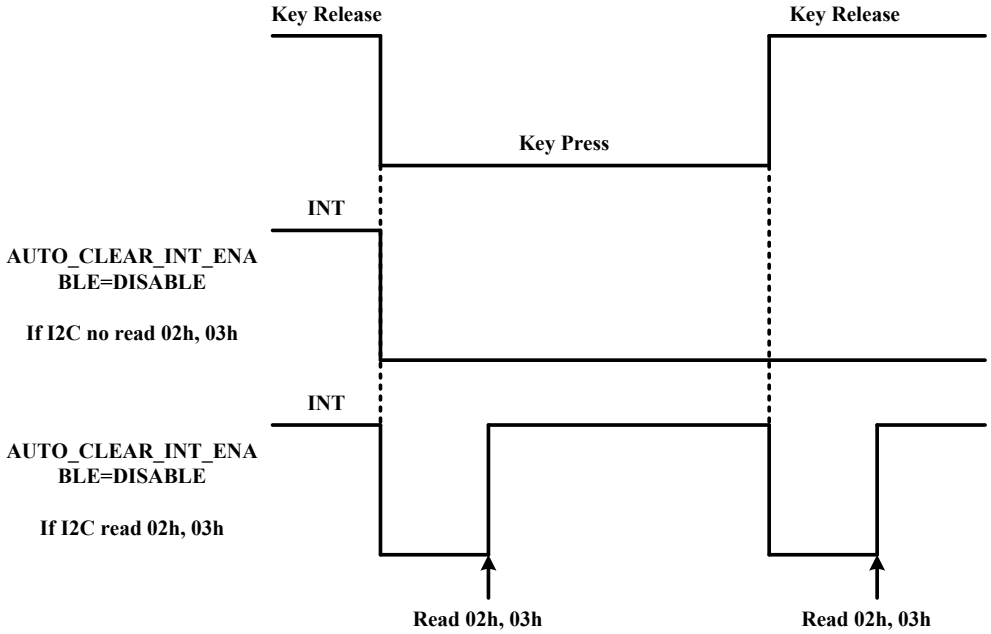

Figure 30: INTB action when AUTO_CLEAR_INT_ENABLE is disabled

When AUTO_CLEAR_INT_ENABLE is enabled, INTB pin will be released by reading 02h (Key Status Register1)

and 03h (Key Status Register2) registers. If 02h and 03h registers are not be read within programmed time AUTO_CLEAR_INT_TIME (10ms~200ms), then IS31FL3800 will release INTB pin after AUTO_CLEAR_INT_TIME time expired.

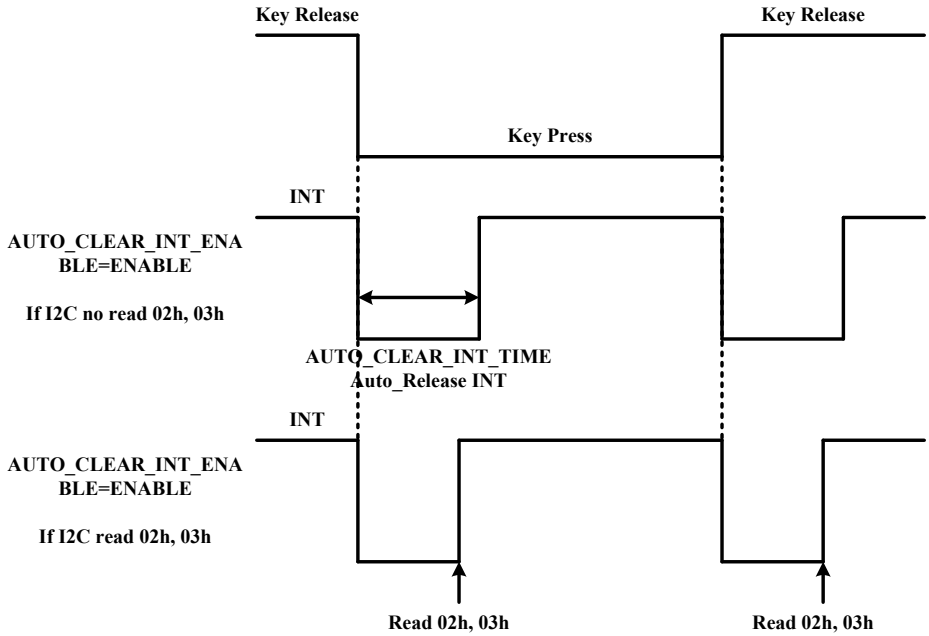

Figure 31: INTB action when AUTO_CLEAR_INT_ENABLE is enabled

Note: KEY12 has been fixed as INTB pin.

