IS31SE5120A EvB User Manual

Revision: D

Publication: June 28, 2023

Copyright © 2020~2023 Integrated Silicon Solution, Inc. All rights reserved. ISSI reserves the right to make changes to this specification and its products at any time without notice. ISSI assumes no liability arising out of the application or use of any information, products or services described herein. Customers are advised to obtain the latest version of this device specification before relying on any published information and before placing orders for products.
Integrated Silicon Solution, Inc. does not recommend the use of any of its products in life support applications where the failure or malfunction of the product can reasonably be expected to cause failure of the life support system or to significantly affect its safety or effectiveness. Products are not authorized for use in such applications unless Integrated Silicon Solution, Inc. receives written assurance to its satisfaction, that:
a.) the risk of injury or damage has been minimized;
b.) the user assumes all such risks; and
c.) potential liability of Integrated Silicon Solution, Inc is adequately protected under the circumstances

1. HARDWARE ENVIRONMENT

1.1 Evaluation Board (SE5120A EvB)

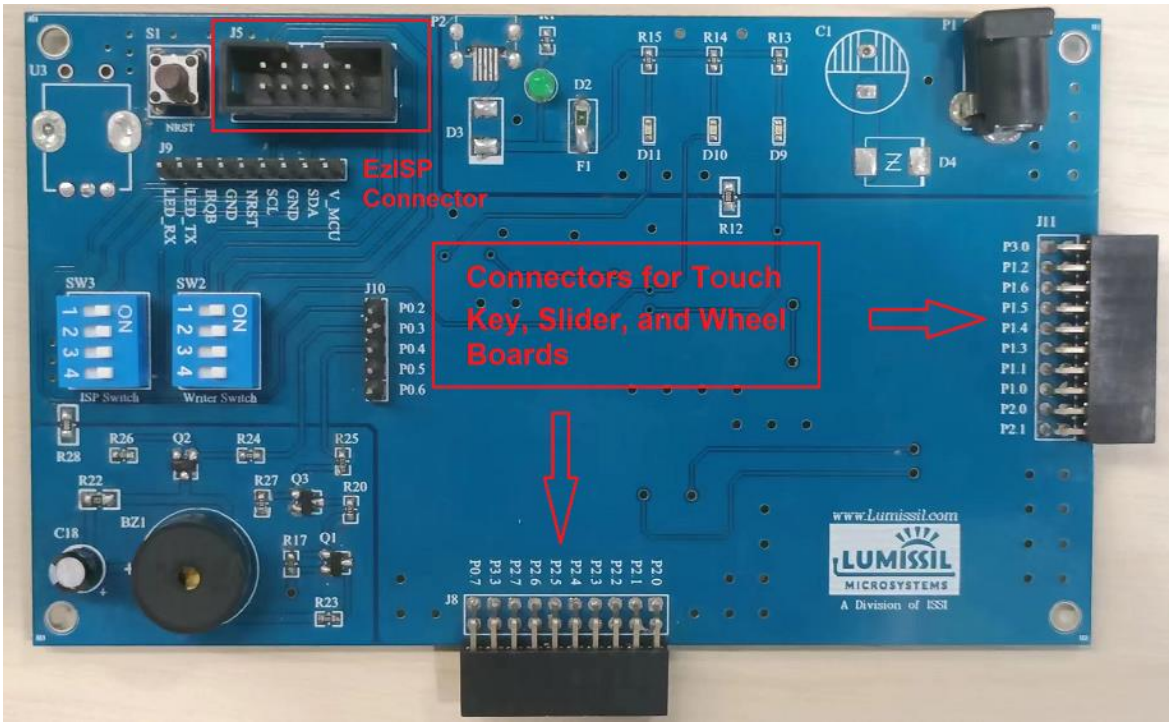


Figure 1-1: Top View of IS31SE5120A Evaluation Board

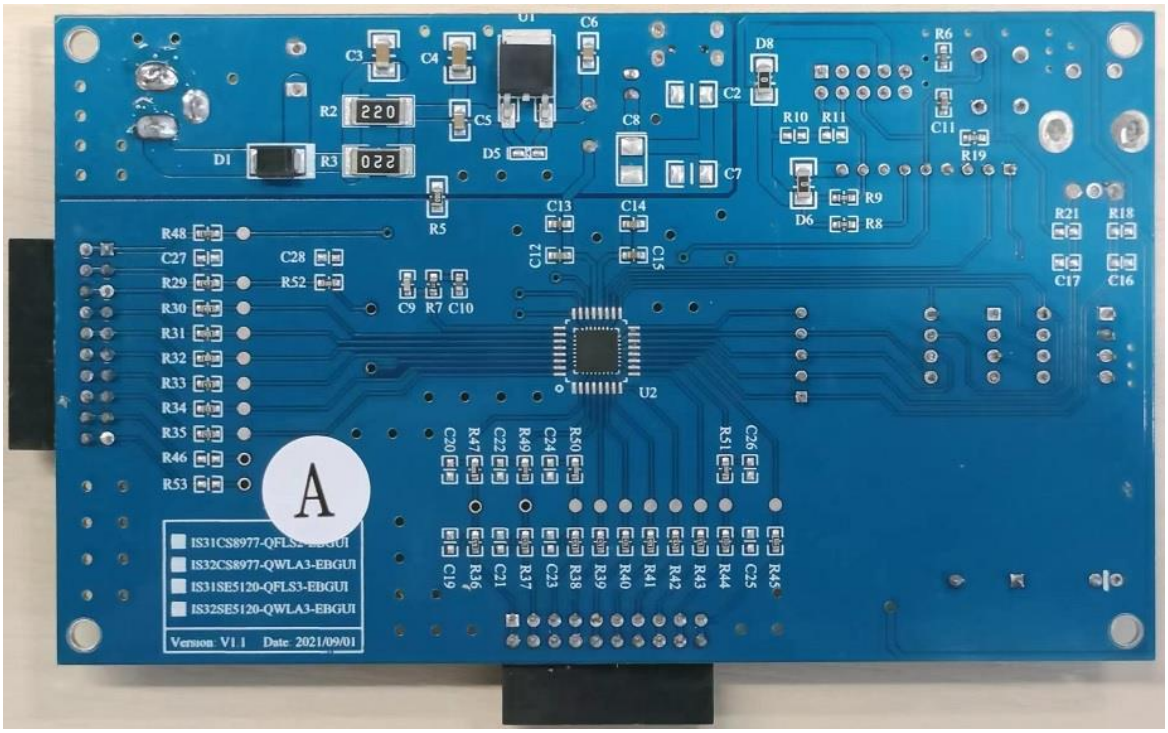


Figure 1-2: Bottom View of IS31SE5120A Evaluation Board

Figure 1-3: Photo of Standard 6 Touch Key Board

Figure 1-4: Photo of Linear Slider Board

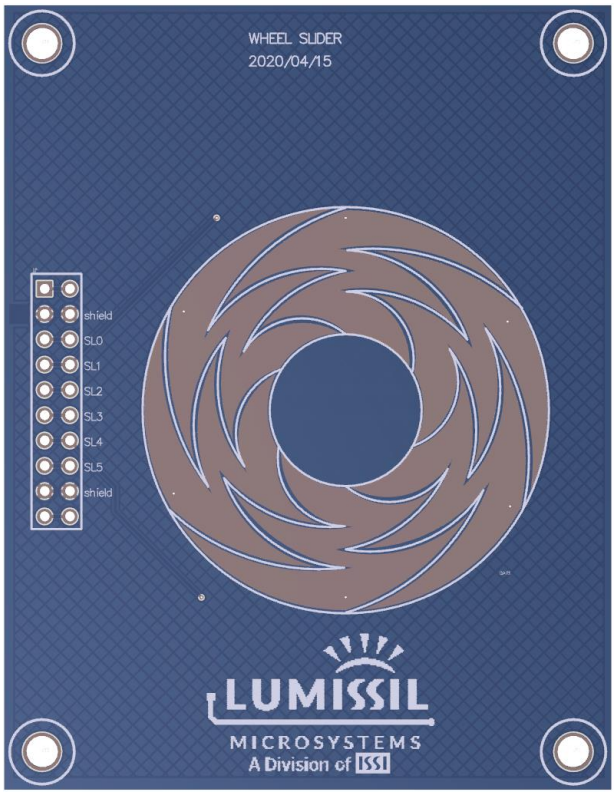


Figure 1-5: Photo of Wheel Slider Board

System Block Diagram:

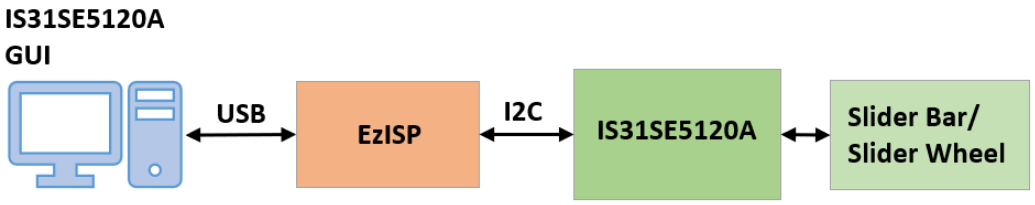
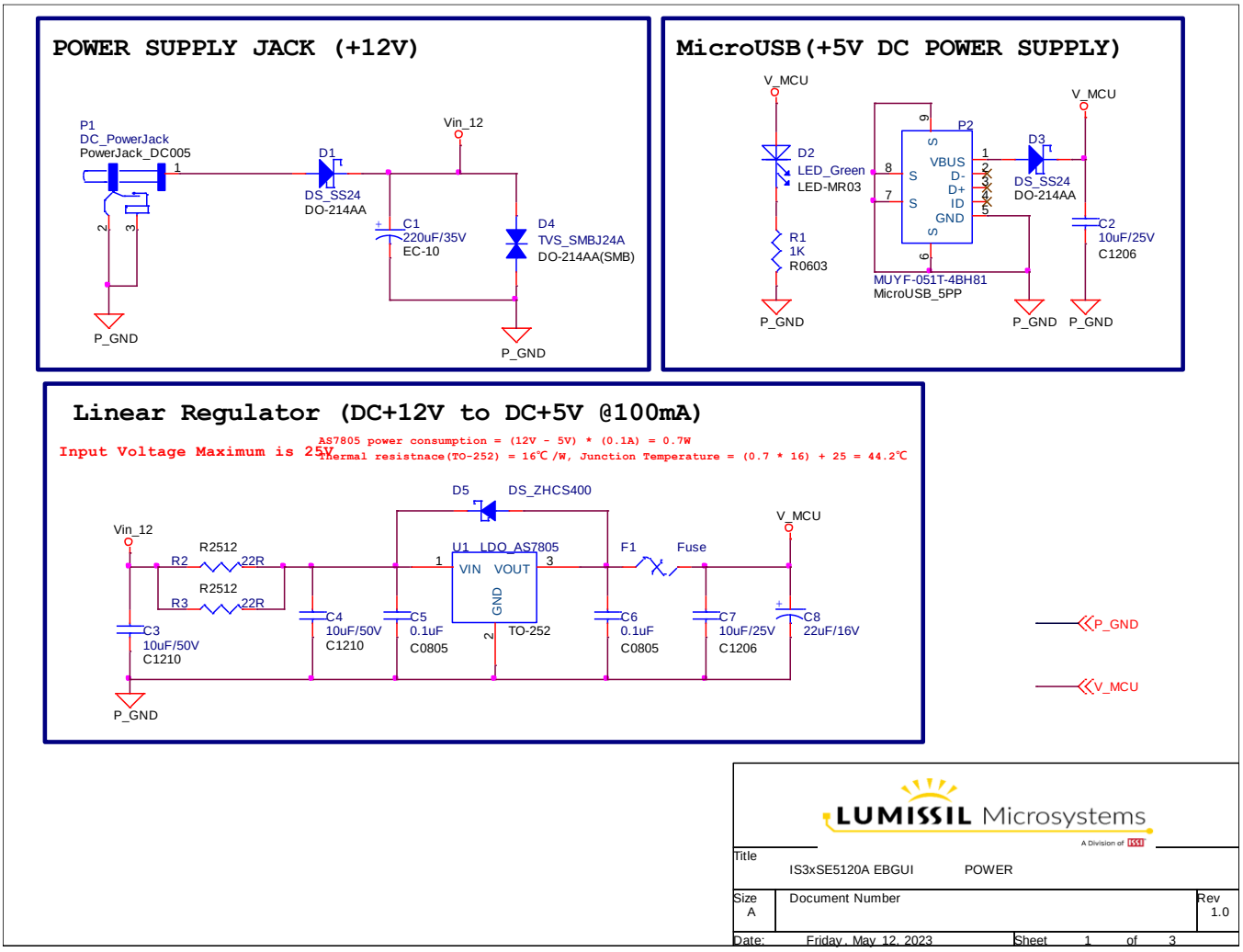
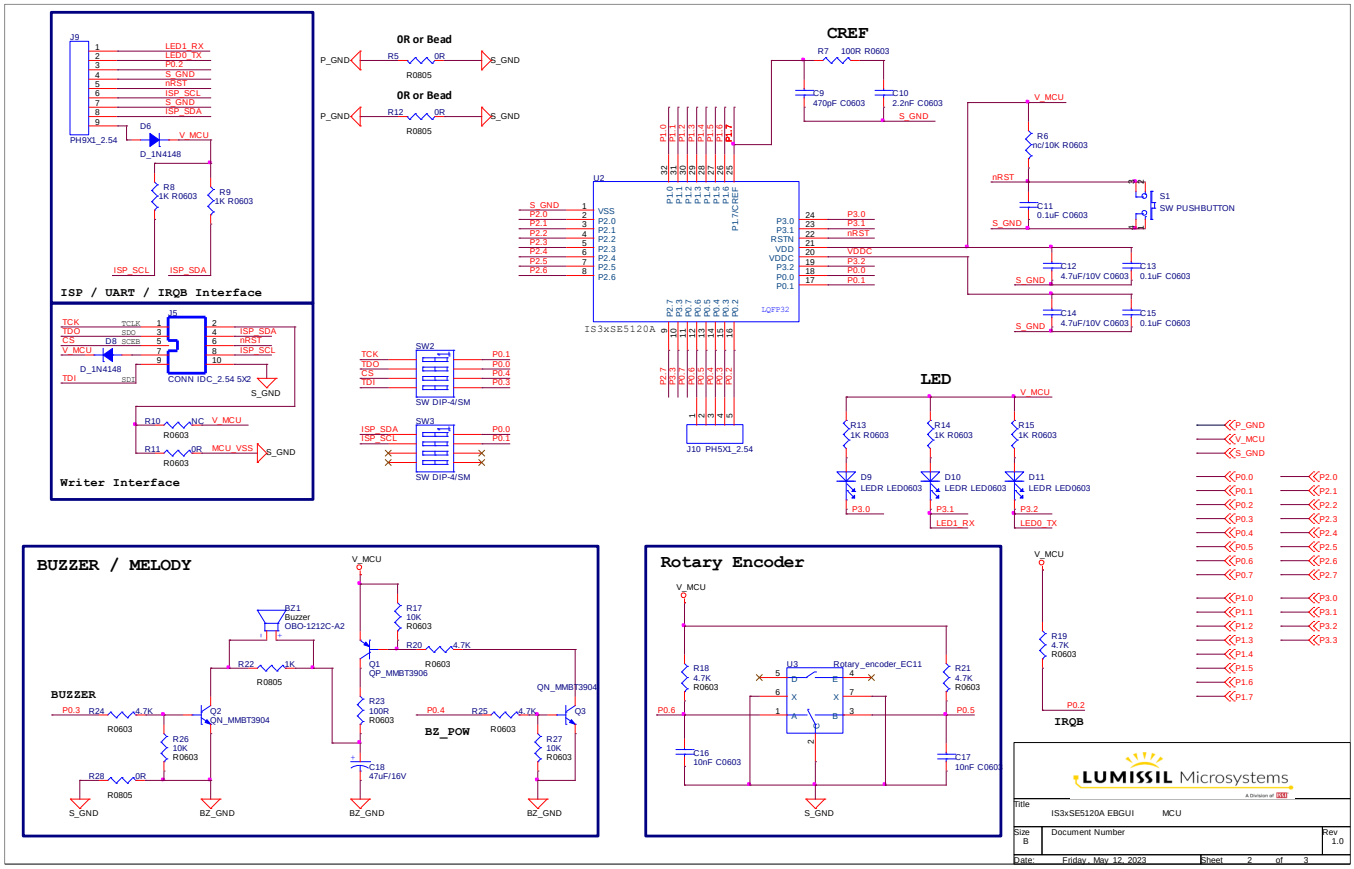


Figure 1-6: IS31SE5120A Evaluation Board Connection Block Diagram

1.2 Schematics of Evaluation Board





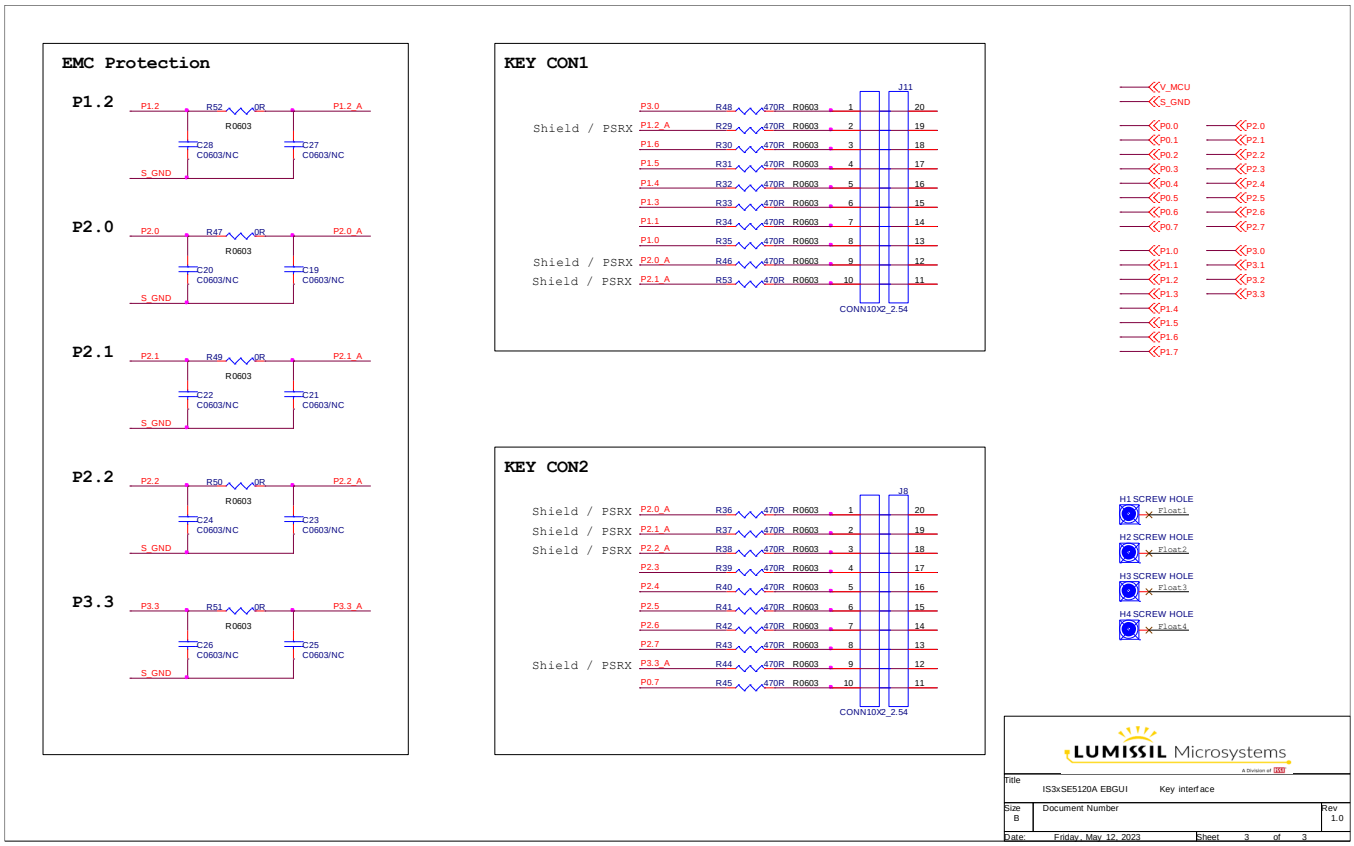


Figure 1-7: Schematic of IS31SE5120A Evaluation Board

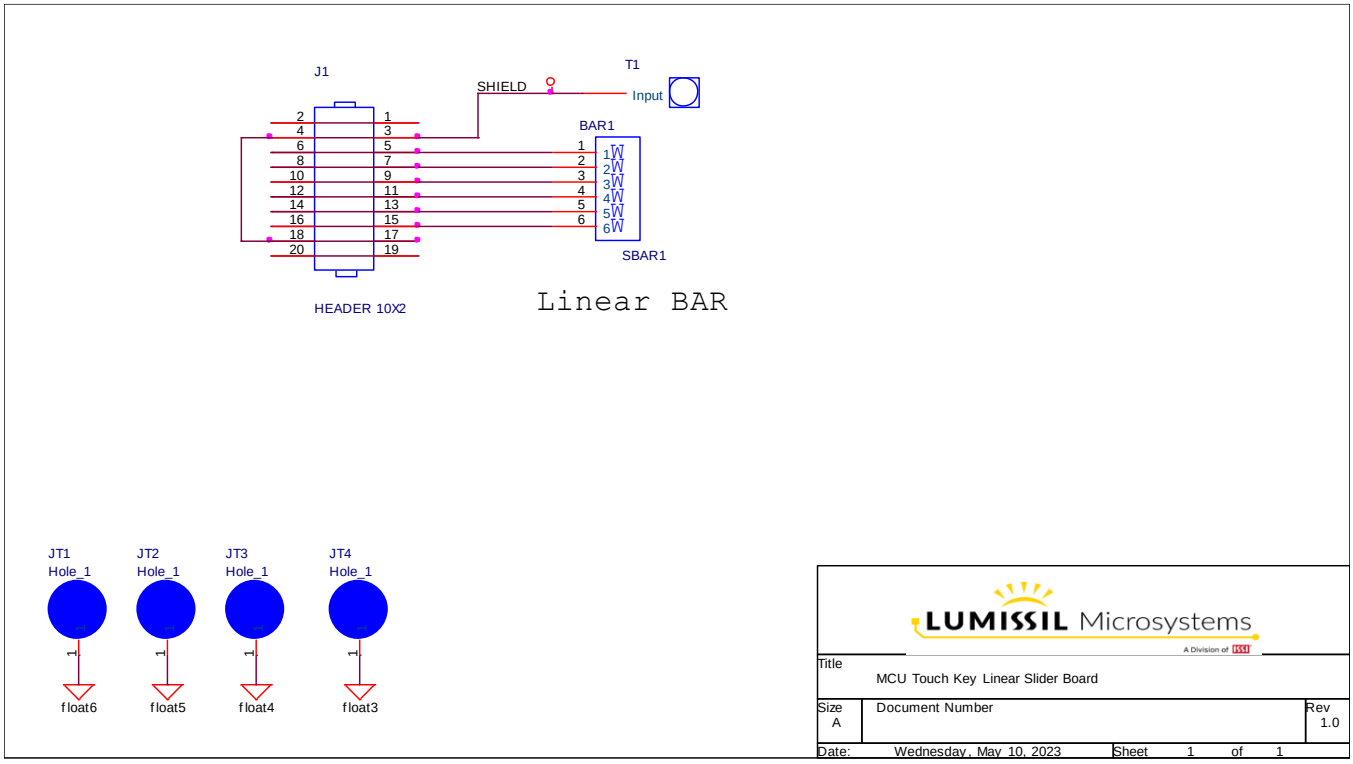


Figure 1-8: Schematic of Slider Board

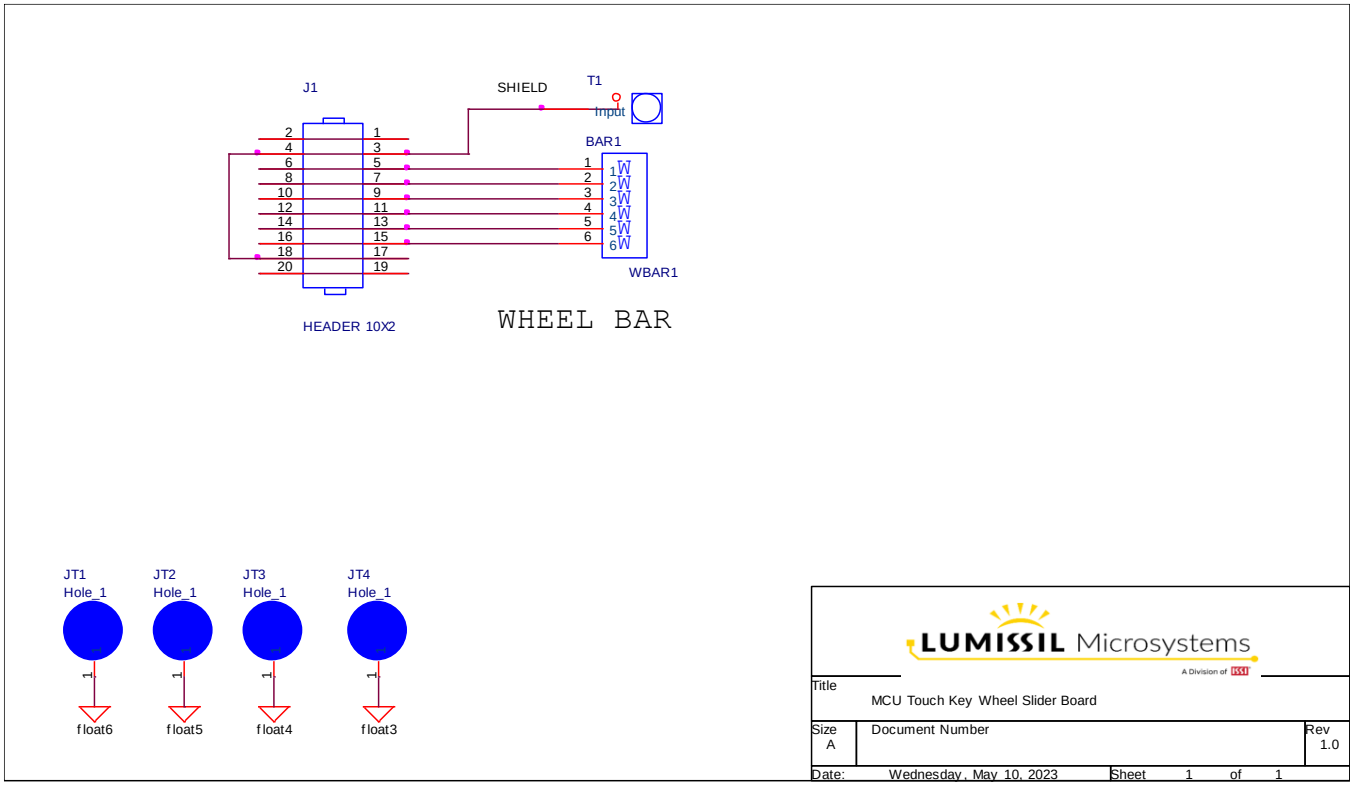


Figure 1-9: Schematic of Wheel Board

1.3 ORDERING INFORMATION

Table 1: Ordering Information

Part No.	Temperature Range	Package
IS31SE5120A-QFLS3-EBGUI	-40°C ~ +105°C	QFN-32, Lead-free

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

2. SOFTWARE SUPPORT

2.1 Software Requirements

Before starting up SE5120A Touch Key GUI software, it is required that the PC be installed with the EzISP USB driver and related files (for example Microsoft Framework and C++ library). Users can unzip file EzISP_V3.3.8.zip and use the setup file for installation. But this installation needs Windows login users with administrator privileges.

