SB358xB

Capacitive Touch Controller IC

Datasheet

Revision 1.0

Jan. 2015

Headquarters
4F-1, No.9, Prosperity Rd.,
Science-based Industrial Park,
Hsinchu City, Taiwan, R.O.C
TEL: 886-3-6662888
FAX: 886-3-6662999
http://www.ene.com.tw

Taipei Office
4F, No.88, Bauchiau Rd.
Shindian City, Taipei,
Taiwan, R.O.C.
TEL: 886-2-89111525
FAX: 886-2-89111523

Revision History

Version	Content	Date
1.0	Initial Release	2015/01

CONTENT

1. Overview

1.1 General description

SB358xB series of capacitive sensor button controller are capable of detecting touch on electrode covered by dielectric material such as glass or plastic. Touch sensing technology is implemented in many fields such as appliances, control panels, or replacing of mechanical switches/buttons. Capacitive sensing technology is more robust and overall cost advantage.

The SB358xB series capacitive sensor button controller consists up to 9 capacitive touch-detected ports which can support up to 9 touch-detected buttons. These ports could be implemented as buttons, sliders, sensing wheels or proximity sensor. Each port can be programmed with its own sensitivity parameter and controlled by firmware.

Detailed functionality is different by parts. Please refer to the following pages for each specification.

1.2 Features

- Capacitive touch sensor controller
- Support up to maximum 9 capacitive touch keys (TK)
- Support up to maximum 12 GPIO with PWM function
- Support I2C serial interface
- Built-in initialization.
- Programmable low-power mode.
- Wakeup function support.
- Programmable threshold for different shapes of sensor buttons.
- Built-in self-calibration of scan baseline.
- RoHS compliant
- Support TK-to-GPIO function

1.3 Selection Table

Part Number	Operation Mode	Inter-face	No of IO	No of TK	TK-to-IO	PWM support	Package
SB3585UB0	Host	I2C	12	9	---	8bx12	QFN20
	Non-host	---	6	6	6 pairs	---	
SB3585WB0	Host	I2C	12	9	---	8bx12	SOP20
	Non-host	---	6	6	6 pairs	---	
SB3586WB0	Host	I2C	9	8	---	8bx9	NSOP16
SB3584WB0	Non-host	---	4	4	4 pairs	---	NSOP16
SB3581WB0	Non-host	---	1	1	1 pair	---	NSOP8

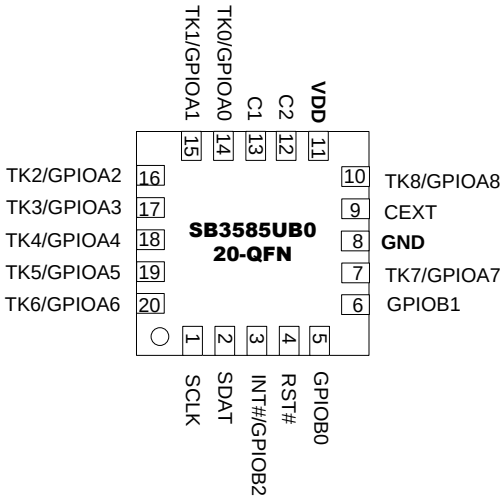
Note: SB358xB supports two modes, Host and Non-host modes. Under host mode, IO function is multiplexed with TK function on the same pin, user is able to assign pin functions through I2C interface. Under non-host mode, pin functions are fixed. Please refer to Chapter 3 for details.