6.3 SPREAD_SPECTRUM_SET：Spread Spectrum Setting

As shown in Figure 32 below,

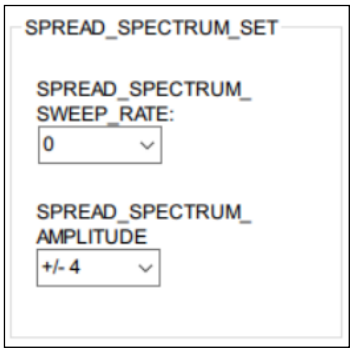

Figure 32: INTB action when AUTO_CLEAR_INT_ENABLE is enabled

SPREAD_SPECTRUM_SWEEP_RATE is defines the spread spectrum sweep rate. If SPREAD_SPECTRUM_SWEEP_RATE = 0, then the spread spectrum is disabled.

SPREAD_SPECTRUM_AMPLITUDE is defines the amplitude of spread spectrum frequency change.

The setting of spread sweep frequency and the spread amplitude should be carefully selected to reduce the effect of EMI.

6.4 MULTI-KEY_SET：Multi-Key Setting

As shown in Figure 33 below,

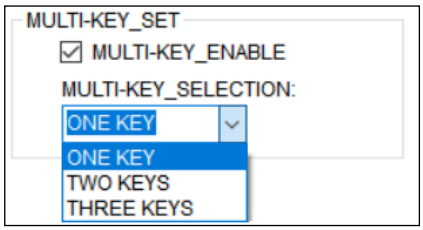

Figure 33: MULTI-KEY_SET option

MULTI-KEY_ENABLE is enabled multi-key function. Checking is enable. When MULTI-KEY_ENABLE is not checking, all keys are available.

MULTI-KEY_SELECTION can be set to ONE KEY, TWO KEYS or THREE KEYS by clicking the arrow in box. In some applications, such as a password lock, the number of keys pressed need to limit at the same time. The MULTI-KEY_SELECTION should be set to ONE KEY to prevent error on password lock.

6.5 INT_REPEAT_SET：Interrupt Repeat Setting

As shown in Figure 34 below,

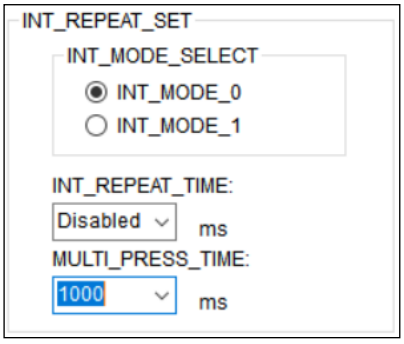

Figure 34: INT_REPEAT_SET option

INT_MODE_SELECT is INT_MODE mode setting. System generates one interrupt only by pressing or releasing keys when pick INT_MODE_0. System generates interrupt repeatly by pressing keys when pick INT_MODE_1 and releaseing key will trigger once interrupt.

INT_REPEAT_TIME is used to set interrupt auto-repeat time. Click arrow in bow to choose different time and unit is milisecond.

MULTI_PRESS_TIME is used to set the time between first and second interrupt. Click arrow to choose different time and unit is milisecond.

If there is a key keeping pressing, second interrupt will be generate untill MULTI_PRESS_TIME after first interrupt and waiting for INT_REPEAT_TIME to trigger third interrupt and going on interrrupt.

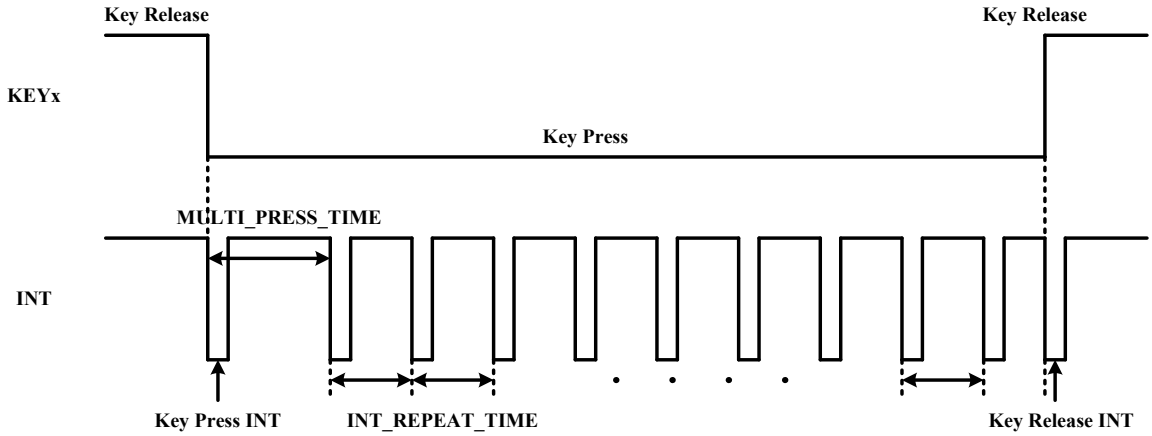

Figure 35: INTB behavior when setting the INT_REPEAT_SET parameter

6.6 SAMPLE_AVERAGE_SET：Sampling Frequency and Average Number Setting

As shown in Figure 36 below,

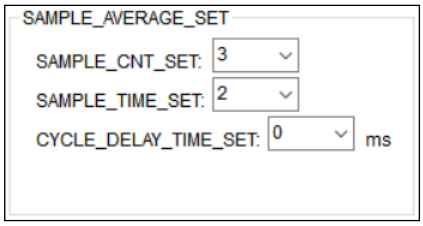

Figure 36: SAMPLE_AVERAGE_SET option

SAMPLE_CNT_SET is to set each button sampling number, the average value of multiple samples is taken as the final key value to improve the scanning stability.

