# A96T418GDN Shield Board

## Shield Board Quick Guide

Version 1.12

# Contents

## List of figures

## List of tables

# 1 Introduction

This document introduces software environment and development recommendations for the ABOV 8-bit Touch/LED-Driver MCU starter kit. It offers an easy way to develop the Home Appliance Touch.

1. The starter kit board contains a debugger called "OCD", which eliminates the need for additional devices for MCU programming and debugging.
2. In this Quick Start Guide, we'll discuss how to operate A96T418GDN Shield Board, as an example, step by step

# 2 User requirements

## 2.1 Hardware

### 2.1.1 Starter Kit

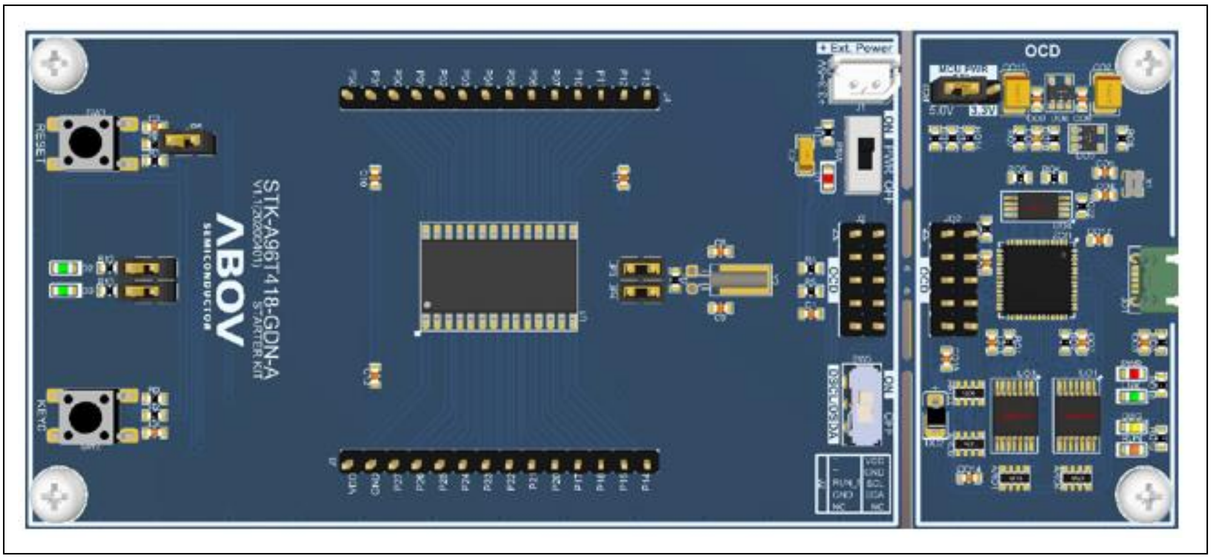


Figure 1. A96T418GDN Starter Kit Board (Hardware)

### 2.1.2 Shield Board

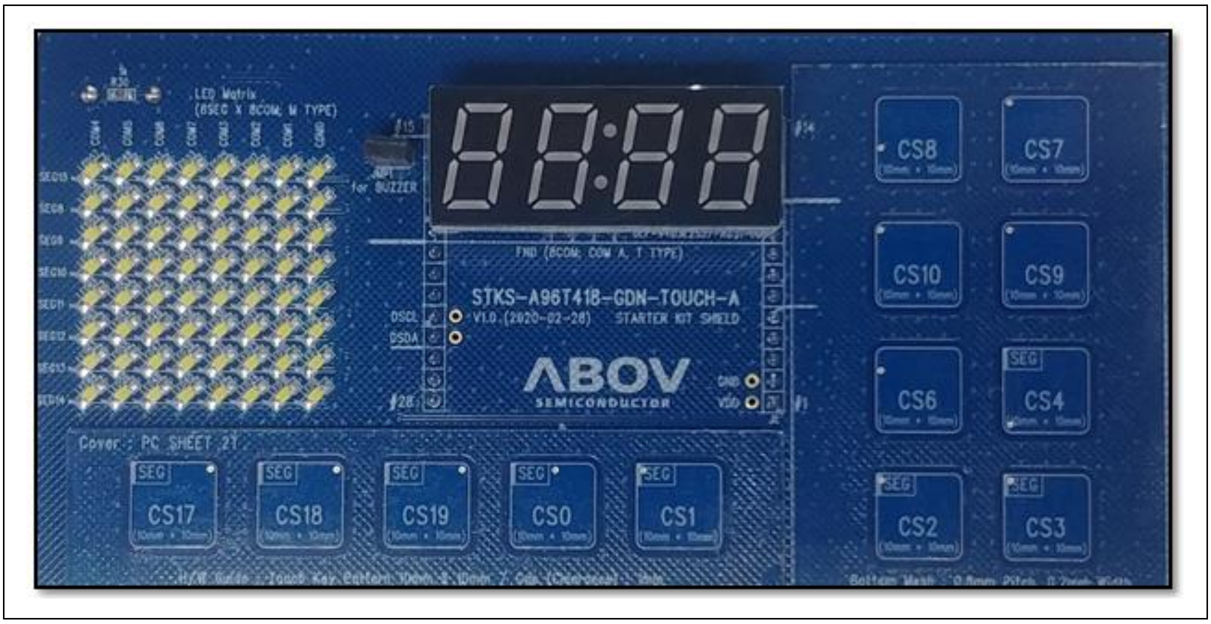
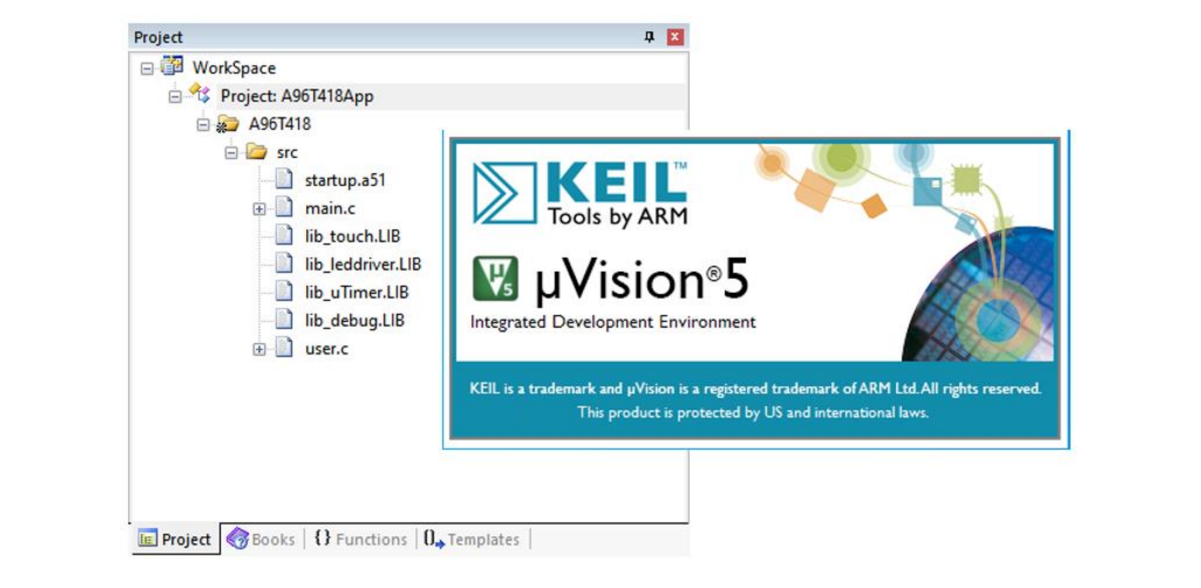


Figure 2. A96T418GDN Shield Board (Hardware)

## 2.2 Software

1. Keil complier (uVision5)
2. EVK (Library) software



```
Project
WorkSpace
  Project: A96T418App
    A96T418
      src
        startup.a51
        main.c
        lib_touch.LIB
        lib_leddriver.LIB
        lib_uTimer.LIB
        lib_debug.LIB
        user.c
```

KEIL™ Tools by ARM

µVision®5

Integrated Development Environment

KEIL is a trademark and µVision is a registered trademark of ARM Ltd. All rights reserved.
This product is protected by US and international laws.

**Figure 3. Compiler (Software)**

## 2.3 Reference documents

1. A96T418 Starter Kit Quick Guide
2. A96T418 Starter Kit HW Manual
3. A96T418 User's Manual
4. T-TYPE LED Structure
5. Touch Basic Algorithm
6. Example code

For detailed information, refer to the categories below.

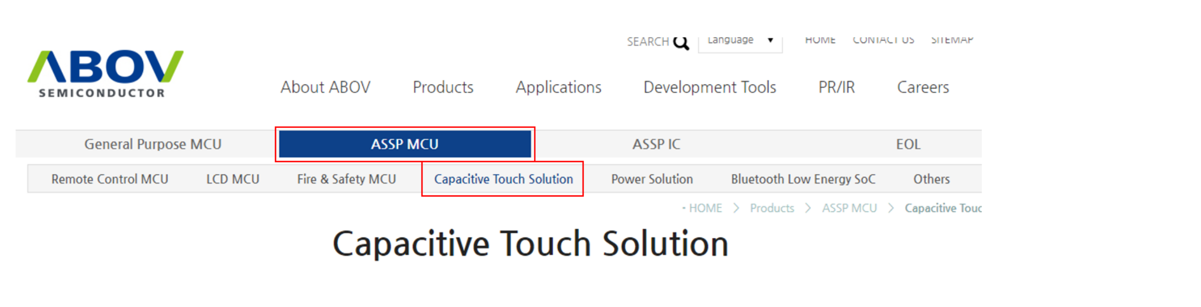


**Figure 4. Reference Documents**

## 2.4 System requirements

1. Windows PC (7, 8, 10)
2. USB micro-B type cable

Figure 5. Window PC & Mini-B cable

## 2.5 ABOV website

For detailed information about corresponding software and documents, you can visit our website at https://www.abovsemi.com.

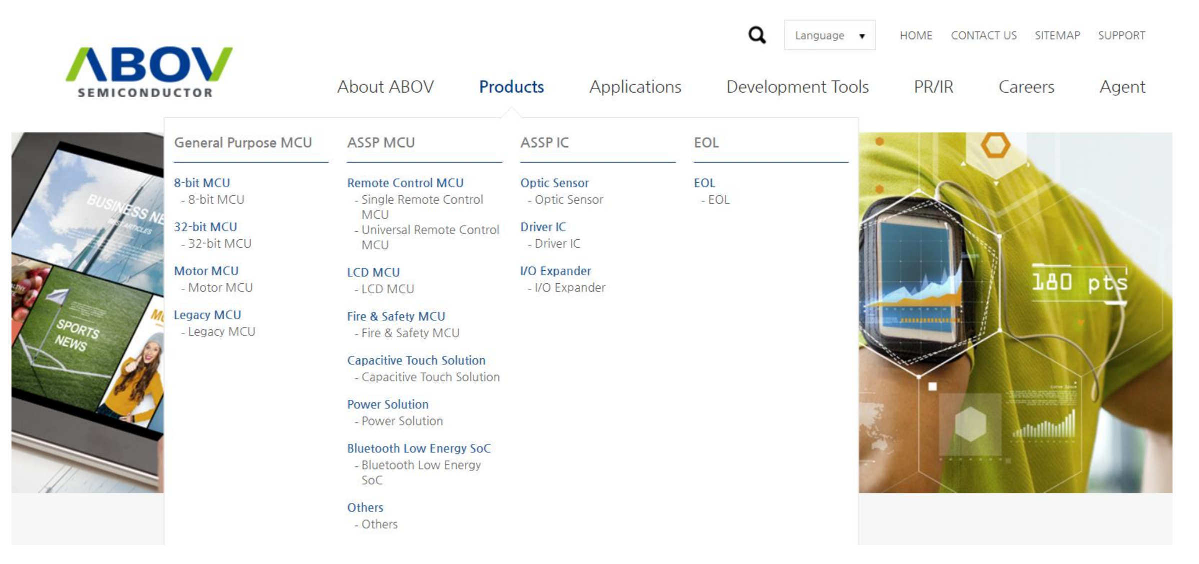


Figure 6. ABOV Semiconductor Website

# 3 Building and running project (Shield Board)

Running application code makes it easier to start the Shield Board. Following the steps below:

Step 1. Prepare the Starter kit & Shield Board

Step 2. Set up the Starter kit & Shield Board

Step 3. Connect the Starter kit to your PC

Step 4. STKS Practice

Step 5. Library description

## 3.1 Prepare the Starter kit & Shield Board

**Hardware components of the Starter Kit**

1. The starter kit board is comprised of a device board and an OCD board. The two boards can be separated if necessary.
2. Device board configuration
   A. ABOV 8-bit MCU
   B. Pin Headers connected to MCU
   C. LED, switch and jumper to check input/output, reset, and debugger pins
3. OCD board configuration
   A. OCD for programming and debugging with ABOV 8-bit MCU(connected to USB port of PC)
   B. Supports OCD Disconnection Mode (In this mode, OCD DSCL and DSDA port can be used as UART TX and UART RX, respectively)

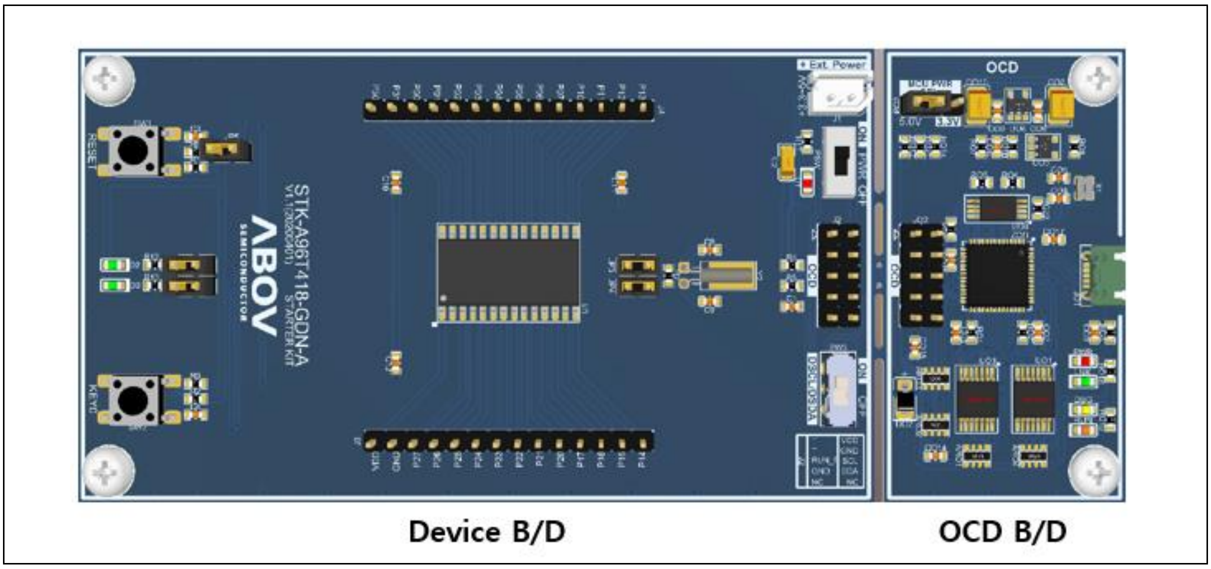


**Figure 7. Starter Kit B/D**

## Hardware components of the Shield Board

Table 1. Shield Board Description

| Function | Description | Remark |
|---|---|---|
| CS6,7,8,9,10 | User SW | Touch Only Pin |
| CS0,1,2,3,4,17,18,19 | User SW | Touch & SEG Pin |
| LED T-Type Segment | User SW | COM0/SEG0~COM7/SEG7 |
| LED M-Type Matrix | User SW | COM0~COM7, SEG8~SEG15 |
| BUZZER | PWM | P11/PWM1O |