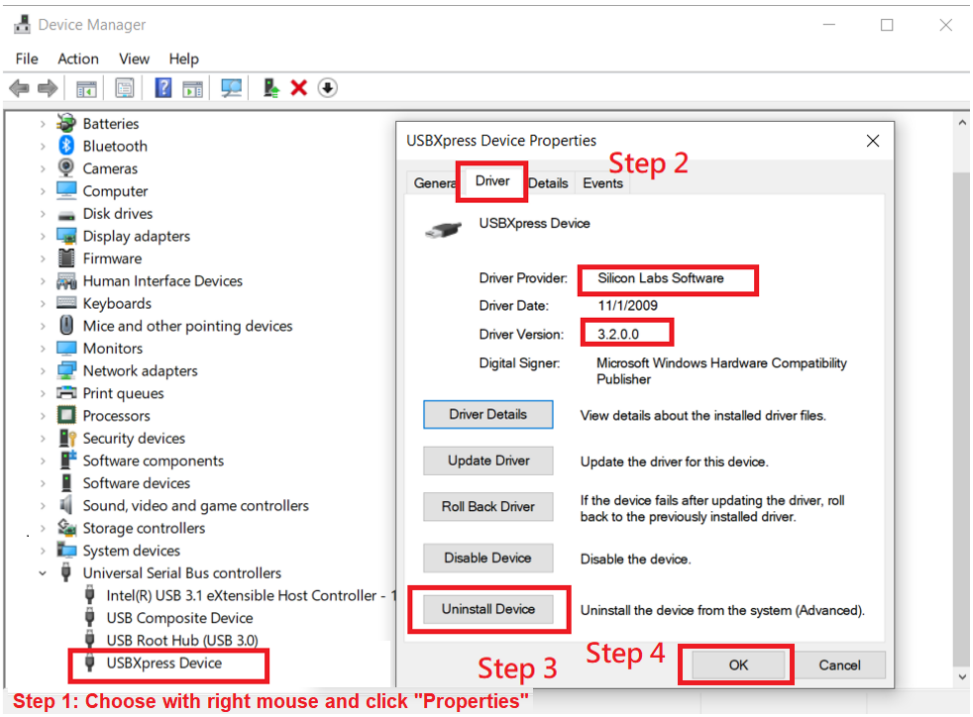
Figure 2-1: Photo of EzISP Board

Windows Software Requirements:

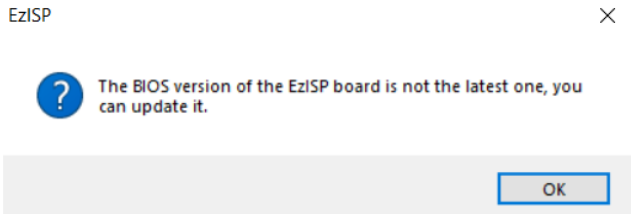
EzISP tool supports Win XP, Vista, Windows 7, Windows 8, Windows 10, and Windows 11. It is required that the Windows-based PC be installed with Microsoft .NET Framework 3.5, Microsoft .NET Framework 4.8, Microsoft Visual C++ 2008, and Silicon Labs' USB driver version 6.7.2.0. After running EzISP setup version 3.3.8 or a later version, Windows will ask the user to install Microsoft .Net Framework or Microsoft Visual C++ 2008 if this PC was not installed with the redistributable packages. Users can follow the pop-up instructions to finish the installation.

Users might need up to seven steps to finish the EzISP installation and configuration after installation for normal operation.

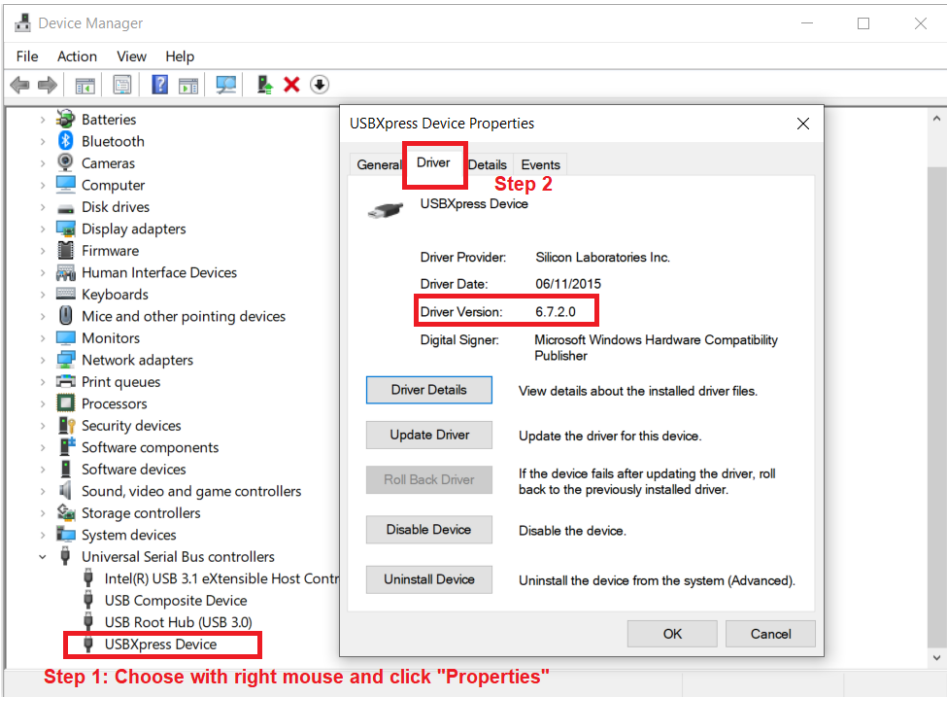
Step 1: For customers who have installed EzISP setup version earlier than 3.3.8 with Silicon Labs' USB driver version 3.2.0.0, USB driver version 3.2.0.0 needs to be uninstalled as the four steps in the below figure before installing the new EzISP setup version 3.3.8.



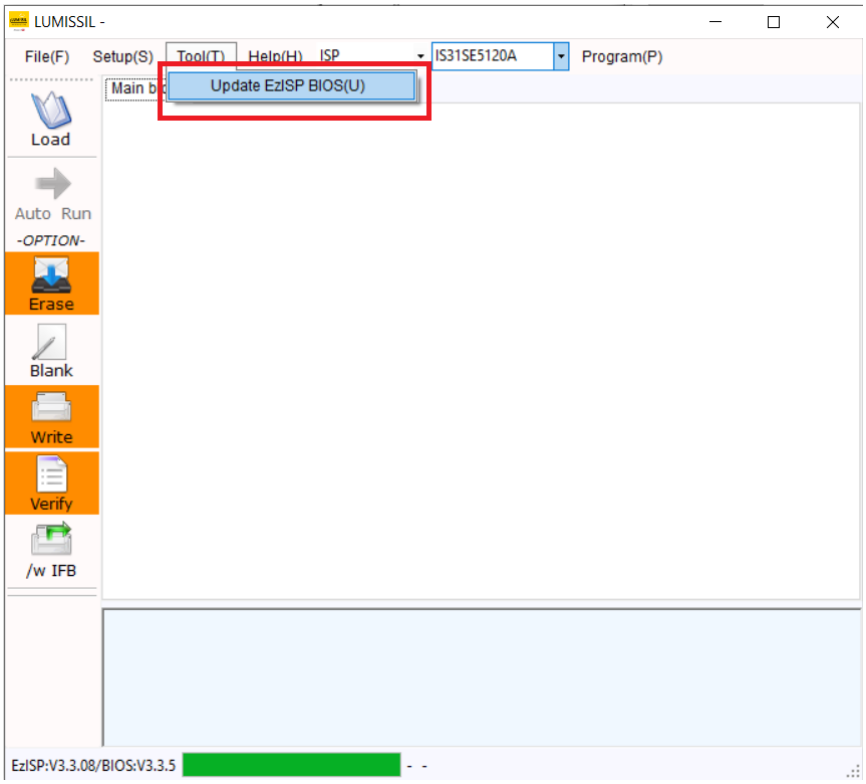
Step 2: Install EzISP setup program. After the installation is done and EzISP is run, users might get below pop-up message and please update with the newer BIOS as Step 3 ~ Step 7.



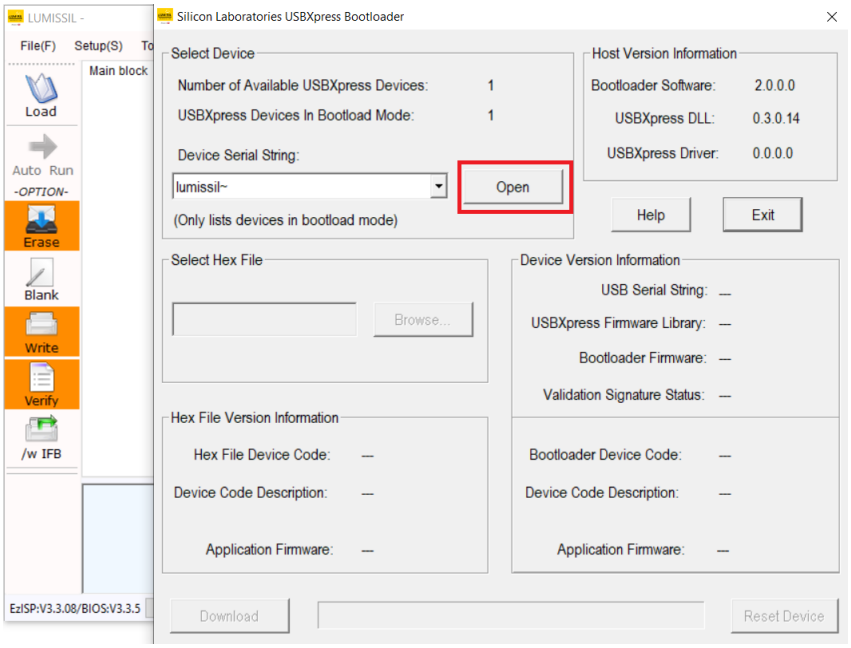
Users can check whether the Silicon Labs' USB driver version 6.7.2.0 is correctly installed or not as the two steps in the below figure.



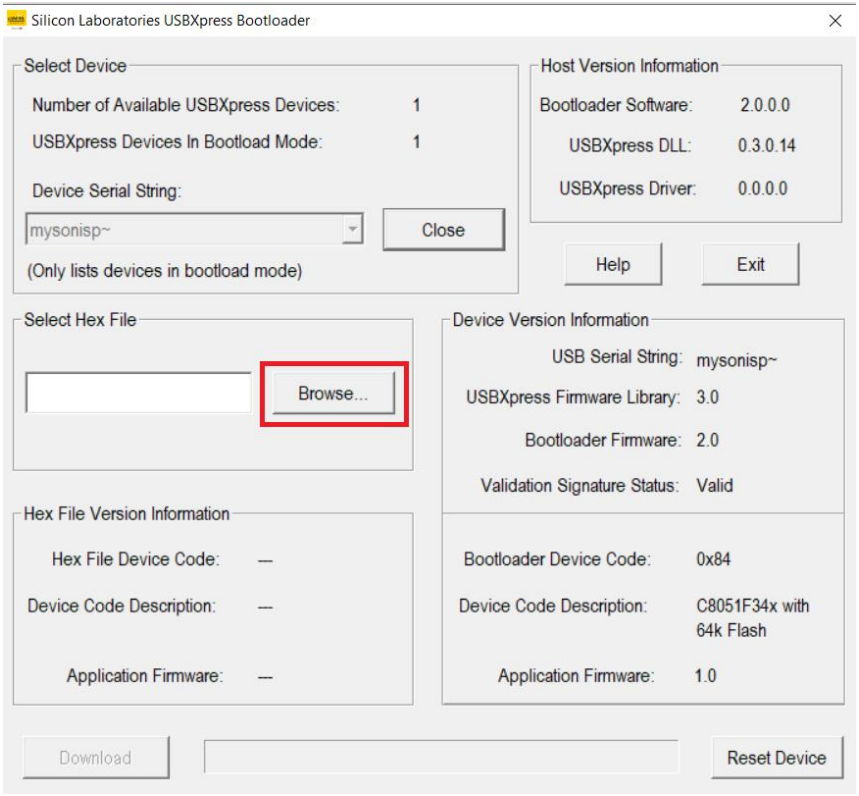
Step 3: Go to "**Tool(T)**" –> "**Update EzISP BIOS(U)**"



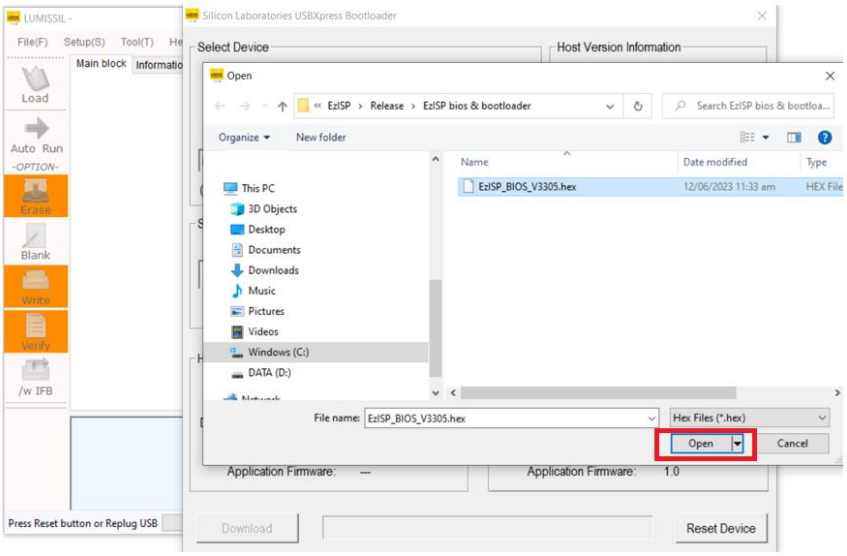
Step 4: Below figure pops out. Click "**Open**" button.



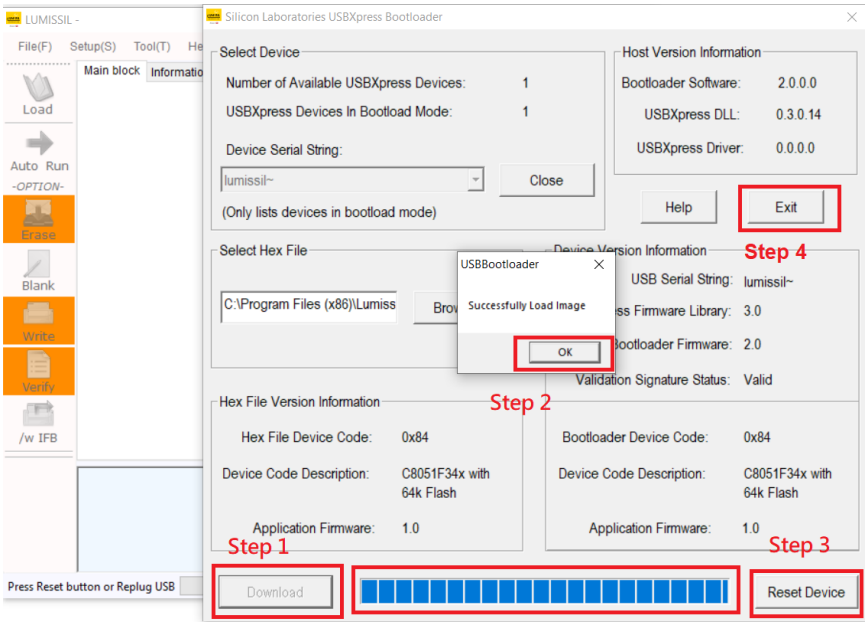
Step 5: Click "**Browse**" button.



Step 6: Below figure pops out. Choose the default pop-out hex file under C:\Program Files (x86)\Lumissil\EzISP\Release\EzISP bios &bootloader and click "**Open**" button.



Step 7: Follow the four steps in the below figure to load Bootloader. Reset the EzISP device to start operation.



2.2 Start up the GUI with EvB

Three-step setup (Please refer to **Figure 1-** IS31SE5120A Evaluation Board Connection Block Diagram.)

1. Connect the USB cable between the USB port of the EzISP Board and the USB port of the PC.
2. Connect a 10-pin 2x5 Socket-Socket 2.54mm IDC cable from the connector on the EzISP Board to the connector J5 of the IS31SE5120A Evaluation Board.

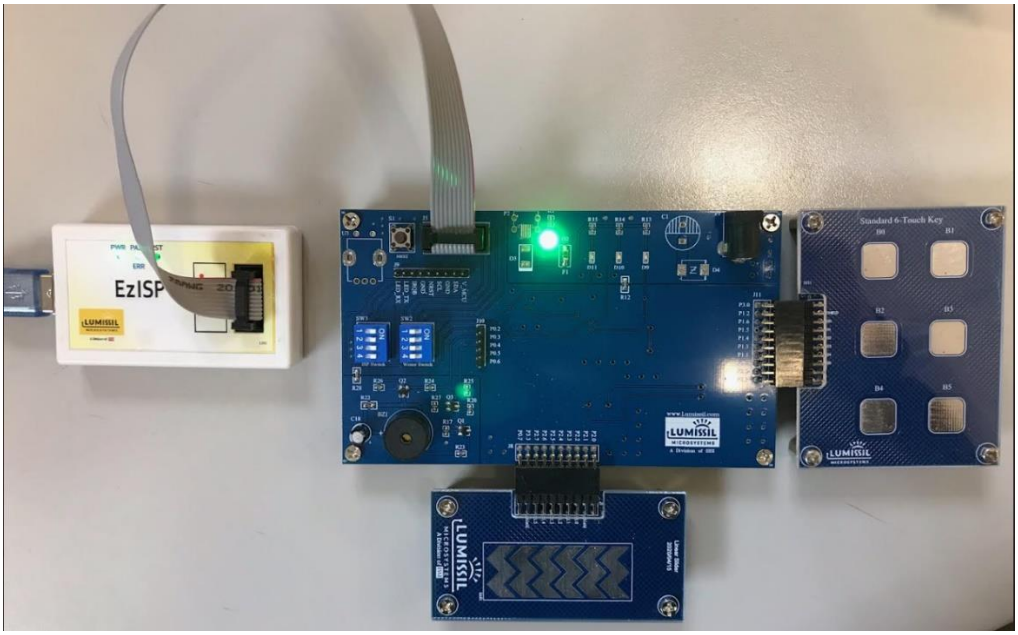


Figure 2-2: Connection of IS31SE5120A Evaluation Board and EzISP Board

3. Execute SE5120A Touch Key GUI program (file name: MC17_SE5120A_GUI_rev2.0.exe).

Current IS31SE5120A EvB board is set at I2C address 0x78.

Table 2: I2C slave address definition

I2C Slave Address*	Write Command	Read Command
	0x78	0x79

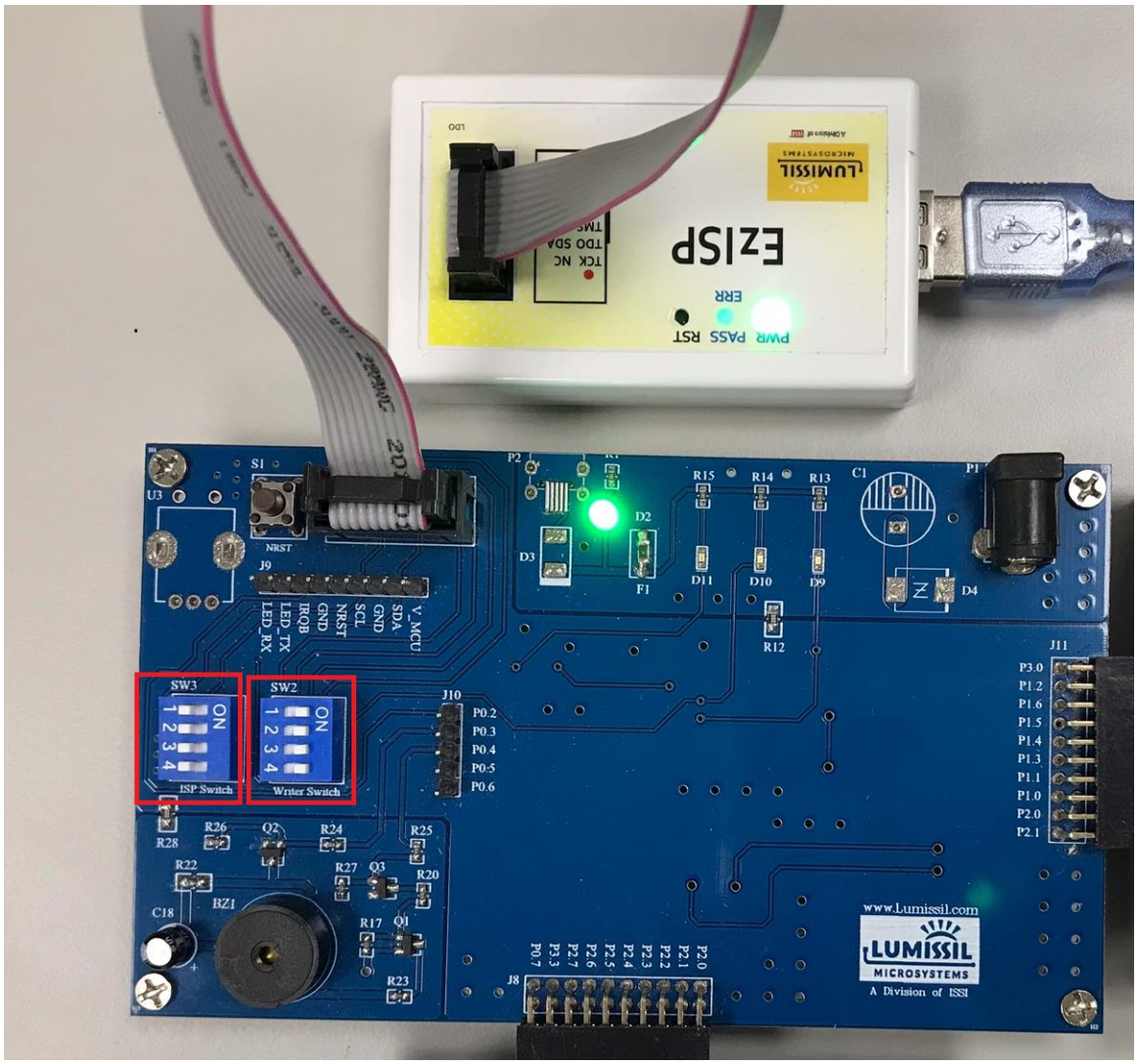
* I2C Slave Address: 7-bit Address + 1 bit(R/W)

2.3 Upgrade Boot Code and Standard Code

This section can help users to upgrade the Boot code and Standard code of SE5120A EvB. In case, Boot code or Standard code of SE5120A EvB is ruined, users can follow the below 4 steps to program code to flash.

Step 1: Turn DIP switch SW2 ON and SW3 OFF

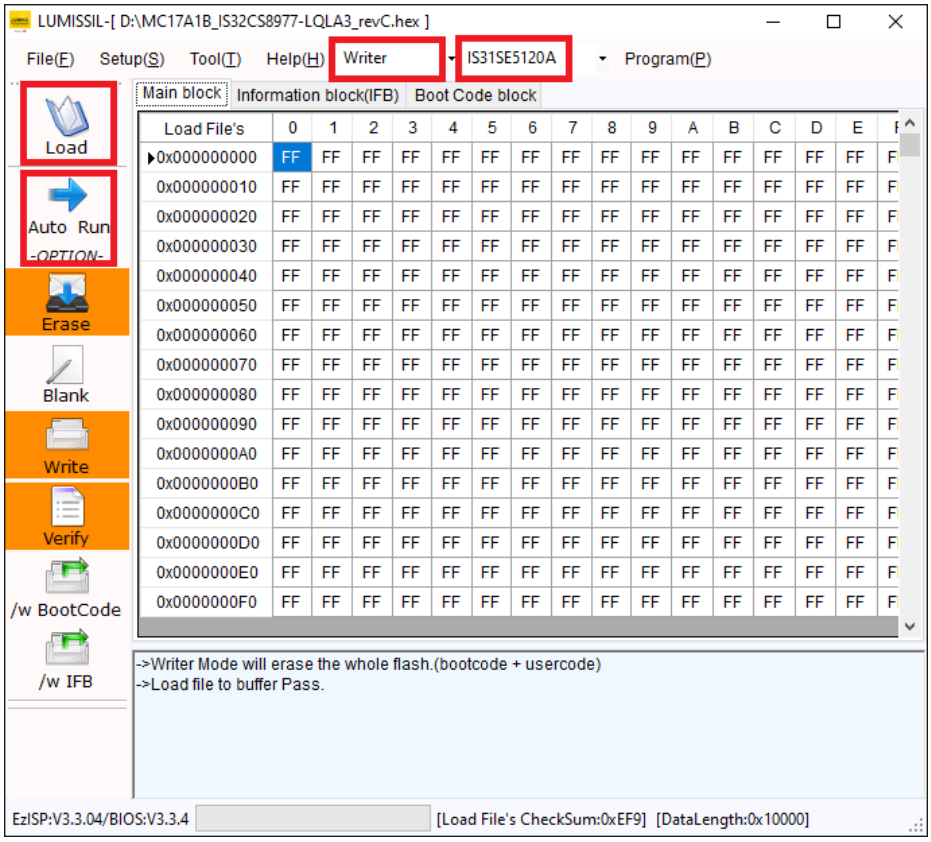
Turn the DIP switch SW2 1/2/3/4 to ON and switch SW3 1/2/3/4 to OFF as the highlighted red rectangles in below SE5120A EvB. SW2 is for writer mode control and SW3 is for ISP mode control.



Step 2: Program boot code

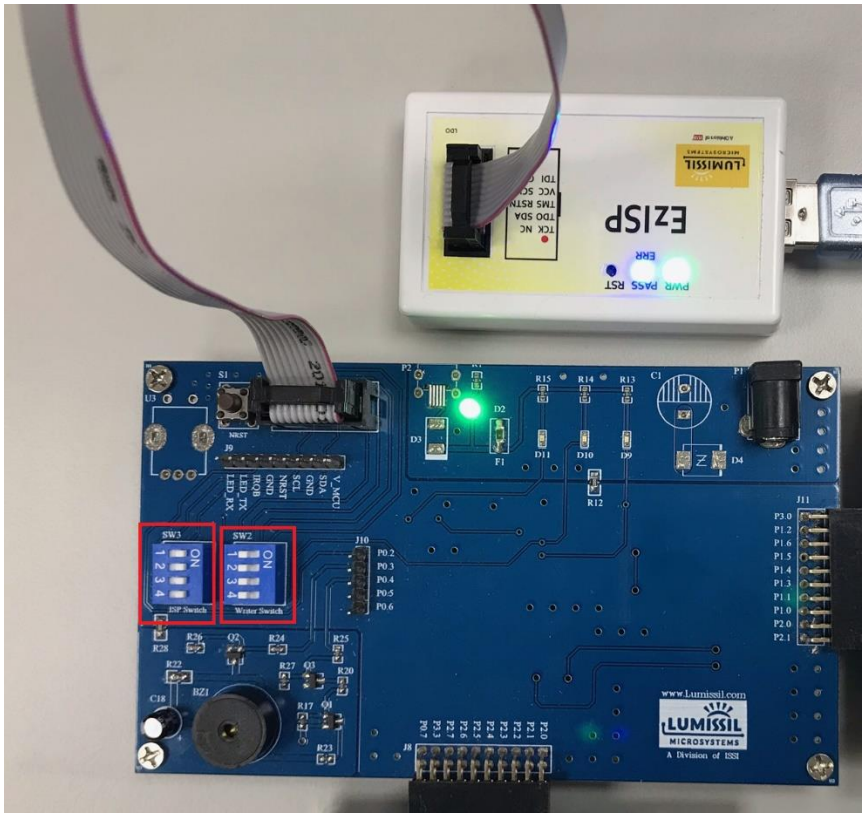
Program boot code " MC17A1B_IS32CS8977-LQLA3_revC.hex" to IS31SE5120A EvB

a. Execute the EzISP Tool
b. Select "**Writer**" and "**IS31SE5120A**" from the drop-down menu as the highlighted red rectangle in below figure
c. Click "**Load**" to load boot code file "MC17A1B_IS32CS8977-LQLA3_revC.hex"
d. Click "**Auto Run**"
e. Close the EzISP program



Step 3: Turn DIP switch SW2 OFF and SW3 ON

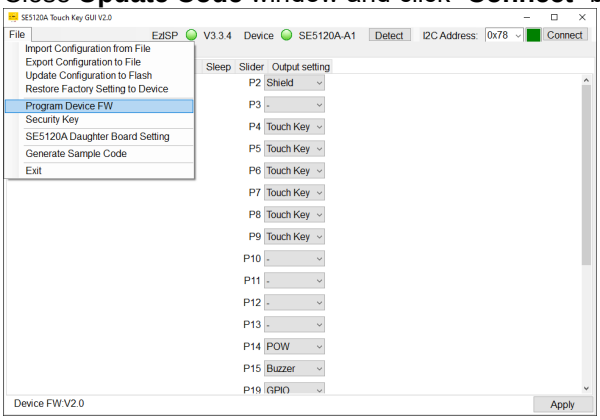
Turn the DIP switch SW2 1/2/3/4 to OFF and SW3 1/2/3/4 to ON as the highlighted red rectangles in below SE5120A EvB.