SAMPLE_TIME_SET is single sampling time for SAMPLE_CNT_SET, then sampling time for each key is: SAMPLE_TIME_SET*SAMPLE_CNT_SET.

The larger value of SAMPLE_CNT_SET and SAMPLE_TIME_SET, the better the reliability of the key. However at the same time the sampling speed of the button will be slower. Please set these values according to the actual application.

CYCLE_DELAY_TIME_SET is a cycle delay time after all keys scanned.

6.7 AUTO_SLEEP_SET：Auto SLEEP Mode Setting

As shown in Figure 37 below,

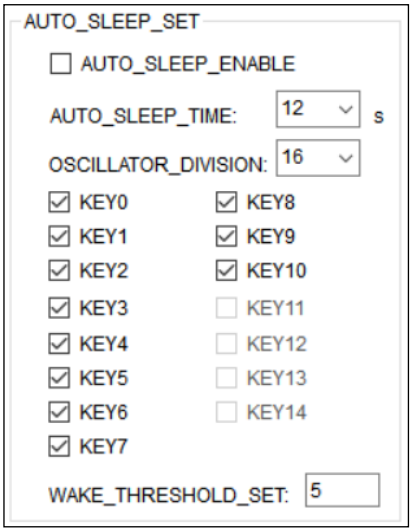

Figure 37: AUTO_SLEEP_SET option

IS31FL3800 integrates AUTO_SLEEP function and the entering time could be configured. System will enter into SLEEP Mode when no action in touch key. It will be waked up by any key action. In some applications that require low power consumption, it can be set to SLEEP mode automatically.

KEY0~KEY10 is enabled to exit SLEEP Mode (Click SLEEP button in Figure 17 to enter into SLEEP Mode in GUI interface). Enable the corresponding key and system will be waked up as key value arrives the wake up threshold (WAKE_THRESHOLD)

WAKE_THRESHOLD_SET is used to set the key threshold for waking up from SLEEP Mode. Input data in box and press Enter or Tab key, or move the cursor to another location will set up the threshold.

6.8 CALIB_SET：Calibration Setting

As shown in Figure 38 below,

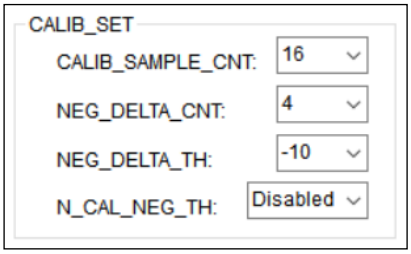

Figure 38: CALIB_SET option

CALIB_SAMPLE_CNT sets times for auto-calibrate cycle. The influence of parasitic capacitance shift on touch key will be reduced by calibrating.

When the variety of continuous sampling is negative and over the negative threshold (NEG_DELTA_TH), the corresponding key will be forced calibrating.

NEG_DELTA_TH is used to set negative threshold.

N_CAL_NEG_TH is the negative threshold for forced calibration.

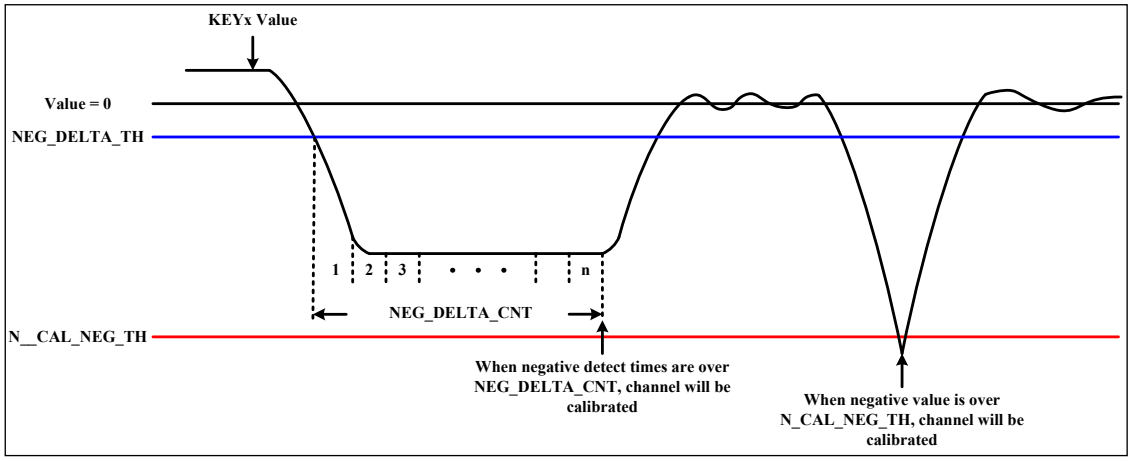

Figure 39: AUTO_SLEEP_SET option

6.9 SCAN_SETTING：Scan And Frequency Setting

Scanning function is set as shown in Figure 40,

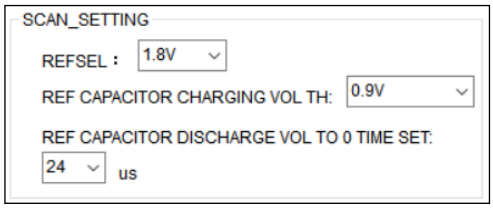

Figure 40: SCAN_SETTING option

REFSEL is VREF selection (Use 1.8V or VDDH as a reference).