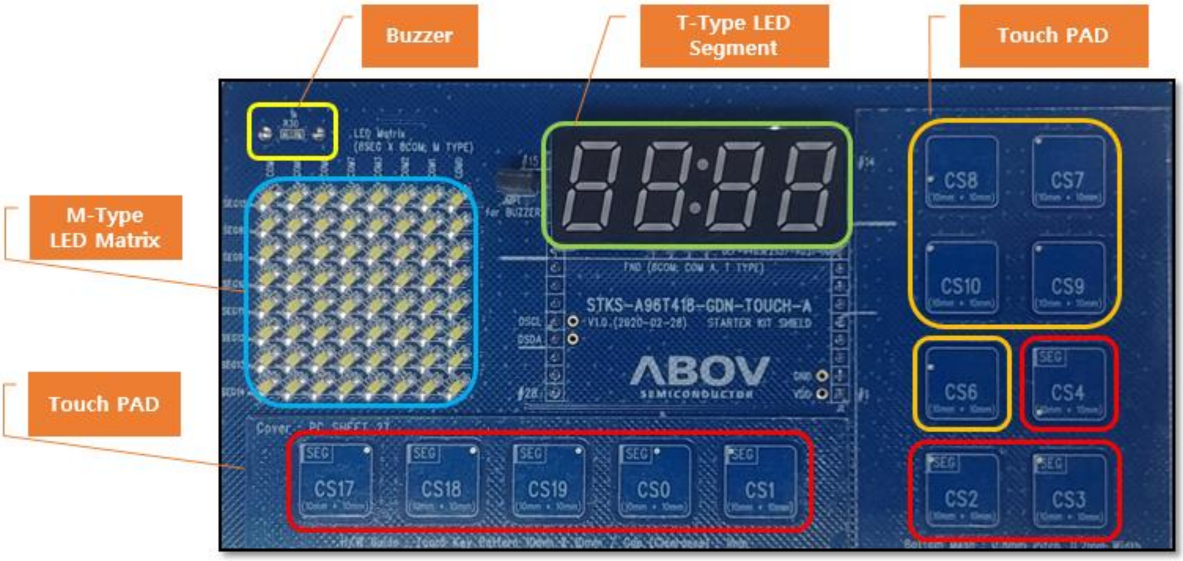


Figure 8. A96T418GDN Shield Board Description

## 3.2 Set up the Starter kit & Shield Board

**Set jumpers to control the Starter Kit**

1. USB 5V Power Output On/Off Jumper
2. OCD/UART Selection Switch
3. External 5V
4. Device Board(A96T418 IC included) Power On/Off Switch
5. OCD_DSCL/OCD_DSDA or UART_TX/UART_RX (Depends on ②OCD/UART SEL SW)
6. JP3~JP7

   : It must be open for touch use.

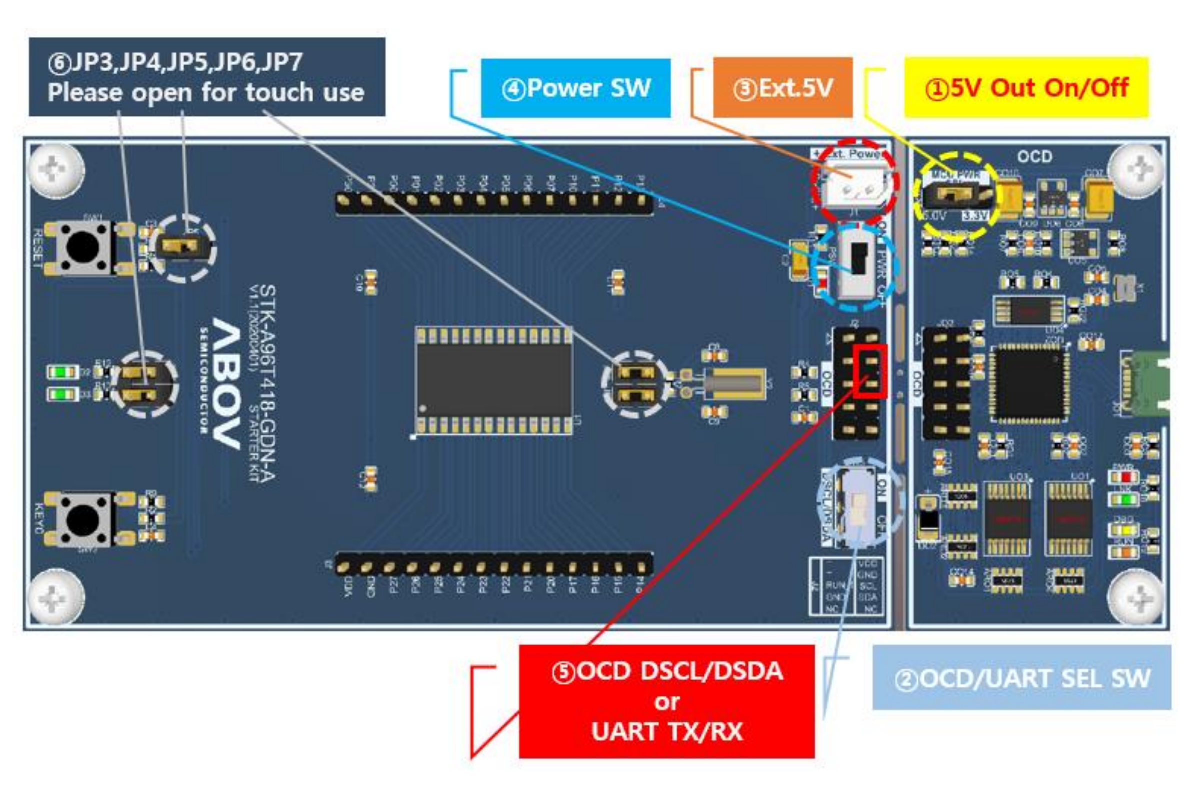


**Figure 9. Starter Kit Jumper & Switch**

## Connect to Starter Kit to use Shield Board

Connect the header socket on the Shield board on the header pin of the Starter Kit Board. As shown in the following Figure 10.

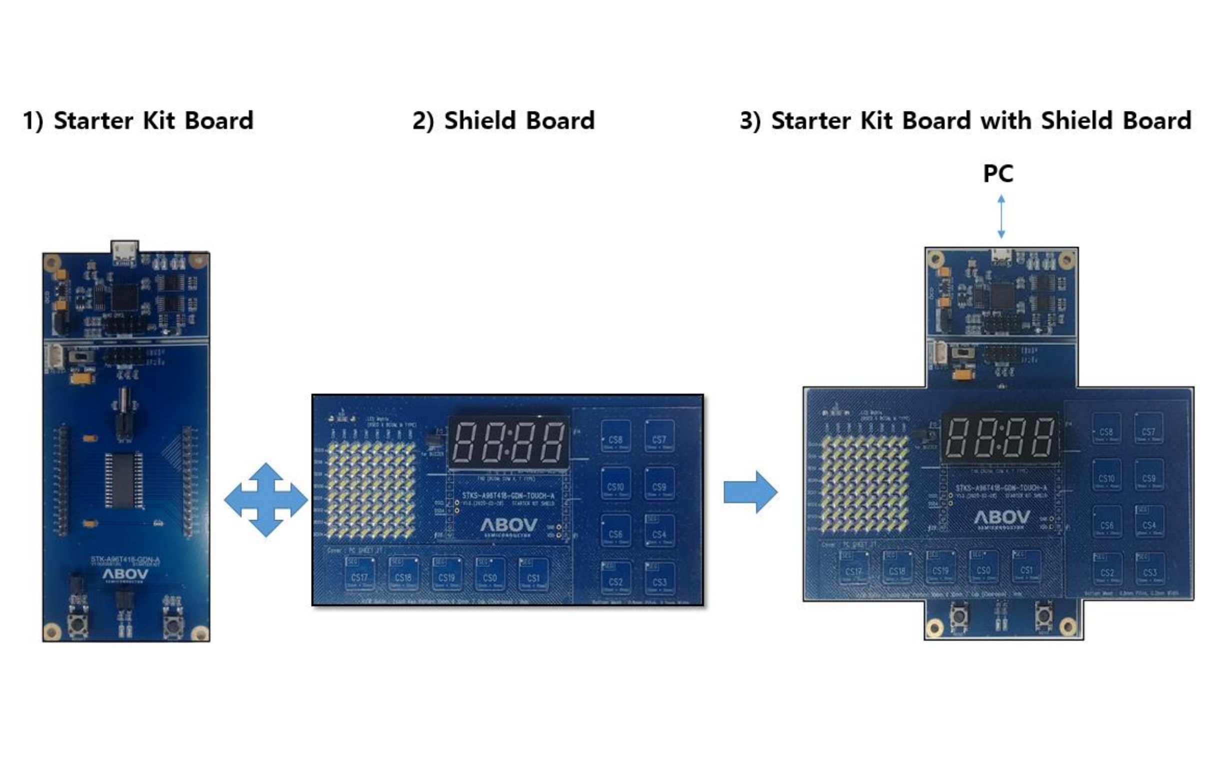


**Figure 10. Connect to Starter Kit to use Shield Board**

## 3.3 Connect the Starter Kit to your PC

**PC connection via USB on the Starter Kit**

Connect the USB 2.0 (micro-B type) cable to the Starter Kit as shown in Figure 11.

1. If the PC is properly connected, the USB Power LED will be turned on. -①
2. If the power switch is moved in the direction shown below, the OCD detects the IC (A96T418) and the LED is turned on. -②

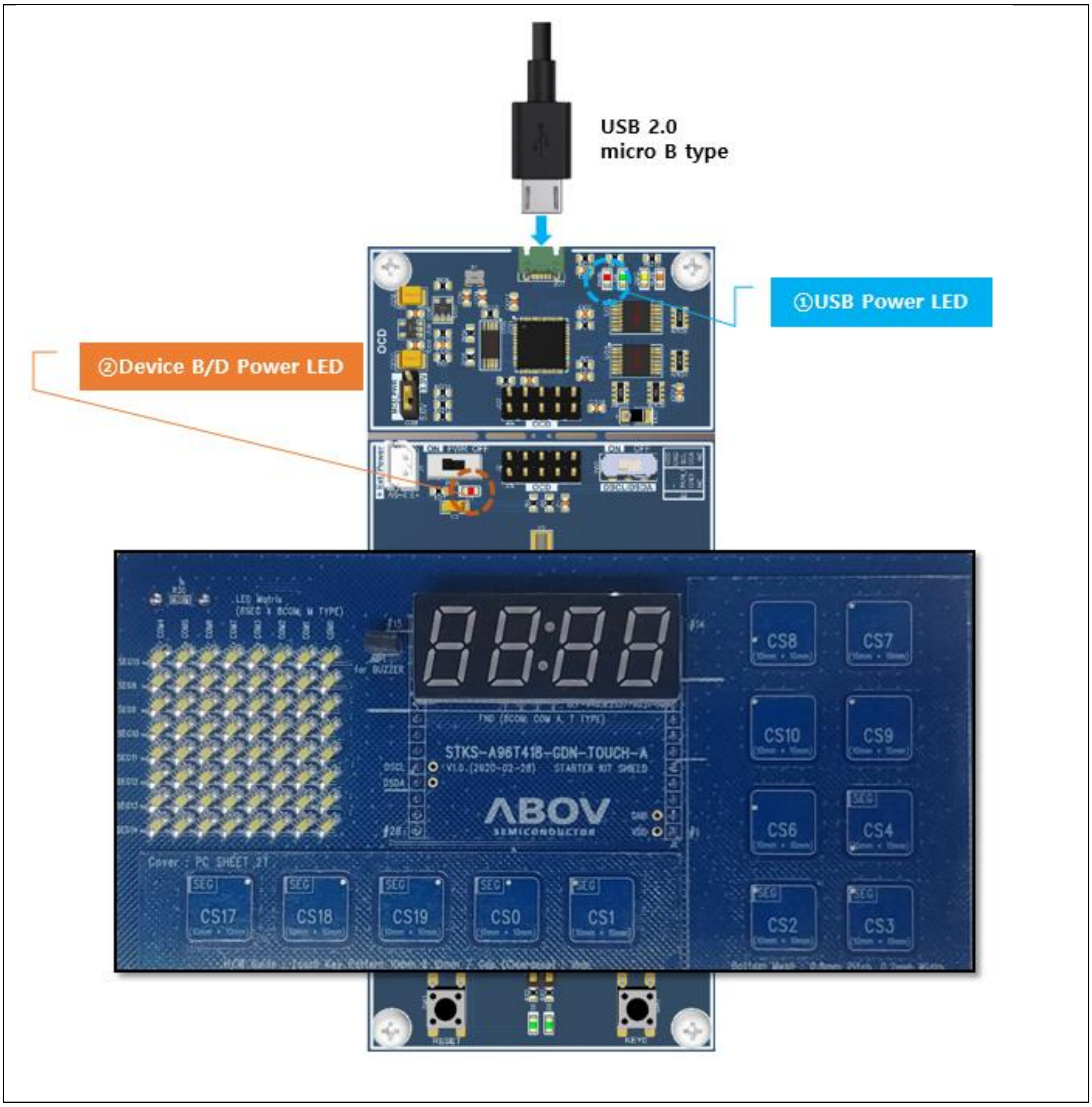
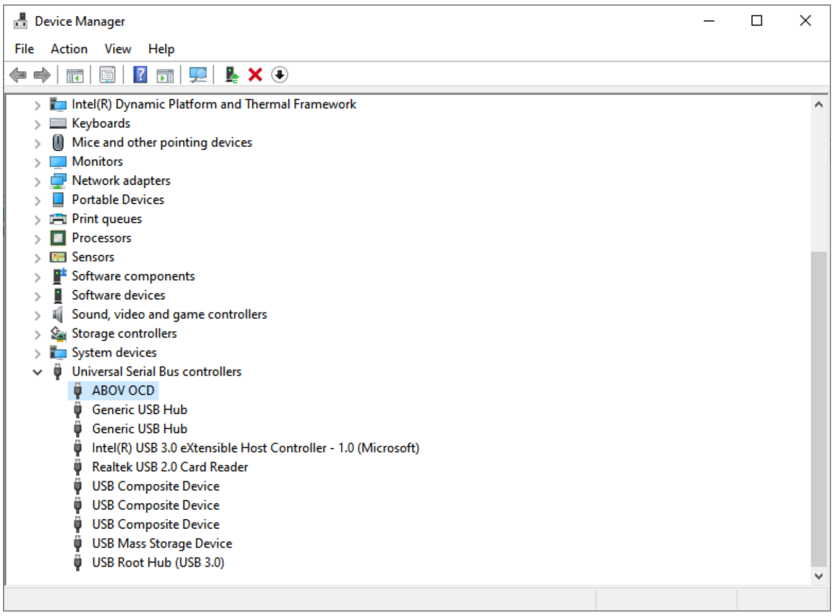


**Figure 11. Starter Kit and USB Connection**

When OCD is connected, it is displayed as **ABOV OCD** in Device Manager.