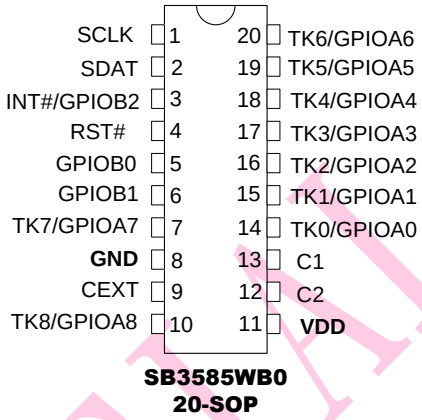
2. Pin Assignment and Description

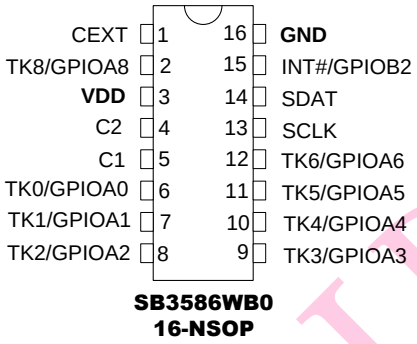
2.1 SB358xB Pin Assignment Table

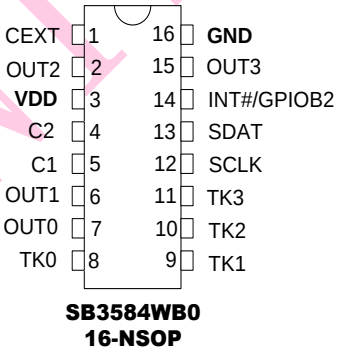
Pin	Pin Name	I/O	Description
1	SCLK	I	I2C clock signal
2	SDAT	I/O	I2C data signal
3	INT#/GPIOB2	I/O	Interrupt
			GPIO and PWM output
4	RST#	I	Reset pin, low active
5	GPIOB0	I/O	GPIO and PWM output
6	GPIOB1	I/O	GPIO and PWM output
7	TK7/GPIOA7	I/O	Capacitive touch channel
			GPIO and PWM output
8	GND	A	Ground
9	CEXT	A	External capacitor.
10	TK8/GPIOA8	I/O	Capacitive touch channel
			GPIO and PWM output
11	VDD	A	Power Pin: 3.0V~5.5V
12	C2	A	External capacitor.
13	C1	A	External capacitor.
14	TK0/GPIOA0	I/O	Capacitive touch channel
			GPIO and PWM output
15	TK1/GPIOA1	I/O	Capacitive touch channel
			GPIO and PWM output
16	TK2/GPIOA2	I/O	Capacitive touch channel
			GPIO and PWM output
17	TK3/GPIOA3	I/O	Capacitive touch channel
			GPIO and PWM output
18	TK4/GPIOA4	I/O	Capacitive touch channel
			GPIO and PWM output
19	TK5/GPIOA5	I/O	Capacitive touch channel
			GPIO and PWM output
20	TK6/GPIOA6	I/O	Capacitive touch channel
			GPIO and PWM output

2.2 SB358xB Top view

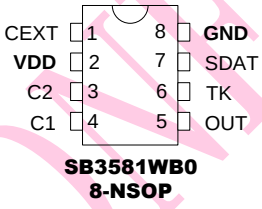








v



3. Functional Description

SB358xB products can operate in two different modes: host mode and non-host mode.

Host mode is accomplished by digital interfaces such as I2C, to communicate with the host (SOC). Host programmed SB358xB internal register settings through I2C, and start setting the chip pins and touch channel resources to achieve the best system performance requirements.

In some relatively simple embedded applications, it may not have the system microcontroller. SB358xB may only need to do a simple touch buttons which do not need to communicate with the system microcontroller. In this situation, SB358xB provides non-host mode to achieve the system design simplification. In the non-host mode (also called TK to GPIO mode), SB358xB starts with the default pin configuration, and after power-on, automatic start the touch access with the default parameter settings for the touch key functions. The default pin configuration is as follows:

SB3585UB0/SB3585WB0:

Touch Key	Output GPIO
TK2	TK1
TK3	TK0
TK4	TK8
TK5	INT#/GPIOB2
TK6	GPIOB0
TK7	GPIOB1

SB3586WB0:

Touch Key	Output GPIO
TK2	TK1
TK3	TK0
TK4	TK8
TK5	INT#/GPIOB2

SB3584WB0:

Touch Key	Output GPIO
TK0	OUT0
TK1	OUT1
TK2	OUT2
TK3	OUT3

SB3581WB0:

Touch Key	Output GPIO
TK	OUT

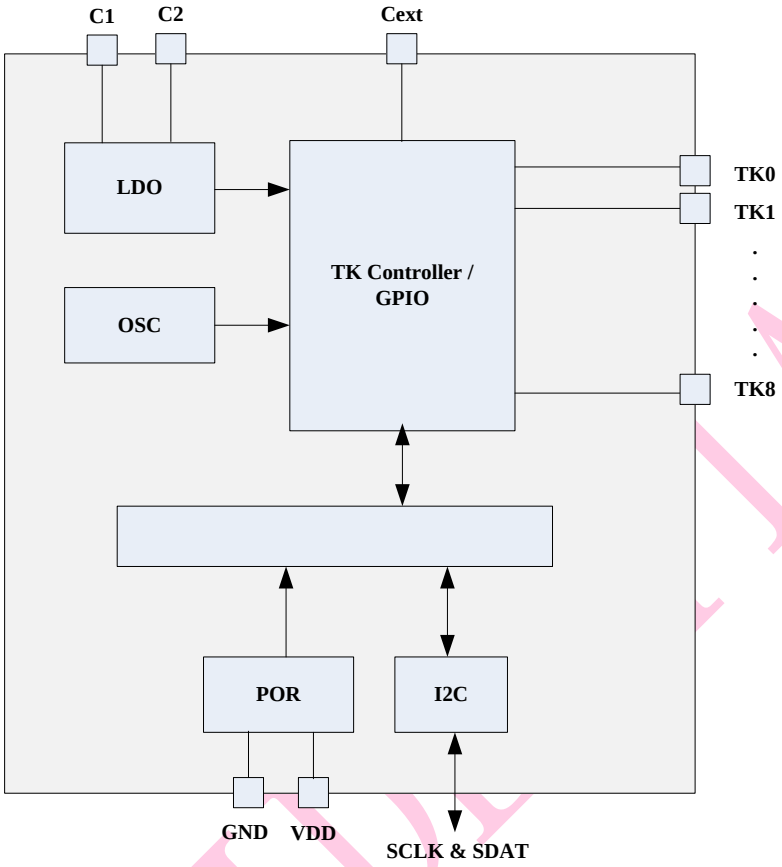
For SB3585B, choice of host mode or non-host mode is decided by the external resistor of GPIOB1. If there is no external pull-down resistor, the default operation mode of SB3585B is non-host mode.