REF CAPACITOR CHARGING VOL TH is set the reference capacitor charging voltage.

Improve Vth will increase detecting time and sensitivity, but might interfere SNR.

REF CAPACITOR DISCHARGE VOL TO 0 TIME SET is set the reset time for reference capacitor discharging.

The reference capacitor should be reset before each detecting period and increase discharging time by actual application.

Scanning frequency is set as shown in Figure 41,

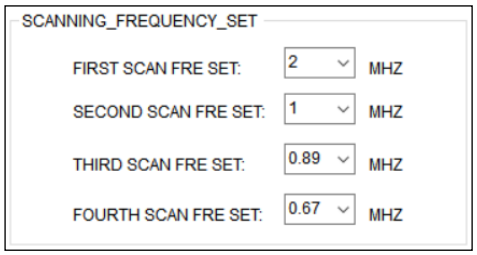

Figure 41: SCANNING_FREQUENCY_SET option

FIRST SCAN FRE SET is the first scanning frequency setting;

SECOND SCAN FRE SET is the second scanning frequency setting;

THIRD SCAN FRE SET is the third scanning frequency setting;

FOURTH SCAN FRE SET is the fourth scanning frequency setting.

A higher scanning frequency will decrease detecting time and not sensitive for alternating supply ripple, precondition is the sensing pad charged enough to VREF or the sensitive will get worse.

These four scanning frequencies are plus total to evaluate the effect.

7 GRAPH

GRAPH is KEY value curves to show the current value of KEY0~KEY15 (Currently IS31FL3800 EVB only supports KEY0~KEY10). As shown in Figure 42, history value of KEYs will be checked by pulling the scroll bar. Users can uncheck the "KEY ENABLE" box in the lower right corner to filter out unwanted key values.

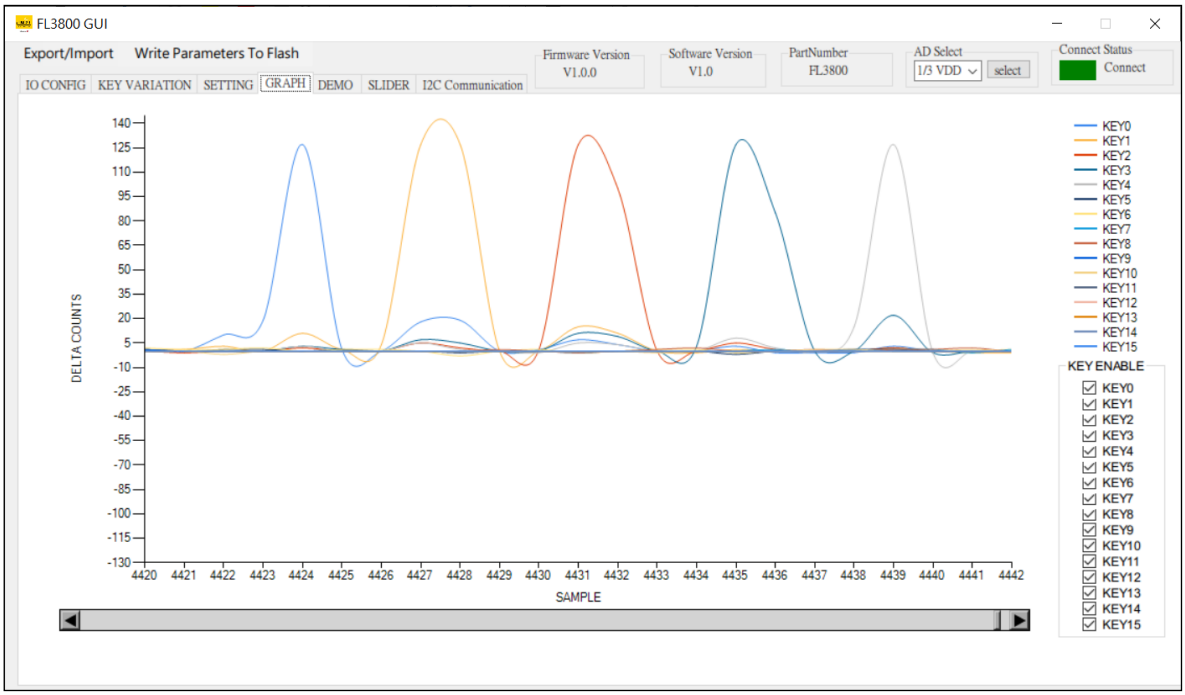

Figure 42: Display the current value of KEY on the GRAPH page

8 DEMO

8.1 Demonstration Type

As shown in Figure 43 below, there are three demo examples to show. The user can select the desired demo to Run, Stop, Pause and Resume. Or use the LoadDemo button to load the LED pattern designed already. When the demo is running, the Demo Info will display relevant information about the current demo. Running Info will display the current number of frames and repeats.