1. OCD is a Universal Serial Bus(USB) controllers class and does not require driver installation on Microsoft Windows



- Intel(R) Dynamic Platform and Thermal Framework
- Keyboards
- Mice and other pointing devices
- Monitors
- Network adapters
- Portable Devices
- Print queues
- Processors
- Sensors
- Software components
- Software devices
- Sound, video and game controllers
- Storage controllers
- System devices
- Universal Serial Bus controllers
  - ABOV OCD
  - Generic USB Hub
  - Generic USB Hub
  - Intel(R) USB 3.0 eXtensible Host Controller - 1.0 (Microsoft)
  - Realtek USB 2.0 Card Reader
  - USB Composite Device
  - USB Composite Device
  - USB Composite Device
  - USB Mass Storage Device
  - USB Root Hub (USB 3.0)

**Figure 12. ABOV OCD at Device Manager**

## 3.4 STKS Practice

### 3.4.1 Compile with Keil uVision5 for C51

1. Install “Keil uVision5 for C51”.
2. Open the project file provided (A96T418Proj.uvmpw).

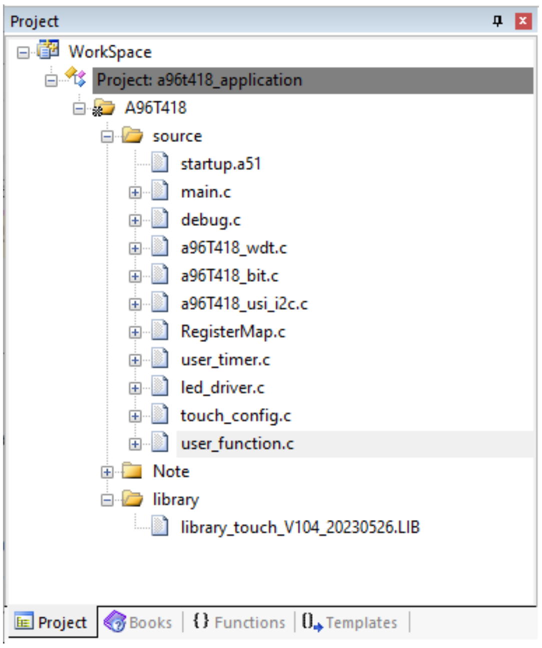
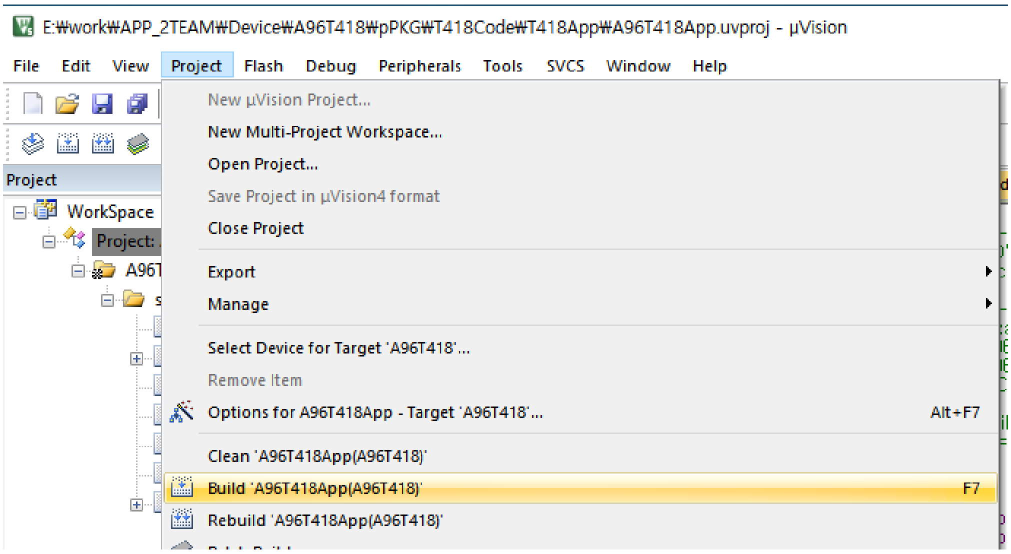


**Figure 13. Execute Keil uVision5 for C51**

3. Build the project


E:₩work₩APP_2TEAM₩Device₩A96T418₩pPKG₩T418Code₩T418App₩A96T418App.uvproj - µVision
File
Edit
View
Project
Flash
Debug
Peripherals
Tools
SVCS
Window
Help
Project
WorkSpace
Project:
A96T
New µVision Project...
New Multi-Project Workspace...
Open Project...
Save Project in µVision4 format
Close Project
Export
Manage
Select Device for Target 'A96T418'...
Remove Item
Options for A96T418App - Target 'A96T418'... Alt+F7
Clean 'A96T418App(A96T418)'
Build 'A96T418App(A96T418)' F7
Rebuild 'A96T418App(A96T418)'

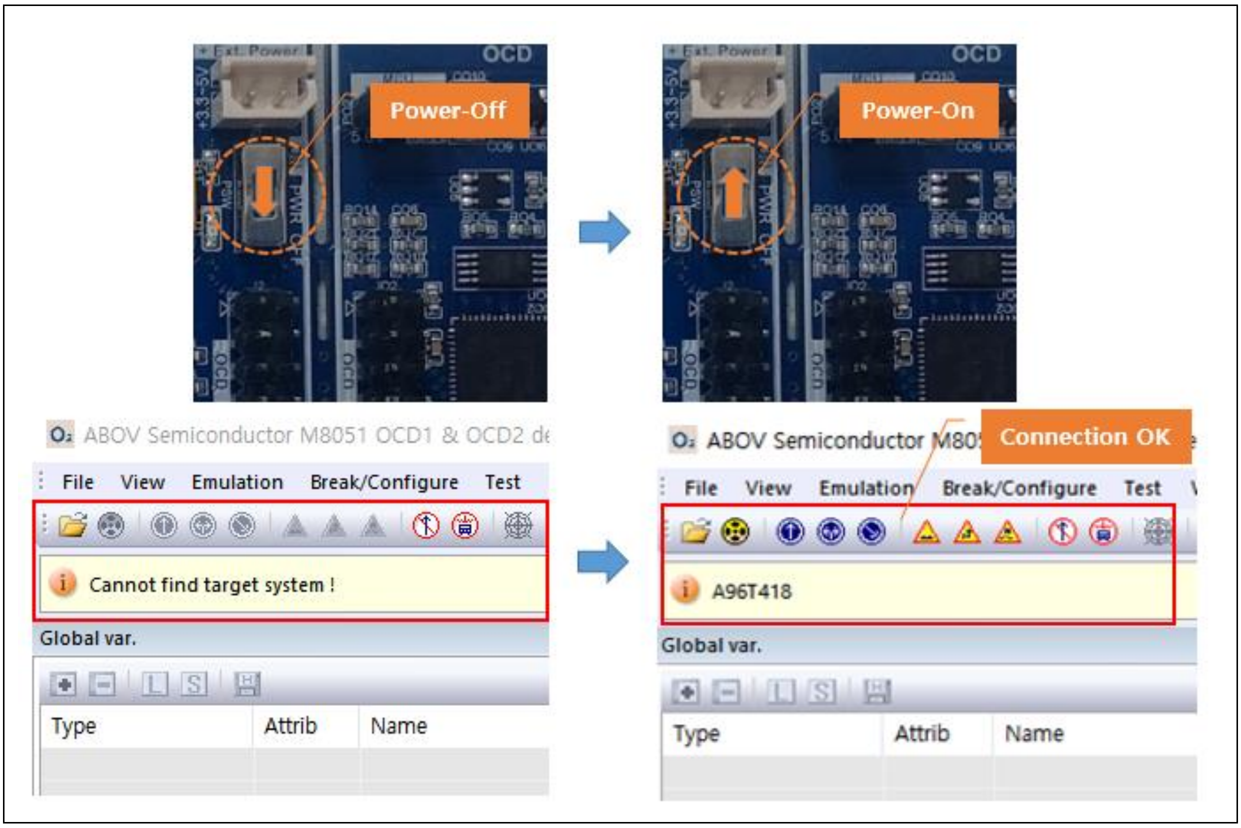
After build, t418code.hex file is created in “T418Code₩T418App₩out” folder

**Figure 14. Build the project**

### 3.4.2 Download & Run with OCD

1. Connection

: To connect the OCD, move the power switch in the direction shown below.



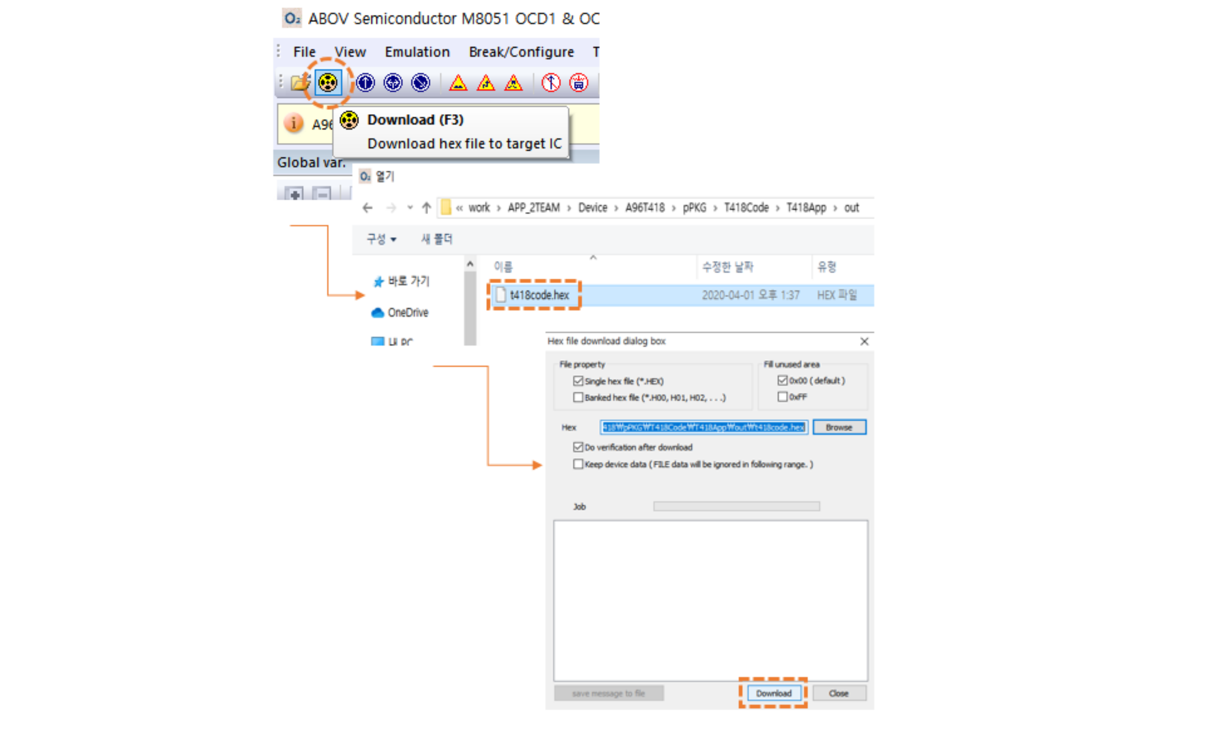
Cannot find target system !

A96T418

**Figure 15. OCD Connection**

2. Download

: Download the hex file (t418code.hex) as shown below.


ABOV Semiconductor M8051 OCD1 & OC
File View Emulation Break/Configure T
A96
Global var.
열기
« work › APP_2TEAM › Device › A96T418 › pPKG › T418Code › T418App › out
구성 ▾ 새 폴더
바로 가기
OneDrive
UI pr
이름
수정한 날짜
유형

Download (F3)
Download hex file to target IC

t418code.hex 2020-04-01 오후 1:37 HEX 파일

Hex file download dialog box

File property
☑ Single hex file (*.HEX)
□ Banked hex file (*.H00, H01, H02, . . .)

Fill unused area
☑ 0x00 ( default )
□ 0xFF

Hex: 418₩pPKG₩T418Code₩T418App₩out₩t418code.hex

☑ Do verification after download
□ Keep device data ( FILE data will be ignored in following range. )

Job

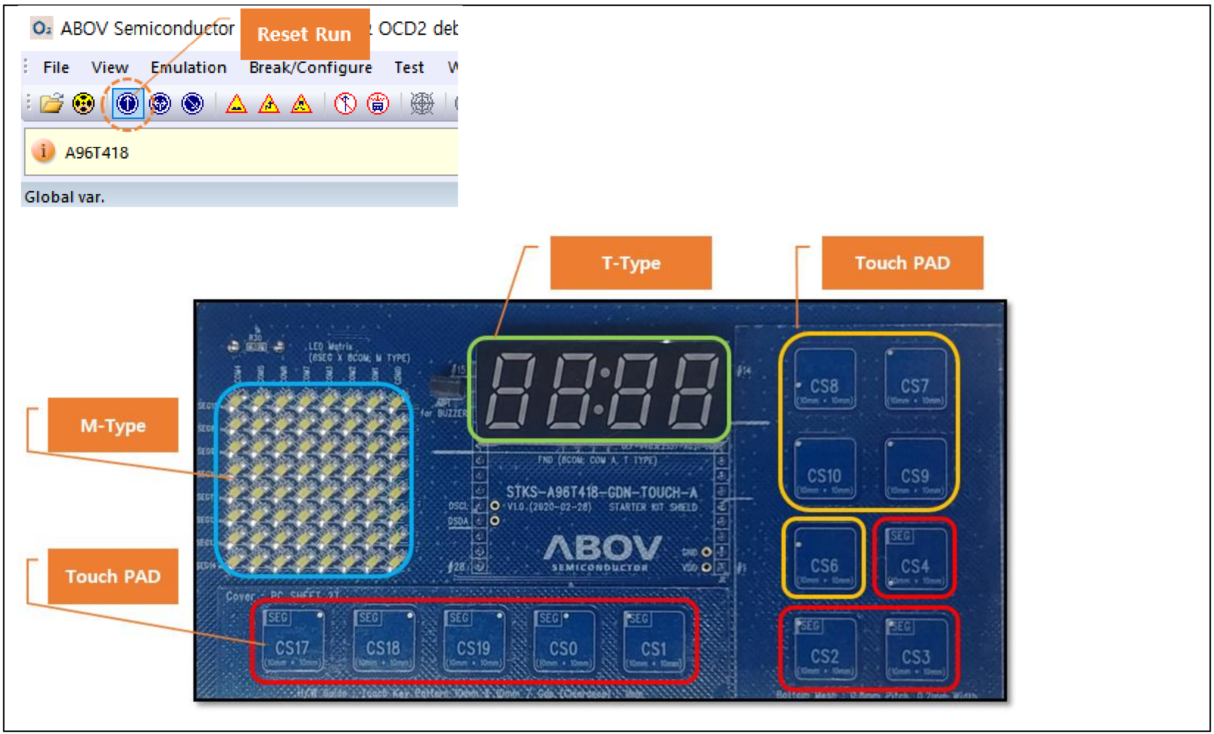
Browse
save message to file
Download
Close