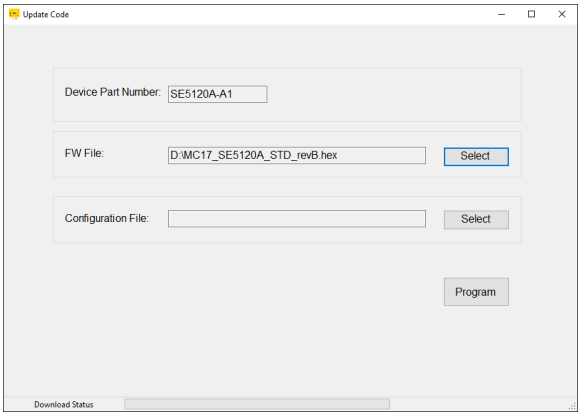


Step 4: Program Firmware

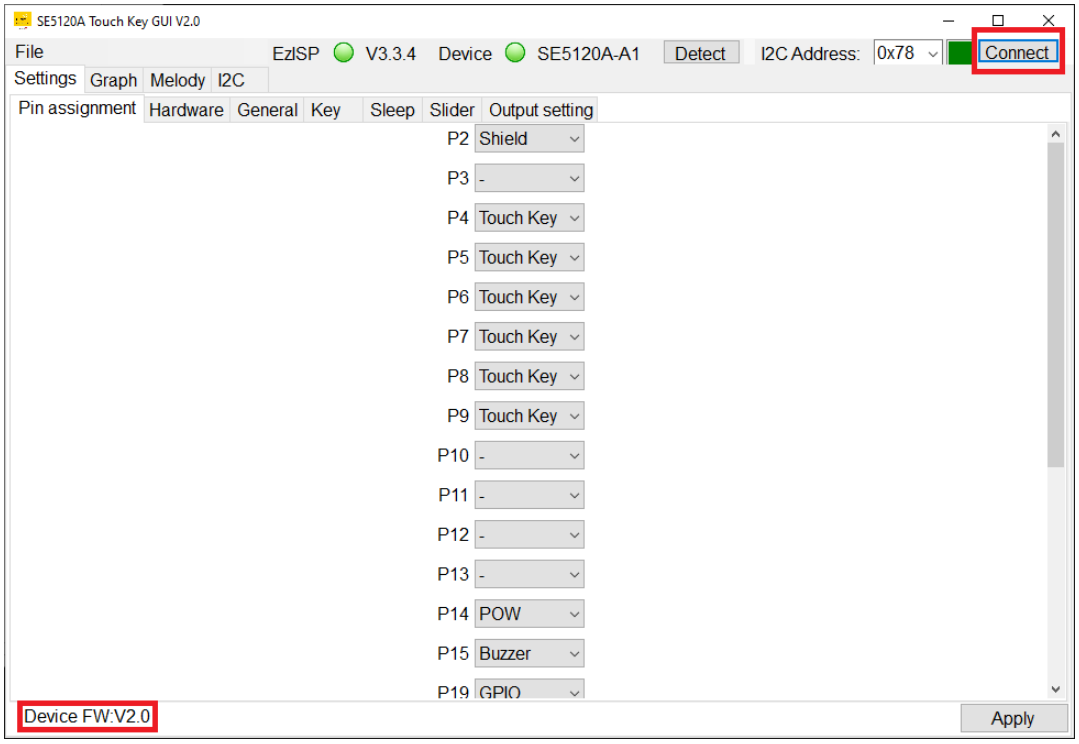
Program Device Firmware into SE5120A EvB

a. Run SE5120A GUI Tool V2.0
b. Click the **Program Device FW** from the **File** drop-down menu
c. Click "**Select**" to open firmware file "MC17_SE5120A_STD_revB.hex" and click "**Program**"
d. Close **Update Code** window and click "**Connect**" button





e. Connect Status will be displayed as "**Connect**" with a green box and this GUI will display the current firmware version after clicking "**Connect**" button.


SE5120A Touch Key GUI V2.0
File
EzISP
V3.3.4
Device
SE5120A-A1
Detect
I2C Address:
0x78
Connect
Settings
Graph
Melody
I2C
Pin assignment
Hardware
General
Key
Sleep
Slider
Output setting
P2 Shield
P3 -
P4 Touch Key
P5 Touch Key
P6 Touch Key
P7 Touch Key
P8 Touch Key
P9 Touch Key
P10 -
P11 -
P12 -
P13 -
P14 POW
P15 Buzzer
P19 GPIO
Device FW:V2.0
Apply

3. SE5120A TOUCH KEY GUI INTERFACE

3.1 Before Use

IS31SE5120A EvB board should have standard code firmware inside. In case there is no standard code firmware inside, users can follow below step 1 to step 3 to program firmware to start evaluation of this board.

Step 1. Click the **Program Device FW** from the **File** drop-down menu.

Step 2. Click "**Select"** to open firmware file, and then click "**Program"**. If **Configuration File** is not selected, then the default settings will be programmed. Users can click "**Export Configuration to File**" to save the current configuration and choose the preferred configuration file before performing "**Program Device FW**".

Step 3. Close the "**Update Code**" window and click "**Connect**" button if the "**Connect**" status is not green. (If IS31SE5120A EvB kit is not connected to the PC before running this software, "**Connect**" status will be with a red box after running this software.)

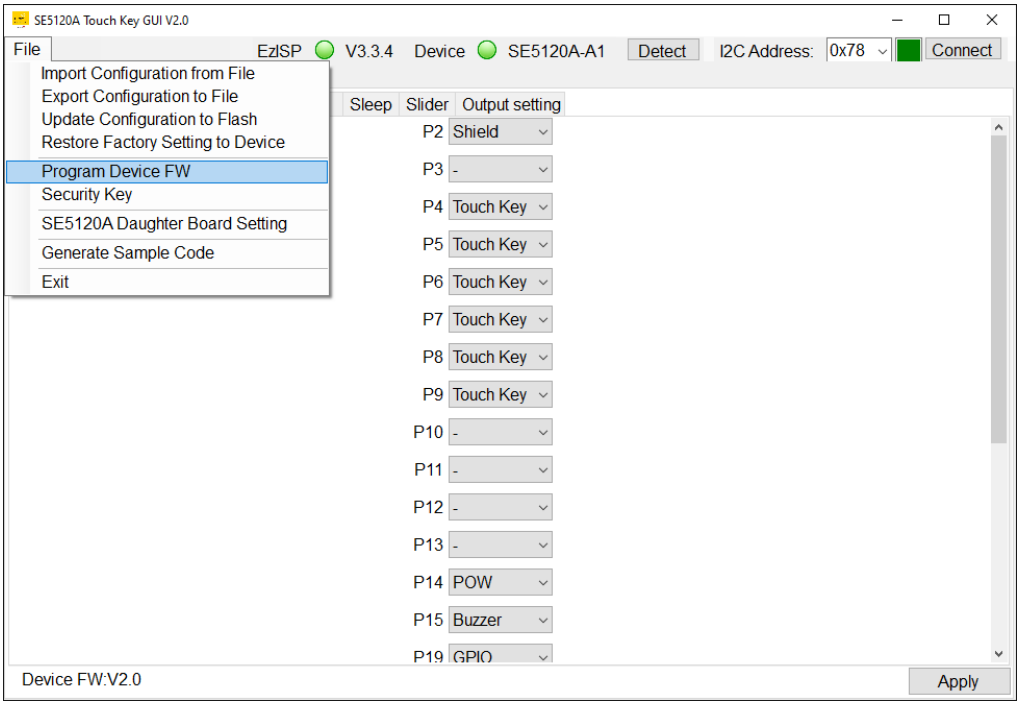


Figure 3-1: Program Device FW Before Use

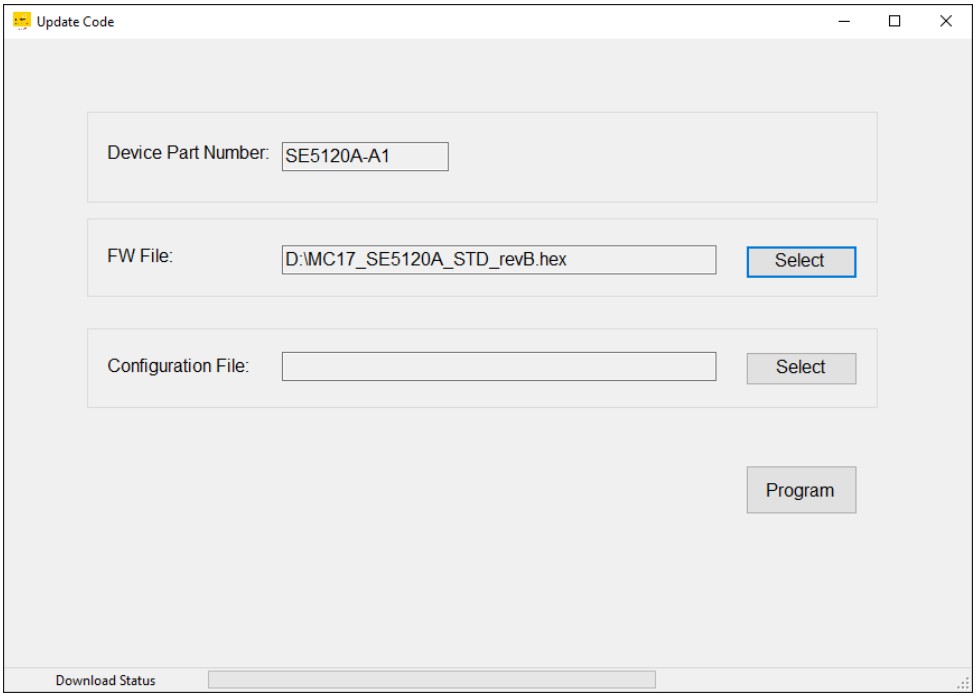


Figure 3-2: Firmware update Code

If the GUI looks small, the following method can improve high-DPI support for the GUI. Users need to find file MC17_SE5120A_GUI_rev2.0.exe, right-click on it and select **Properties**.

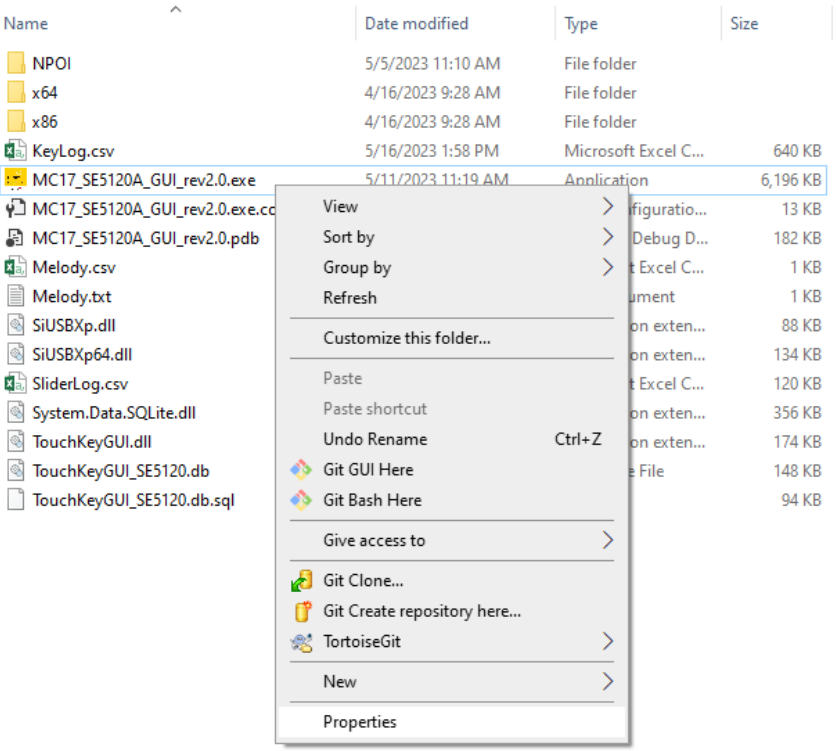


Figure 3-3: Set GUI program Properties

Go to the **Compatibility** tab and click **Change high DPI settings**. Select **Override high DPI scaling behavior. Scaling performed by: System or System (Enhanced)** and click **OK**.

Restart the GUI to see if there is an improvement.

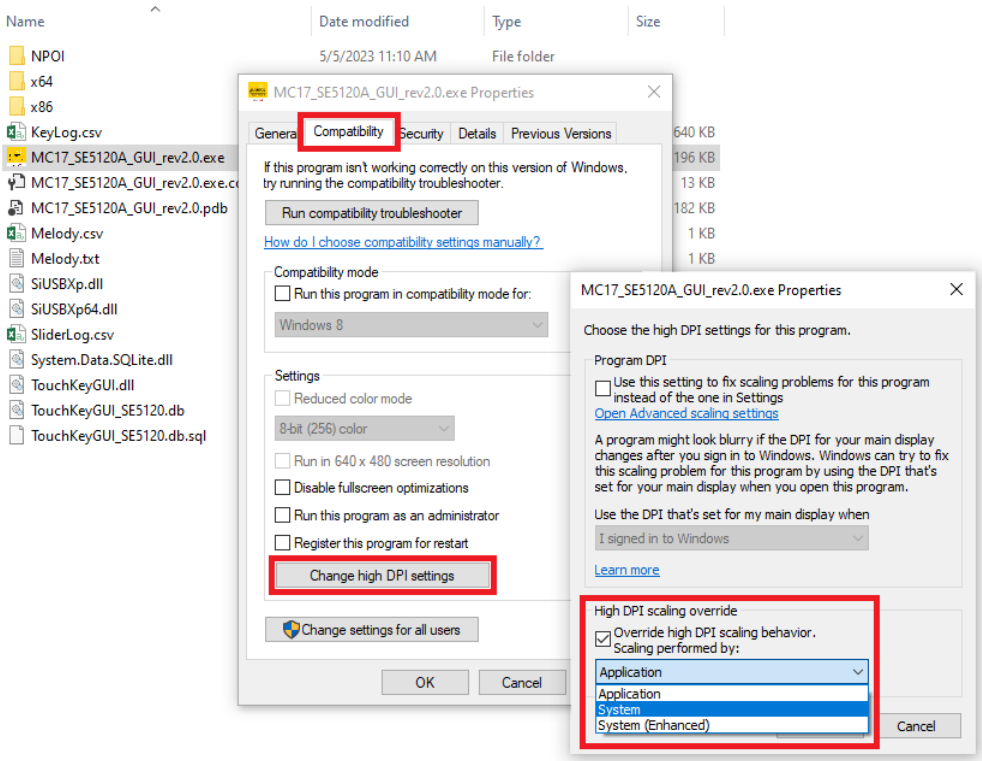


Figure 3-4: Choose High DPI settings

3.2 Connect Status

The I2C salve address is fixed at 0x78, as shown in **Figure 3-5**: GUI Connect Status Shows Disconnect. with IS31SE5120A EvB connected to PC via EzISP board. Users need to click the "**Connect"** button for normal operation. If the user finds the "**Device"** status in red, please try to re-connect the device. connection with PC USB port and run Touch Key GUI tool again. If the "Device" status still shows red, it might be caused by the boot code not being correctly burned. If the "**Connect"** status is still in yellow color after trying to re-connect devices connection with PC USB port and running Touch Key GUI tool again, it might be caused by standard code not being correctly burned.

The Connect Status might be displayed as "**Connect**" with red box and cannot change any setting if IS31SE5120A EvB kit is not connected to the PC before the GUI software is run. Users can click "**Connect**" button and the status will turn green.

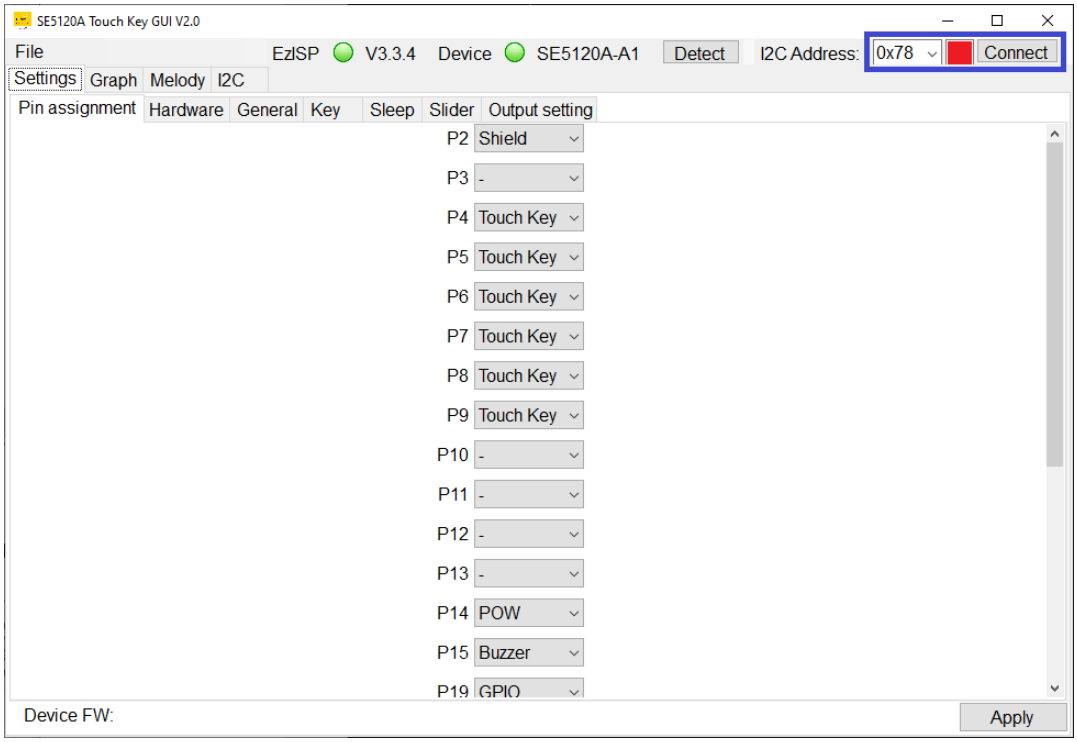


Figure 3-5: Connect Status Shows Disconnect

As shown in **Figure 3-6** below, the Connect Status will be displayed as "**Connect**" (with a green box) and display current firmware version after clicking "**Connect**" button.

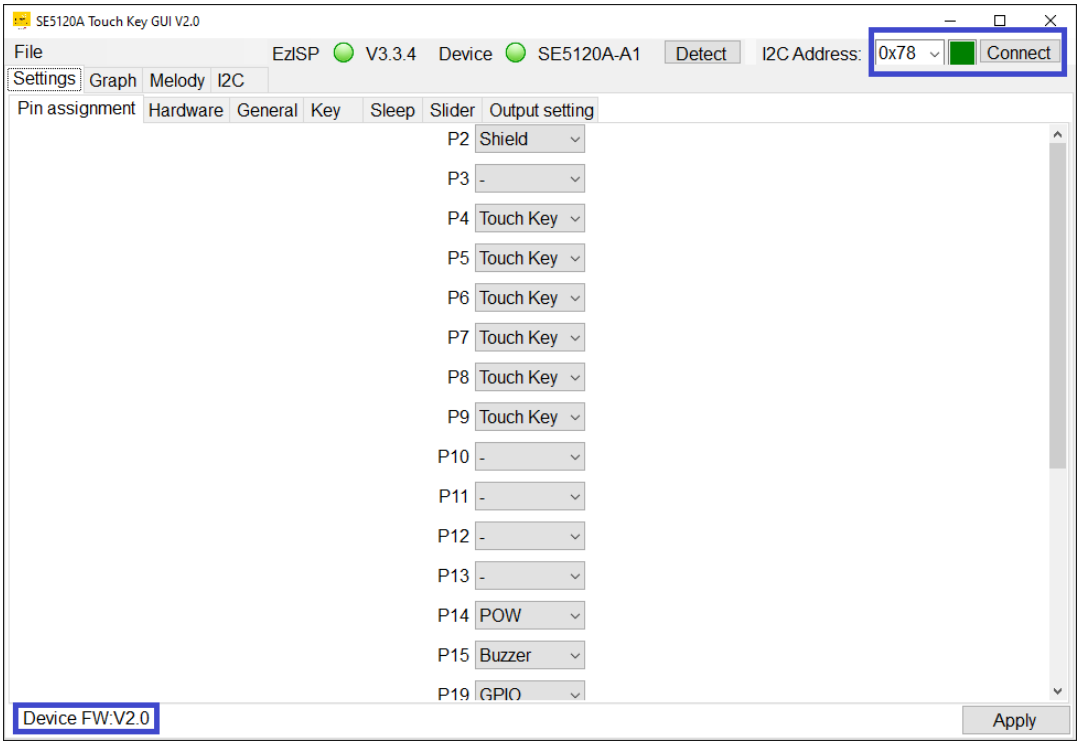


Figure 3-6: Connect Status Shows Connect

3.3 Export/Import Configuration File

As shown in **Figure 3-7** below, **Import Configuration from File** is used to import the configuration setting from the preferred IS31SE5120A GUI settings file into the device. Click **Apply** to apply the parameters to the device.

Export Configuration to File is used to export the configuration of IS31SE5120A and save in a file. By exporting the configuration, we can save the parameters settings in the GUI to a file (The default file name is an excel file "register values" and users can define file name for version control). It also simultaneously creates a BIN file in the same folder and the file can be used for future manufacture.

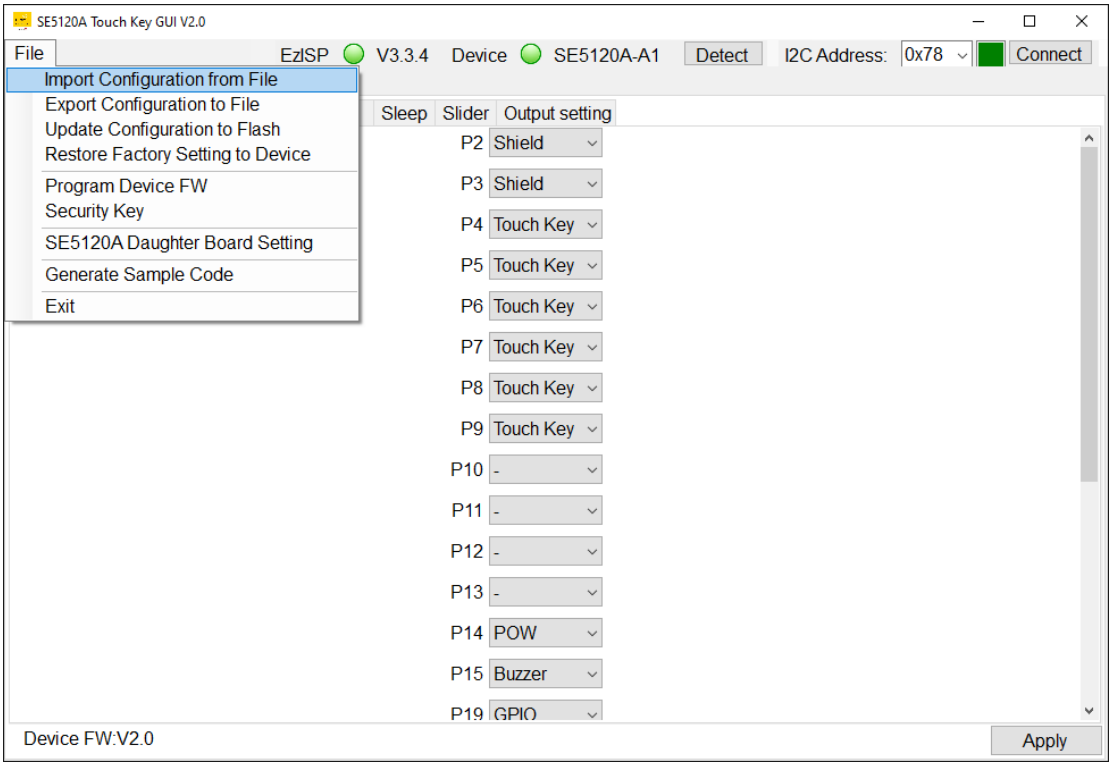


Figure 3-7: Import/Export Options

3.4 Update Configuration to Flash

As shown in **Figure 3-8** below, users can write the parameters settings in the GUI into the flash memory of the IS31SE5120A chip. After clicking the **Update Configuration to Flash** from the **File** drop-down menu, writing configuration settings to the flash memory can be completed in about 4 seconds.

The embedded Flash Memory has the capability to hold the saved data even when the power is off. When IS31SE5120A is powered on again, the parameters previously written to the flash memory will become the default values.

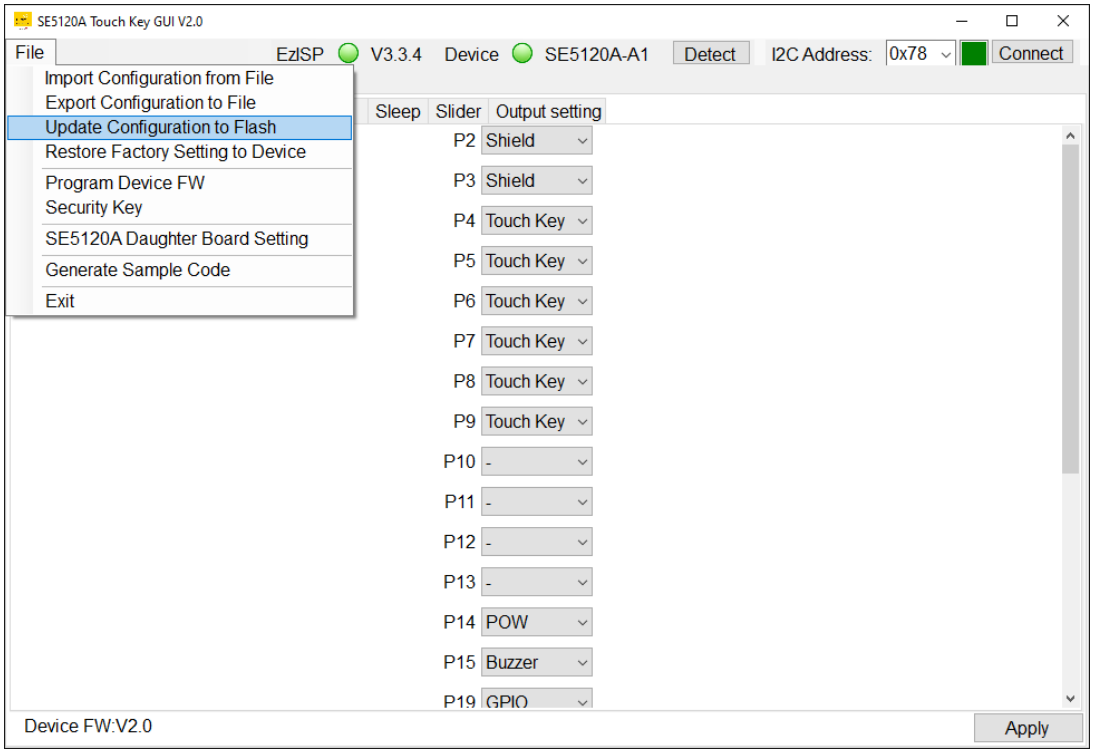


Figure 3-8: Update Configuration to Flash

3.5 Restore Factory Setting to Device

As shown in **Figure 3-9** below, after clicking the **Restore Factory Setting to Device** from the **File** drop-down menu, writing factory default parameters to the flash memory can be completed in about 4 seconds.