In the demo example, you can also enable the single step option and press the run button. Each time you press the Next button, the current frame will increase by 1 to observe the LED pattern change.

Demo1 (file name: Moving): The square pattern moving left and right.

Demo2 (file name: Square): The square pattern moves up and down, left and right.

Demo3 (file name: Color): RGB lights flash sequentially.

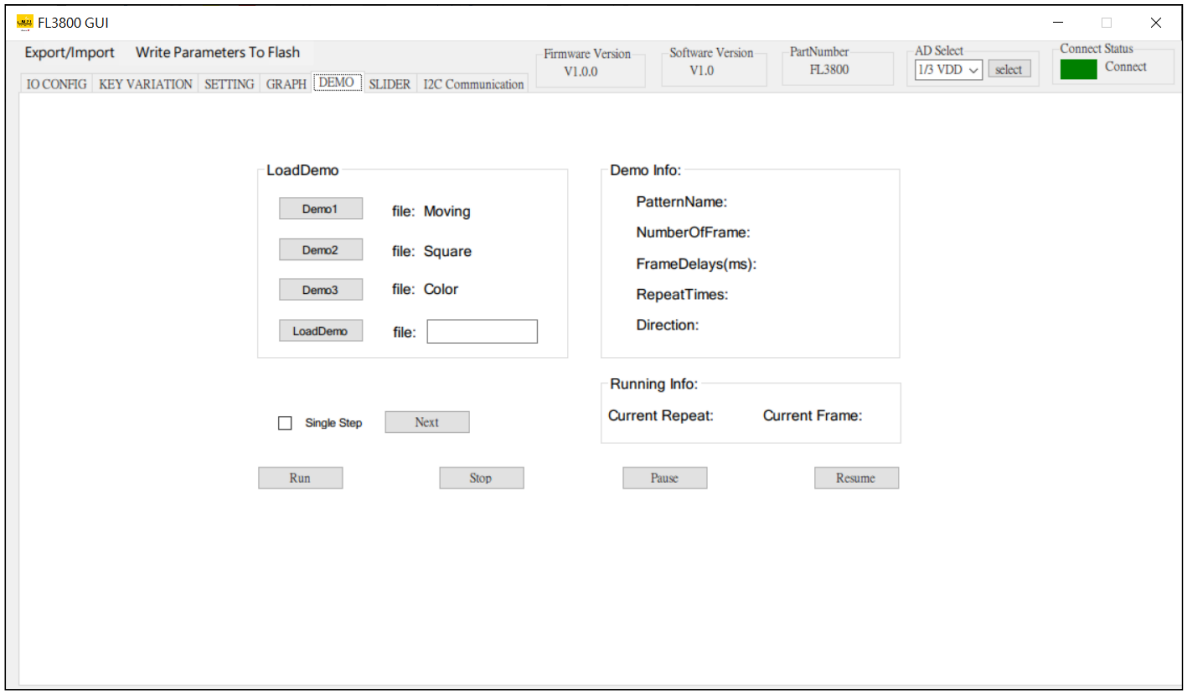

Figure 43: Demo page of GUI

8.2 Load a Self-Designed Demo File

As shown in Figure 44 below, users can design the desired LED pattern by themselves. Open the folder of 3800 GUI, there are three txt files with LED patterns. Here is an explanation with the 3800 moving LED pattern.

```
PatternName:    Moving
NumberOfFrame:  12
FrameDelays(ms):    100
RepeatTimes:    3
Direction:  LeftToRight

#1 Frame:
    FF  FF  FF  0   0   0   0   0   0   0   0   0   0   0   0   0   0
    FF  FF  FF  0   0   0   0   0   0   0   0   0   0   0   0   0   0
    FF  FF  FF  0   0   0   0   0   0   0   0   0   0   0   0   0   0
    FF  FF  FF  0   0   0   0   0   0   0   0   0   0   0   0   0   0
#2 Frame:
    0   0   0   FF  FF  FF  0   0   0   0   0   0   0   0   0   0   0
    0   0   0   FF  FF  FF  0   0   0   0   0   0   0   0   0   0   0
    0   0   0   FF  FF  FF  0   0   0   0   0   0   0   0   0   0   0
    0   0   0   FF  FF  FF  0   0   0   0   0   0   0   0   0   0   0
#3 Frame:
    0   0   0   0   0   0   FF  FF  FF  0   0   0   0   0   0   0   0
    0   0   0   0   0   0   FF  FF  FF  0   0   0   0   0   0   0   0
    0   0   0   0   0   0   FF  FF  FF  0   0   0   0   0   0   0   0
    0   0   0   0   0   0   FF  FF  FF  0   0   0   0   0   0   0   0
#4 Frame:
    0   0   0   0   0   0   0   0   0   FF  FF  FF  0   0   0   0   0
    0   0   0   0   0   0   0   0   0   FF  FF  FF  0   0   0   0   0
    0   0   0   0   0   0   0   0   0   FF  FF  FF  0   0   0   0   0
    0   0   0   0   0   0   0   0   0   FF  FF  FF  0   0   0   0   0
#5 Frame:
    0   0   0   0   0   0   0   0   0   0   0   0   FF  FF  FF  0   0
    0   0   0   0   0   0   0   0   0   0   0   0   FF  FF  FF  0   0
    0   0   0   0   0   0   0   0   0   0   0   0   FF  FF  FF  0   0
    0   0   0   0   0   0   0   0   0   0   0   0   FF  FF  FF  0   0
#6 Frame:
    0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   FF  FF
    0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   FF  FF
    0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   FF  FF
    0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   FF  FF
```