**Figure 16. OCD Connection**

A. Display on LED-Matrix(M-type) & LED-Segment(T-type)

B. Display on LED-Matrix with the rules of 0,1,2,3,4 …

C. Display on LED-Segment with the rules of 1000,2000,4000,8000,0100 …



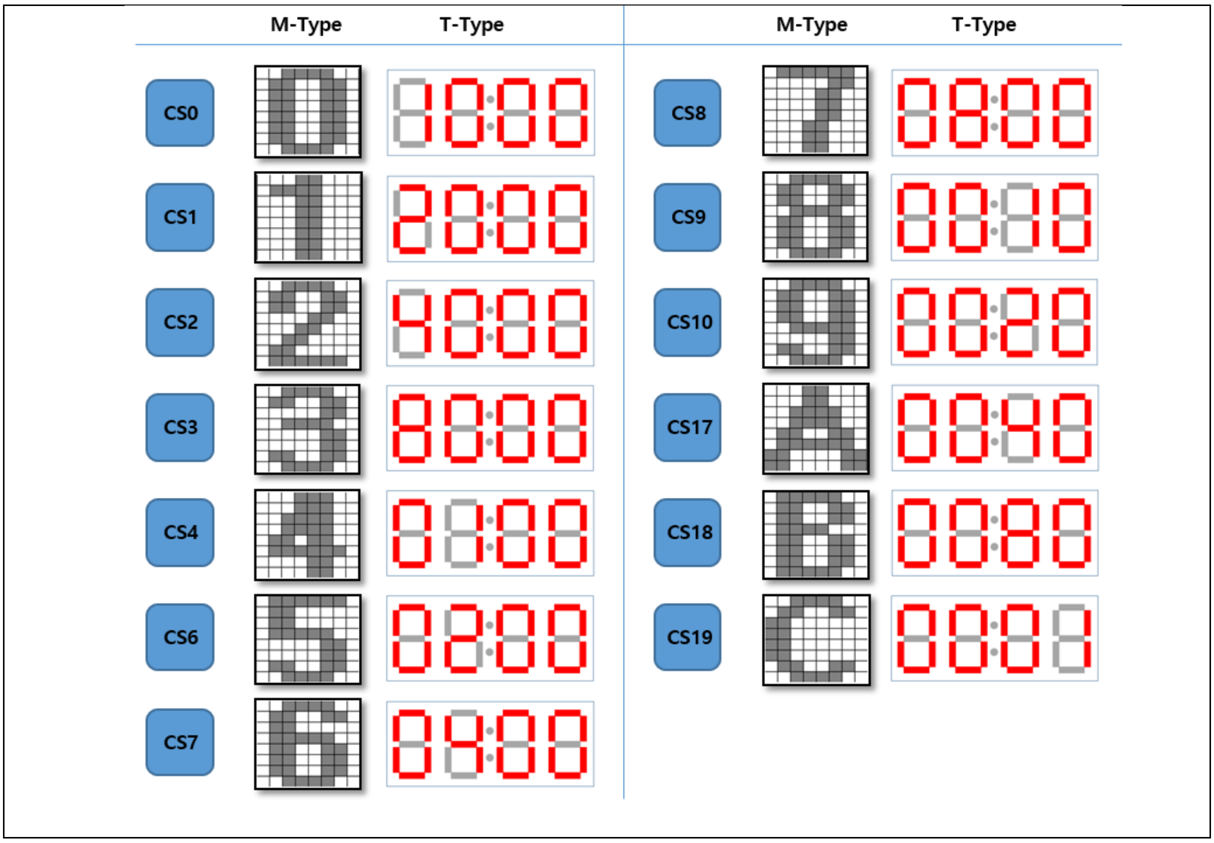


Figure 17. Touch & Display

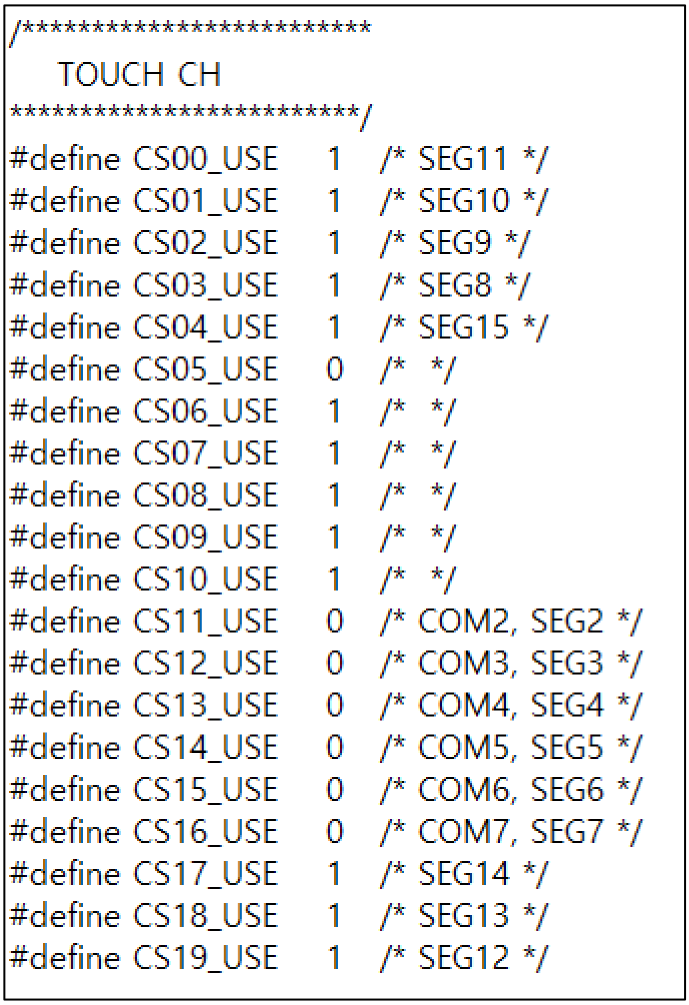
# 4 Function description (Using Touch library)

## 4.1 Touch Parameter Description

### 4.1.1 touch_config.h

#### 4.1.1.1 #define CSXX_USE X

— If you used touch channel, set X=1, otherwise set X=0.

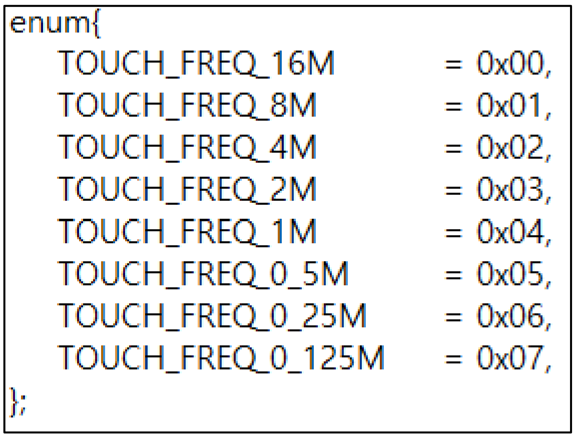


```
/*************************
   TOUCH CH
*************************/
#define CS00_USE    1   /* SEG11 */
#define CS01_USE    1   /* SEG10 */
#define CS02_USE    1   /* SEG9 */
#define CS03_USE    1   /* SEG8 */
#define CS04_USE    1   /* SEG15 */
#define CS05_USE    0   /*  */
#define CS06_USE    1   /*  */
#define CS07_USE    1   /*  */
#define CS08_USE    1   /*  */
#define CS09_USE    1   /*  */
#define CS10_USE    1   /*  */
#define CS11_USE    0   /* COM2, SEG2 */
#define CS12_USE    0   /* COM3, SEG3 */
#define CS13_USE    0   /* COM4, SEG4 */
#define CS14_USE    0   /* COM5, SEG5 */
#define CS15_USE    0   /* COM6, SEG6 */
#define CS16_USE    0   /* COM7, SEG7 */
#define CS17_USE    1   /* SEG14 */
#define CS18_USE    1   /* SEG13 */
#define CS19_USE    1   /* SEG12 */
```

**Figure 18. Touch CH activation (in touch_config.h)**

#### 4.1.1.2 #define TOUCH_CLK_VAL [n]

- Touch sensing clock divide select.
- Default setting : TOUCH_FREQ_4M (0x02)



```
enum{
    TOUCH_FREQ_16M        = 0x00,
    TOUCH_FREQ_8M         = 0x01,
    TOUCH_FREQ_4M         = 0x02,
    TOUCH_FREQ_2M         = 0x03,
    TOUCH_FREQ_1M         = 0x04,
    TOUCH_FREQ_0_5M       = 0x05,
    TOUCH_FREQ_0_25M      = 0x06,
    TOUCH_FREQ_0_125M     = 0x07,
};
```

**Figure 19. Touch clock setting value (in touch_lib.h)**

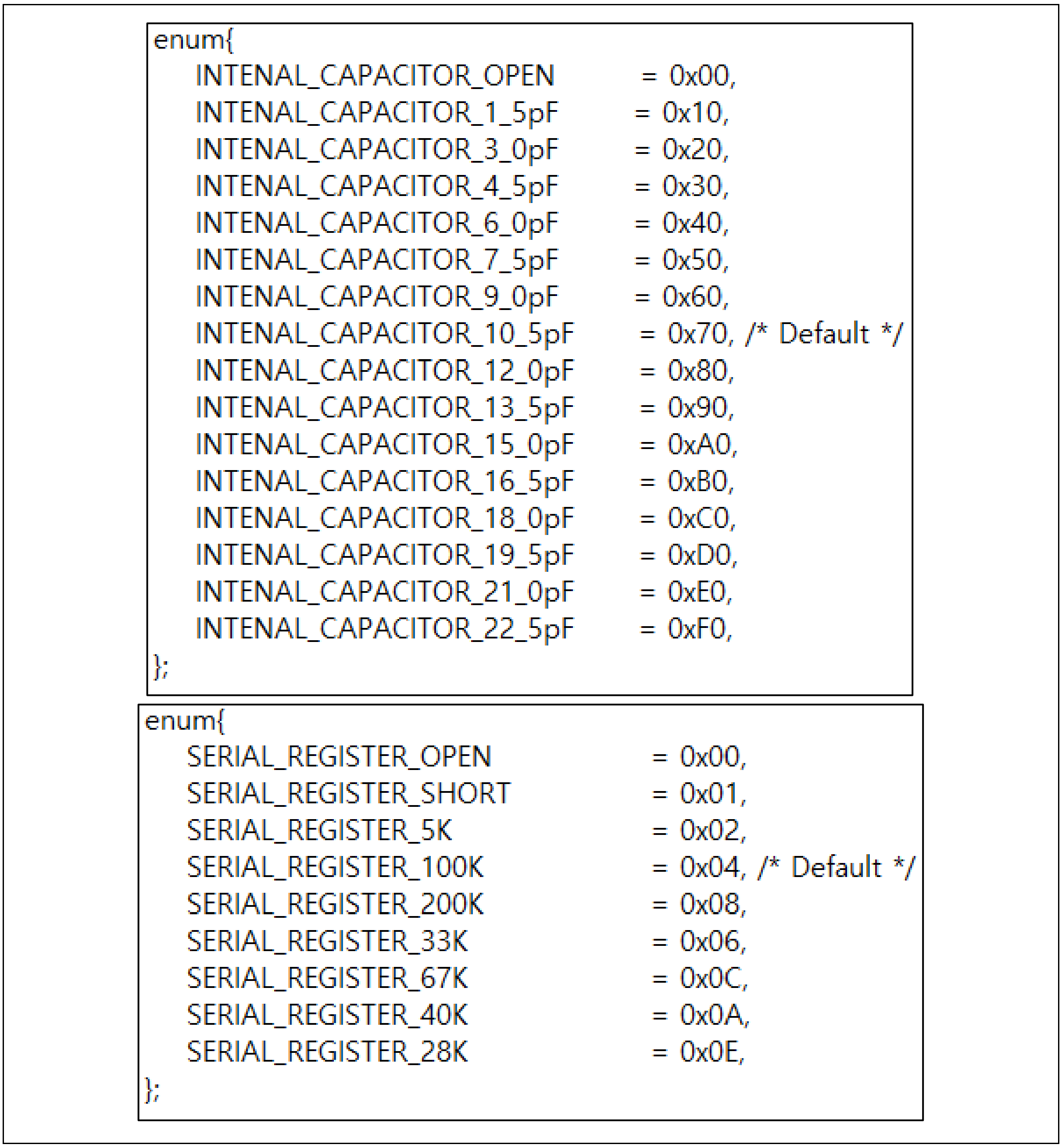
#### 4.1.1.3 #define TOUCH_CLK_OFFSET_VAL [n]

- Basically, the touch sensor performs touch sensing twice at two frequencies. Set the offset between the two touch frequencies with the value of "TOUCH_CLK_OFFSET_VAL".
- Setting range [0H ~ 3FH]
- Frequency calculation formula
  If (20H + TOUCH_CLK_OFFSET_VAL) < 40H
  : 16MHz(Touch Frequency) *(1+(TOUCH_CLK_OFFSET_VAL*0.75%))
  If (20H + TOUCH_CLK_OFFSET_VAL) >= 40H
  : 16MHz(Touch Frequency) *(1+(TOUCH_CLK_OFFSET_VAL-40H)*0.7%))
- Default setting : 4H (4.12Mhz)
  : 4MHz *(1+((4H)*0.75%)) = 4.12Mhz, When TOUCH_CLOCK_VAL is TOUCH_FREQ_4M
  : Freq 0 : 4.12Mhz, Freq 1 : 4Mhz

#### 4.1.1.4 #define TOUCH_LPF_C_VAL [n]

#### 4.1.1.5 #define TOUCH_LPF_R_VAL [n]

— Internal Low pass filter value setting parameters



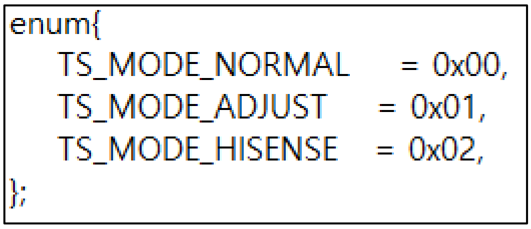
```
enum{
    INTENAL_CAPACITOR_OPEN          = 0x00,
    INTENAL_CAPACITOR_1_5pF         = 0x10,
    INTENAL_CAPACITOR_3_0pF         = 0x20,
    INTENAL_CAPACITOR_4_5pF         = 0x30,
    INTENAL_CAPACITOR_6_0pF         = 0x40,
    INTENAL_CAPACITOR_7_5pF         = 0x50,
    INTENAL_CAPACITOR_9_0pF         = 0x60,
    INTENAL_CAPACITOR_10_5pF        = 0x70, /* Default */
    INTENAL_CAPACITOR_12_0pF        = 0x80,
    INTENAL_CAPACITOR_13_5pF        = 0x90,
    INTENAL_CAPACITOR_15_0pF        = 0xA0,
    INTENAL_CAPACITOR_16_5pF        = 0xB0,
    INTENAL_CAPACITOR_18_0pF        = 0xC0,
    INTENAL_CAPACITOR_19_5pF        = 0xD0,
    INTENAL_CAPACITOR_21_0pF        = 0xE0,
    INTENAL_CAPACITOR_22_5pF        = 0xF0,
};
```

```
enum{
    SERIAL_REGISTER_OPEN            = 0x00,
    SERIAL_REGISTER_SHORT           = 0x01,
    SERIAL_REGISTER_5K              = 0x02,
    SERIAL_REGISTER_100K            = 0x04, /* Default */
    SERIAL_REGISTER_200K            = 0x08,
    SERIAL_REGISTER_33K             = 0x06,
    SERIAL_REGISTER_67K             = 0x0C,
    SERIAL_REGISTER_40K             = 0x0A,
    SERIAL_REGISTER_28K             = 0x0E,
};
```