Need to click "**Connect**" button to refresh parameters in the GUI.

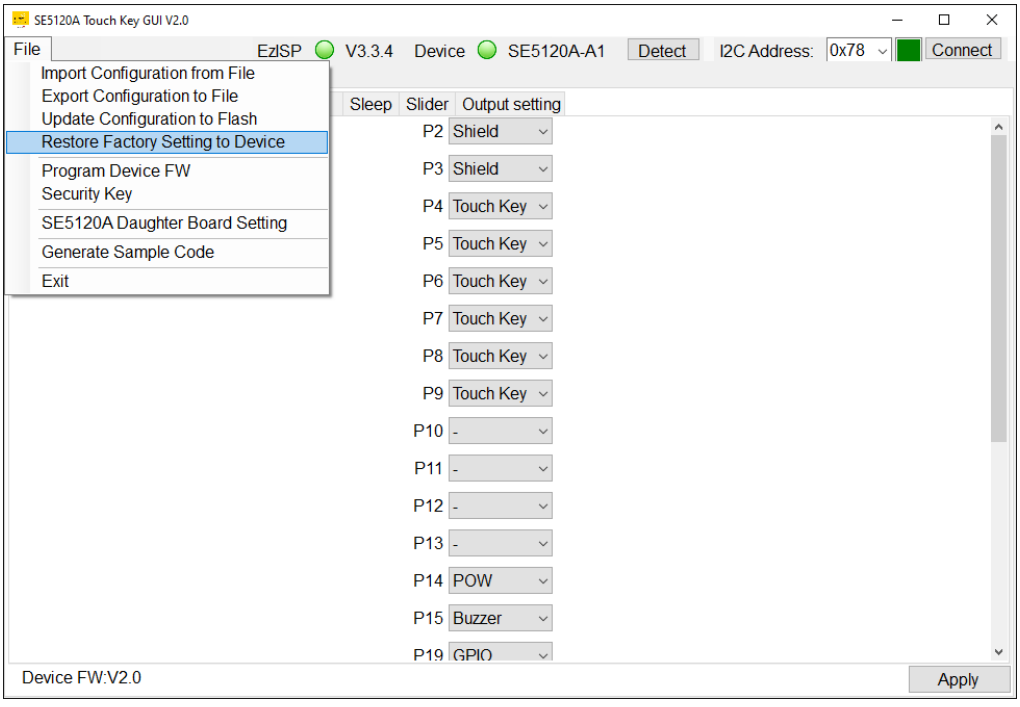


Figure 3-9: Restore Factory Setting to Device

3.6 Program Device Firmware

After users click the **Program Device FW** from the **File** drop-down menu, there comes out a pop-up window as shown in **Figure 3-10** below. Click **Select** to open firmware file and configuration file. Configuration will be restored to the factory settings if configuration file is not selected. Click **Program** to update firmware code.

Need to click **Connect** button to reconnect device after programming device firmware.

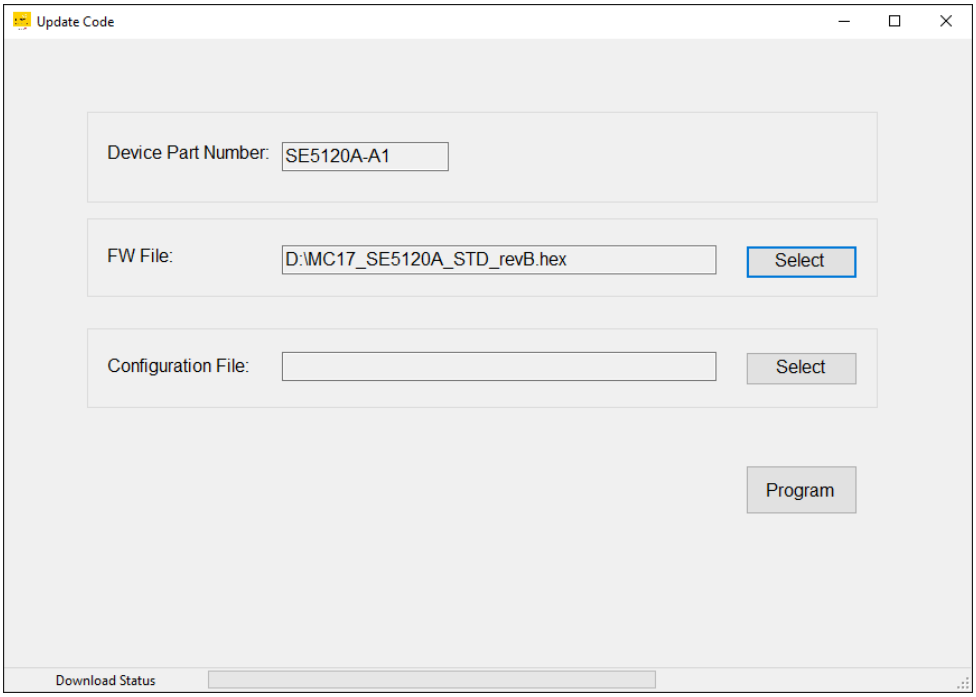


Figure 3-10: Update Code

3.7 Security Key

After clicking the **Security Key** from the **File** drop-down menu, a pop-up window will appear as shown in **Figure 3-11** below.

There are 8-bytes **Lock/Unlock Key** and the default setting is 0xFFFFFFFFFFFFFFFF.

Lock Key will be updated with the programmed device firmware.

The security key is used to protect user's defined device parameters. Invalid security key inhibits reading the chip information, but programming new FW is allowed, i.e., users can reset the security key by programming a new FW.

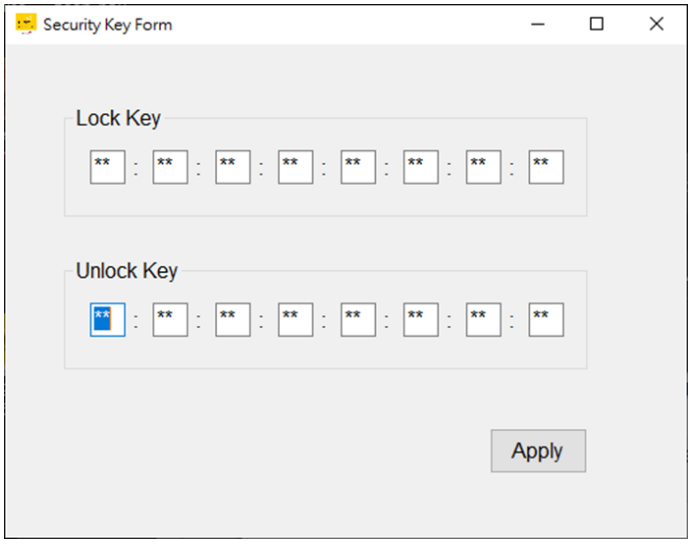


Figure 3-11: Security Key

3.8 Generate Sample Code

As shown in **Figure 3-12: Generate Sample Code** below, "**Generate Sample Code"** is used to create a sample code for configuring SE5120A via I2C bus. Customers can adjust SE5120A Touch Key GUI for the required parameters to meet their own application needs. After system level verification, please click **Apply** to apply the parameters to the device before clicking **Generate Sample Code**.

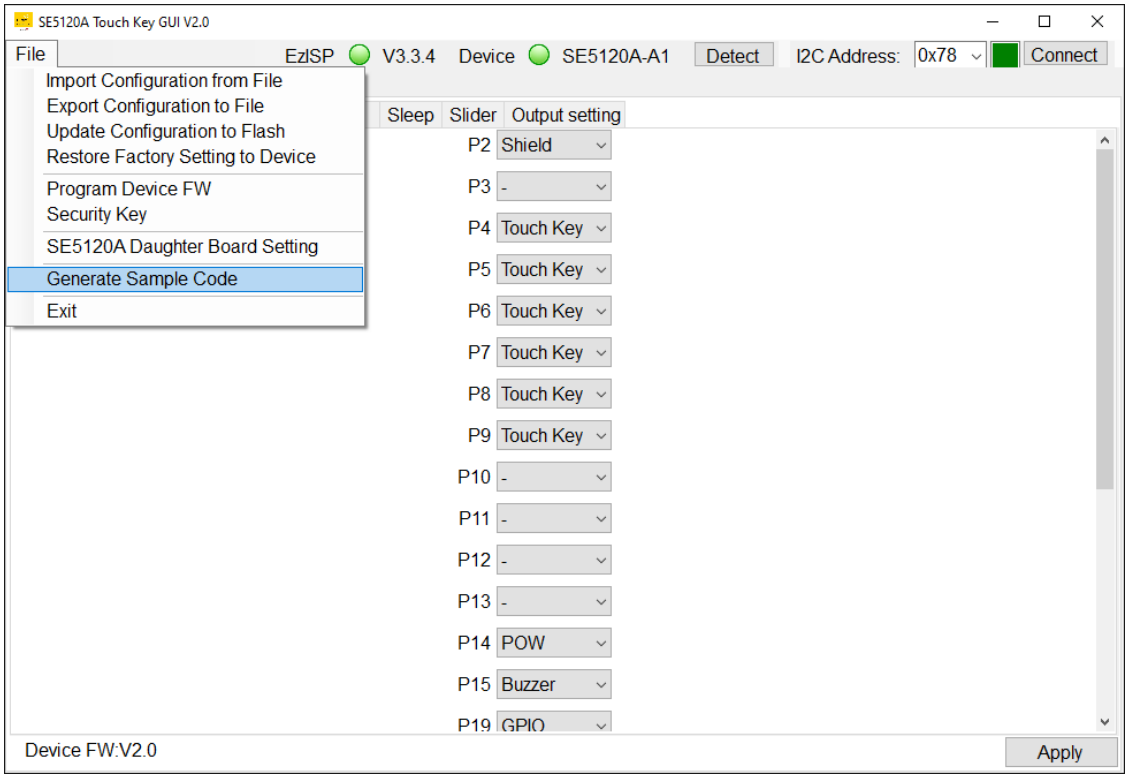


Figure 3-12: Generate Sample Code

3.9 How to Use IS31SE5120A Touch Key GUI

The following steps explain how to use IS31SE5120A Touch Key GUI. **We recommend using a round copper bar instead of a human finger for the touch key parameters testing and setting.**

Step 1. Run MC17_SE5120A_GUI_rev2.0.exe and click **Connect** button to establish connection.
Parameters will be uploaded from the device.

Step 2. Click **Import Configuration from File** from **File** menu and click **Apply** to apply the parameters to the device (optional).

Step 3. Follow Section 4. SETTINGS to configure setting of pins, hardware, key parameters and so on.

Step 4. Click **Apply** to apply the parameters to device and check the result by **Graph** or other methods.

Step 5. Click **Update Configuration to Flash** from **File** menu if users want to use these parameters as device default setting or click **Export Configuration to File** from **File** Menu to save current GUI parameters.
"**Export Configuration to File**" also creates a BIN file in addition to saving a user-defined csv file in the same folder for future manufacture.

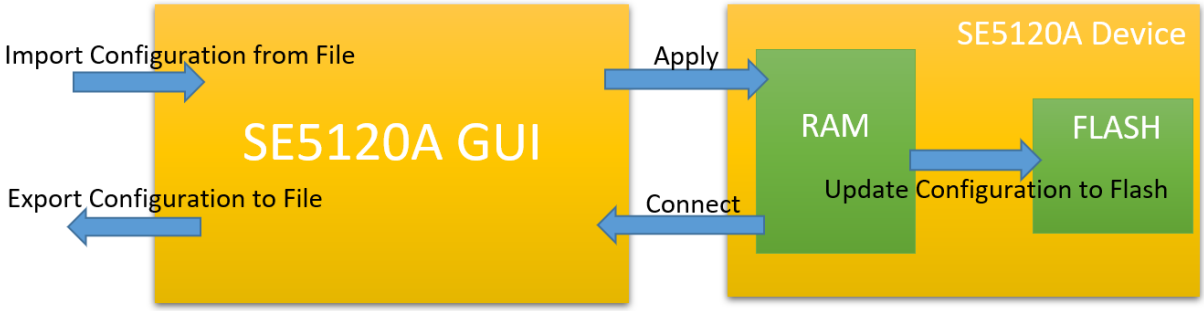


Figure 3-13: Interaction between GUI Configuration and Device

4. SETTINGS

4.1 Pin Assignment

As shown in **Figure 4-1: Default Pin Assignment** below, after clicking **Connect** button, the Connect Status shows "**Connect**" with a green box and the EvB board has been successfully connected. Users can choose different pin assignments from the drop-down menu.

Users must click **Apply** button located at the right bottom corner of below figure to submit selected settings.

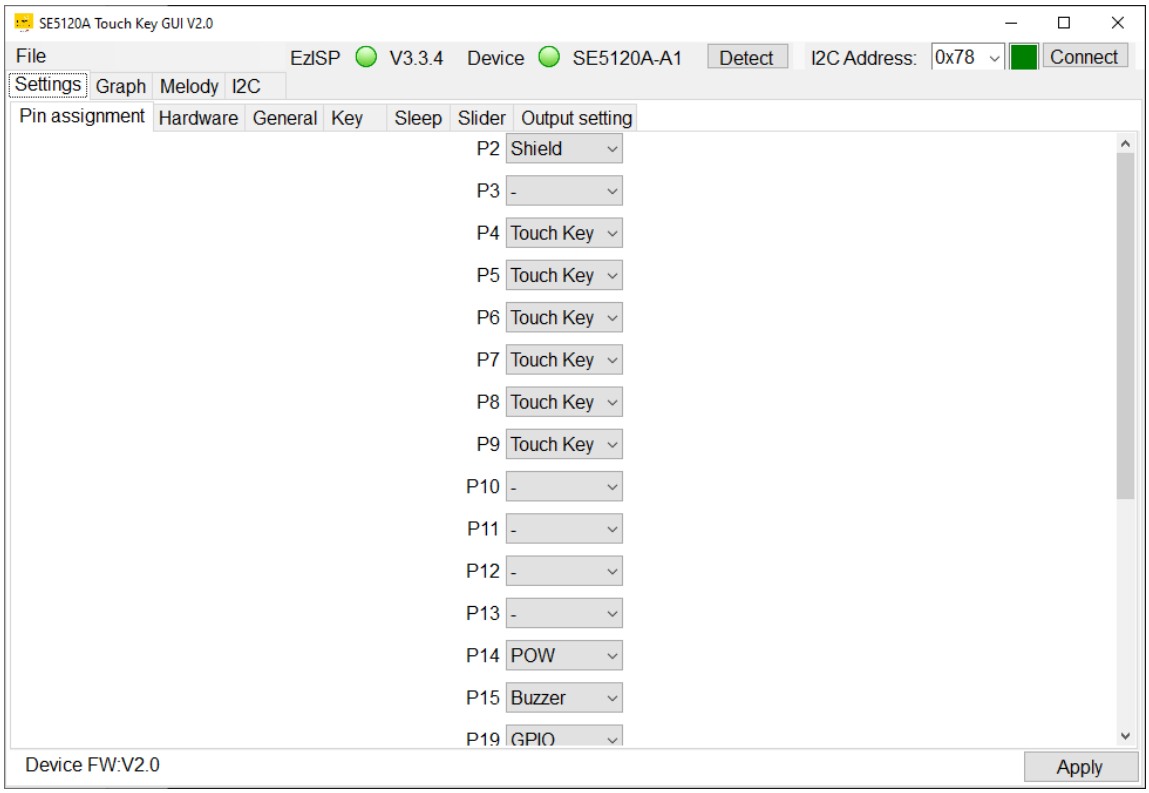


Figure 4-1: Default Pin Assignment

Touch Key : Touch Sense Channel
Shield : Shield Output
Buzzer : Buzzer Output
POW : Melody Power Output
GPIO : General Purpose Input/Output

Figure 4-2: Pin Drop-Menu

4.2 Hardware Setting

As shown in **Figure 4-3: Hardware Setting Page** below, TK3 stands for Touch Key technology generation III for Lumissil. TK3 uses dual-slope technology to detect key touch via charging and discharging among an internal charge capacitor, an external reference capacitor and the Touch Key capacitor. Several parameters for the TK3 can be set to improve touch key detection. All settings will be applied by clicking the **Apply** button.

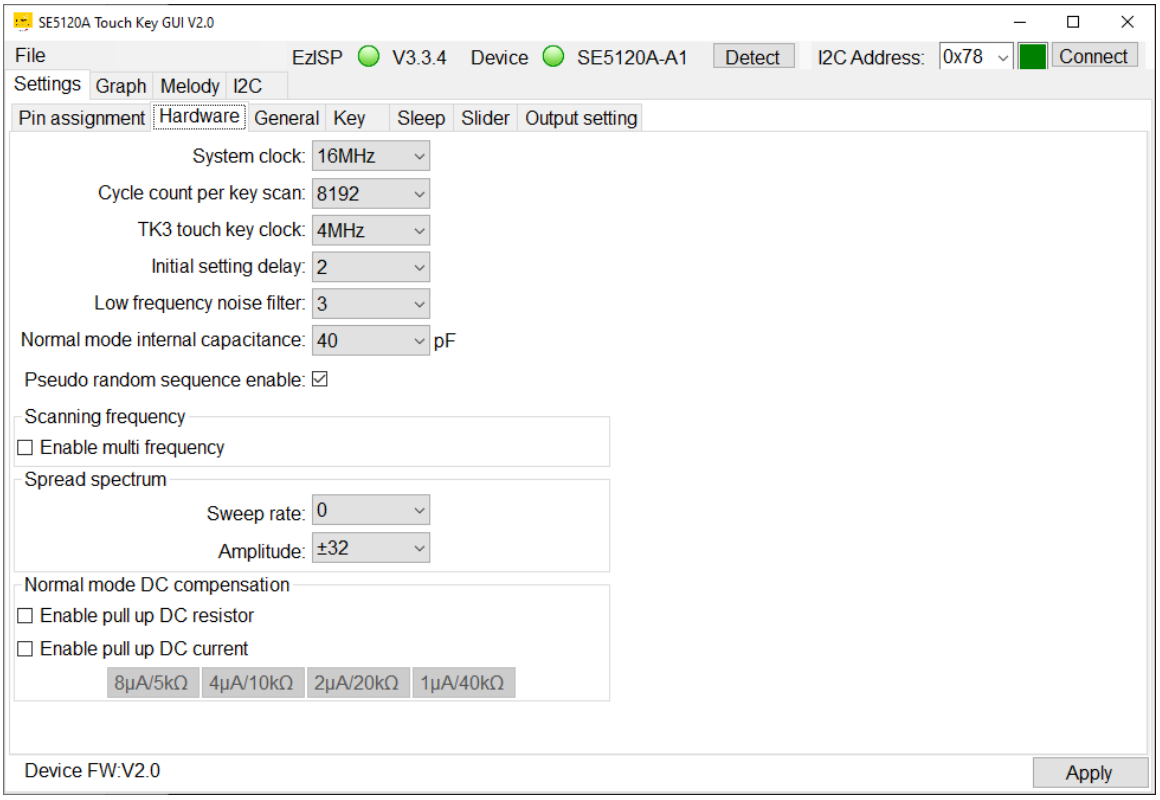


Figure 4-3: Hardware Setting Page

4.2.1 System Clock

Select **System clock** to change device's clock and the default setting is 16MHz. The lower system clock may cause I2C communication error.

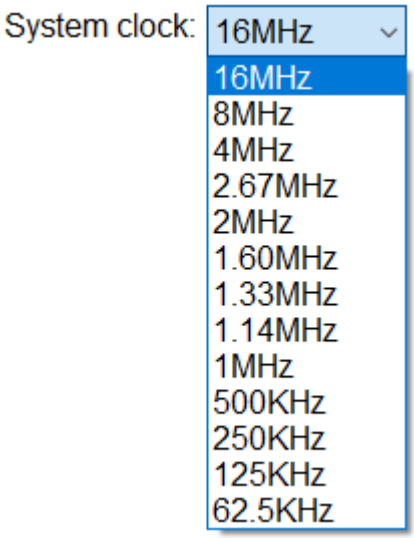


Figure 4-4: System Clock

4.2.2 Cycle Count Per Key Scan

For example: Cycle Count = 8192, total keys = 5, TK3 touch key clock = 4MHz

One scan cycle = (1/4M)* 8192* 5 = 10.24ms

Higher cycle count per key scan will increase key's resolution for detect, but it will have a longer scan time. The default setting is 8192. Choosing different cycle count setting, users might need to modify key detect parameters like finger threshold to have correct key detect.

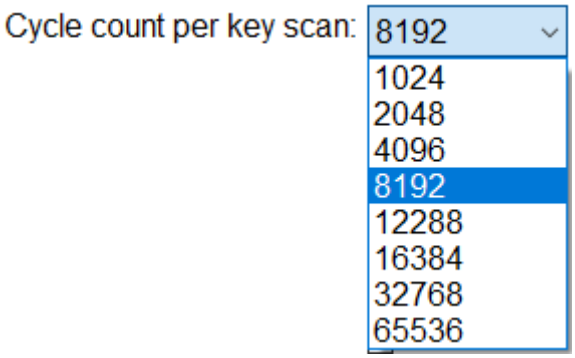


Figure 4-5: Cycle Count Per Key Scan

4.2.3 TK3 Touch Key Clock

Select **TK3 touch key clock** to change touch key clock and the default setting is system clock/4 = 16MHz/4 = 4MHz

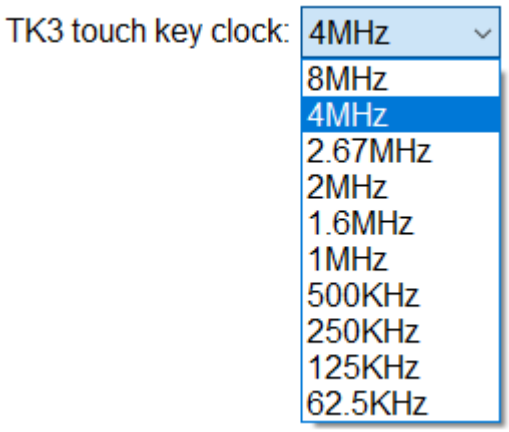


Figure 4-6: TK3 Touch Key Clock

4.2.4 Initial Setting Delay (INI)

Initial setting delay defines the number of TK3 clock (system clock/4) period for initial settling of IS31SE5120A pin Cref (Cref, the integration capacitor of the charge transfer). The delay time is set to (INI + 1) * 4 * TK3 clock time.

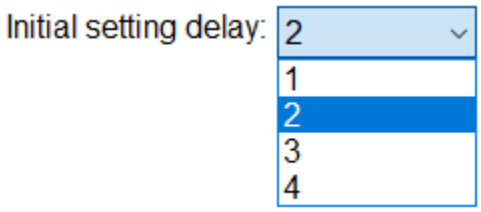


Figure 4-7: Initial Setting Delay

4.2.5 Low Frequency Noise Filter (LFNF)

Regarding low frequency noise rejection, users can use this setting to reject a continuous high or low comparator output that exceeds the setting duration. If the scan count with noise injection detection is larger than (LFNF * 8), the scan result is ignored.

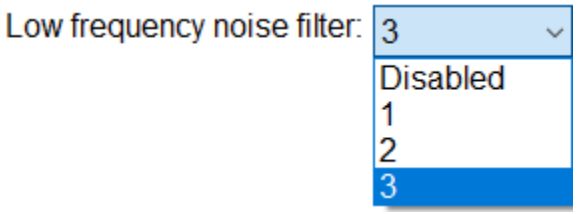


Figure 4-8: Low Frequency Noise Filter

4.2.6 Normal Mode Internal Capacitance

It is suggested to set the charge capacitance value at around twice the capacitance value of touch pad and we can evaluate the setting value by checking the raw count without touching keys.

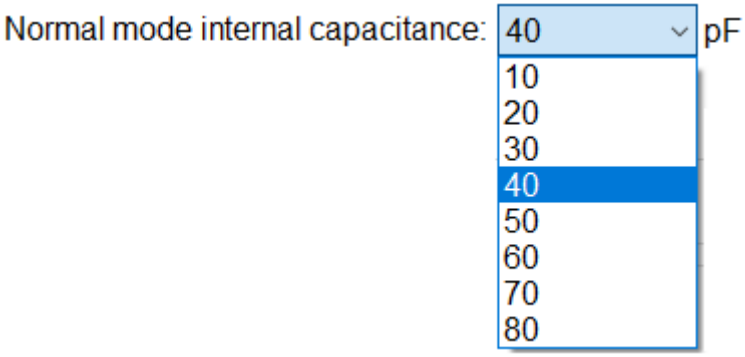


Figure 4-9: Normal Mode Internal Capacitance

4.2.7 Pseudo Random Sequence Enable

IS31SE5120A applies a pseudo-random sequence that randomizes the charge and discharge timing sequences to prevent high frequency noise rejection.

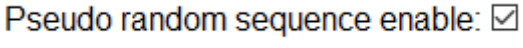


Figure 4-10: Pseudo Random Sequence Enable

4.2.8 Scanning Frequency

Scan one key at three different frequencies to prevent noise, however it will increase scan time.

Figure 4-11: Scanning Frequency

4.2.9 Spread Spectrum

The spread is achieved by varying the output to the oscillator circuit, thus effectively changing the oscillation frequency.

When SS (Spread Spectrum) is enabled, the frequency of IOSC varies according to time and setting, and therefore the accuracy of IOSC frequency cannot be guaranteed.

As Spread Spectrum suggests, the total EMI energy is not reduced, but the energy is spread over a wider frequency. It is recommended that SS usage should be carefully evaluated and the setting of spread amplitude and the sweep rate/frequency should be chosen carefully for reducing the EMI effect.

Sweep rate

Define the spread spectrum sweep rate and the spread spectrum function is disabled when **Sweep rate** = 0.

Amplitude

Setting to adjust the range of spread spectrum frequency. Be careful to select the setting of Sweep rate and Amplitude to reduce EMI.