Figure 44: 3800Moving.txt parameter description

As shown in the blue box above, the parameter description is as follows:

PatternName: LED pattern name.

NumberOfFrame: Number of Frame.

FrameDelays(ms): Delay time between frames.

RepeatTimes: The number of times all frames are repeated.

Direction: LED display direction. There are two options: LeftToRight or RightToLeft.

As shown in the red box above. The red, green and blue LED PWM data are arranged in order. The LED PWM data is a hexadecimal value. The maximum value is FF (brightest brightness), and the minimum value is 00 (darkest brightness).

9 Slider

As shown in the blue box in Figure 45, black line is key threshold value, gray line is key vlaue.

When the selected key is combined to the slider (Set the slider type through the IO CONFIG page in Figure 16). The keys used as a slider will be updated as the finger moves. The purple box on the right and the green box below will also be updated immediately.

Sliding keys can also change the order. As shown in the red box below, the key sequence is Key5 to Key10. Users can change the order of slider keys as needed.

Note: The number of Sliding keys must be 6 keys to be combined.

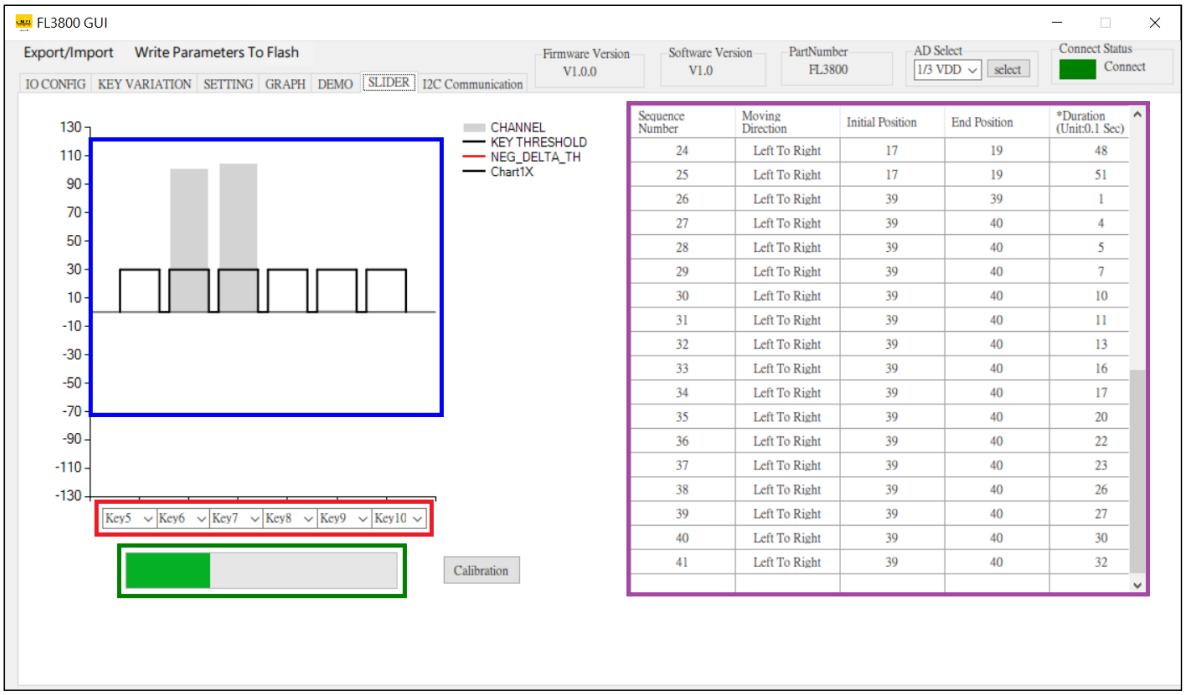

Figure 45: Slider page of GUI

9.1 Slider Type

As shown in Figure 46 below, Users can set the following slider types (Set the slider type through the IO CONFIG page in Figure 16).

Slider Bar (Open-end): Used in Slider Board (As shown on the left side of Figure 2).

Slider Wheel (Close-end): Used in Wheel Board (As shown on the right side of Figure 2).