**Figure 20. Low Pass Filter Register and Capacitor setting value (in touch_lib.h)**

#### 4.1.1.6 #define TOUCH_MODE_VAL [n]

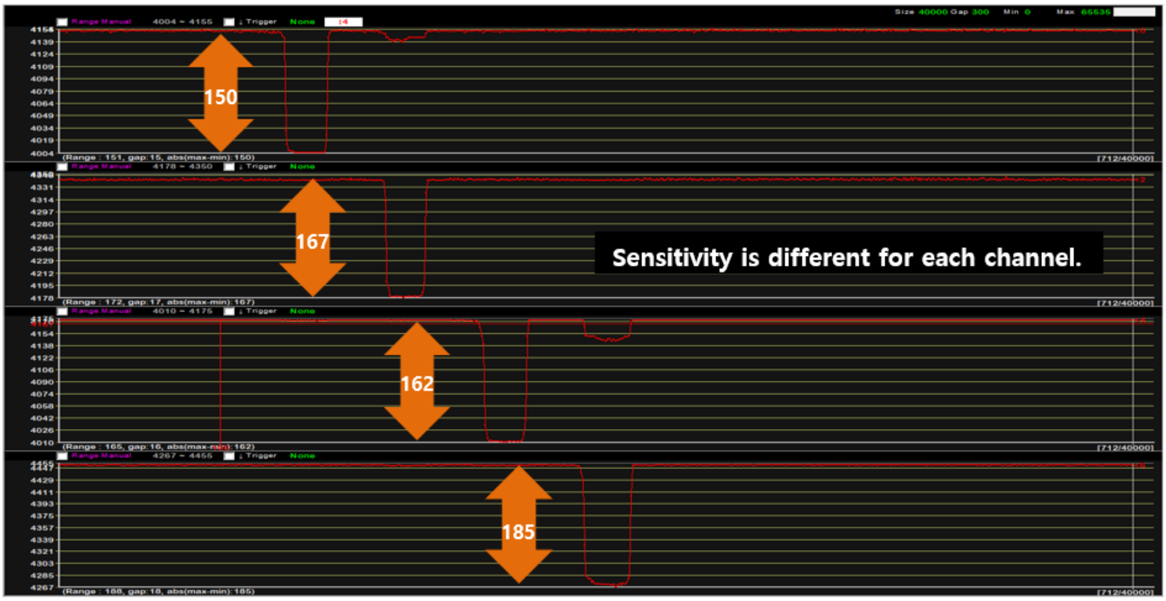
— Touch sensing mode setting value



```
enum{
    TS_MODE_NORMAL     = 0x00,
    TS_MODE_ADJUST     = 0x01,
    TS_MODE_HISENSE    = 0x02,
};
```

**Figure 21. Touch sensing mode setting value (in touch_lib.h)**

— TS_MODE_NORMAL
: In this mode, the parasitic capacitance value for each channel is not adjusted.
: The most common mode of operation. Parameter settings are applied to each channel.
: The sensitivity and threshold are set according to the H/W conditions of each channel.
: Sensitivity for each channel is different.
: Since the sensitivity is different for each channel, the threshold should be applied for each channel.



Sensitivity is different for each channel.

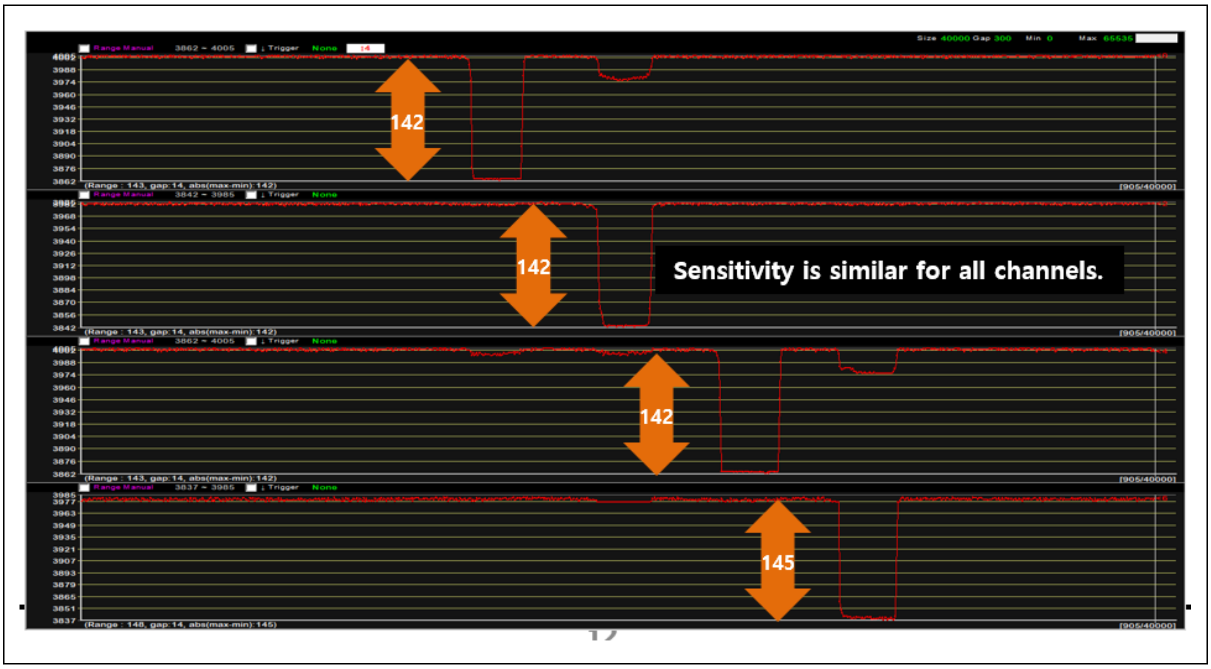
**Figure 22. Sensitivity for each channel in normal mode**

— TS_MODE_ADJUST

: It is a mode that automatically adjusts the capacitance value of each channel to be the same using the IC internal capacitance.

: The capacitance values for each channel are adjusted equally, so the sensitivity for each channel is adjusted similarly.

: However, the reference channel is the channel with the lowest sensitivity (the channel with the highest capacitance), so you should consider the H/W conditions of each channel.


142
142
142
145
(Range : 143, gap: 14, abs(max-min): 142)
(Range : 148, gap: 14, abs(max-min): 145)

Sensitivity is similar for all channels.

**Figure 23. Sensitivity for each channel in normal mode**

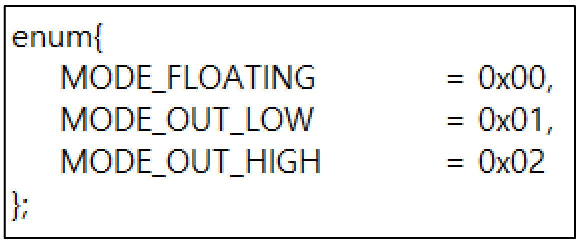
— TS_MODE_HISENSE

: If the touch sensitivity is insufficient in Normal mode, use Hisense mode.

: However, using this feature can make you more vulnerable to noise and take longer to sense than in Normal mode.

#### 4.1.1.7 #define TOUCH_PORT_SEL_VAL [n]

- Touch sensing channel port setting
- Default output Low



```
enum{
    MODE_FLOATING        = 0x00,
    MODE_OUT_LOW         = 0x01,
    MODE_OUT_HIGH        = 0x02
};
```

**Figure 24. Touch sensing channel port setting value (in touch_lib.h)**

#### 4.1.1.8 #define TOUCH_VHS_VAL [n]

- Applies to 'Hisense Mode' only
- A value greater than the value of ' TOUCH_COMP_REFV_VAL ' must be set.
- Setting range (3000 ~ 5000) : default 4000

#### 4.1.1.9 #define TOUCH_COMP_REFV_VAL [n]

- the actual use area should only be used for an area larger than VDD/2.
- Setting range (2700 ~ 4500) : default 3700

### 4.1.1.10 #define DETECT_DEBOUNCE_CNT_VAL [n]

### 4.1.1.11 #define RELEASE_DEBOUNCE_CNT_VAL [n]

- Debounce is used to remove glitch noise.
- If the debounce count is too large, it takes a lot of time for touch recognition and release recognition, so you need to set the appropriate value.
- Setting range (0 ~ 5) : default 0

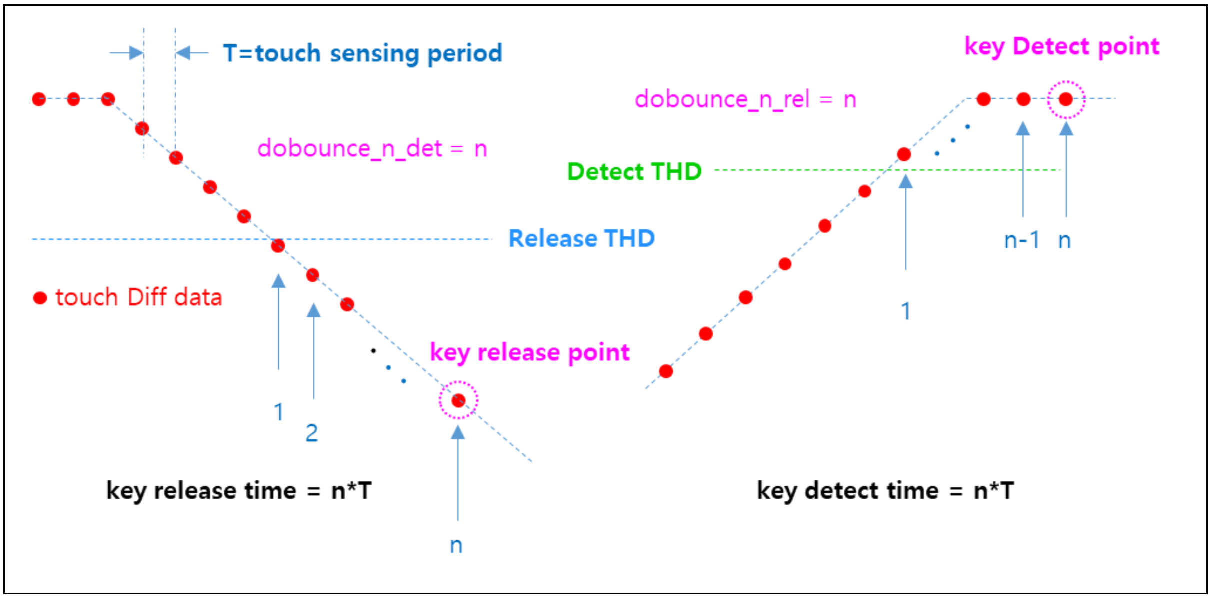


Figure 25. Debounce Count

### 4.1.1.12 #define FILTER_IIR_GAIN_VAL [n]

- This function sets offset value of the IIR filter.
- data(n) = (data(n-1) * offset_cnt + data(n)) / (offset_cnt + 1)
- The recommended setting is 1, which may affect operability if set too high.
- Setting range (0 ~ 5) : default 0

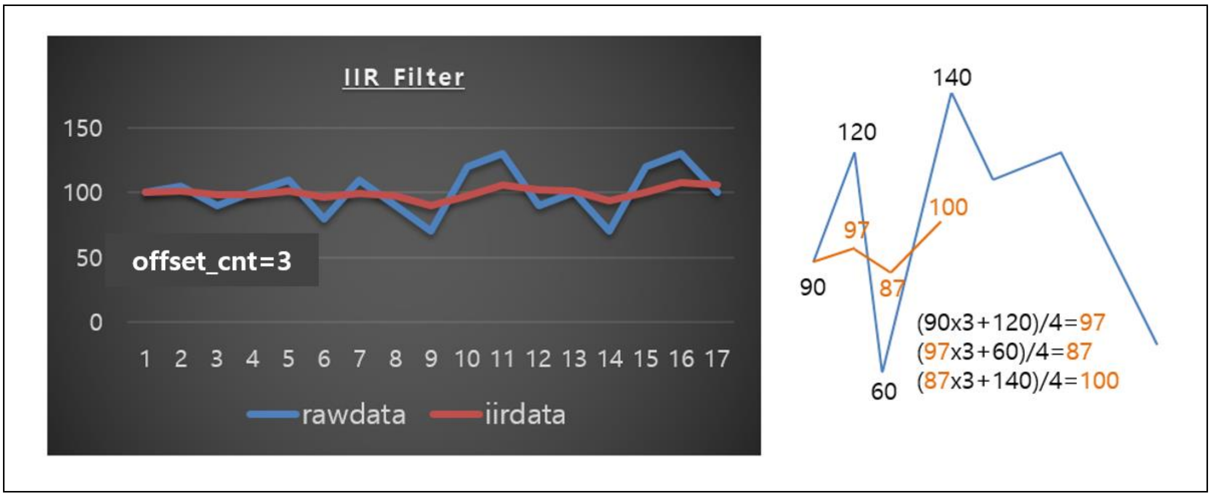


Figure 26. IIR-Filter

#### 4.1.1.13 #define TIMER_STARTUP_VAL [n]

- Set stabilization time of touch sensor after IC power on (unit : 1 ms)
- Generate Touch Key values after a set time.
- Setting range (100 ~ 500) : default 300

#### 4.1.1.14 #define TIMER_BASE_SAMPLE_VAL [n]

- Baseline update cycle setting (unit : 1 ms)
- Setting range (200 ~ 1000) : default 500

#### 4.1.1.15 #define BASE_RELEASE_HOLD_CNT_VAL [n]

- Baseline is not updated as many as [n] frames during touch release.
- Set for stabilization of touch data.
- Setting range (0 ~ 5) : default 3

### 4.1.1.16 #define TIMER_REVERSE_BASE_SAMPLE_VAL [n]

- This feature prevents malfunction when Rawdata is outside the baseline.
- This function sets the reverse holding time (unit: ms).
- After this time, rawdata and basedata are matched.
- The default setting is 100ms, setting it too long or too short may affect operability.
- Setting range (100 ~ 500) : default 100

### 4.1.1.17 #define THD_REVERSE_RATE_VAL [n]

- This function sets the level for key reverse situation (rawdata > basedata+α) judgement.
- Key reverse situation level is rate % of the THD (threshold).
- By default, it is set to 50% of THD, too high or too low may affect touch behavior.
- Setting range (50 ~ 200) : default 50