To operate SB3585B in host mode, user needs to reset SOC using I2C. Note that from power on till I2C configuration is completed, SB3585B TK scan is continuous. System designer needs to be aware if the pin external connection is initiated.

SB3584WB0 and SB3581WB0 only support non-host mode. The SDAT and SCLK pins in the pin assignment are for hardware configuration. Please see chapter 3.2 for details.

Chapter 3 and 4 outline details on controller register definitions and examples, please contact ENE FAE for support.

3.1 Block Diagram



3.2 Hardware Function Configuration

Pin Name	Description
GPIOB0	7 bits I2C address selection 0: 0xA0 (1010_000x) 1: 0x80 (1000_000x)
INT#/GPIOB2	Test mode enable 0: Test mode enabled (debug enabled) 1: Normal mode
SDAT	GPIO High/Low Level Active Selection 0: High-Level Active 1: Low-Level Active
SCLK	Selection of scan parameter for different background capacitor 0: High capacitor 1: Low capacitor
GPIOB1	Host mode selection 0: Host Mode 1: Non-host mode

Note 1:
These pins are internally pulled-up when reset. All of them are read as '1' in normal state. ◦ The internal pull-up resistor of INT#/SCLK/SDAT is 4.7KΩ，and GPIOB0/GPIOB1 is 40KΩ ◦ It needs to add an additional pull-down resistor if intends to reset to 0.

Note 2:
These pins are default pulled-up, host can turn them off by clear the registers respectively ◦

Note 3:
The internal pull-up is weak, so that it does not allow current leakage on those pins when reset. (ex. connect to an un-powered chip, connect to a stronger low signal before powered-on)

Note 4:
The major difference between host mode and non-host mode is:
When the non-host mode is set, IC is default set to "button to GPIO" mode, and after power-up, the default touch channel will automatically start operation, does not require the setup and activation from system microcontroller (master). Setting adjustment can also be accomplished via I2C.
When the host mode is set, IC must be configured by the system microcontroller (master) via I2C interface to set and wait for the host to start the touch channel.
Refer to Section 3.4 the default pin function descriptions.

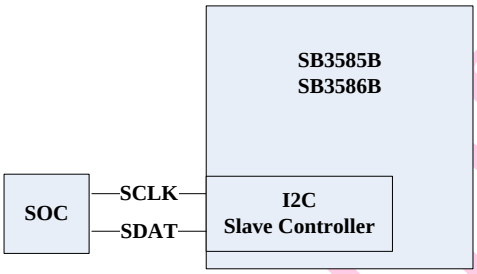
3.3 Basic Function

SB358xB capacitive sensor controller series is capable of detecting touches on materials such as glass or plastic. To provide more flexibility for system design, SB358xB is able to operate in two different modes: host mode and non-host mode. Host mode is accomplished by digital interfaces such as I2C, to communicate with the host (SOC).

In the non-host mode (also called TK to GPIO mode), SB358xB can operate as touch key only without communicating with SOC. SB358xB starts with the default pin configuration, and after power-on, automatic start the touch access with the default parameter settings for the touch key functions.

3.3.1 I2C

SB358xB supports H/W I2C slave controller to communicate with SOC, the I2C protocol supports byte read and write operations.

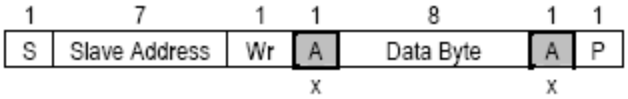


SCLK: I2C clock signal

SDAT: I2C data signal

3.3.1.1 I2C Protocol

SB358xB supports byte read and write operations, the I2C protocol is depicted as followings:



S	Start Condition
Sr	Repeated Start Condition
Rd	Read (bit value of 1)
Wr	Write (bit value of 0)
x	Shown under a field indicates that that field is required to have the value of 'x'
A	Acknowledge (this bit position may be '0' for an ACK or '1' for a NACK)
P	Stop Condition
PEC	Packet Error Code
	Master-to-Slave
	Slave-to-Master
...	Continuation of protocol

Slave Address:

Address	Name	Bit	Type	Description	Default
0x081	SMBADDR	**I2C address**			0x80
		7	RO	Retrieve '1'	
		6	RO	Retrieve ' 0'	
		5	RO	Select 7 bit I2C slave address is decided by HW trap (GPIOB0) 0: 0xA0 (1010_000x) 1: 0x80 (1000_000x)	
		4-1	RO	Retrieve ' 0'	
		0	RO	0: Write 1: Read	

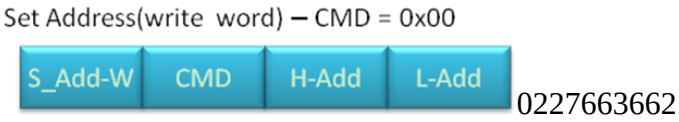
Slave Address:

1	**0**	**HW Trap**	**0**	**0**	**0**	**0**	**R/W**

3.3.1.2 The Target Address Setting for Read/Write Commands

User needs to set correct target address in order to read the internal register of SB IC.
Example command of setting target address as follows:

Example for setting address of **Write Word** command (CMD=**0x00**):



0227663662

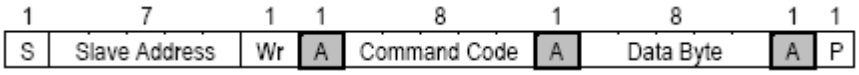
Note: S_Add-W = slave_address(bit7~1) + write_bit (bit 0 = 0)