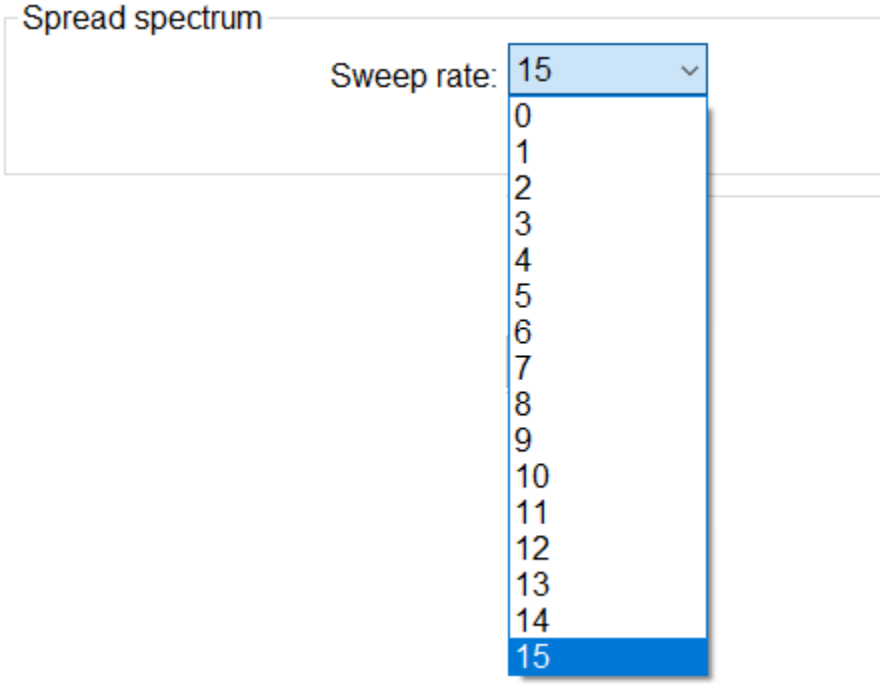


Figure 4-12: Spread Spectrum Sweep Rate

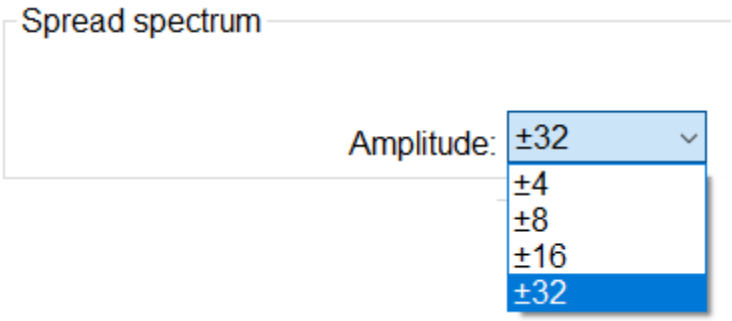


Figure 4-13: Spread Spectrum Amplitude

4.2.10 DC Compensation

The hardware can be configured a constant DC pull-up/pull-down (PUD) on IS31SE5120A pin Cref to allow high capacitance touch-key detection. A DC pull-up/pull-down can compensate for the equivalent resistance caused by a high capacitance key. Connecting a switching current source or a resistor can thus maintain touch key detection sensitivity.

For DC current, PUD can enable 8uA/4uA/2uA/1uA current source. For Resistor, PUD can enable 5K/10K/20K/40K resistor. Users can choose any of the settings 8uA/5KΩ, 4uA/10KΩ, 2uA/20KΩ, 1uA/40 KΩ or a combination of multiple selections at one time.

Figure 4-14: DC Compensation

4.2.11 Raw Count Calibration

Step 1. Select **Cycle Count Per Key Scan** (1024 - 65536) as 4.2.2

Step 2. Select larger **Normal Mode Internal Capacitance** (10pF – 80pF) to increase the raw count value.

Step 3. Enable **DC Compensation** if needed. But it will increase the raw count.

Step 4. Click **Apply** button located at the right bottom corner of the GUI to submit user's setting.

Step 5. Check current raw count/baseline from **GRAPH**.

Step 6. It is good to have Raw count and Baseline count closer to 1/2 of "**Cycle count per key scan**" count. If not, go to step 2.

Example 1: Set Cycle count per key scan = 8192 and compare raw counts for different internal capacitance settings.

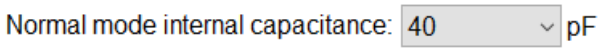


Figure 4-15: Normal mode internal capacitor

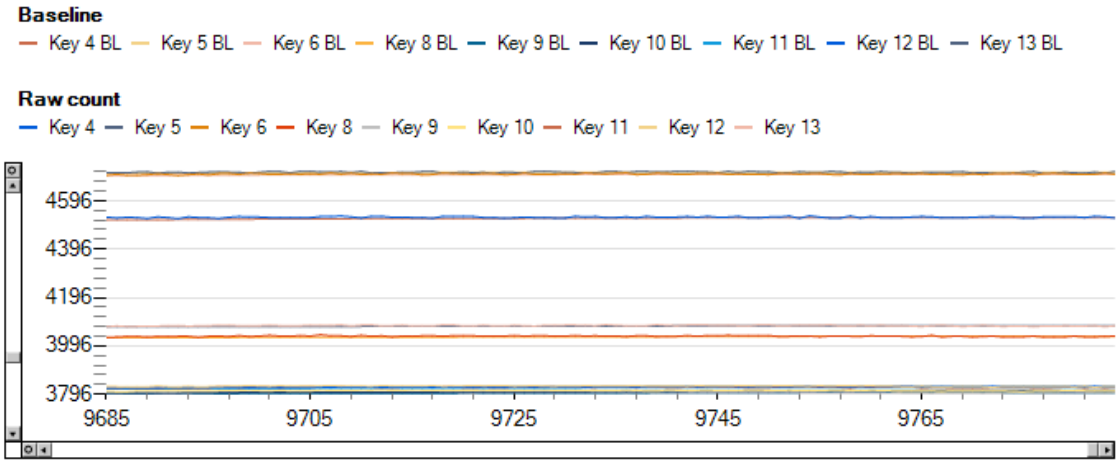


Figure 4-16: Raw Count Value when Internal Cap = 40pF

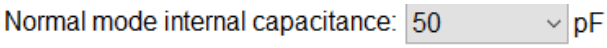


Figure 4-17: Normal mode internal capacitor = 50pF

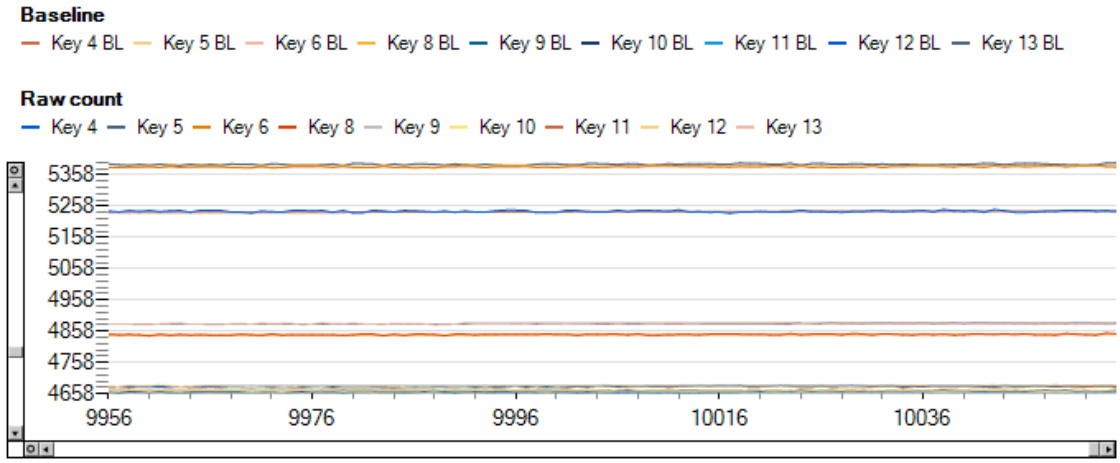


Figure 4-18: Raw Count Value when Internal Cap = 50pF

Example 2: Set **Cycle count per key scan** = 8192 and compare baseline and raw counts for different settings of DC compensation.

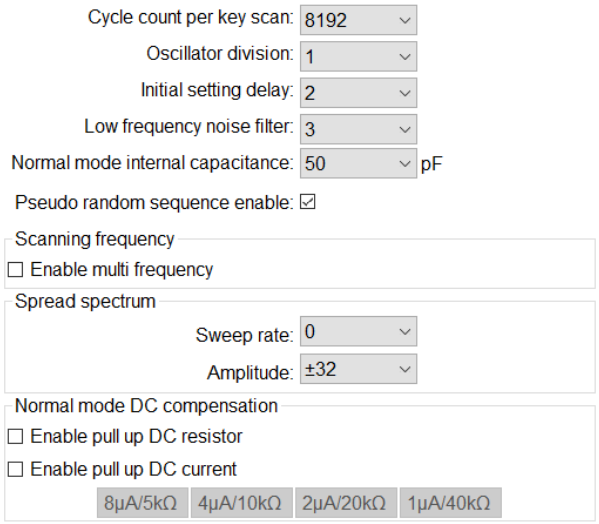


Figure 4-19: Normal mode internal capacitor = 50pF

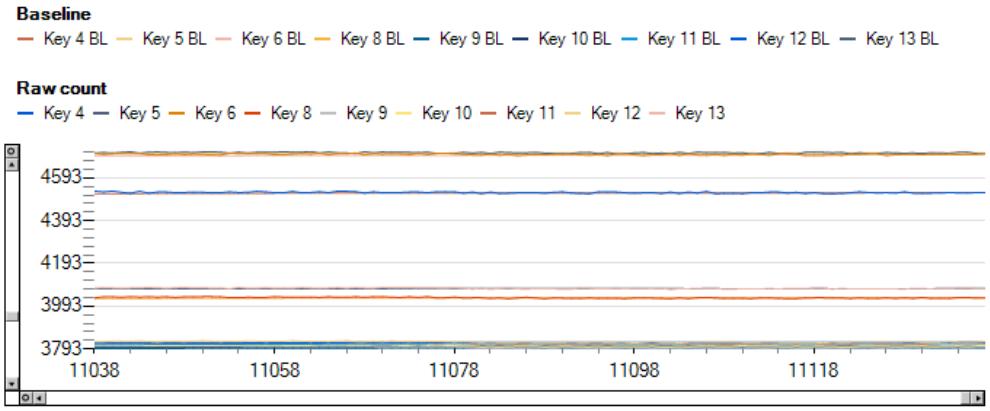


Figure 4-20: Disable DC Compensation

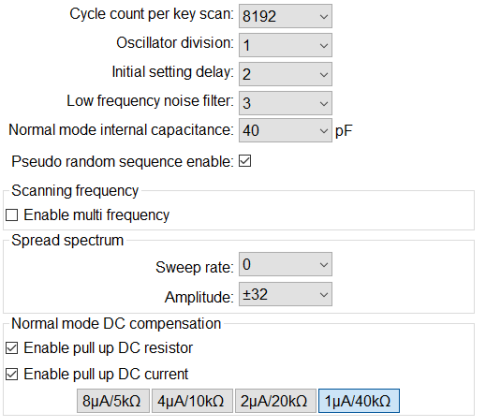


Figure 4-21: DC Current and Resistor setting

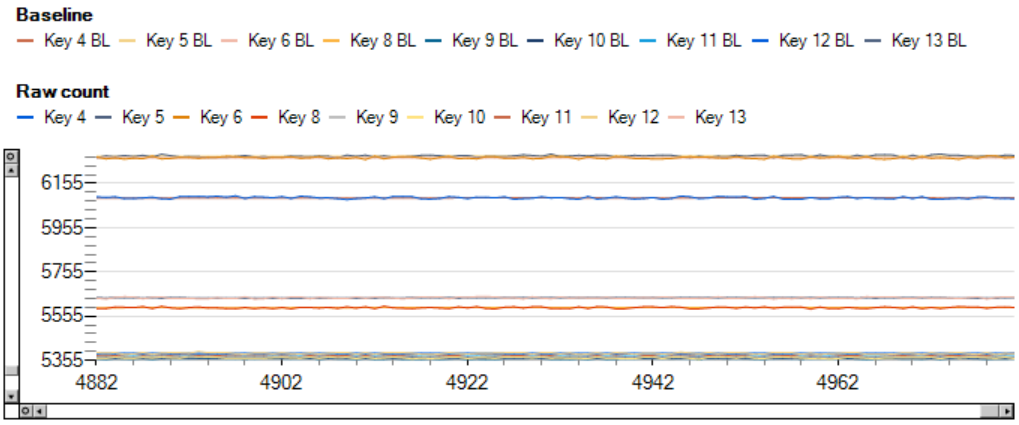


Figure 4-22: Enable DC Current 1uA and Resistor 40KΩ

4.3 General

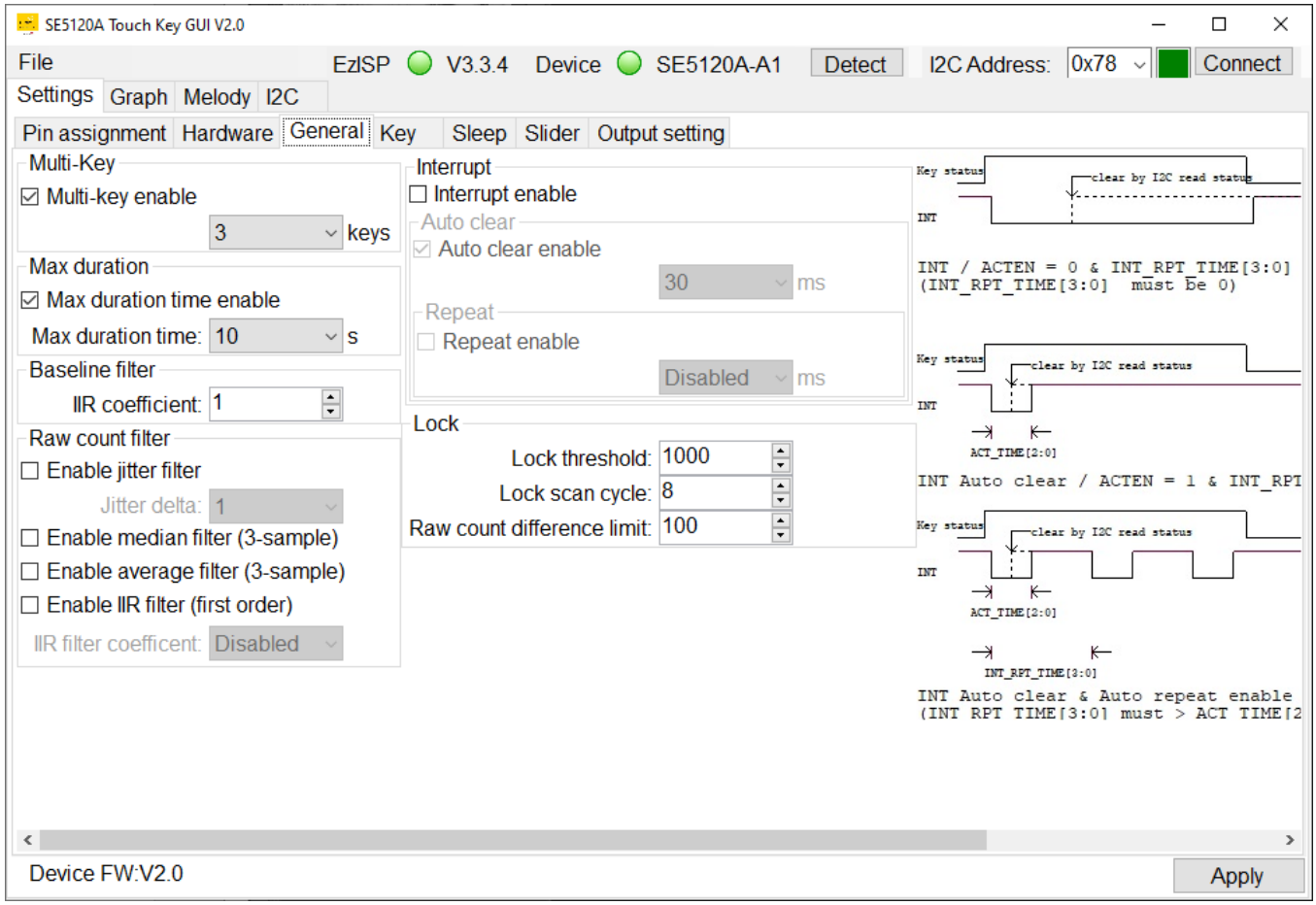


Figure 4-23: General Page

4.3.1 Multi Key

As shown in below **Figure 4-24: Multi-Key**, **Multi-key enable** is to enable multi-key function. When **Multi-key enable** is not checked, only one key is detectable. **Multi-key enable** selection can be set to TWO KEYS, THREE KEYS or All KEYS. In some applications, such as a password lock, the number of keys pressed can be set via this setting. The **Multi-key enable** should be unchecked by default to prevent error on password lock.

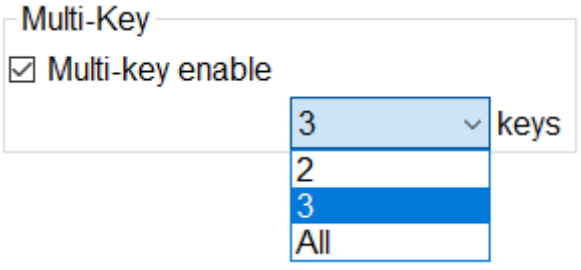


Figure 4-24: Multi-Key

4.3.2 Maximum Duration

Max duration time enable is the maximum pressing duration time setting in seconds. When the pressing time is longer than the maximum duration time, the device will be forced to do calibration automatically on the pressed key.

This function is mainly used to recover a touch key mis-triggered by the environmental factor. For example, a water droplet falls on a touch button and causes a trigger status. That key will stay triggered until a reset occurs and hence not usable. However, if the maximum duration time is set, the mis-triggered key will be turned off since a forced calibration of the baseline will occur after the maximum duration time is expired. Once the forced calibration is done, the touch key with the water droplet can be used again.

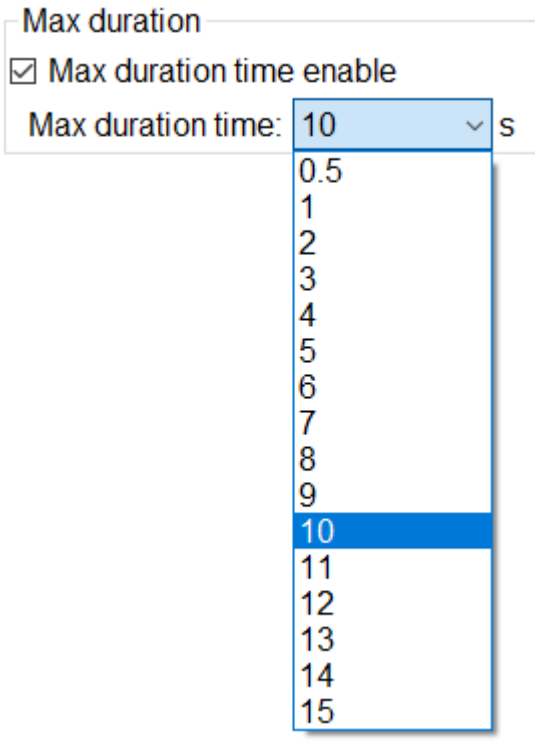


Figure 4-25: Max Duration

4.3.3 Baseline Filter

IIR coefficient

Return the total value of coefficient/256 value of the current raw count plus (256 - coefficient)/256 of the previous filter value.

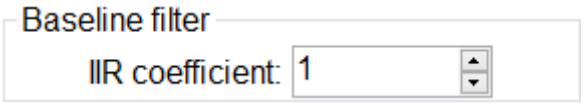


Figure 4-26: Baseline Filter

4.3.4 Raw Count Filter

There are four types of filters for raw count. One or more filters can be selected and adopted following the sequences **Jitter filter → Median filter →Average filter →IIR filter**.

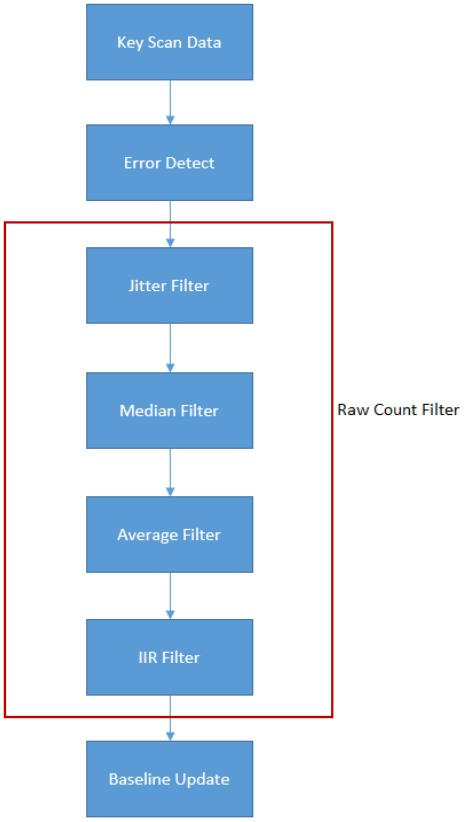


Figure 4-27: Raw Count Filter

Jitter

This filter is to remove centroid noise for slider or touch key buttons.

If Input > (previous output + jitter delta), output value is (Input – jitter delta).

If Input < (previous output - jitter delta), output value is (Input + jitter delta).

Otherwise, output value is the previous filter value.

Median

Return the middle value after sorting the last three raw counts.
This is a nonlinear filter. The median is computed using a two-step process. First, the buffer values are sorted from the smallest to the largest; and the middle value is selected from the ordered list. The buffer is scanned for the median with each update of the buffer.

Average

Return the average value of the last three raw counts.

It is used in the case where is necessary to smooth data that carries high frequency distortion.

IIR

Return the value of 1/coefficient * the current raw count + (coefficient – 1)/coefficient * the previous filter value.
The IIR filter (Infinite Impulse Response filter) is a recursive filter because the output from the filter is computed by using the current and previous outputs.

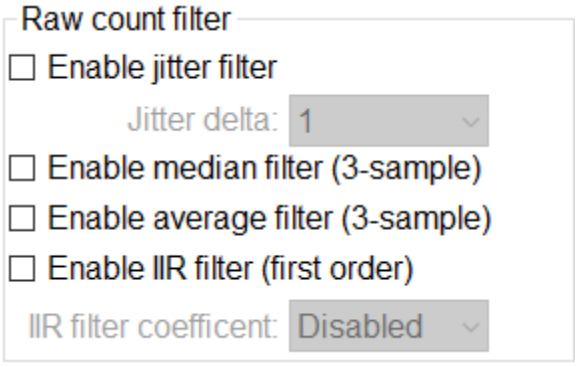


Figure 4-28: Raw Count Filter Settings

4.3.5 Interrupt

Interrupt is the output for interrupt request (IRQ) and interrupt Pin has been fixed at Pin 16 INTB

From **Settings -> General**, click **Interrupt enable** checkbox. Optional select: **Auto clear enable /Repeat enable**

All changes will be active by clicking the **Apply** button**.**

Reading key status register through I2C from the host has the high priority to clear INT.

Interrupt enable is a global interrupt setting. If it is disabled, interrupt for all keys will be turned off even though some key is pressed.

Figure 4-29: Interrupt Menu

Case 1. Click Interrupt Enable

Interrupt output is low if any key touch is detected. Interrupt output is high (by external pull high resistor) if key status registers (07h, 08h and 09h) are read or no key touch is detected.

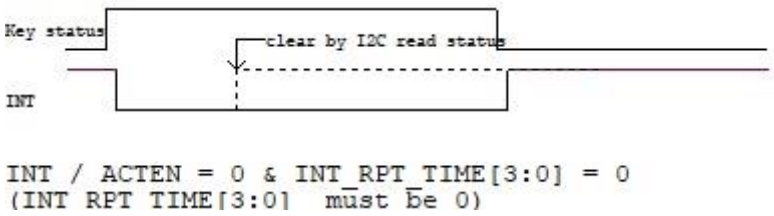


Figure 4-30: INT Output Low by Key ON Status

Case 2. Click Interrupt Enable and Auto Clear Enable

Auto clear enable is to enable auto clear interrupt function. Check to enable.

Auto clear time is to set auto clear interrupt time interval in milliseconds.

When **Auto clear enable** is enabled and 07h & 08h & 09h registers are not read within the programmed **Auto clear time** (10ms~200ms), IS31SE5120A will release INT pin after **Auto clear time** is expired. INT pin can also be released by reading registers 07h, 08h, and 09h (Key Status Register).

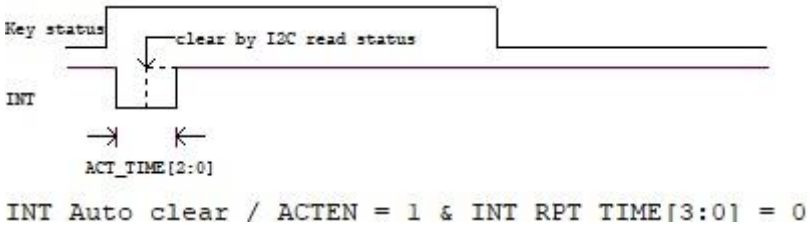


Figure 4-31: Auto Clear INT

Case 3. Click Interrupt Enable, Auto Clear Enable and Repeat Enable

Repeat Enable is used to set interrupt auto-repeat time. Click drop down menu to choose a different time and the unit is millisecond. If there is a key keeping pressed, the second interrupt will be generated after the repeat time is expired.

If Multi Key function is enabled, INT signal will be immediately low when there is a 2nd key touch detection even the repeat INT process as the below figure is on-going.

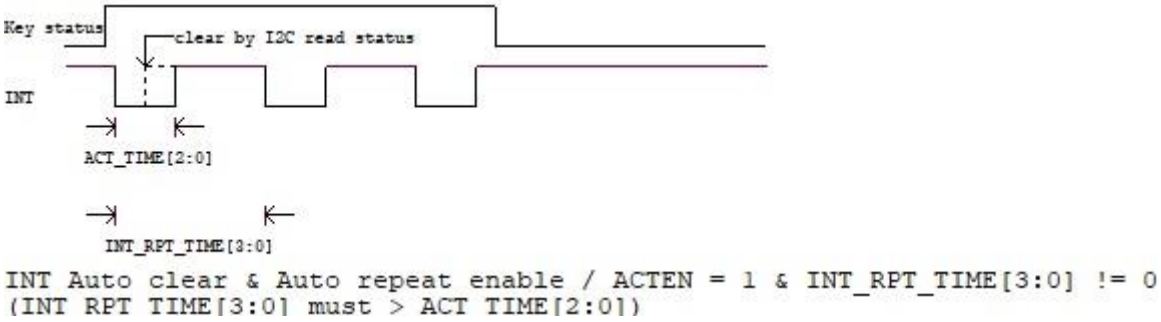


Figure 4-32: Repeat INT

4.3.6 Lock

Lock threshold

Ignore touch key scan if the count difference between baseline count and raw count exceeds the lock threshold.

Lock scan cycle

When Lock threshold is reached, ignore key scanning for the following "**Lock scan cycle**" setting time to prevent noise interference.

Raw count difference limit

Ignore the key scan data if the difference between previous raw count and current raw count exceeds the limit setting.