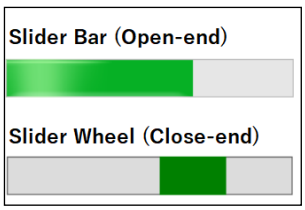

Figure 46: Slider Bar(Open-end) and Slide Wheel(Close-end)

9.2 Slider Key Calibration

When the user develops the slider PCB (Slider Bar or Slider Wheel), the calibration button can be used to calibrate the moving distance of the slider. When the calibration button is pressed, as shown in the blue box in Figure 47, the red blocks will be displayed in order. The user must move the slider with a finger at a constant speed until the red blocks are displayed to the end.

Note: If no touch is detected within 3 seconds after pressing the calibration button, the calibration process will end.

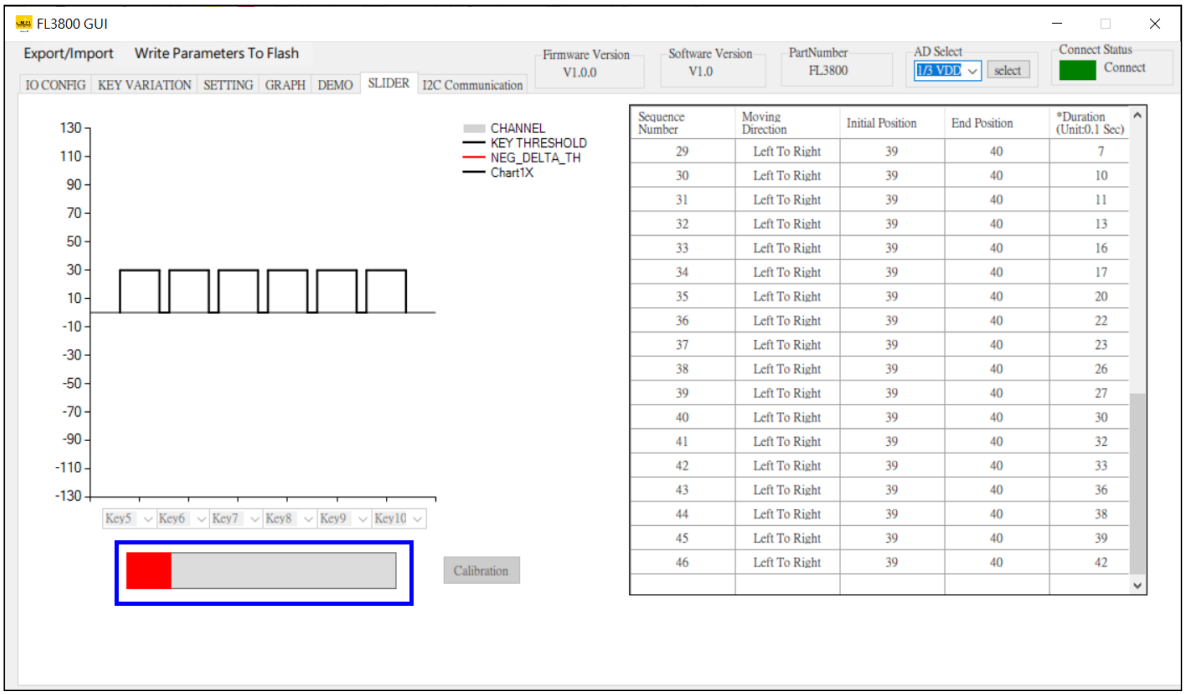

Figure 47: Slide key calibration

10 I2C Communication

I2C communication allows users to change the register data of I2C commands of different headers by writing commands (Touch Key or LED part, 55h means Touch Key, AAh means LED). All TK parts can read the register, but the LED part can only write data.

As shown in the blue box in Figure 48, the user can issue I2C commands to the EVB here. On the right, you can choose to load, store, send and clear actions. I2C command set can be composed of several bytes. These bytes can be separated by space or comma. I2C command set can be terminated by a line wrap.

As shown in the red box, you can choose to save or clear the data log on the right after the result of the I2C command is issued. The green box indicates the delay time setting between each command and command, the delay setting range is 10ms~1000ms.

If the switch of the AD pin is set to "ON" (As shown in Figure 10), the setting of 1/3VDD is used. The communication address of EVB is Write command: 7Ah, Read command: 7Bh.

Note: Please select the correct I2C address according to the setting of the AD pin switch on the EVB (Please refer to Table 2).