To read current address to be read or written, using **Read Word** command (CMD=**0x11)** as an example:

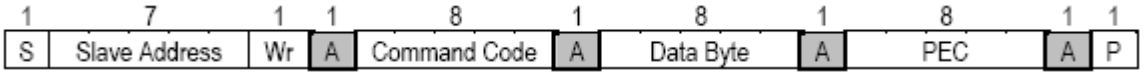
Note: S_Add-R = slave_address(bit7~1) + read_bit (bit 0 = 1)

3.3.1.3 I2C Write Protocol

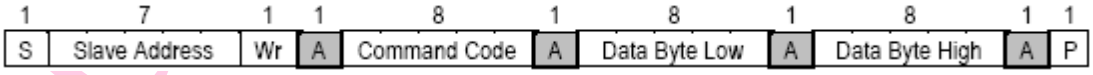
Write Byte: CMD = 0x01



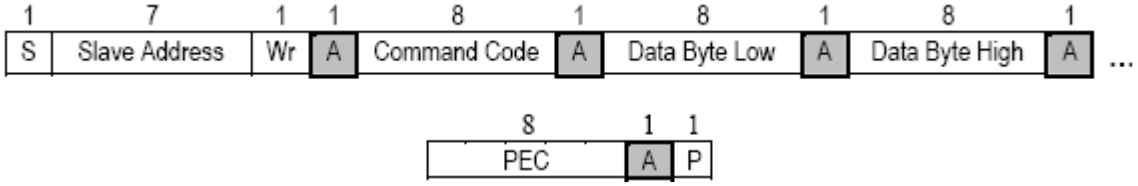
Write Byte with PEC: CMD = 0x01



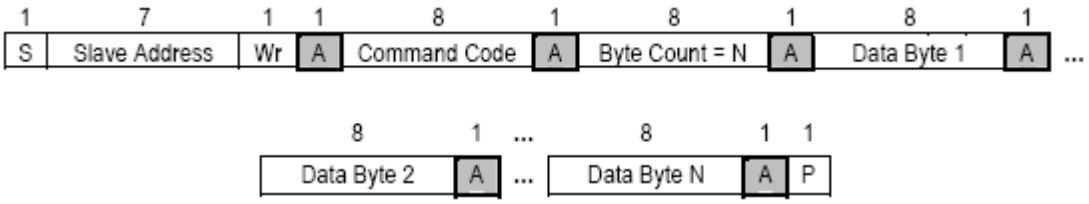
Write Word: CMD = 0x02



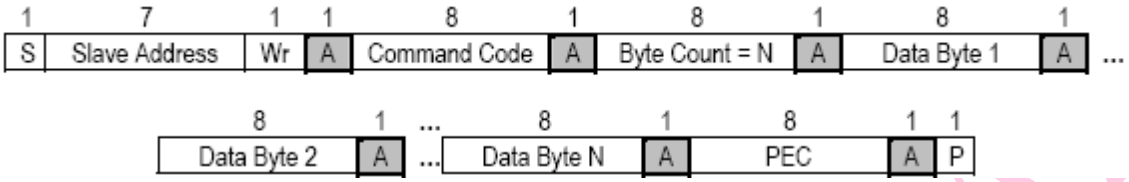
Write Word with PEC: CMD = 0x02



Write Block: CMD = 0x03 (Byte Count = 0~31, 5bits)

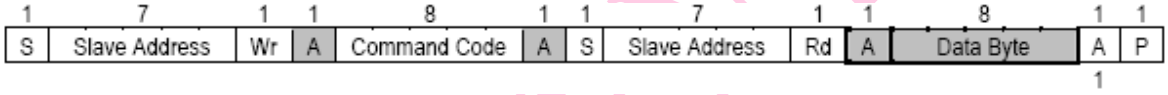


Write Block with PEC: CMD = 0x03 (Byte Count = 0~31, 5bits)