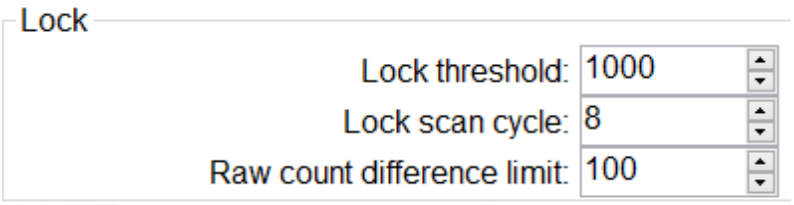


Figure 4-33: Lock

4.4 Key

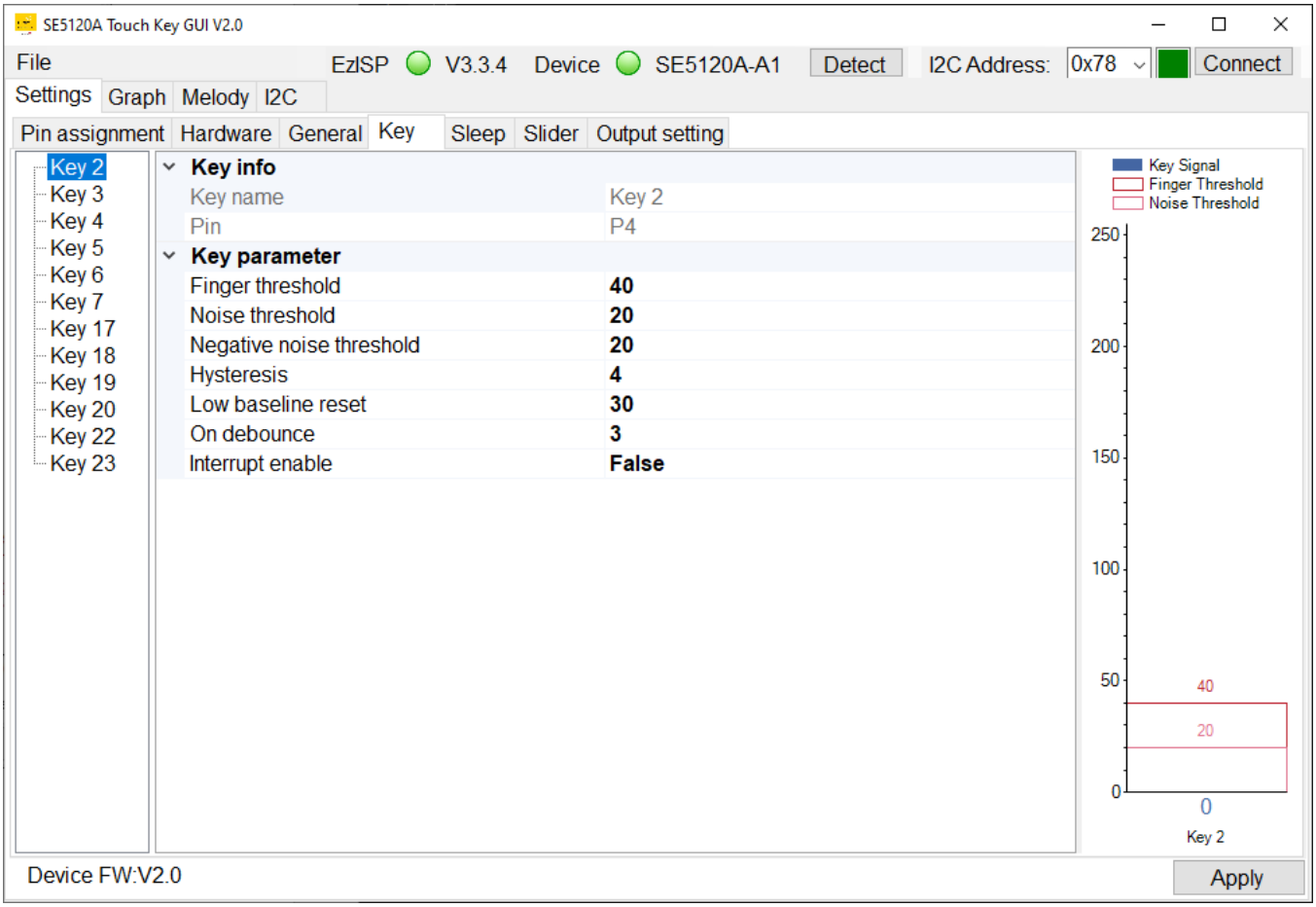


Figure 4-34: Key Page

4.4.1 Key Info

It includes key name and pin number.

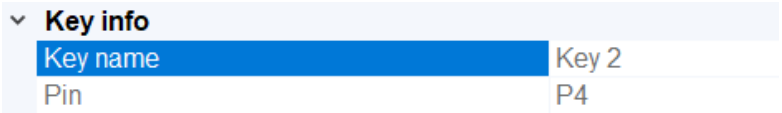


Figure 4-35: Key Info

4.4.2 Key Parameter

Finger threshold

It is used with hysteresis to determine the key status.

Noise threshold

Baseline is updated if the difference count of baseline count and raw count is below the noise threshold.

Negative noise threshold

It is used with the low baseline reset count to reset baseline count to the current raw count. Please refer to the description of **Low baseline reset.**

Hysteresis

Touch key is triggered when its signal is bigger than the finger threshold plus hysteresis value and is released when its signal is smaller than the finger threshold minus the hysteresis value.

Low baseline reset

Low baseline reset count of each key. A reset count increases one if the absolute |raw count – baseline| > absolute |negative noise threshold|. Once the reset count exceeds the low baseline reset register value, the baseline is reset to the current raw count. The reset count will be reset to 0 if the absolute | raw count – baseline| <= absolute |negative noise threshold|.

On debounce

When one key's on debounce count is reached, this key is triggered.

Interrupt enable

The Interrupt enable determines whether a key is detected and causes the interrupt pin to be asserted.

Key parameter	
Finger threshold	**40**
Noise threshold	**20**
Negative noise threshold	**20**
Hysteresis	**4**
Low baseline reset	**30**
On debounce	**3**
Interrupt enable	**False**

Figure 4-36: Key Parameter

4.4.3 Key Signal Bar Chart

It shows individual key signals. The corresponding value will be displayed in the box when pressing KEYs on EvB. The key number X turns red if its signal value is over finger threshold plus hysteresis and it means this key is triggered. The maximum value is 254. GUI will keep 254 if the input value is over 254.

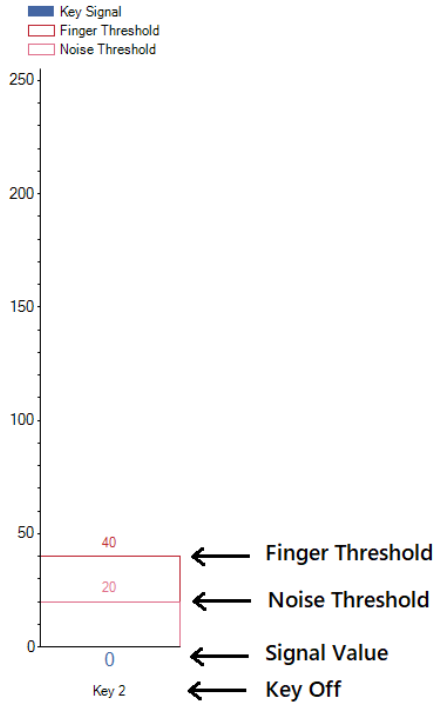


Figure 4-37: Key Signal Bar Chart

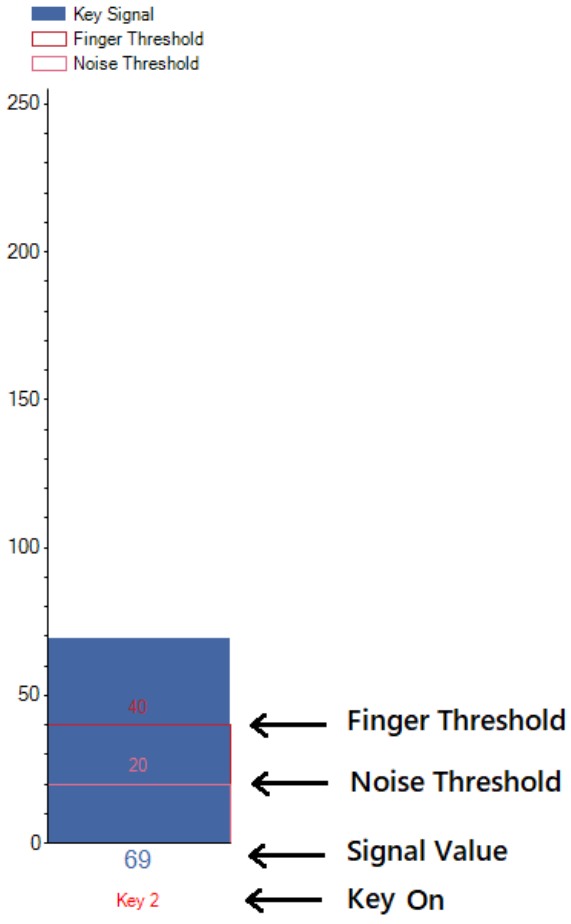


Figure 4-38: Signal Bar Char with Finger Touch

4.5 Sleep

IS31SE5120A integrates auto sleep function, and the entering-sleep time can be configured. Device will enter sleep mode when there is no touch key detected. It will be woken up by the key touch. For applications that require low power consumption, this device can be set to sleep mode automatically.

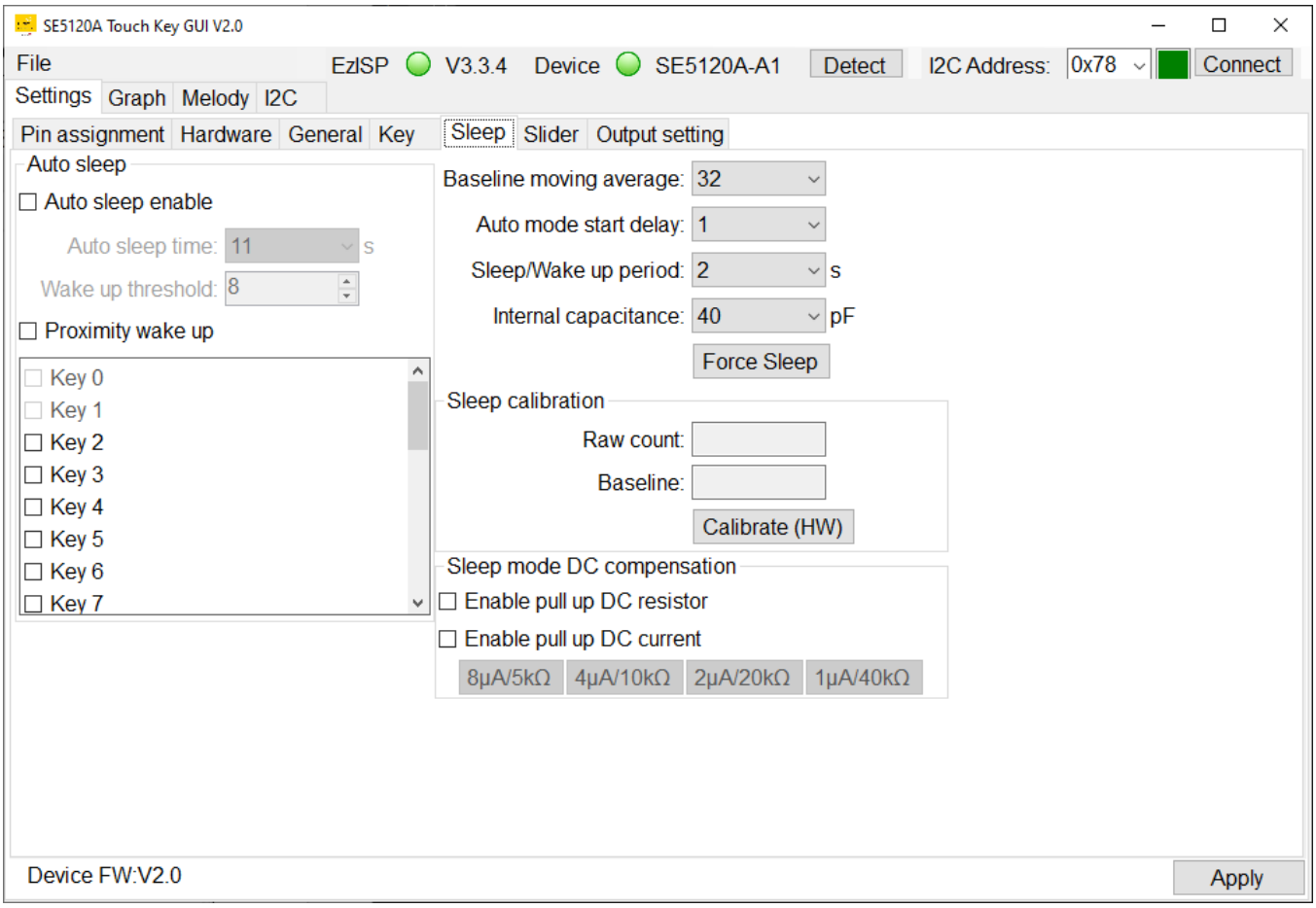


Figure 4-39: Sleep Page

4.5.1 Auto Sleep

IS31SE5120A 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.

Enable the corresponding key and the device will be woken up as the key value arrives at the **Wake-up threshold**.

Wake-up threshold is to set the key threshold of waking up from sleep mode. Input setting in box and pressing enter, tab key, or moving the cursor to another location will save the setting.

Proximity wake-up

Apply non-contact detection to wake up the device.

Disable: Approaching device won't wake up device and device behavior will follow settings in □**Auto sleep enable**.

Device wakes up -> scans key once -> goes to sleep again if no key is detected.

Enable: Approaching the device will wake up the device. Device wakes up -> generates INT signal (optional) -> goes to sleep after **Auto sleep time** is expired if no key is detected.

Generate the INT signal for wake-up event if **Interrupt enable** option is checked in General Tab.

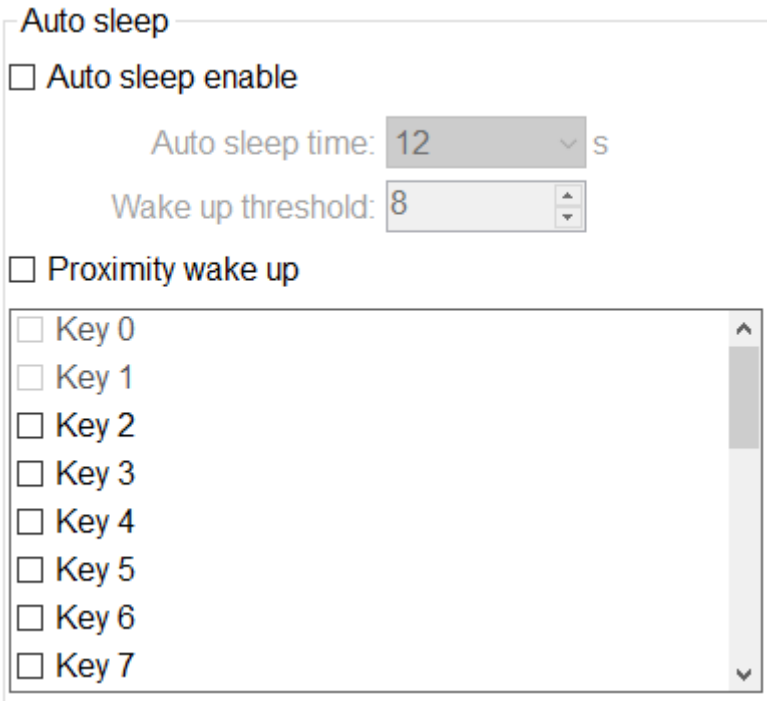


Figure 4-40: Auto Sleep Setting

The Connect status will be displayed with a yellow box whenever device enters sleep mode.

Figure 4-41: I2C Address

The GUI cannot operate in sleep mode. Device needs to return to normal mode by pressing the selected wake-up key or "**Connect**" button.

Exit Sleep Mode

Step 1. Press the selected wake-up key and the connect status will be displayed with a green box.

Step 2. Hold the key pressed if needed and it will keep the device in normal mode.

Step 3. Disable sleep mode by unchecking **Auto Sleep enable** box and clicking **Apply**.

4.5.2 Baseline Moving Average

Device will enter sleep mode if baseline exceeds the setting Baseline moving average.

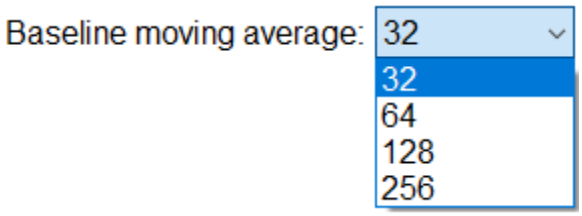


Figure 4-42: Baseline Moving Average

4.5.3 Auto Mode Start Delay (ASTDLY)

ASTDLY inserts an inter-sequence idle time of (ASTDLY+1) * 256 * TK3 clock time at each sequence start. This delay allows the stabilization time from normal mode to sleep mode.

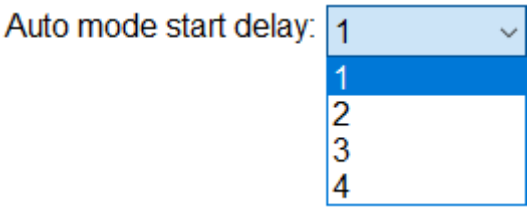


Figure 4-43: Auto Mode Start Delay

4.5.4 Sleep/Wake Up Period

Set "**Sleep/wake up period**" and the touch key baseline will be dynamically adjusted after the setting time is expired. The default setting is 2 seconds.

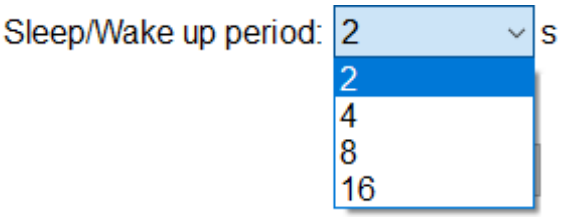


Figure 4-44: Sleep/Wake Up Period

4.5.5 Internal Capacitance

This setting is to select internal charging capacitance value.

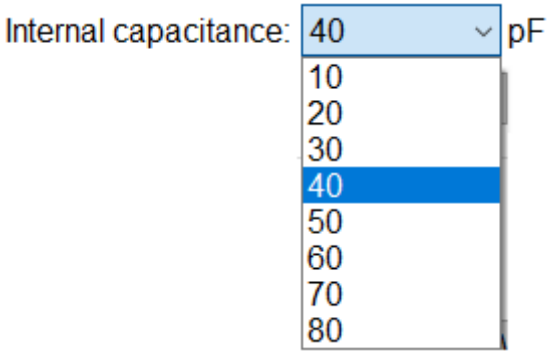


Figure 4-45: Internal Capacitance

4.5.6 Force Sleep

When IS31SE5120A EvB is in normal mode, the "**Connect**" button in GUI will turn green. After clicking **Force Sleep**, IS31SE5120A EvB will enter sleep mode and show connecting status with yellow box. Touch the assigned wake-up key or click "**Connect**" button, device will return to normal state.

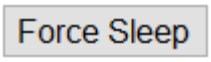


Figure 4-46: Force Sleep

4.5.7 Sleep Calibration

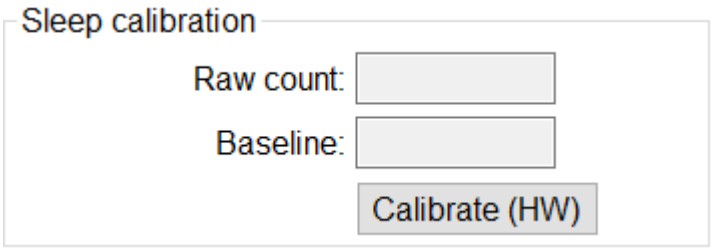


Figure 4-47: Sleep Calibration

Sleep Calibration Method – Use calibration result to set up Wake-up threshold.

Step 1. Select available Key 0~Key 23 for wakeup. Suggest choosing one Key or raising internal capacitor setting value, otherwise there might not come out raw count and baseline value when users click **Calibrate (HW)** button.

Step 2. Select related settings (Internal capacitance, DC compensation, ……). Users must click **Apply** button located at GUI's right bottom corner to submit setting.

Step 3. Click **Calibrate (HW)** button to get raw count and baseline data and click button again to stop calibration. Need to stop calibration before going to the next step.

Step 4. It is good to have raw count and baseline count closer to 1/2 of "Cycle count per key scan". If not, go to step 2.

Step 5. Record the raw count without finger touch and with finger touch and set the **Wake-up threshold** by the difference value of both counts.

Example:

Make sure the device is connected (with a green box). If not, click "**Connect"** button.

Step 1. Select wake-up key = Key 2.

Step 2. Select internal capacitance = 60pF.

Step 3. Click **Apply**.

Step 4. Click **Calibrate (HW)**, raw count = 4312.

Step 5. Click **Calibrate (HW)** again to stop calibration.

Step 6. Record the raw count without finger touch and with finger touch and set the **Wake-up threshold** by the difference value of both counts.

Case 1: Sleep calibration raw count = 4312 without finger touch when Internal capacitance = 60pF

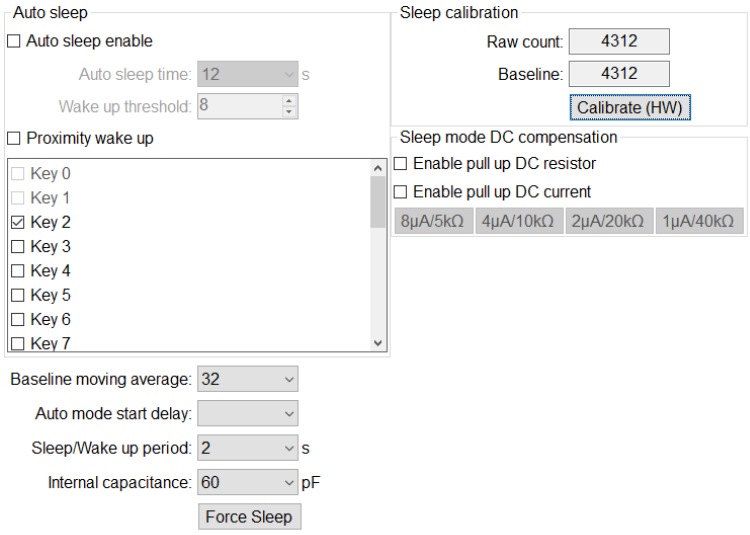


Figure 4-48: Sleep Calibration case 1

Case 2: Sleep calibration raw count = 5251 without finger touch when Internal capacitance = 80pF

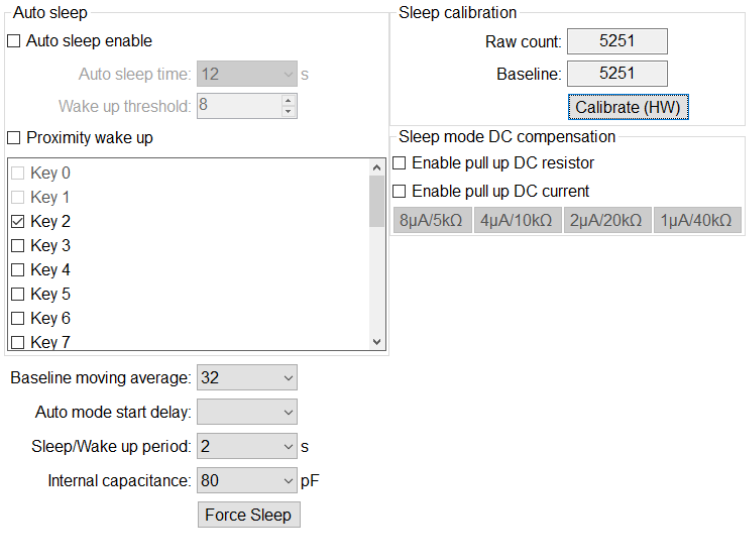


Figure 4-49: Sleep Calibration case 2

Case 3: Sleep calibration raw count = 5193 with finger touch when Internal capacitance = 80pF

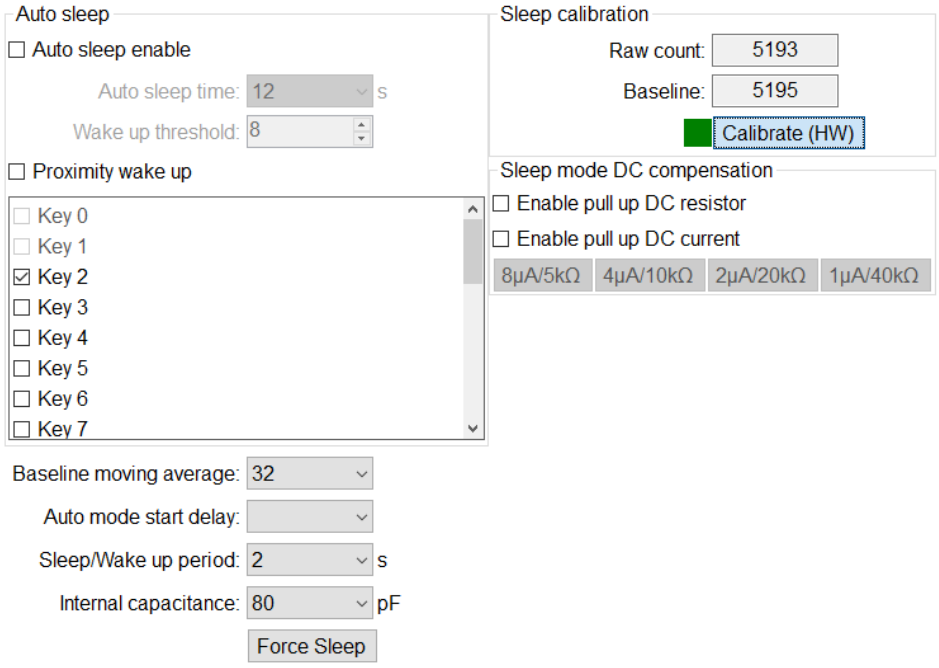


Figure 4-50: Sleep Calibration case 3

4.5.8 DC Compensation

DC compensation in sleep mode is the same as in the normal mode. But DC compensation in the sleep mode will be performed based on total effects of all the keys users selected.

The hardware is to configure a constant DC pull-up/pull-down (PUD) on IS31SE5120A pin Cref to allow high capacitance touch-key detection. A DC pull-up/pull-down can compensate for the equivalent resistance which is caused by a high capacitance key. Connecting a switching current source or resistor can thus maintain sensitivity of touch key detection.

For DC current, PUD can enable 8uA/4uA/2uA/1uA current source with Cref pin connecting with 5K**Ω**/10K**Ω**/20K**Ω**/40K**Ω** resistor respectively.

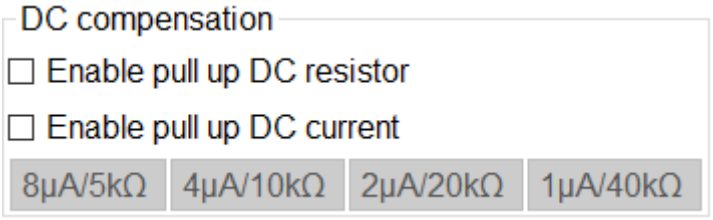


Figure 4-51: DC Compensation

Example:

Make sure the device is connected (with a green box). If not, click "**Connect"** button.

Step 1. Select wake-up key = Key 2, 3, 4, 5, 6.

Step 2. Select internal capacitance = 40pF.

Step 3. Enable pull up DC current by selecting the checkbox and click **2uA/20kΩ**.

Step 4. Click **Apply**.

Step 5. Click **Calibrate (HW)**, raw count = 6677.

Step 6. Click **Calibrate (HW)** again to stop calibration.