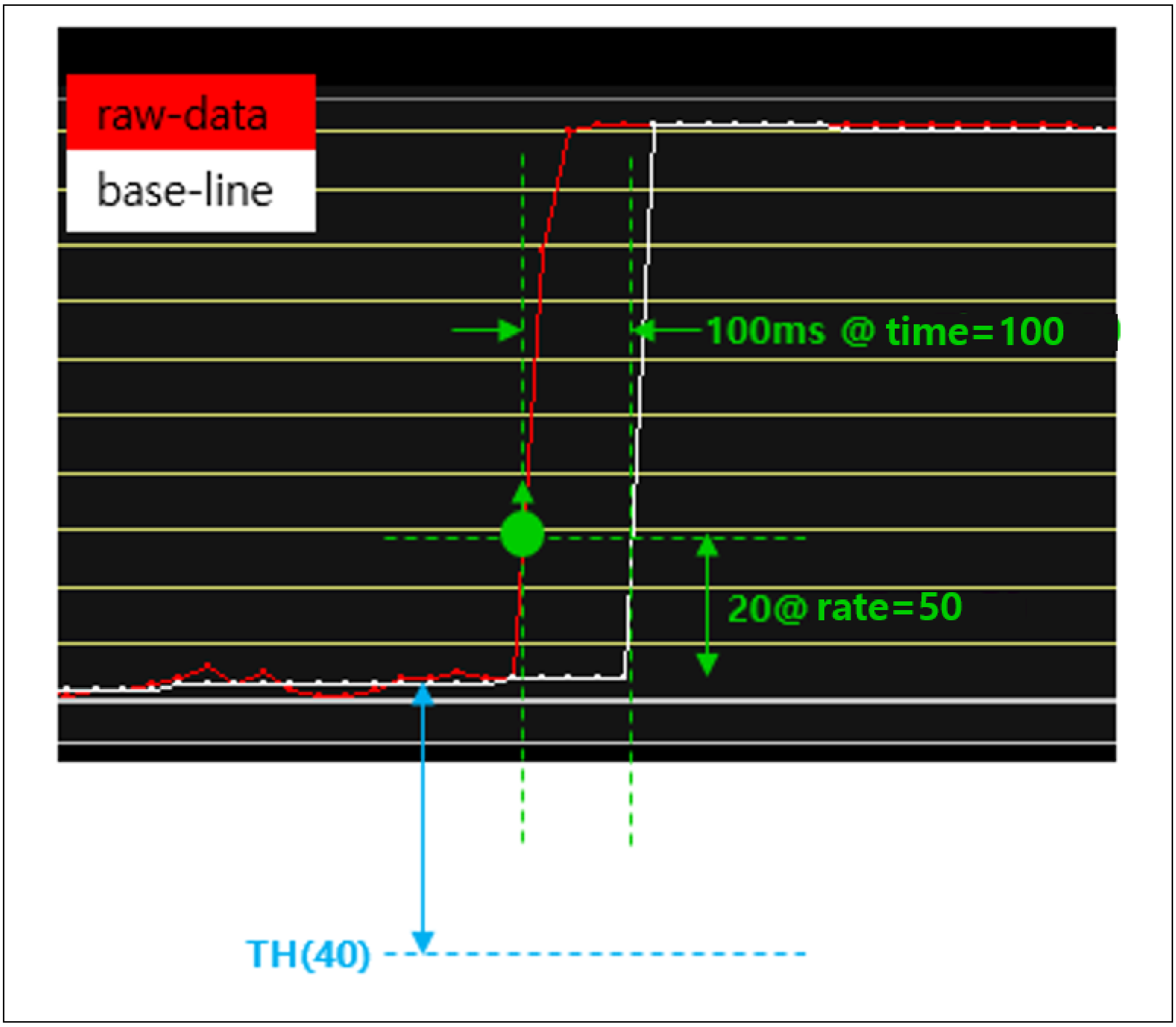


Figure 27. Reverse situation parameters

### 4.1.1.18 #define THD_DETECT_CHXX [n]

- Set the threshold for each channel.
- If the Diff data exceeds Threshold (THD), a Key detect event will occur.
- Setting range (50 ~ 2000)

### 4.1.1.19 #define THD_RELEASE_RATE_VAL [n]

- function sets the level for key-release judgement.
- Key release judgement level is rate % of the THD (threshold).
- By default, it is set to 70% of THD, too high or too low may affect touch behavior.
- Setting range (50~80) : default 70

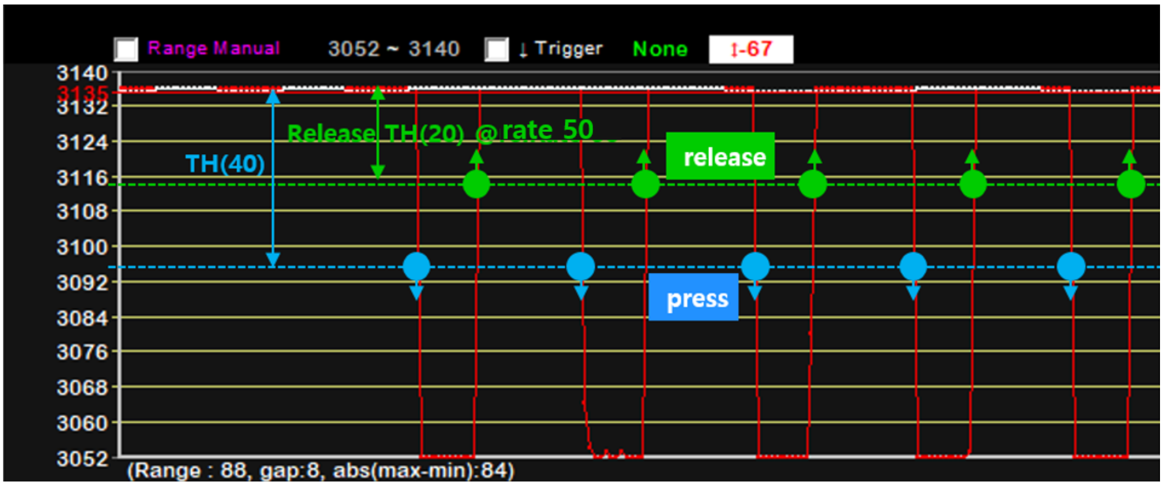


**Figure 28. Press Threshold & Release threshold (set 50% rate)**

#### 4.1.1.20 #define TOUCH_SUM_CNT_CHXX [n]

- A function that allows you to adjust the touch sensitivity.
- SUM_COUNT determines the number of consecutive sensing times for a channel.
- The greater the setting value, the greater the sensitivity.
- SUM_COUNT can act as a low pass filter, but it has a disadvantage that the sensing time becomes longer.
- Setting range (3 ~ 15) : default 6

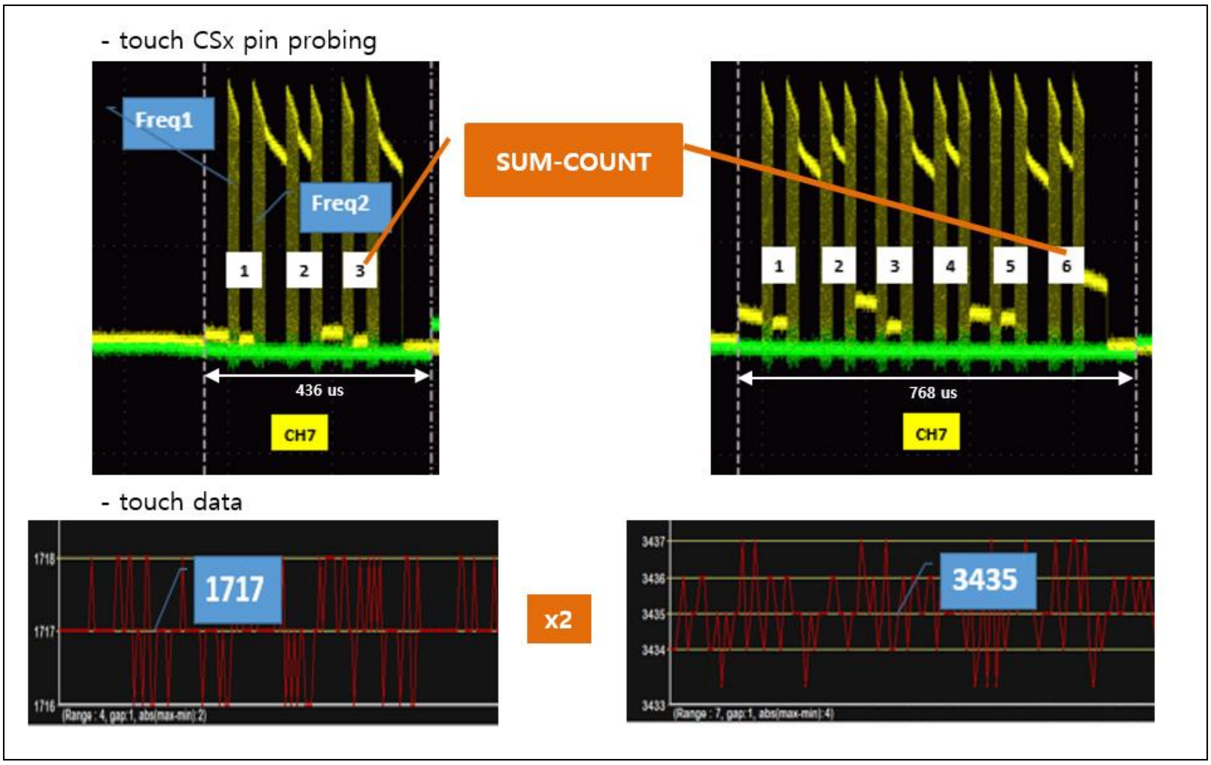


Figure 29. Change of touch data according to TOUCH_SUM_CNT_CH07

### 4.1.1.21 #define BASE_TRACE_SPEED_CHXX [n]

- Sets the ratio between the existing data and the current data when updating the baseline.
- Previous value : Current value = N : 1
- If it is too high or too low, it may affect touch behavior.
- Setting range (1 ~ 7) : default 1

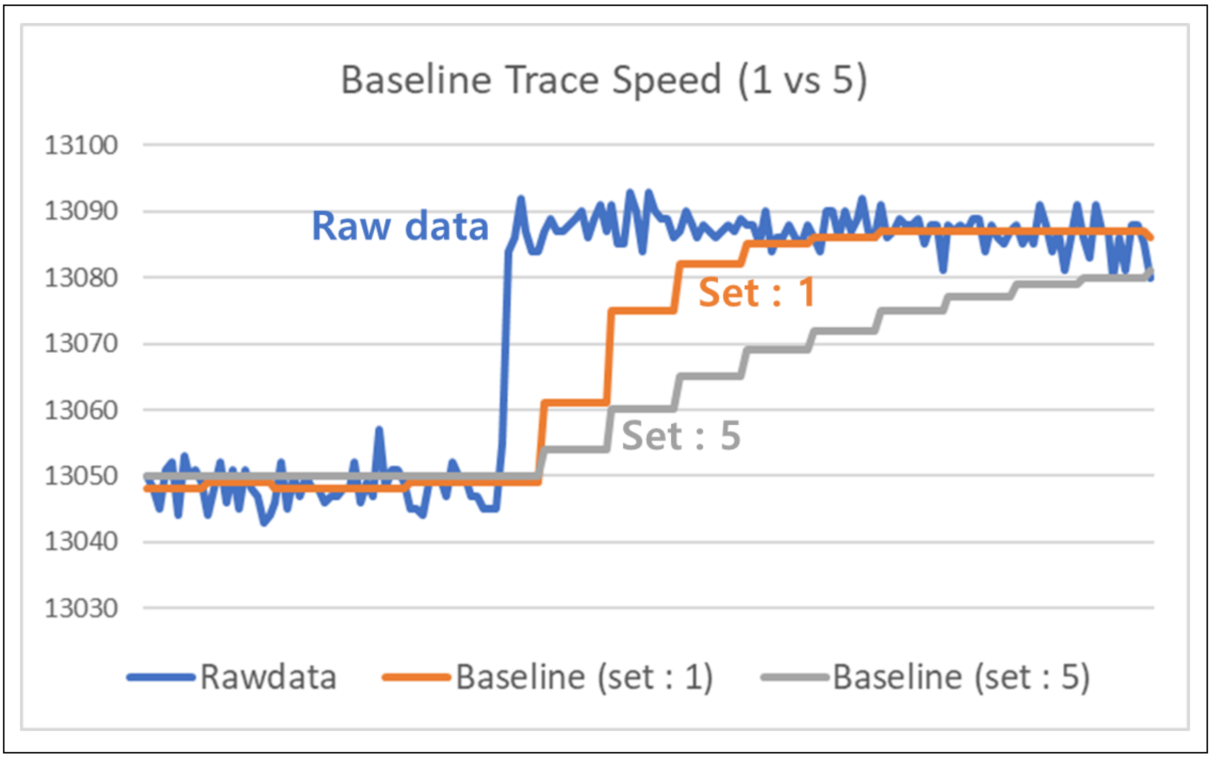


Figure 30. Baseline Trace Speed (1 vs 5)

### 4.1.2 user_function.c

#### 4.1.2.1 void Library_Data_Assign(void)

- Data connection function of the variable pointer used by the Touch library.
- If you change the contents of the function, it may affect touch sensing, so don't change it.

#### 4.1.2.2 void Touch_Config_Set (void)

- Touch register setting function used by the Touch library.
- If you change the contents of the function, it may affect touch sensing, so don't change it.

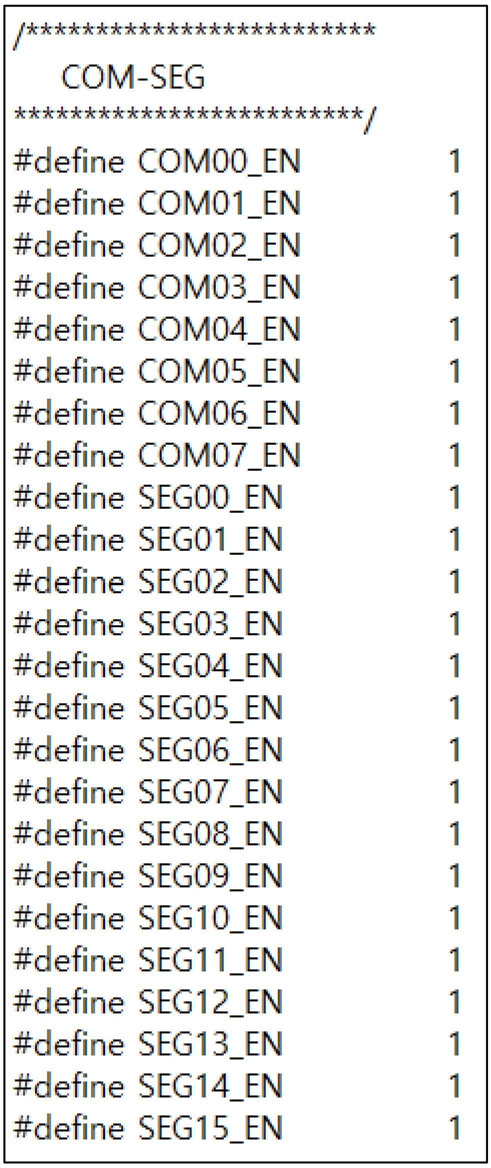
## 4.2 LED Parameter Description

### 4.2.1 user_function.h

#### 4.2.1.1 #define COMXX_EN X

#### 4.2.1.2 #define SEGXX_EN X

— If you used COM/SEG channel, set X=1, otherwise set X=0.