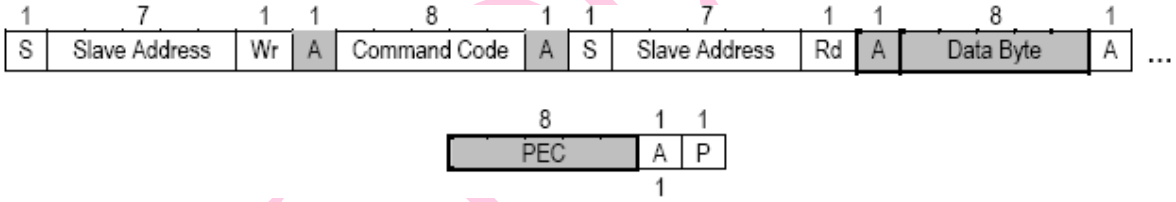


3.3.1.4 I2C Read Protocol

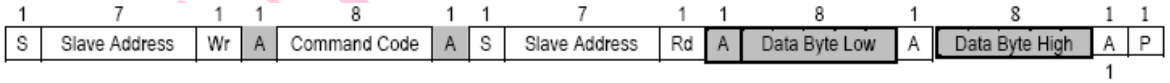
Read Byte: CMD = 0x81



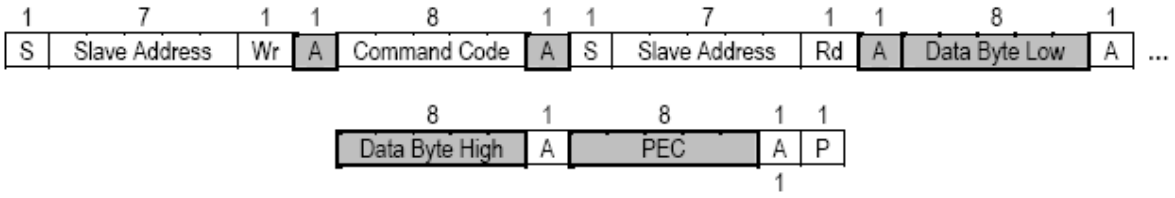
Read Byte with PEC: CMD = 0x81



Read Word: CMD = 0x82



Read Word with PEC: CMD = 0x82



Read Block: CMD = 0x80 + Byte Count (0~31, 5bits)

Read Block with PEC: CMD = 0x80 + Byte Count (0~31, 5bits)

3.3.1.5 I2C Command code

I2C Command	Description
0x00	Set read/write address
0x01	Write 1 byte
0x02	Write 1 bit
0x03	write 1 block (Length: CNT) (Byte Count = 0~31, 5bits)
0x11	Read configured address
0x81	Read 1 byte
0x82	Read 1 bit
0x80+CNT	Read 1 block (Length: CNT) (Byte Count = 0~31, 5bits)