Case 4: Sleep calibration raw count = 6677 without finger touch when Internal cap = 40pF, DC current = 2uA

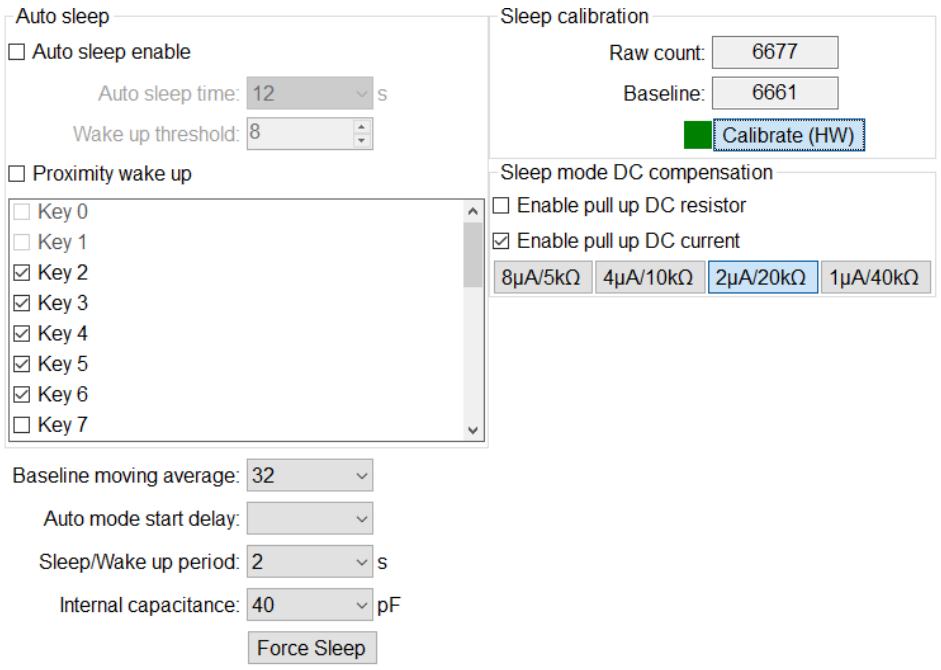


Figure 4-52: Sleep Calibration case 4

Case 5: Sleep calibration raw count = 6591 with finger touch when Internal cap = 40pF, DC current = 2uA

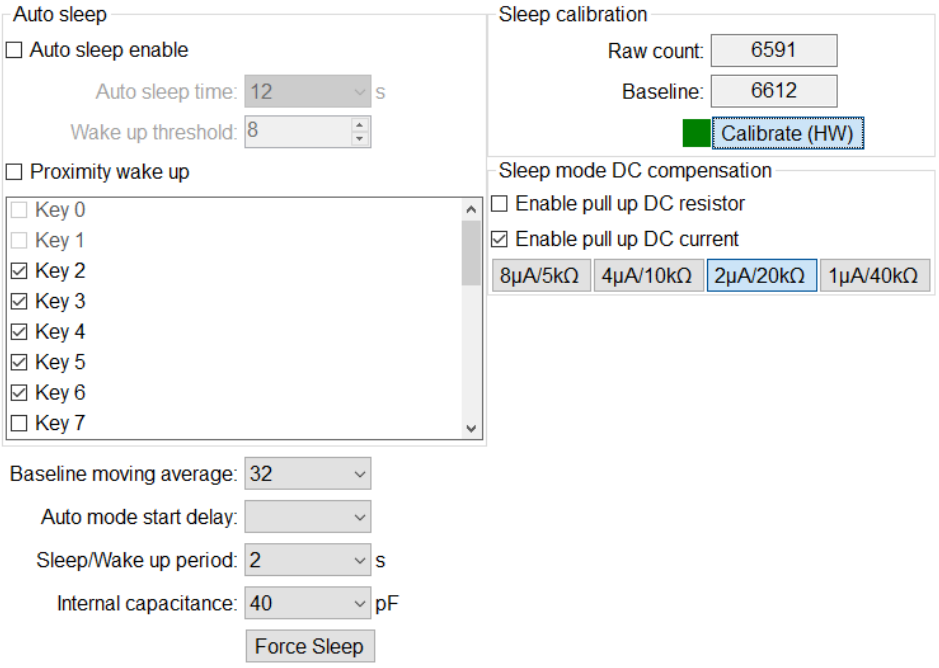


Figure 4-53: Sleep Calibration case 5

4.6 Slider

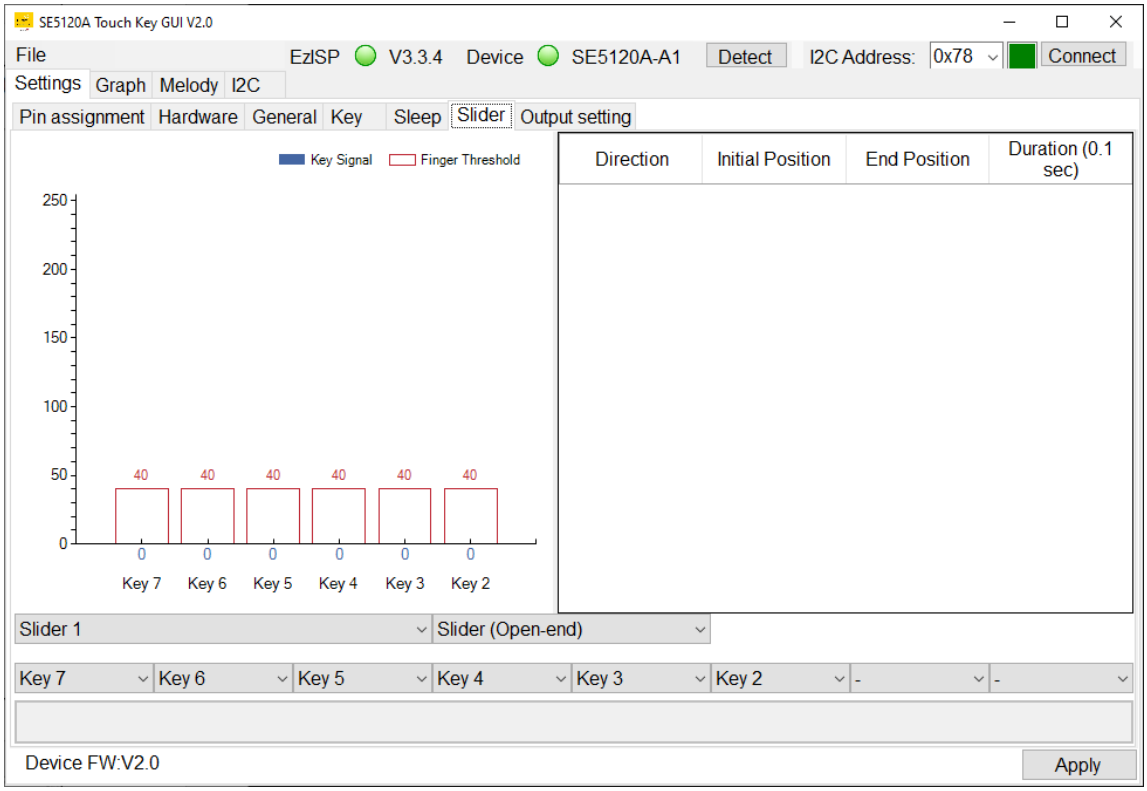


Figure 4-54: Slider Page

There are two slider groups support now. Each slider can support 3 ~ 8 keys. Users can switch the order of the selected keys based on user's need.

IS31SE5120A EvB kit can support linear slider and wheel slider as below figures.

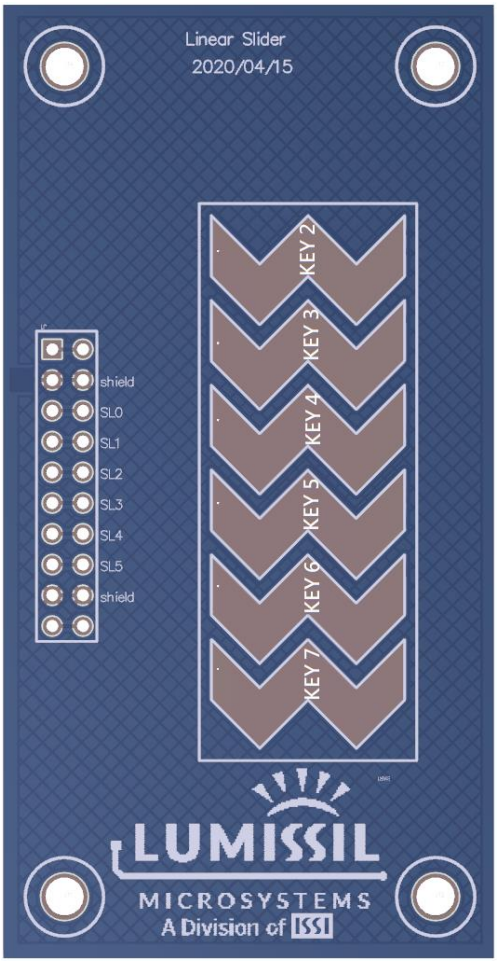


Figure 4-55: Open end Linear Slider

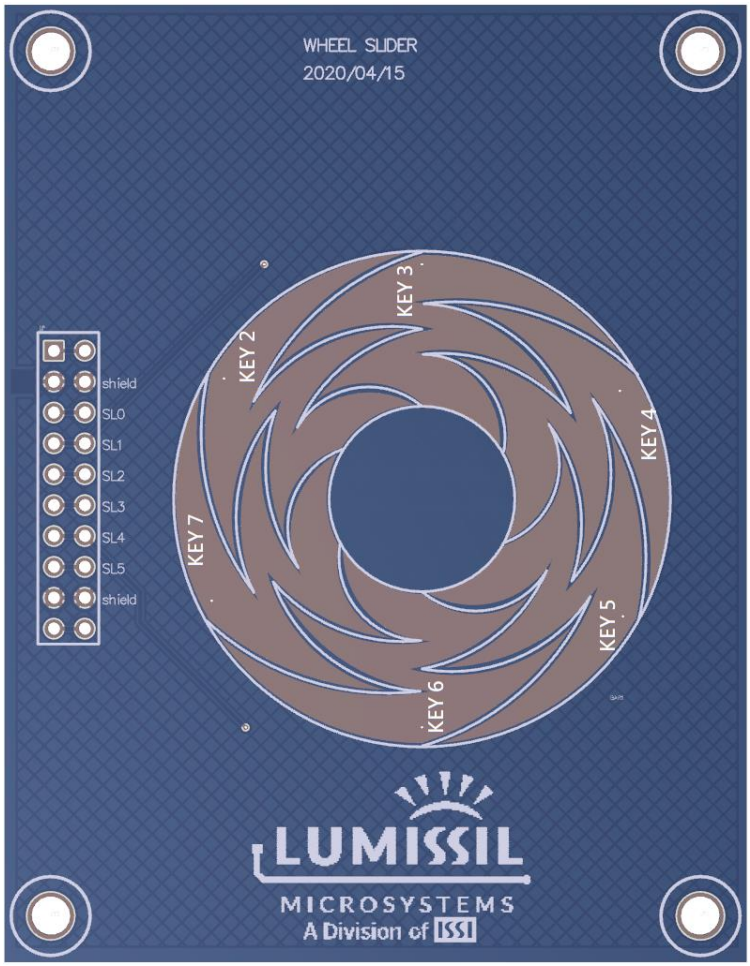


Figure 4-56: Closed end Wheel Slider

4.7 Output setting

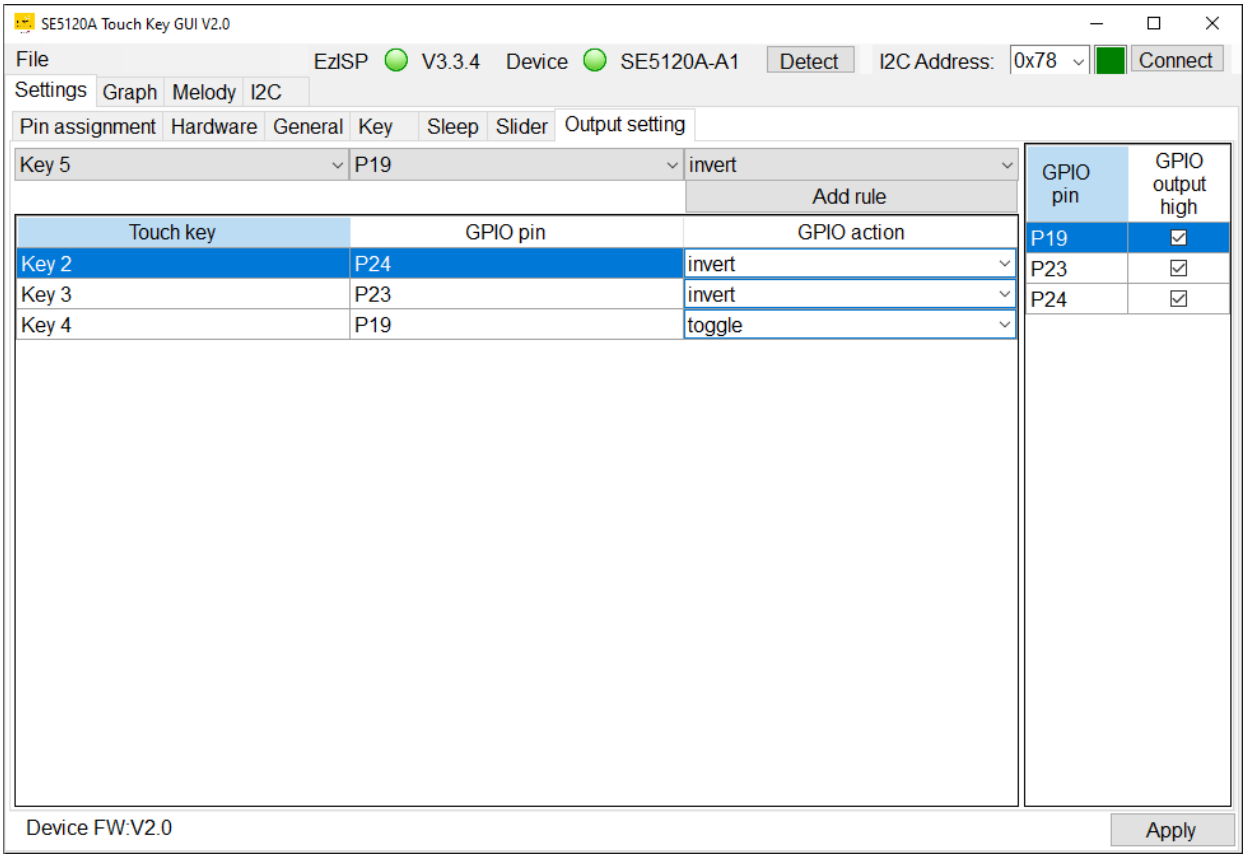


Figure 4-57: Output Setting Page

4.7.1 Add GPIO rule

Users can click "**Add rule**" and select Key x, Px and invert/toggle settings to add more Key setting as the highlighted red rectangles in below figure. Key x is the key number in IS31SE5120A EvB or daughter boards. Px is the pin number of SE5120A IC part. Remember to click **Apply** to save the setting.

In case users need to remove some **Add rule** like for Key 4, select the whole Key 4 row as the blue rectangle in below figure, and press "**delete**" key in the keyboard.

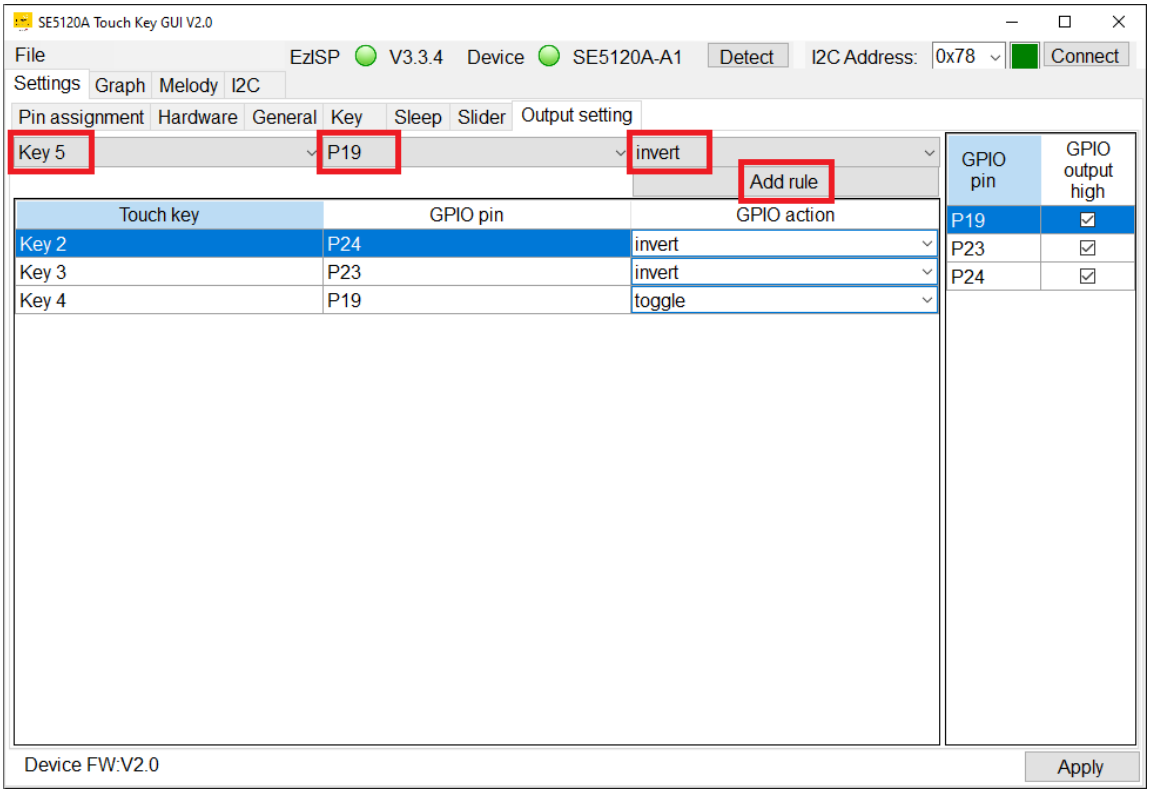


Figure 4-58: Add and remove rule for Output setting

4.7.2 GPIO setting

When Key 4 is set "toggle", the red LED D11 on the top of Key 4 as the below figure will turn on and keep on until Key 4 is triggered again. Then the LED will turn off.

When Key 4 is set "invert", the red LED near Key 4 will turn on and then off when Key 4 is triggered. LED D9~D11 can be configured for different Key based on the setting rule.

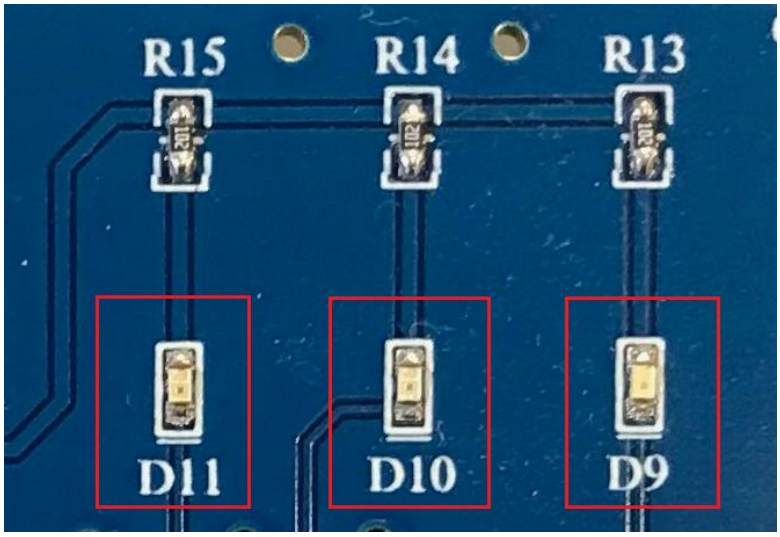


Figure 4-59: Key 2~4 and LED 9~11

When "GPIO output high" is enabled, the default GPIO output is HIGH, and the corresponding LED will be OFF.

If "GPIO output high" is disabled, the GPIO output is LOW, and the corresponding LED will be ON.

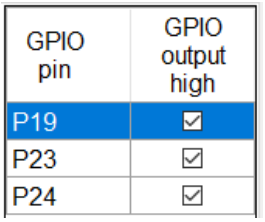


Figure 4-60: GPIO output high setting

5. GRAPH

As shown in **Figure 5-1: Graph Line Chart** below, Line Chart is the KEY value curves to show the current value of KEY2~KEY23. The historical value of KEYs will be checked by pulling the scroll bar. Users can uncheck the key enable box in the higher right corner to remove unwanted key information.

Reset baseline: Click this button, baseline will be replaced by raw count.

□**I2C sync:** If the option is enabled, baseline and raw count will be synchronized.

GUI polling period: Default is 50ms. It can be increased or decreased by 10ms per click.

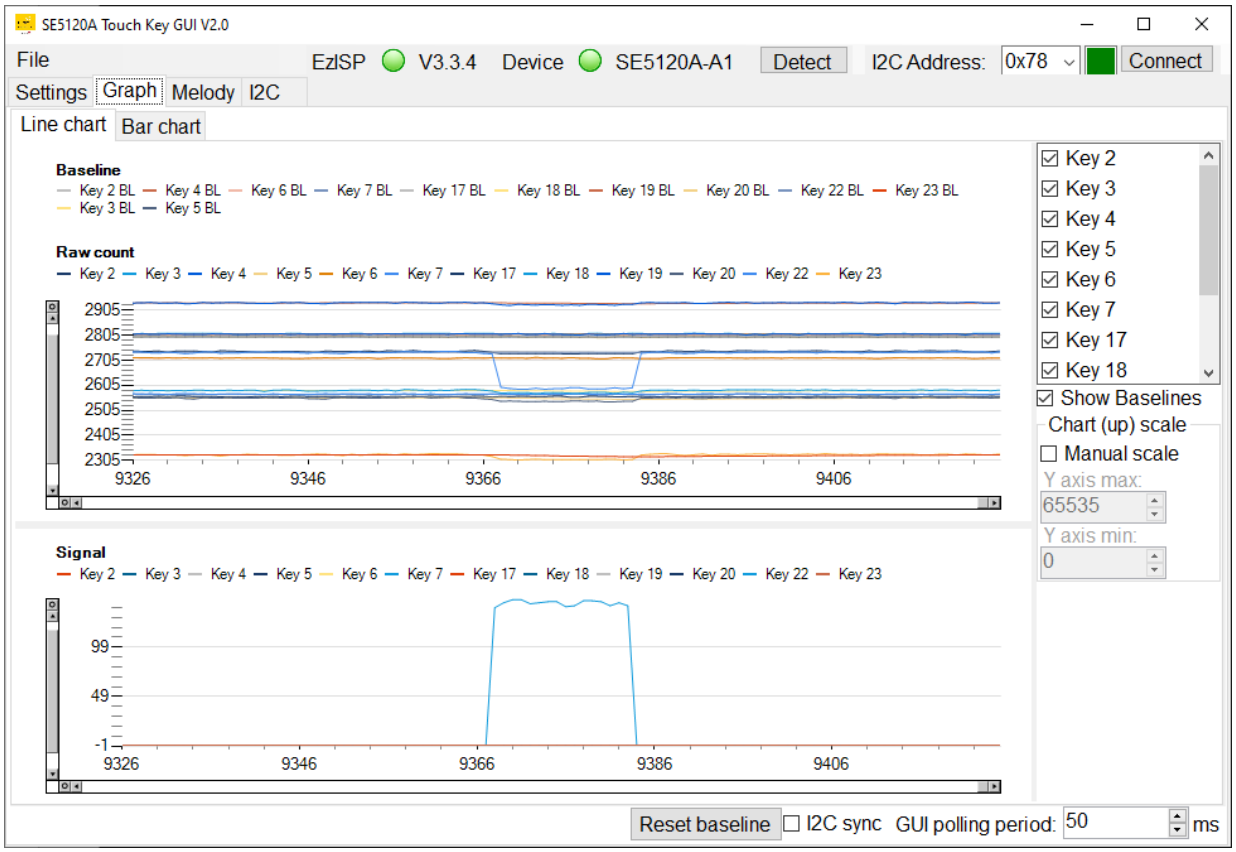


Figure 5-1: Graph Line Chart

As shown in Figure 5-2: Graph Bar Chart below, Bar Chart is to show the current signal value of KEY2~KEY23. Red color on Key 19 means the device has detected a finger touch on Key 19.

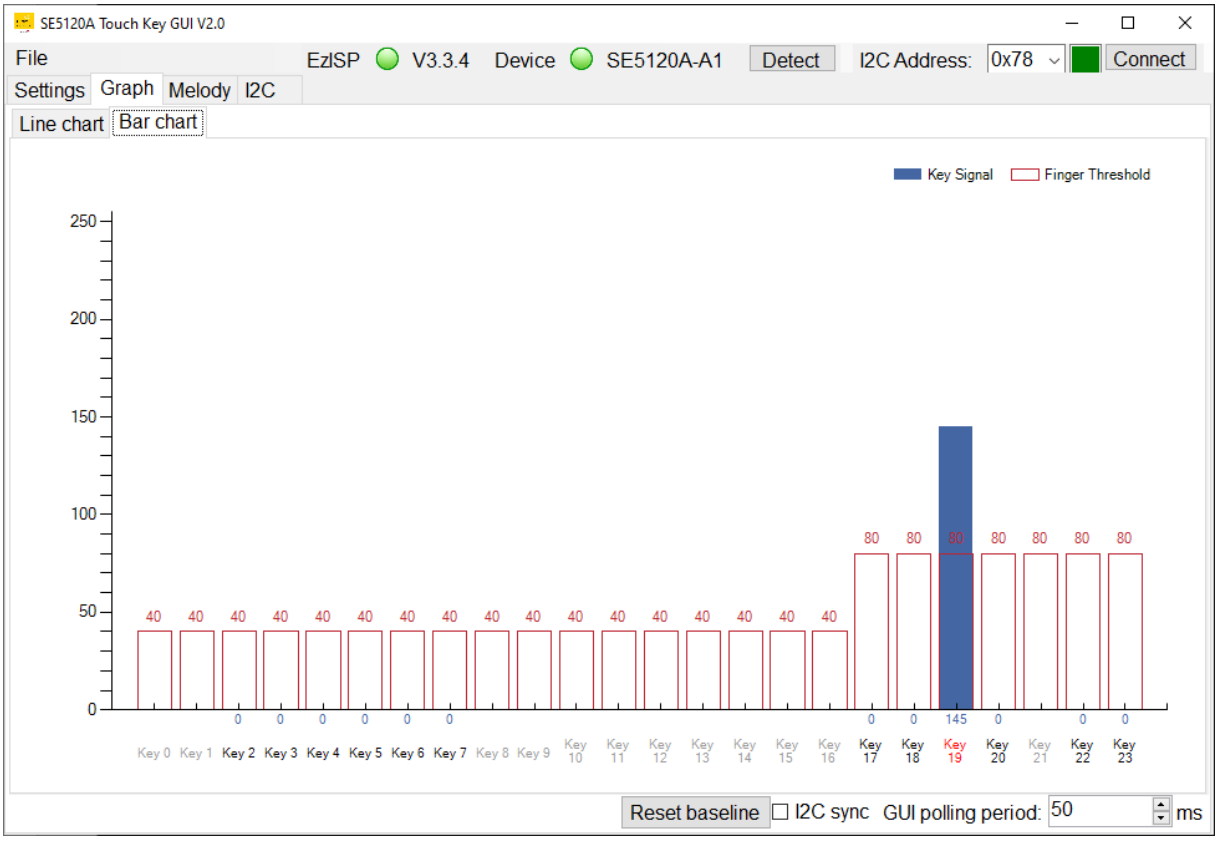


Figure 5-2: Graph Bar Chart

6. MELODY

As shown in **Figure 6-1: Melody Page of GUI**Figure 6-1: Melody Page of GUI below, the built-in melody function has 10 FIFOs, and the host controller can send the melody data up to 10 notes.