Figure 48: I2C Communication page of GUI

10.1 I2C Communication Example1 (Touch Key)

Command list:

Command	Description
7A AA 04	Read data from register address 0x04
7B	
7A AA 05	Read data from register address 0x05
7B	
7A AA 04 07	Write data 0x07 to register address 0x04
7A AA 05 03	Write data 0x03 to register address 0x05

Log list:

Time	Direction	Command/Data	Description
22:16:45:170	>>	7A AA 04	Read data from register address 0x04
22:16:45:197	>>	7B	
22:16:45:197	<<	FF	The data read out is FFh
22:16:45:218	>>	7A AA 05	Read data from register address 0x05
22:16:45:238	>>	7B	
22:16:45:238	<<	07	The data read out is 07h
22:16:45:259	>>	7A AA 04 07	Write data 0xa9 to register address 0x69

22:16:45:279	>>	7A AA 05 03	Write data 0xa9 to register address 0x69

10.2 I2C Communication Example2 (LED)

Command list:

Command	Description
7A,55,FE,C5	Write 0xC5 to the FEh register to unlock.
7A,55,FD,01	Point to page1 (FDh register).
7A,55,51,FF	Set the global current (Page1, 51h register).
7A,55,50,01	Configure the operation mode (Page1, 50h register), SSD=1.
7A,55,01,FF,FF,FF,FF,FF,FF,FF,FF, FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF, FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF, FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF, FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF, FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF, FF,FF,FF,FF,FF,FF,FF,FF,FF	Set scaling for each LED (Page1, 01h~48h register).
7A,55,FE,C5	Write 0xC5 to the FEh register to unlock.
7A,55,FD,00	Point to page0 (FDh register).
7A,55,01,FF,00,00,FF,00,00,FF,00, 00,FF,00,00,FF,00,00,FF,00,00,FF, 00,00,FF,00,00,FF,00,00,FF,00,00, FF,00,00,FF,00,00,FF,00,00,FF,00, 00,FF,00,00,FF,00,00,FF,00,00,FF, 00,00,FF,00,00,FF,00,00,FF,00,00, FF,00,00,FF,00,00,FF,00,00	Light up all red LEDs. Set PWM for each LED (Page0, 01h~48h register).

Log list:

Time	Direction	Command/Data	Description
11:04:46:739	>>	7A,55,FE,C5	
11:04:46:759	>>	7A,55,FD,01	
11:04:46:780	>>	7A,55,51,FF	
11:04:46:800	>>	7A,55,50,01	
11:04:46:821	>>	7A,55,01,FF,FF,FF,FF,FF,FF,FF,FF, FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF, FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF, FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF, FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF, FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF, FF,FF,FF,FF,FF,FF,FF,FF,FF	
11:04:46:845	>>	7A,55,FE,C5	
11:04:46:866	>>	7A,55,FD,00	
11:04:46:886	>>	7A,55,01,FF,00,00,FF,00,00,FF,00, 00,FF,00,00,FF,00,00,FF,00,00,FF, 00,00,FF,00,00,FF,00,00,FF,00,00,	

		FF,00,00,FF,00,00,FF,00,00,FF,00, 00,FF,00,00,FF,00,00,FF,00,00,FF, 00,00,FF,00,00,FF,00,00,FF,00,00, FF,00,00,FF,00,00,FF,00,00	

11 Debug Customer Target Board Based On GUI

11.1 Connection Block Diagram

As shown in Figure 49 below, the EzISP board can be connected to the customer target board to configure touch keys or LED parameters through the GUI interface.

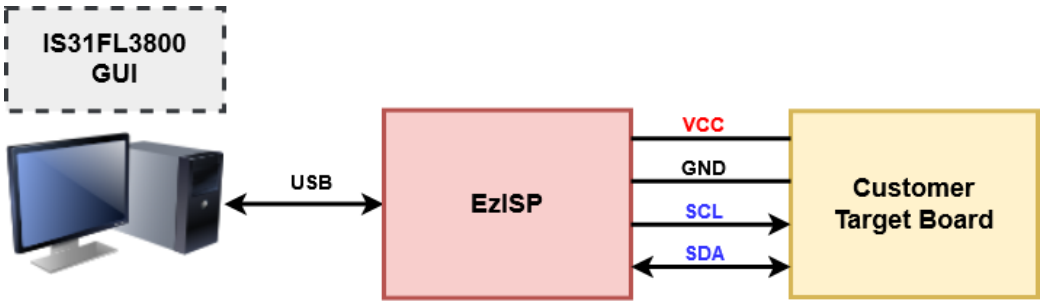

Figure 49: Block diagram of EzISP Board connected to the custom target board

As shown in Figure 50 below, the EzISP Board only needs 4 pins to connect to the custom target board, which are VDD, GND, SCL, and SDA.

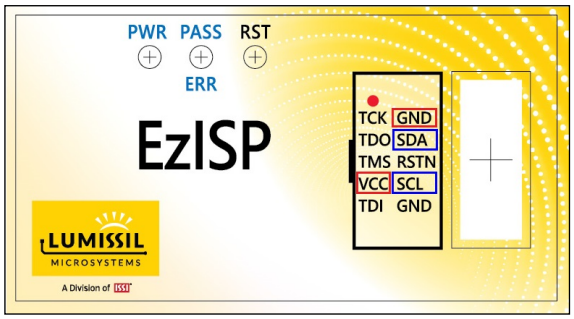

Figure 50: Pin configuration on EzISP board

11.2 Customer Target Board Configuration by GUI

First, you can use the GUI interface to connect to the customer's target board to adjust the parameters of the touch keys and set the LED pattern without having to develop code in the customer's MCU at the beginning. This can shorten development time.