3.3.1.6 I2C Example procedure

```
//    SB358xB_I2C.c
//SCL & SDA The procedure may be different to work with various SoC
#define SDA                                {/* customer code */}
#define SCL                                {/* customer code */}

#define SET_SDA_OUTPUT_MODE          {/* customer code */}
#define SET_SDA_INPUT_MODE           {/* customer code */}
#define SET_SCL_OUTPUT_MODE          {/* customer code */}
#define SET_GPIO_MODE                {/* customer code */}

#define SET_SDA_HIGH                 {/* customer code */}
#define SET_SDA_LOW                  {/* customer code */}
```



```
#define IF_SDA_IS_HIGH                  {/* customer code */}
#define IF_SDA_IS_LOW                   {/* customer code */}

#define SET_SCL_HIGH                    {/* customer code */}
#define SET_SCL_LOW                     {/* customer code */}
#define I2C_Wait()                      {/* customer code */}

// SB358x SMBUS slave address
#define         SB358x_ADDR         0x80     // device's slave address
// SB358x SMBUS commands
#define     SB358x_SETADDR          0x00    // set address cmd
#define     SB358x_WR1BYTE          0x01    // write 1 byte cmd
#define     SB358x_WRBLOCK          0x03    // write 1 byte cmd
#define     SB358x_RD1BYTE          0x81    // read 1 byte cmd
#define     SB358x_RD1WORD          0x82    // read 1 word cmd
#define     SMBUS_WRITE_SIZE         32
#define     SMBUS_READ_SIZE          32
#define     SMBUS_WR                  0
#define     SMBUS_RD                  1

BYTE SB358x_I2C_WriteByte(WORD Addr, BYTE Data)
{
    BYTE result = 1;                                    //1= success

    I2C_start();
    if (I2C_write(SB358x_ADDR|SMBUS_WR))                //SB358x chip address
        return 0;

    if (I2C_write(SB358x_SETADDR))                      // CMD = 00       -- set address
        return 0;

    if (I2C_write(Addr >> 8))                           // addr high
        return 0;

    if (I2C_write(Addr & 0xFF))                         // addr low
        return 0;
    I2C_stop();

    I2C_start();
```



```
    if (I2C_write(SB358x_ADDR|SMBUS_WR))          //SB358x chip address
        return 0;

    if (I2C_write(SB358x_WR1BYTE))                // CMD = 01      -- write 1 byte
        return 0;

    if (I2C_write(Data))                          // write 1 byte data
        return 0;
    I2C_stop();

    return 1;

}

BYTE SB358x_I2C_ReadByte(WORD Addr, BYTE *p_data)
{
    I2C_start();
    if (I2C_write(SB358x_ADDR|SMBUS_WR))          // SB358x chip address
        return 0;

    if (I2C_write(SB358x_SETADDR))                // CMD = 00      -- set address
        return 0;

    if (I2C_write(Addr >> 8))                     // addr high
        return 0;

    if (I2C_write(Addr & 0xFF))                   // addr low
        return 0;
    I2C_stop();

    I2C_start();
    if (I2C_write(SB358x_ADDR|SMBUS_WR))          // SB358x chip address
        return 0;

    if (I2C_write(SB358x_RD1BYTE))                // CMD = 81      -- read 1 byte
        return 0;


    I2C_start();                                  // re-start
```



```
    if (I2C_write(SB358x_ADDR|SMBUS_RD))          //SB 358x chip address (+1 for Read )
        return 0;

    *p_data = I2C_read();
    I2C_Send_ACK(1);                              // Send NACK

    I2C_stop();
    return 1;

}

BYTE SB358x_I2C_ReadWord(WORD Addr, BYTE *p_data)
{
    I2C_start();
    if (I2C_write(SB358x_ADDR|SMBUS_WR))          //SB 358x chip address
        return 0;

    if (I2C_write(SB358x_SETADDR))                // CMD = 00      -- set address
        return 0;

    if (I2C_write(Addr >> 8))                     // addr high
        return 0;

    if (I2C_write(Addr & 0xFF))                   // addr low
      return 0;
    I2C_stop();

    I2C_start();
    if (I2C_write(SB358x_ADDR|SMBUS_WR))          // SB358x chip address
        return 0;

    if (I2C_write(SB358x_RD1WORD))                // CMD = 82   -- read 1 word
        return 0;

    I2C_start();                                  // re-start
    if (I2C_write(SB358x_ADDR|SMBUS_RD))          // SB358x chip address (+1 for Read )
        return 0;

    *(p_data +1) = I2C_read();                    // because 8051 is big endian
```



```
    I2C_Send_ACK(0);                        // Send ACK

    *p_data = I2C_read();
    I2C_Send_ACK(1);                        // Send NACK

    I2C_stop();
    return 1;

}


BYTE SB358x_I2C_ReadBlock(WORD usAddr, BYTE ucLength, BYTE *p_data)
{
    BYTE length;
     if(ucLength == 0)
    {
        return 0;      //FAIL
    }
    while (ucLength > 0)
    {
        I2C_start();
        if (I2C_write(SB358x_ADDR|SMBUS_WR))        //SB 358x chip address
            return 0;

        if (I2C_write(SB358x_SETADDR))               // CMD = 00       -- set address
            return 0;

        if (I2C_write(usAddr >> 8))                  // addr high
            return 0;

        if (I2C_write(usAddr & 0xFF))                // addr low
            return 0;
        I2C_stop();

        if(ucLength > SMBUS_READ_SIZE)
        {
            length = SMBUS_READ_SIZE;
            ucLength -= SMBUS_READ_SIZE;
            usAddr += SMBUS_READ_SIZE;
```



```
        }
        else
        {
            length = ucLength;
            ucLength = 0;
        }

        I2C_start();
        if (I2C_write(SB358x_ADDR|SMBUS_WR))        // SB358x chip address
            return 0;

        if (I2C_write(0x80 + length))               // CMD = (0x80 + n)    -- read n byte
            return 0;

        I2C_start();                                // re-start
        if (I2C_write(SB358x_ADDR|SMBUS_RD))        //SB358x chip address (+1 for Read )
            return 0;

        I2C_read();                                 // first read byte count
        I2C_Send_ACK(0);
        while ((length--) > 1)
        {
            *(p_data++) = I2C_read();
            I2C_Send_ACK(0);
        }

        *(p_data++) = I2C_read();
        I2C_Send_ACK(1);                            // Send NACK

        I2C_stop();
    }

    return 1;

}

BYTE SB358x_I2C_WriteBlock(WORD usAddr, BYTE ucLength, BYTE *pWriteArray)
{
    BYTE length;
```



```
 if(ucLength == 0)
{
      return 0;                                    //FAIL
}

while(ucLength > 0)
{
      I2C_start();
      if (I2C_write(SB358x_ADDR|SMBUS_WR))       //SB 358x chip address
            return 0;

      if (I2C_write(SB358x_SETADDR))              // CMD = 00       -- set address
            return 0;

      if (I2C_write(usAddr >> 8))                 // addr high
            return 0;

      if (I2C_write(usAddr & 0xFF))               // addr low
            return 0;
      I2C_stop();

       if(ucLength > SMBUS_WRITE_SIZE)
      {
            length = SMBUS_WRITE_SIZE;
            ucLength -= SMBUS_WRITE_SIZE;
            usAddr += SMBUS_WRITE_SIZE;
      }
      else
      {
            length = ucLength;
            ucLength = 0;
      }

      I2C_start();
      if (I2C_write(SB358x_ADDR|SMBUS_WR))       //SB358x chip address
            return 0;
      if (I2C_write(SB358x_WRBLOCK))              // CMD = 03       -- write 1 byte
            return 0;
      if (I2C_write(length))                      // data length
```



```
            return 0;

        while(length--)
        {
            if(I2C_write(*pWriteArray++))
            {
                return 0;
            }
        }

        I2C_stop();

    }
    return 1;

}

BYTE I2C_read(void)                     //I2C reads a 8-bit number
{
    BYTE i;
    BYTE tmp;

    SET_SDA_INPUT_MODE;
    tmp = 0;

    for(i = 0;i < 8; i++)
    {
        SET_SCL_LOW;
        I2C_Wait();
        SET_SCL_HIGH;
        I2C_Wait();
        tmp <<= 1;
        IF_SDA_IS_HIGH
            tmp++;
    }
    SET_SCL_LOW;
    SET_SDA_OUTPUT_MODE;
    I2C_Wait();
    return tmp;
```



```
}

BYTE I2C_write(BYTE tmp)                    //I2C writes a 8-bit number
{
    BYTE i;
    BYTE ack;

    for(i = 0;i < 8; i++){
        SET_SCL_LOW;
        I2C_Wait();
        if (tmp & 0x80)
            SET_SDA_HIGH;
        else
            SET_SDA_LOW;
        tmp <<= 1;
        I2C_Wait();
        SET_SCL_HIGH;
        I2C_Wait();
    }
    SET_SCL_LOW;
    I2C_Wait();

    ack = I2C_Check_ACK();

    return ack;

}

void I2C_start(void)
{
    // First pull SDA and SCL to High
    SET_SDA_HIGH;
    SET_SCL_HIGH;
    //I2C_Wait();
    // Pull SDA to low when SCL is high
    SET_SDA_LOW;
    I2C_Wait();
    SET_SCL_LOW;
```



```
    I2C_Wait();
}


void I2C_stop(void)
{
   // First pull SDA and SCL to low
   //SET_SCL_LOW
   //I2C_Wait();
   SET_SDA_LOW;
   I2C_Wait();
   SET_SCL_HIGH;
   I2C_Wait();
    // Pull SDA to high when SCL is high
   SET_SDA_HIGH;
   I2C_Wait();

}

BYTE I2C_Check_ACK()
{
      BYTE ucAck;
      ucAck = 1;                                  // set default to NACK


      SET_SDA_INPUT_MODE;
      I2C_Wait();
      SET_SCL_HIGH;
      I2C_Wait();
      //slave respond ACK(0)
      IF_SDA_IS_LOW
            ucAck = 0;
      SET_SCL_LOW;
      I2C_Wait();
      SET_SDA_OUTPUT_MODE;
      I2C_Wait();
      SET_SDA_HIGH;
      return ucAck;
}
```



```
/* send ACK(0) or NACK(1) */
void I2C_Send_ACK(BYTE ack)
{
   //SDA=ack;
   if (ack)
        SET_SDA_HIGH;
   else
        SET_SDA_LOW;
   I2C_Wait();
   SET_SCL_HIGH;
   I2C_Wait();
   SET_SCL_LOW;
}

void I2C_Init()
{
      SET_GPIO_MODE;
      SET_SDA_OUTPUT_MODE;
      SET_SCL_OUTPUT_MODE;
      SET_SCL_HIGH;
      SET_SDA_HIGH;
}
```