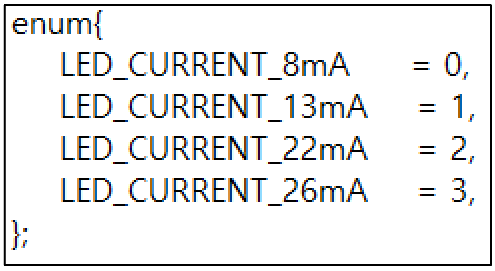


```
/*************************
   COM-SEG
*************************/
#define COM00_EN            1
#define COM01_EN            1
#define COM02_EN            1
#define COM03_EN            1
#define COM04_EN            1
#define COM05_EN            1
#define COM06_EN            1
#define COM07_EN            1
#define SEG00_EN            1
#define SEG01_EN            1
#define SEG02_EN            1
#define SEG03_EN            1
#define SEG04_EN            1
#define SEG05_EN            1
#define SEG06_EN            1
#define SEG07_EN            1
#define SEG08_EN            1
#define SEG09_EN            1
#define SEG10_EN            1
#define SEG11_EN            1
#define SEG12_EN            1
#define SEG13_EN            1
#define SEG14_EN            1
#define SEG15_EN            1
```

**Figure 31. LED COM/SEG port activation**

### 4.2.1.3 #define LED_CURRENT [n]

- This is the driving current setting parameter of the LED driver.



```
enum{
    LED_CURRENT_8mA     = 0,
    LED_CURRENT_13mA    = 1,
    LED_CURRENT_22mA    = 2,
    LED_CURRENT_26mA    = 3,
};
```

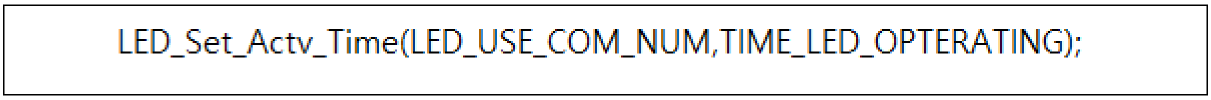
**Figure 32. LED port current setting value**

### 4.2.1.4 #define LED_USE_COM_NUM [n]

- Number of COM ports used.
- Used as a parameter for the LED_Set_Actv_Time (uint8_t com_cnt, uint8_t duration) function.

### 4.2.1.5 #define TIME_LED_OPTERATING [n]

- LED operation time setting parameter (unit: ms)
- Used as a parameter for the LED_Set_Actv_Time (uint8_t com_cnt, uint8_t duration) function.



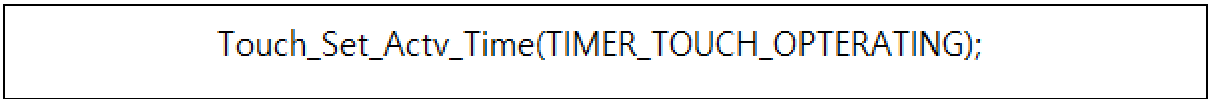
```
LED_Set_Actv_Time(LED_USE_COM_NUM,TIME_LED_OPTERATING);
```

**Figure 33. Set LED operating time**

- TIME_LED_OPTERATING / LED_USE_COM_NUM must not exceed 4.

### 4.2.1.6 #define TIMER_TOUCH_OPTERATING [n]

- Touch sensing operation time setting parameter (unit: ms)
- Used as a parameter for the Touch_Set_Actv_Time (uint8_t slot_ms) function.
- Used only when "LED_DRV_EN" and "TS_LED_TIME_DIV" are 1.



```
Touch_Set_Actv_Time(TIMER_TOUCH_OPTERATING);
```

**Figure 34. Touch sensing operation time setting**

- If the TIME_LED_OPTERATING + TIMER_TOUCH_OPTERATING exceeds 16 ms, LED flickering may occur.
- 60 Hz => 16.6 ms

## 4.3 User Function Feature Description

### 4.3.1 user_function.h

#### 4.3.1.1 #define LED_DRV_EN X

— An option to control the LED via the LED driver.

— If you are using led driver, set X=1, otherwise set X=0.

— Setting the top-level options for LED driver operations. If set to 0, LED driver related operations are not performed.

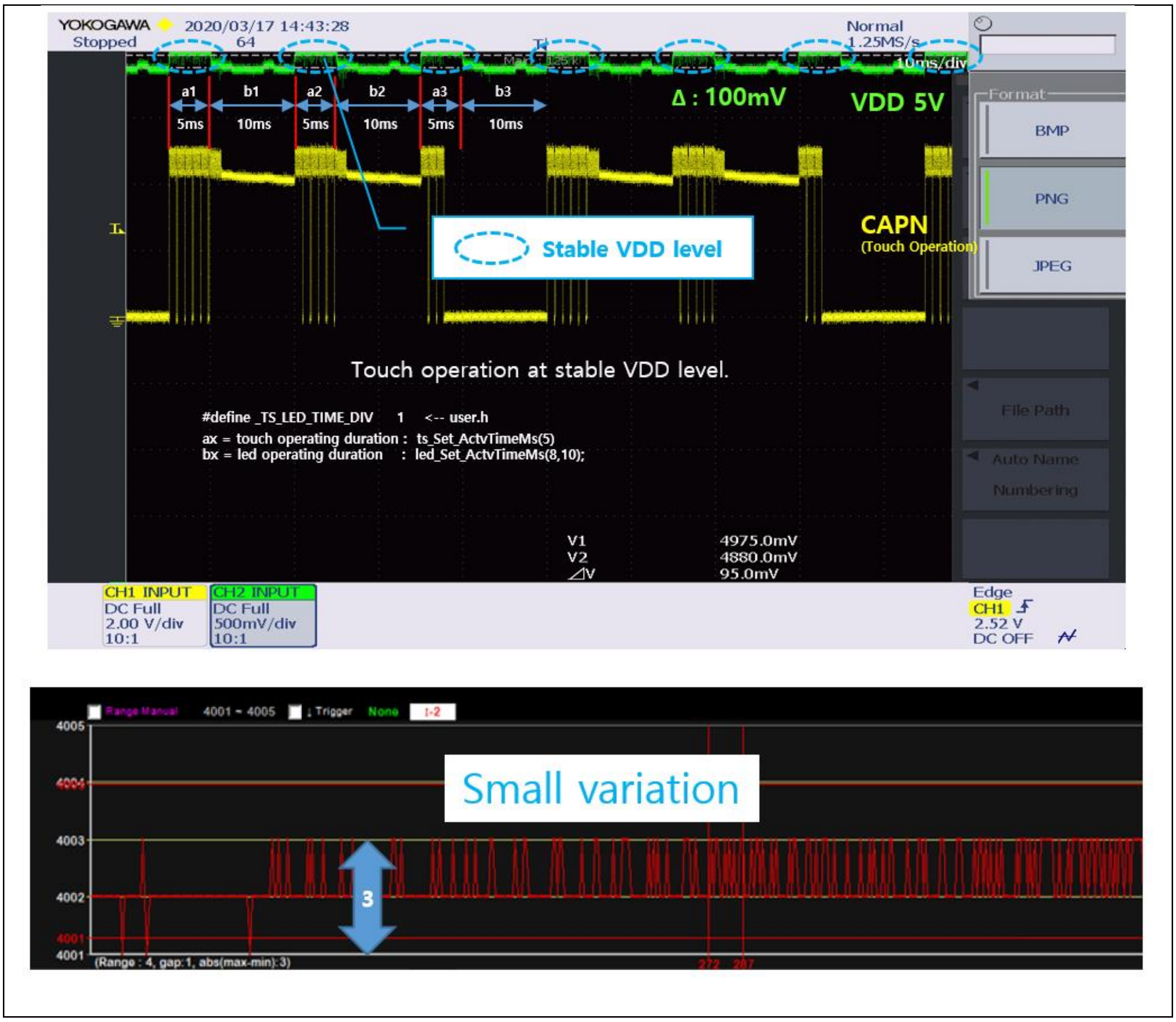
#### 4.3.1.2 #define TS_LED_TIME_DIV X

— #LED_DRV_EN must be enabled first.

— If you are using Touch & LED time division, set X=1, otherwise set X=0.

— It is an optional function that divides time so that touch sensing and LED can be used together as one PIN.

— Touch operation is carried out in a stable VDD section (LED - OFF section). Therefore, the change in the touch data is relatively small and the LED brightness is relatively dark.



Touch operation at stable VDD level.

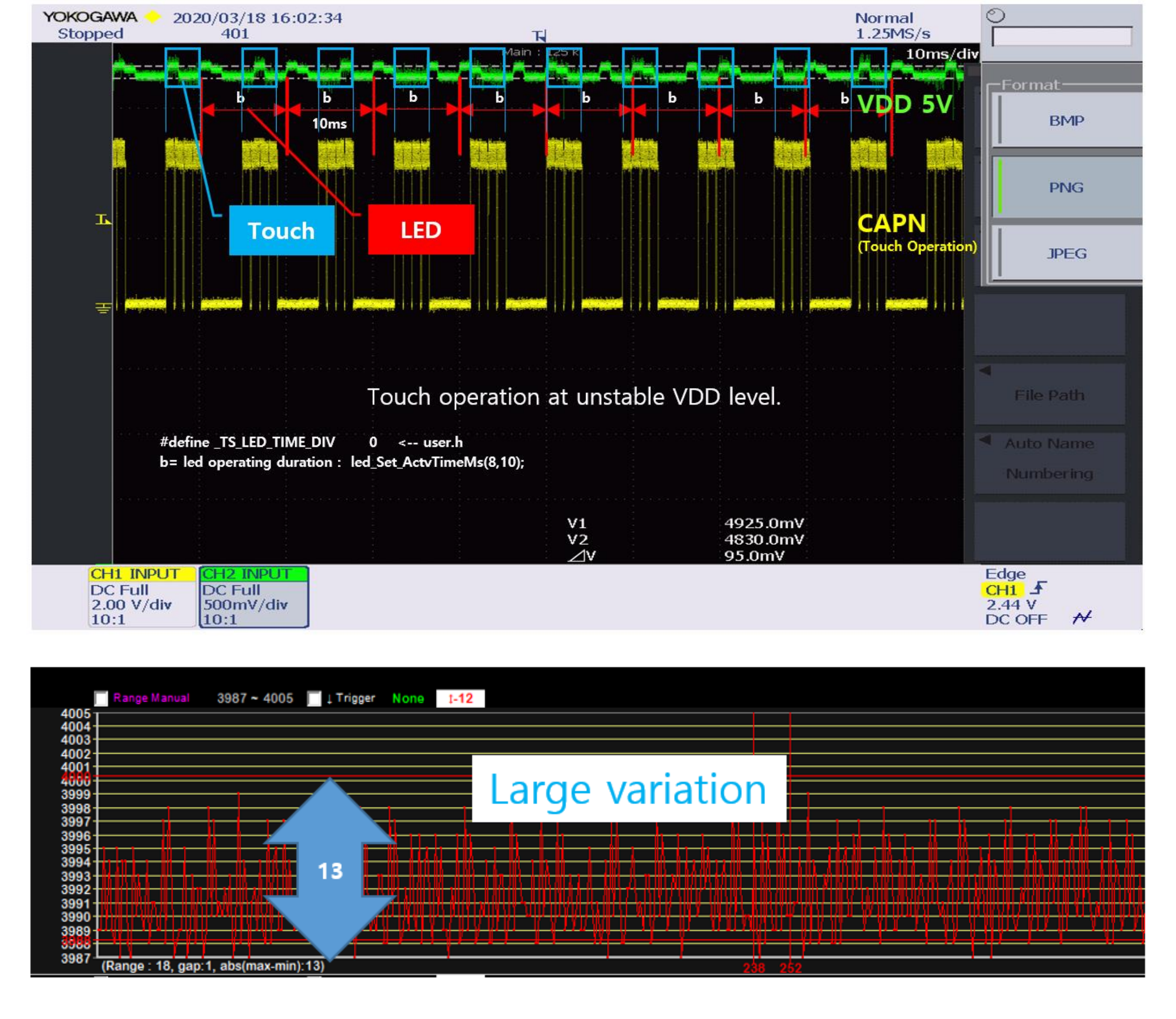
```
#define _TS_LED_TIME_DIV     1    <-- user.h
ax = touch operating duration :  ts_Set_ActvTimeMs(5)
bx = led operating duration    :  led_Set_ActvTimeMs(8,10);
```

```
V1    4975.0mV
V2    4880.0mV
⊿V    95.0mV
```

(Range : 4, gap:1, abs(max-min):3)

Figure 35. Touch/LED Time-Division Mode

— #define TS_LED_TIME_DIV 0 : More noise is generated than when used.



Touch operation at unstable VDD level.

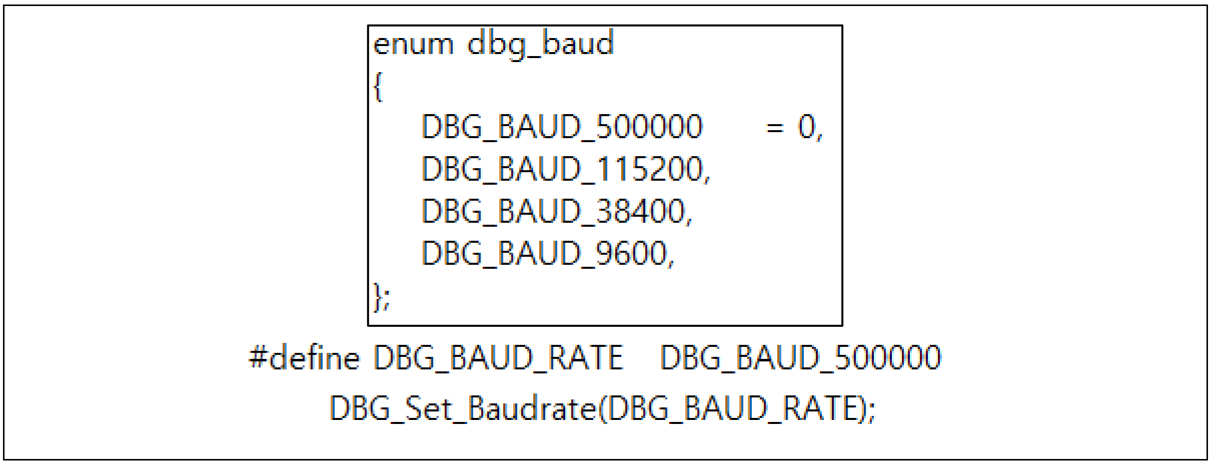
```
#define _TS_LED_TIME_DIV    0    <-- user.h
b= led operating duration :  led_Set_ActvTimeMs(8,10);
```

(Range : 18, gap:1, abs(max-min):13)

**Figure 36. Touch/LED Independent Mode**

#### 4.3.1.3 #define UART_ENABLE X

- Data on Touch can be checked during project development.
- If you use debug.c (UART), set X=1, otherwise set X=0.
- The UART baud rate setting uses the parameter "#define DBG_BAUD_RATE" and is used as a parameter for the function "DBG_Set_Baudrate (uint8_t baud)".



```
enum dbg_baud
{
    DBG_BAUD_500000     = 0,
    DBG_BAUD_115200,
    DBG_BAUD_38400,
    DBG_BAUD_9600,
};
#define DBG_BAUD_RATE   DBG_BAUD_500000
DBG_Set_Baudrate(DBG_BAUD_RATE);
```

**Figure 37. UART baud rate setting**

#### 4.3.1.4 #define I2C_ENABLE X

- Data on Touch can be checked during project development.
- If you are using I2C Debug, set X=1, otherwise set X=0.