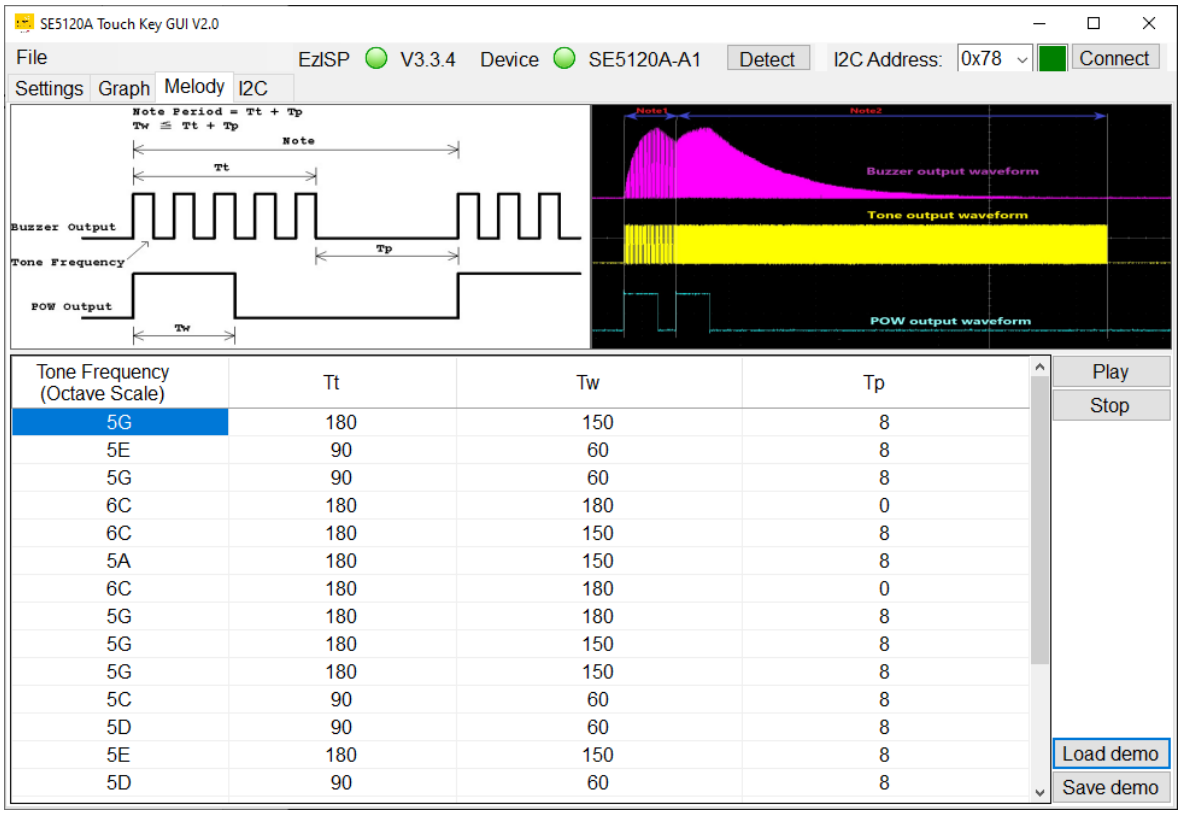


Figure 6-1: Melody Page of GUI

Support scale from 3A to 8G#

Frequencies for equal-tempered scale, A4 = 440 Hz "Middle C" is C4												
	3	freq	divisor	freq error	4	freq	divisor	freq error	5	freq	divisor	freq error
C					3	261.6	1911	0.01%	15	523.3	956	-0.05%
C#					4	277.2	1804	-0.01%	16	554.4	902	-0.01%
D					5	293.7	1703	-0.02%	17	587.3	851	0.04%
D#					6	311.1	1607	0.00%	18	622.3	804	-0.06%
E					7	329.6	1517	-0.01%	19	659.3	758	0.06%
F					8	349.2	1432	-0.02%	20	698.5	716	-0.02%
F#					9	370.0	1351	0.03%	21	740.0	676	-0.05%
G					10	392.0	1276	-0.04%	22	784.0	638	-0.04%
G#					11	415.3	1204	-0.01%	23	830.6	602	-0.01%
A	0	220.0	2273	-0.01%	12	440.0	1136	0.03%	24	880.0	568	0.03%
A#	1	233.1	2145	0.01%	13	466.2	1073	-0.04%	25	932.3	536	0.05%
B	2	246.9	2025	-0.01%	14	493.9	1012	0.04%	26	987.8	506	0.04%

	6	freq	divisor	freq error	7	freq	divisor	freq error	8	freq	divisor	freq error
C	27	1046.5	478	-0.05%	39	2093.0	239	-0.05%	51	4186.0	119	0.37%
C#	28	1108.7	451	-0.01%	40	2217.5	225	0.21%	52	4434.9	113	-0.23%
D	29	1174.7	426	-0.08%	41	2349.3	213	-0.08%	53	4698.6	106	0.39%
D#	30	1244.5	402	-0.06%	42	2489.0	201	-0.06%	54	4978.0	100	0.44%
E	31	1318.5	379	0.06%	43	2637.0	190	-0.21%	55	5274.0	95	-0.21%
F	32	1396.9	358	-0.02%	44	2793.8	179	-0.02%	56	5587.7	89	0.54%
F#	33	1480.0	338	-0.05%	45	2960.0	169	-0.05%	57	5919.9	84	0.55%
G	34	1568.0	319	-0.04%	46	3136.0	159	0.28%	58	6271.9	80	-0.35%
G#	35	1661.2	301	-0.01%	47	3322.4	150	0.33%	59	6644.9	75	0.33%
A	36	1760.0	284	0.03%	48	3520.0	142	0.03%				
A#	37	1864.7	268	0.05%	49	3729.3	134	0.05%				
B	38	1975.5	253	0.04%	50	3951.1	127	-0.36%				

I2C command format (Each node is composed of 4-byte data and the incomplete node will be ignored.)

0x78, 0xF0, (scale_id, Tt, Tw, Tp), (scale_id, Tt, Tw, Tp), ….

0x78, 0xF0, 0xFF Stop the melody play and clear the FIFO

0x78, 0xF0 Set the register no to 0xF0
0x79 Read FIFO remaining length

Scale ID 0 is 3A, 1 is 3A#, 2 is 3B ….

Tt, Tw, Tp range from 0 to 255 and each value increase equals to 4 ms.

7. I2C COMMUNICATION

In the blue highlighted rectangular box for "**Command list**" in **Figure 7-1: I2C Communication Page of GUI** , users can issue I2C commands to the EvB. On the GUI right-hand side, choose to **Send**, **Clear**, **Save** and **Load**. 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 highlighted rectangular box on the right bottom, users can choose to save or clear the data log after the I2C command is issued. The data Log content is under the "**Command list**" box. The green highlighted rectangular box on the right middle indicates the delay time setting between each command and the delay setting range is 10ms~1000ms.

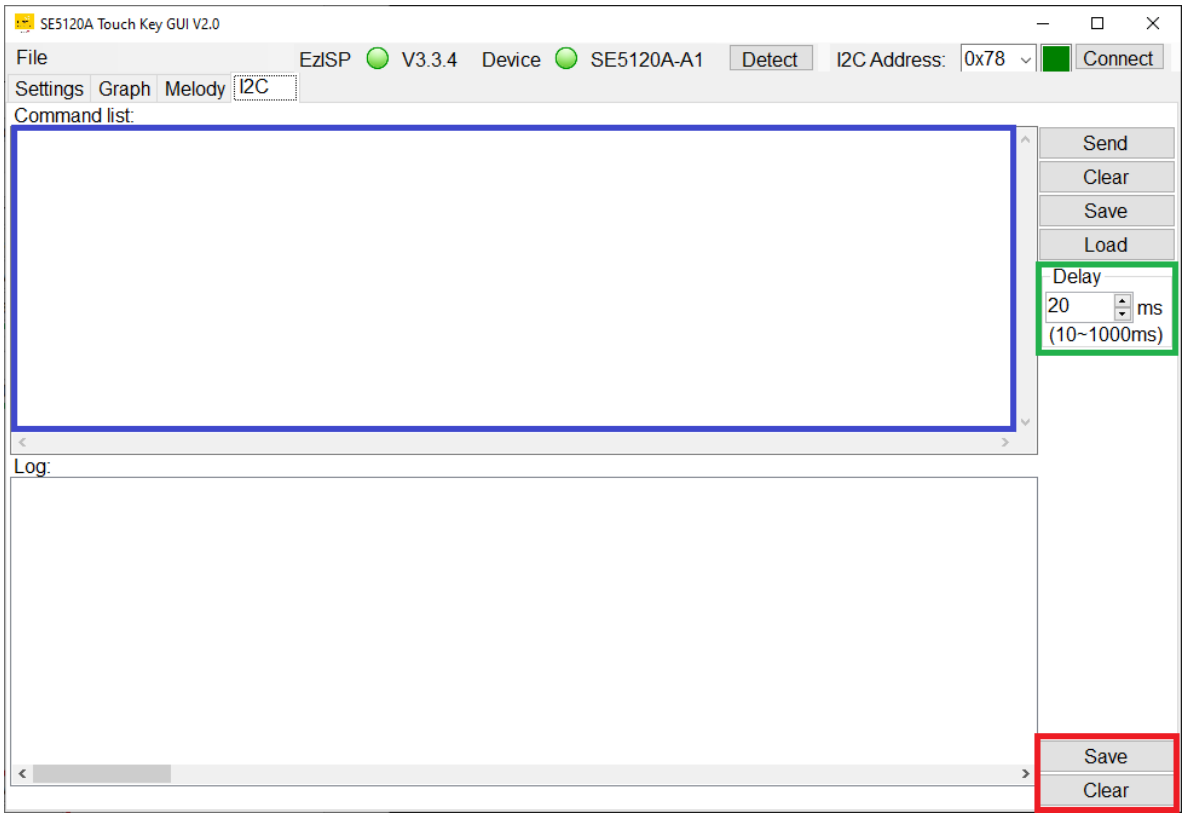


Figure 7-1: I2C Communication Page of GUI

7.1 I2C Communication Example (Touch Key)

Command list:

78 82 0A 3C	Write data 0x0A to register no. 82
	Write data 0x3C to register no. 83
78 82	Write register no. 82
79	Read data from register no. 82
78 83	Write register no. 83
79	Read data from register no. 83

Log list:

22:16:45:170	>>	78 82 0A 3C	Write data 0x0A to register no. 82
			Write data 0x3C to register no. 83
22:16:45:197	>>	78 82	Write register no. 82
22:16:45:218	<<	0A	Read data from register address 82
22:16:45:238	>>	78 83	Write register no. 83
22:16:45:238	<<	3C	Read data from register address 83

Note:

All the expressed number will be treated as hex, i.e., 16 is 22 in decimal.

Only one byte data can be read at one command.

8. SAFETY SELF-TEST

Lumissil built a set of library functions to support IEC60730-1 Class B Self-Test Library (STL) which is for MCU self-test according to the requirements of IEC60730-1 Standard Annex H and the hardware resources of MCU itself. Through the MCU self-test, the reliability and security of the system operation can increase.

Lumissil provides a sample code containing Self-Test Library to help software developers better understand STL and accelerate the development of products that require IEC60730-1 certification. Developers can refer to the usage of STL and integrate it into their own projects.

The GUI provides Safety Self-Test items in addition to Touch Key function GUI support. For detailed Safety Self-Test application, please refer to Lumissil's IEC60730-1 ClassB STL Application Note for 8 bit MCU.pdf.

8.1 Safety Self-Test

Click the **Safety Self-Test** from the **File** drop-down menu as below Figure 8-1.

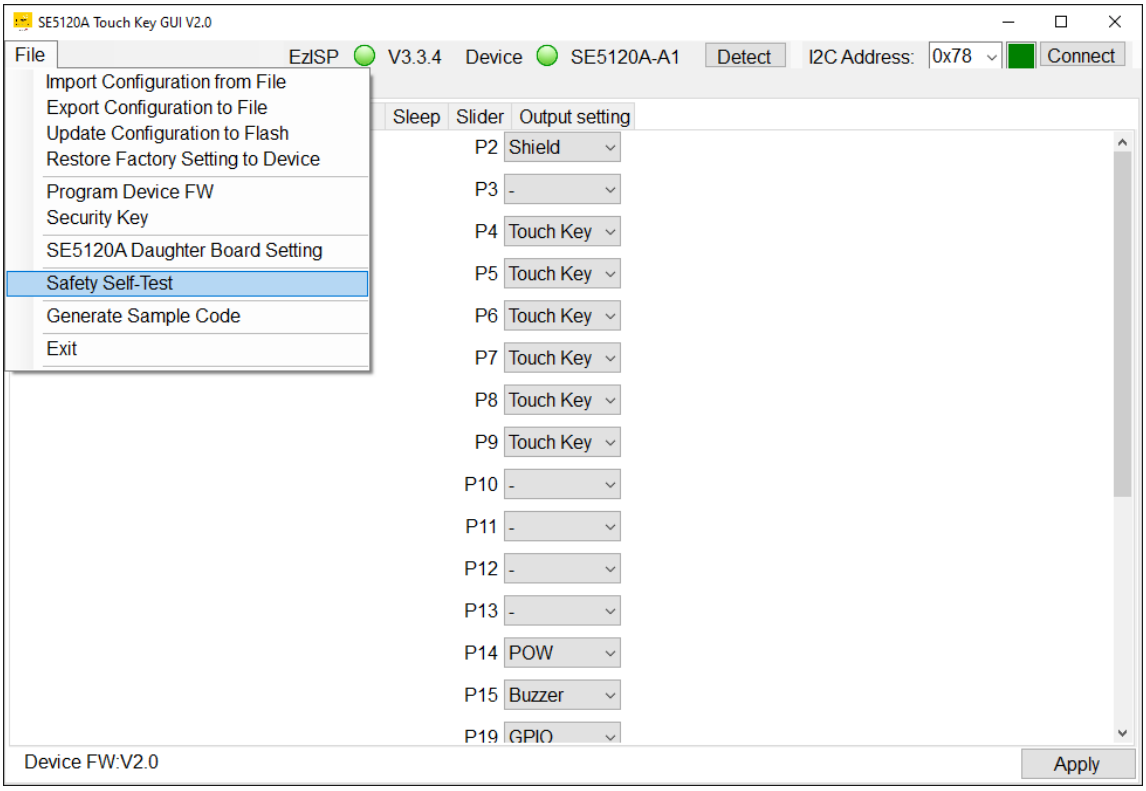


Figure 8-1: Safety Self-Test

Below **Safety Self-Test** window will pop out and the above SE5120A Touch Key GUI "**Connect**' status will turn yellow color. **User needs to click "Connect "button again. The connect status will turn green color and Safety Self-Test can be performed.**

When the user chooses any test item, the bottom box of the Safety Self-Test window will provide detailed information for the selected test item. For example, a CPU test item is selected and there is CPU register test information in the message box at the bottom. And CPU test will start.

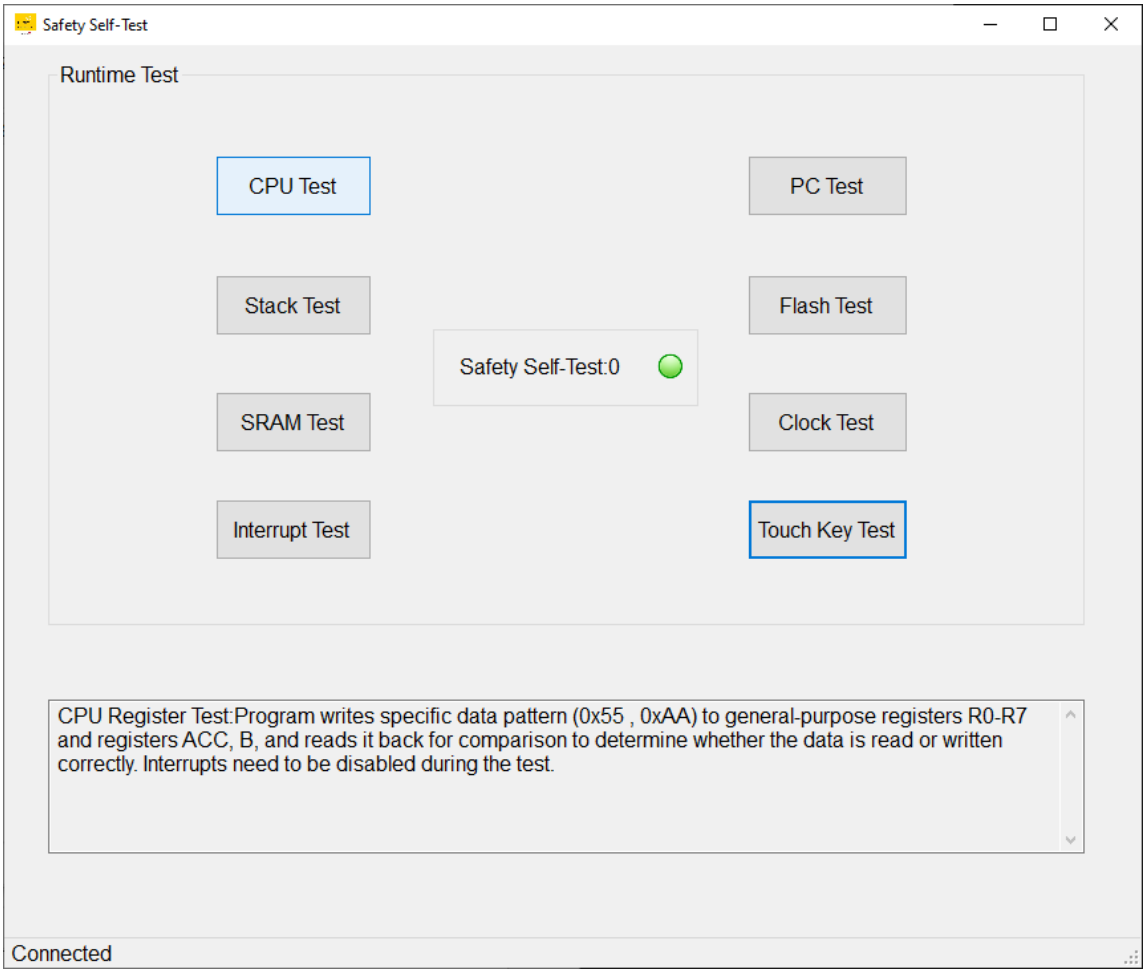


Figure 8-2: Safety Self-Test options

If the selected test item passed, then user would see "**Safety Safe-Test :5a Pass**" with green color status as above figure. If the test fails, "**Safety Safe-Test :1 Fail**" status will come with red color.

For the default setting, the Flash test and SRAM test will test the whole memory space at one time, and it will take a long time. If the user wants to reduce the testing time, the memory space can be split into continuous segments, and perform testing based on the inquiry.

8.2 Safety Self-Test commands

Below is Safety Self-Test Command and Address table for your reference.

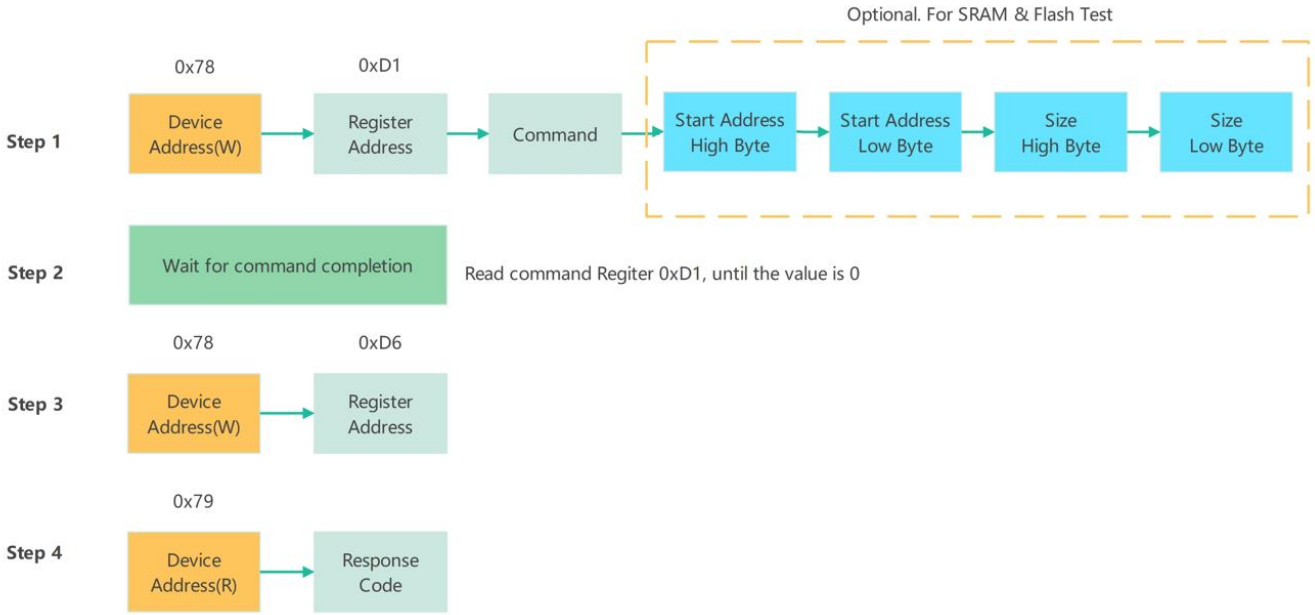
Address	Name	Definition of Parameters		R/W	Default Value
0xD1	Self-Test command	CPU	0x01	R/W	0x00
		PC	0x02		
		Stack	0x03		
		Flash*	0x04		
		SRAM*	0x05		
		Clock	0x06		
		Interrupt	0x07		
		Touch Key	0x08		

0xD2	Start address of Flash and SRAM	Start address high byte		W	
0xD3		Start address low byte			
0xD4	Test size of Flash and SRAM	Test size high byte			
0xD5		Test size low byte			
0xD6	Self-Test result	Pass	0x5A	R	
		Fail	0x01		
* Start address and test size of Flash and SRAM are defined at addresses 0xD2 - 0xD5.					

Safety Self-Test Command & Address Table

Type	Min. Start Address	Max. End Address	Max. Size
SRAM	0x0008	0x7BF	0x7B8
FLASH	0x0000	0xEFF7	0xEFF8

Below are 4 steps to place a self-test command and get the test result.



8.3 Flash test example

User can refer to following sample code for Flash Self-Test.

Flash test example

Device address : 0x78
Start address : 0x0000
Size : 0xEFF8

I2C command sequence

// Switch to page 0
0x78,**0x06,0x00**

// write command, start address high byte, start address low byte,
// test size high byte and test size low byte.
0x78, **0xD1**, **0x04, 0x00, 0x00, 0xEF, 0xF8**

// Wait command completion, repeat read the value of register 0xD1, until the value is zero.
// set address of command (0xD1)
0x78,**0xD1**
// Read result
0x79

// check if the value is 0, if not, repeat to read 0xD1.
// if the value is 0, continue to the next step.

// set address of result (0xD6)
0x78,**0xD6**
// Read the result
0x79

// check the result
// if the result value is 0x5a, test passed.
// if the result value is 0x01, test failed.

9. DEBUG TARGET BOARD VIA GUI SOFTWARE

9.1 Connection Block Diagram

As shown in **Figure 9-1: Block diagram of EzISP Board connected to the custom target board** below, the EzISP board can be connected to the customer target board to configure touch keys through the GUI interface.

Figure 9-1: Block diagram of EzISP Board connected to the custom target board

As shown in **Figure 9-2: Pin configuration on EzISP board** below, the EzISP Board only needs five pins to connect to the custom target board. The five pins are VDD, GND, RSTN, SCL, and SDA.

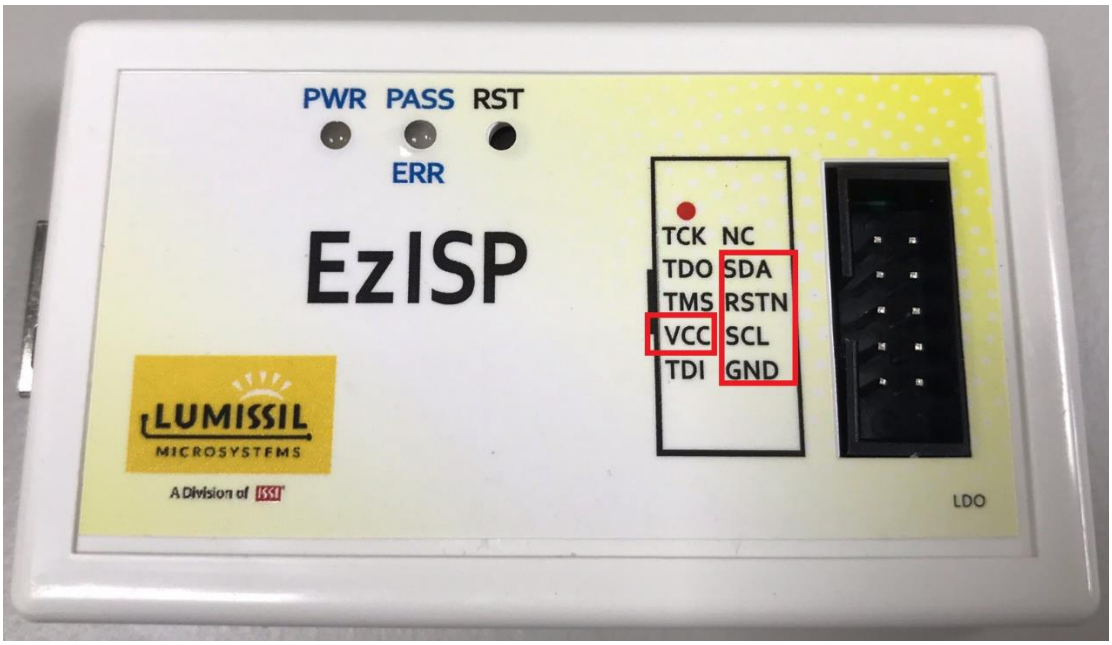


Figure 9-2: Pin configuration on EzISP board

9.2 Customer Target Board Configuration by GUI

Users can run SE5120A Touch Key GUI software for their own target boards to adjust the parameters of the touch keys without having to develop code for the MCU at the beginning. This can shorten the development time.

REVISIONS

Revision	Detailed Information	Date
A	First Formal Release	2022.05.30
B	1. Add RSTN pin along with VDD, GND, SDC and SDL pins for EzISP board connecting with SE5120 board to run SE5120 Touch Key GUI tool. 2. Support two slider groups and each slider with 3~8 keys support. 3. Update IS31SE5120 with IS3xSE5120 except IS31SE5120 EvB order number and EvB board name	2022.07.20
C	Add Section 2.3 Upgrade Boot Code and Standard Code	2022.12.15
D	1. Support SE5120A Touch Key GUI 2.0 2. Add more descriptions for "Detect" and "Connect" statuses handling in 3.2 Connect Status 3. Change export/import Configuration File format from csv to excel. Please refer to 3.3 Export/Import Configuration File 4. Add 3.8 Generate Sample Code 5. Add 4.2.1 System Clock 6. Add information about system clock and duration per touch key scan in 4.2.2 Cycle Count Per Key Scan 7. Add 4.2.3 TK3 Touch Key Clock 8. Add information about how spread spectrum affects IOSC frequency accuracy and EMI in 4.2.9 Spread Spectrum 9. Add more information about Jitter, Media, Average, and IIR filters in 4.3.4 Raw Count Filter 10. Add INT description for Multi-Key enable for Case 3. Click Interrupt Enable, Auto Clear Enable and Repeat Enable in 4.3.5 Interrupt 11. For firmware 2.0, register 7, register 8 and register 9 need to be read to acquire key status and interrupt will be clear after register 9 is read. Please refer to Case1. Click Interrupt Enable of 4.3.5 Interrupt For firmware 1.0, register 6, register 7 and register 8 need to be read to acquire key status and interrupt will be clear after register 8 is read 12. Add Chapter 8 SAFETY SELF-TEST 13. Update SE5120 EVB order information from "IS31SE5120-QFLS3-EBGUI" to "IS31SE5120A-QFLS3-EBGUI" for new firmware 3.0 support 14. Update 2.1 Software Requirements to support Silabs USB driver version 6.7.2.0 and detailed EzISP setup steps and configuration after setup installation. Besides, update Windows software requirement of Microsoft .NET Framework 4.6.2 as Microsoft .NET Framework version 4.8.	2023.06.28