3.3.2 Register Configuration

This Chapter outlines information regarding register configuration for SB3585B and SB3586B.

3.3.2.1 Touch Key

BTNSCANEN0 [0x09D] initiation of TK7 to TK0;

BTNSCANEN1 [0x09E][0] initiation of TK8;

3.3.2.2 TK scan register

SB358xB supports up to 9 touch keys (TK).

SB358xB adopts charge transfer design scheme, which complete touch scan cycle includes **Charge**, **Charge Transfer** and **Discharge** phases**.** These 3 phases can be controlled by S0, S1 and S2 registers. The corresponding control registers are S0TIMER[0x083], S1TIMER[0x084] and PREDISTIM[0x088]. Configuration of S0 and S1 will affect the raw-count values, and S2 is to adjust the sensitivity of TK. Considering the performance and system design flexibility, each TK is controlled (enable/disable) by an individual register, the shielding function and de-bounce number are programmable by controlled registers. De-bounce is controlled by DEBNVERF[0x086], where bits[3:0] sets the de-bounce number below reference voltage level, and bits[7:4] sets the de-bounce number beyond reference voltage level. Increasing de-bounce number is able to enhance the noise immunity but will lower the TK reaction speed at the same time.

There are some parameters must be configured cautiously, please refer to the details below:

For specific system design requirements and TK performance adjustments, please refer to individual AP note for detail register information or contact ENE FAE support.

Recommended configuration for TK registers**:**

Register Name [offset]	Recommendation	Description
S0TIMER[0x083]	0x01	S0 phase, SB IC charges TK during this phase.
S1TIMER[0x084]	0x01	S1 phase, charge transfers during this phase.
IDLETIMER[0x085]	0x00	S0, S1 idle time
SCANSET[0x089]	0x1A	TK scan configuration
SCANEN[0x0A0]	0x01	Bit0 scan initiation
PREDISTIM[0x088]	0xFF	S2 phase, SB IC discharge Cext during this phase.
DEBNVREF[0x086]	0x31	De-bounce Number

3.3.2.3 Window Mode

TK controller will read the raw counts of each corresponding Scan Window:

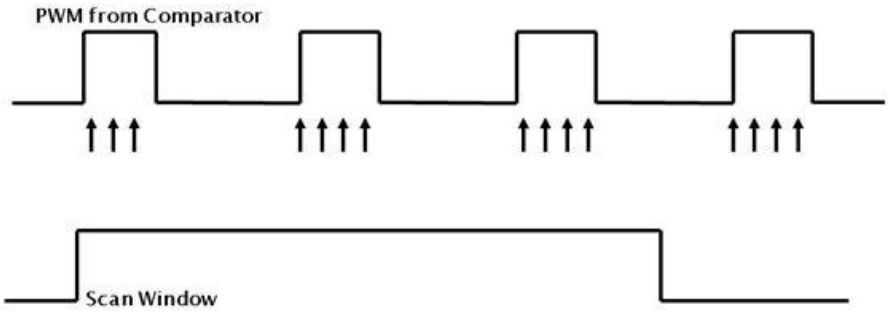


Diagram: Window Mode

At Window Mode, SOC will set an appropriate scan window length to read the raw counts on touch keys. The length of scan window is defined by SCANWIN[0x087]. Bigger the scan window may have larger raw counts, correspondingly, the TK total scan time is longer.