#### 4.3.1.5 #define WDT_ENABLE X

- If you are using the WDT function, set X=1, otherwise set X=0.
- It is currently set to the default 4sec setting, and you can change the time using the WDT_Set_4sec_Reset()

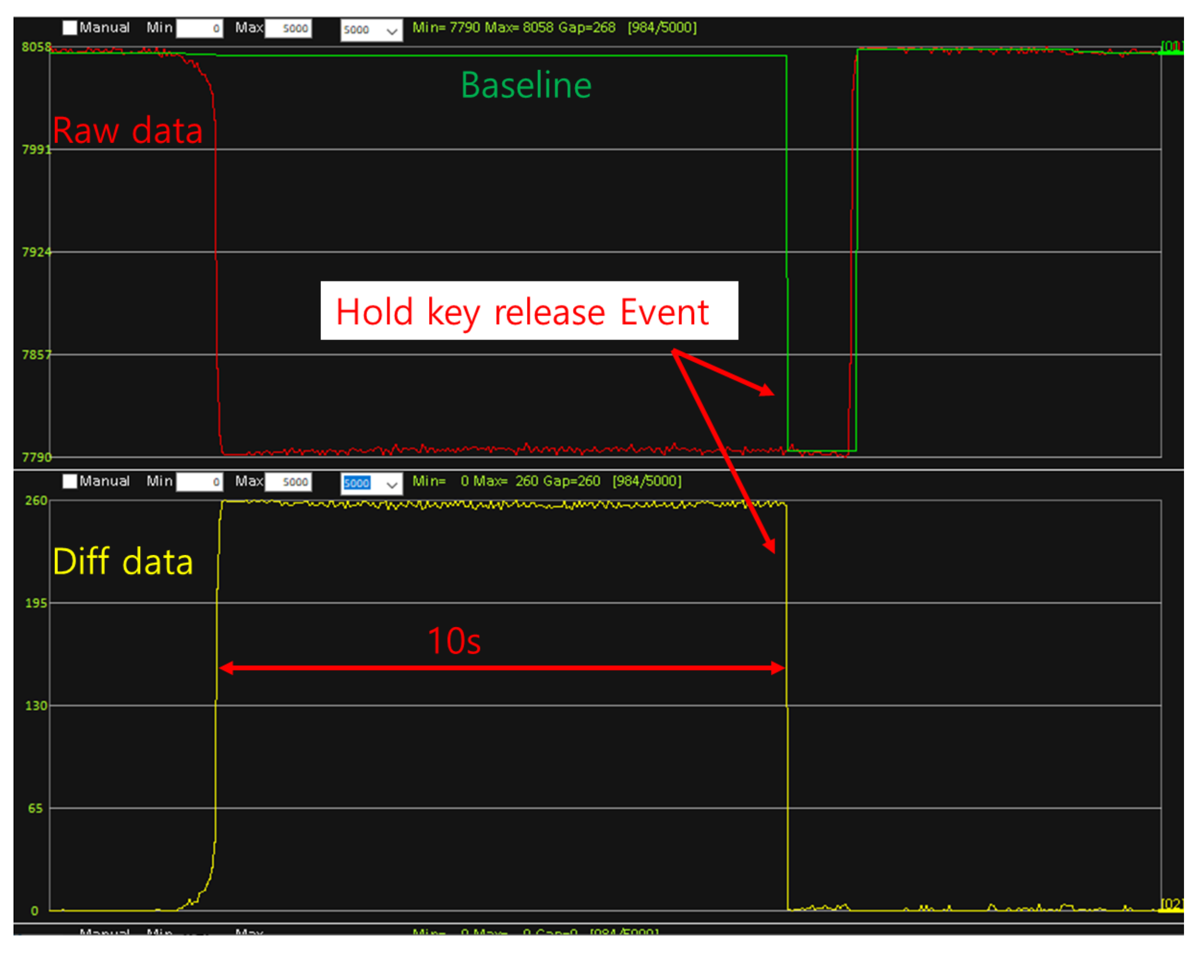
### 4.3.1.6 #define HOLD_KEY_RELEASE_EN X

— If you are using Hold Key Release function, set X=1, otherwise set X=0.

— If one or more keys are continuously pressed for a certain period of time (default 10s), the function is judged to be abnormal and the entire key is initialized.

— #define TIMER_HOLD_KEY_RELEASE 10000 : hold key release time 10s (n x 1ms). The recommended range for setup is [3000 to 60000].

— The function of "void Touch_Key_Scenario (void)" in "user_function.c" is activated to activate the Hold key release function.

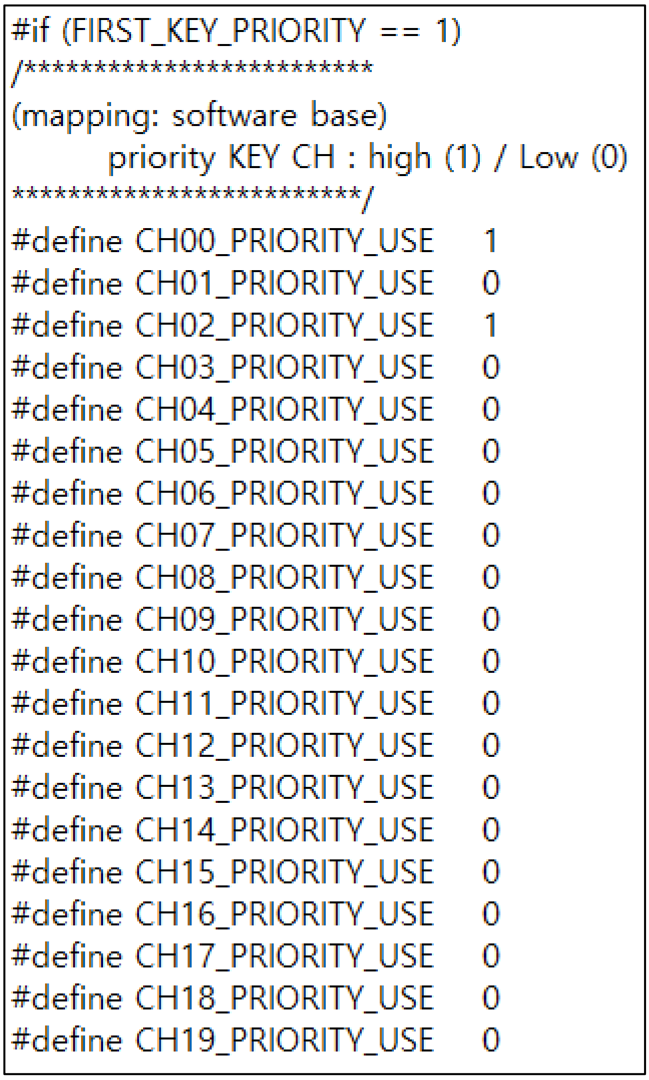

□ Manual Min 0 Max 5000 5000 Min= 7790 Max= 8058 Gap=268 [984/5000]

□ Manual Min 0 Max 5000 5000 Min= 0 Max= 260 Gap=260 [984/5000]

Figure 38. Hold Key release Event

#### 4.3.1.7 #define FIRST_KEY_PRIORITY X

- If you are using Key Priority set function, set X=1, otherwise set X=0.
- If you enter a key set to priority, other keys are ignored (the keys with the same priority are not ignored; they are displayed in the order in which they were entered first).
- #define CHxx_PRIORITY_USE [1 : High, 0 : Low]
- The function of "void Touch_Key_Scenario (void)" in "user_function.c" is activated to activate the first key priority function.



```
#if (FIRST_KEY_PRIORITY == 1)
/*************************
(mapping: software base)
        priority KEY CH : high (1) / Low (0)
*************************/
#define CH00_PRIORITY_USE    1
#define CH01_PRIORITY_USE    0
#define CH02_PRIORITY_USE    1
#define CH03_PRIORITY_USE    0
#define CH04_PRIORITY_USE    0
#define CH05_PRIORITY_USE    0
#define CH06_PRIORITY_USE    0
#define CH07_PRIORITY_USE    0
#define CH08_PRIORITY_USE    0
#define CH09_PRIORITY_USE    0
#define CH10_PRIORITY_USE    0
#define CH11_PRIORITY_USE    0
#define CH12_PRIORITY_USE    0
#define CH13_PRIORITY_USE    0
#define CH14_PRIORITY_USE    0
#define CH15_PRIORITY_USE    0
#define CH16_PRIORITY_USE    0
#define CH17_PRIORITY_USE    0
#define CH18_PRIORITY_USE    0
#define CH19_PRIORITY_USE    0
```

**Figure 39. Touch First Key Priority define**

### 4.3.1.8 #define MULTI_KEY_RESET X

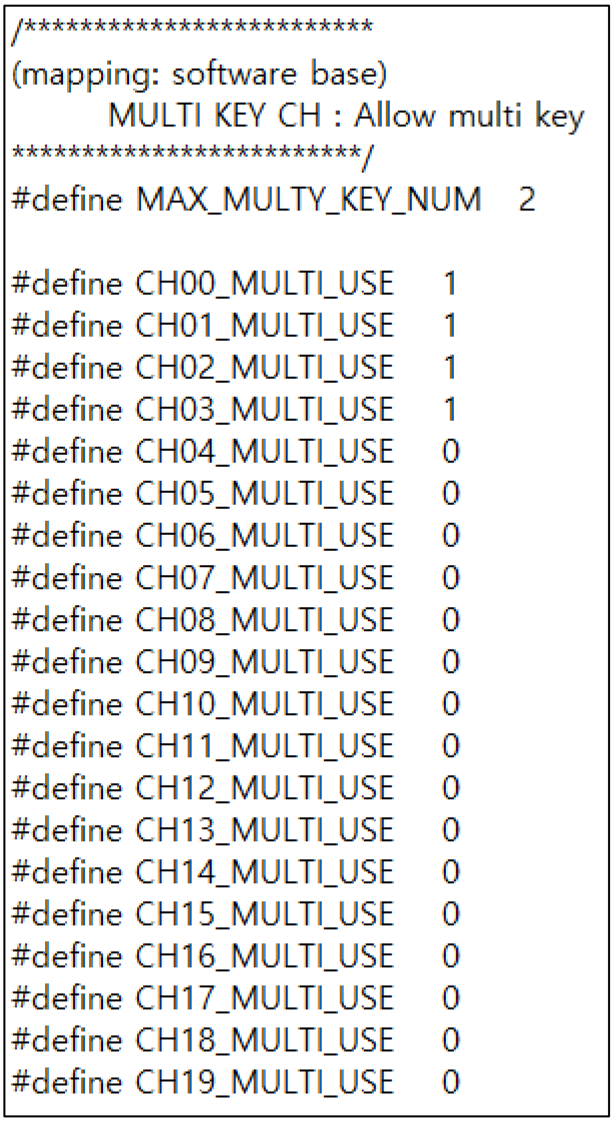
— Set X=1 if you are using the multi-key reset function, otherwise set X=0.

— Pressing more than a set number of channels with the multi-key reset function enabled will reset the key.

— #define NOISE_COUNT_MAX [n] : Initializes keys when [n] or more Multi key reset use channels are entered.

— #define CHxx_MULTI_USE [1: Multi-key-reset use ch, 0: Multi-key-reset no use ch]

— The function of "void Touch_Key_Scenario (void)" in "user_function.c" is activated to activate the multi key reset function.



```
/*************************
(mapping: software base)
        MULTI KEY CH : Allow multi key
*************************/
#define MAX_MULTY_KEY_NUM   2

#define CH00_MULTI_USE    1
#define CH01_MULTI_USE    1
#define CH02_MULTI_USE    1
#define CH03_MULTI_USE    1
#define CH04_MULTI_USE    0
#define CH05_MULTI_USE    0
#define CH06_MULTI_USE    0
#define CH07_MULTI_USE    0
#define CH08_MULTI_USE    0
#define CH09_MULTI_USE    0
#define CH10_MULTI_USE    0
#define CH11_MULTI_USE    0
#define CH12_MULTI_USE    0
#define CH13_MULTI_USE    0
#define CH14_MULTI_USE    0
#define CH15_MULTI_USE    0
#define CH16_MULTI_USE    0
#define CH17_MULTI_USE    0
#define CH18_MULTI_USE    0
#define CH19_MULTI_USE    0
```

**Figure 40. Multi Key Reset Use channel define**

#### 4.3.1.9 #define TOUCH_RREQ_SEL_EN X

- If you are using Touch Frequency Change function, set X=1, otherwise set X=0.
- This function calculates Diff data by selecting a frequency with a small noise among the two frequencies.
- The function "void Touch_Freq_Sel(void)" in "user_function.c" is activated.

### 4.3.2 main.c

#### 4.3.2.1 void Init_GPIO(void)

- Default GPIO set, output Low

#### 4.3.2.2 void Init_Touch(void)

- Touch sensing mode and touch related data assign function.
- If you change the contents of the function, it may affect touch sensing, so don't change it.

#### 4.3.2.3 void Touch_Config_Set (void)

- Function to apply touch sensing parameter value.
- If you change the contents of the function, it may affect touch sensing, so don't change it.

#### 4.3.2.4 void Init_User_Function (void)

- A function that sets whether user_function functions are used or not and the initial value.
- If you change the contents of the function, it may affect touch sensing, so don't change it.

#### 4.3.2.5 void Set_User_Function (void)

- A function that sets the user setting value of the user_function functions.
- Change the settings in user_function.h.
- If you change the contents of the function, it may affect touch sensing, so don't change it.

# Revision history

| Date | Revision | Description |
|---|---|---|
| 20.04.10 | 1.00 | Document created |
| 21.01.04 | 1.01 | Function name, variable name modified |
| 21.01.04 | 1.01 | ts_Get_Key() Function deleted. ts_detect_key variable added |
| 21.01.04 | 1.01 | TS_Set_CH_THD(u8 ch_idx, s16 tdh), u8 ch -> u8 ch_idx |
| 22.11.01 | 1.02 | Revised the font of this document |
| 22.12.28 | 1.10 | Library code Change structure and add functionality |
| 23.06.21 | 1.11 | Library code Change structure and add functionality |
| 24.12.02 | 1.12 | Updated the disclaimer. |

**Korea**

**Regional Office**, Seoul
R&D, Marketing & Sales
8th Fl., 330, Yeongdong-daero,
Gangnam-gu, Seoul,
06177, Korea

Tel: +82-2-2193-2200
Fax: +82-2-508-6903
www.abovsemi.com

**HQ**, Ochang
R&D, QA, and Test Center
93, Gangni 1-gil, Ochang-eup,
Cheongwon-gun,
Chungcheongbuk-do,
28126, Korea

Tel: +82-43-219-5200
Fax: +82-43-217-3534
www.abovsemi.com

**Domestic Sales Manager**
Tel: +82-2-2193-2206
Fax: +82-2-508-6903
Email: sales_kr@abov.co.kr

**Global Sales Manager**
Tel: +82-2-2193-2281
Fax: +82-2-508-6903
Email: sales_gl@abov.co.kr

**China Sales Manager**
Tel: +86-755-8287-2205
Fax: +86-755-8287-2204
Email: sales_cn@abov.co.kr

**ABOV Disclaimer**

**IMPORTANT NOTICE – PLEASE READ CAREFULLY**

ABOV Semiconductor ("ABOV") reserves the right to make changes, corrections, enhancements, modifications, and improvements to ABOV products and/or to this document at any time without notice. **ABOV DOES NOT GIVE WARRANTIES AS TO THE ACCURACY OR COMPLETENESS OF THE INFORMATION INCLUDED HEREIN.** Purchasers should obtain the latest relevant information of ABOV products before placing orders. Purchasers are entirely responsible for the choice, selection, and use of ABOV products and ABOV assumes no liability for application assistance or the design of purchasers' products. **NO LICENSE, EXPRESS OR IMPLIED, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY ABOV HEREIN. ABOV DISCLAIMES ALL EXPRESS AND IMPLIED WARRANTIES AND SHALL NOT BE RESPONSIBLE OR LIABLE FOR ANY INJUIRES OR DAMAGES RELATED TO USE OF ABOV PRODUCTS IN SUCH UNAUTHORIZED APPLICATIONS.** ABOV and the ABOV logo are trademarks of ABOV. For additional information about ABOV trademarks, please refer to https://www.abov.co.kr/en/about/corporate_identity.php. All other product or service names are the property of their respective owners. Information in this document supersedes and replaces the information previously supplied in any former versions of this document.