Recommended register configuration under Window Mode is as below:

Register Name [offset]	Recommendation	Description
SCANWIN[0x087]	0x60	Length of scan window
SCANEN[0x0A0]	0x01	Scan initiation

3.3.2.4 Low Power Mode

To initiate SB low power mode by bit3 of SCANEN[0x0A0]. Under low power mode, IC operates at lower clock speed to save power. Duration of low power mode is configured by register WAKETMR[0x0A8]. After low power mode is expired, IC will initiate TK scan to see if there is any TK is initiated during the low power mode, the number of scan is configured by bits[7:2] of SLPTMR1[0x098]. If the scan result indicate any is TK is initiated, IC will return to normal operation mode. Otherwise, IC will enter low power mode again after the scan is finished and repeat next WAKETMR[0x0A8] cycle.

Example**:**

1. Configure low power mode duration

Write WAKETMR[0x0A8] as 0x10, in this case, the timer is set to be 16, the duration for low power mode is 16ms x (16+1) = 272ms, where 16ms is the clock for low power mode.

2. Enable low power mode

   Write SCANEN[0x0A0] as 0x09

3. Set number of scan

Write SLPTMR1 [7:2] as 0b000001=1, in this case, the scan number is (1+1) = 2, which means SB IC will wake up every 272ms, then issue TK scan 2 times to decide whether or not to stay under low power mode.

Recommended register setting for low power mode

Register Name [offset]	Recommendation	Description
SLPTMR0[0x097]	0x00	Sleep timer
SLPTMR1[0x098]	0x07	Sleep timer MSB (bits[1:0]) Interrupt event check period
WT2SCAN[0x099]	0x02	OSC 6MHz stable time
WAKETMR[0x0A8]	0x10	Wake up timer
SCANEN[0x0A0]	0x09	Bit3 is to initiate low power mode

3.3.2.5 GPIO

TK pins can also act as IO (GPIO) and PWM. Function selection can be configured by BTNSCANENx (x represents 0~8). The control registers control individual TK pins and may be set independently.

When BTNSCANENx is set to "1", TK function is initiated. When BTNSCANENx is set to "0", TK pin is initiated as GPIO or PWM. GPIOMD0[0x100] and GPIOMD1[0x101] is to select GPIO or PWM. Set GPIOMDx to "00" to initiate GPIO function. User needs to set related registers GPIOIE0~GPIOIE1, GPIOOE0~GPIOOE1, andGPIODO0~GPIODO1 to complete the configuration of GPIO as input or output mode. Control registers GPIODI0~GPIODI1 contain the input data through GPIO.

Pin Name	Pin function	Register
TK	Touch Key	BTNSCANEN='1'
	GPIO	BTNSCANEN='0' & GPIOMD='0'
	PWM	BTNSCANEN='0' & GPIOMD='1'

Please refer to Chapter 3 for detail register definitions.

Example**:**

This example demonstrates how to set TK0~TK3 as GPIO output mode; set TK4~TK7 as touch keys; set TK8/GPIOB0/GPIOB1/GPIOB2 as GPIO input mode:

1. To initiate TK
   A. Write BTNSCANEN0[0x09D] as 0xF0
   B. Write BTNSCANEN1[0x09E] as 0x00
2. Set TK pin GPIO
   A. Write GPIOMD0[0x100] as 0x00
   B. Write CPIOMD1[0x101] as 0x00
3. Set GPIO pin as output or input mode
   A. Write GPIOIE1[0x107] as 0x0F
   B. Write GPIOOE0[0x108] as 0x0F
4. Read or Write data through GPIO
   A. Read GPIODI1[0x103] data that is input from TK8/GPIOB0/GPIOB1/GPIOB2
   B. Write data to GPIODO0[0x104][3:0], then output through TK0~TK3

3.3.2.6 PWM

To initiate PWM function via GPIOMD0[0x100] & GPIOMD1[0x101].

PWMOM0[0x10C] & PWMOM1[0x10D] are used to select push-pull mode or open-drain mode.

Register PWMBCK is for selecting PWM clock, recommended value is 0xBA. PWM clock is a gated clock, which can be configured through register GPIOMISC[0x1A0] bit0. GPIOMISC[0x1A0] description as follows:

Address	Name	Bit	Type	Description	Default
0x1A0	GPIOMISC	**GPIO miscellaneous**			0x00
		7-2	RSV	Reserved	
		1	RW	Reserved	
		0	RW	When PWM mode is disable, PWM clock status: 0: Normal 1:Stop	

SB series support up to 12 GPIO/PWM pins. Each PWM pin has its indicidual register controller: [0x110]~[0x11B]，[0x180]~[0x18B].

PWMBCK[0x10E] is for PWM clock configuration, PWMMCL [0x10F] is for PWM cycle length configuration.

PWM frequency and duty configuration

After complete the related register settings, corresponding GPIO pins will output the configured PWM cycle. These registers include duty control registers ([0x180] ~ [0x18B]), PWM clock register (PWMBCK[0x10E]) and cycle length register (PWMMCL[0x10F]).

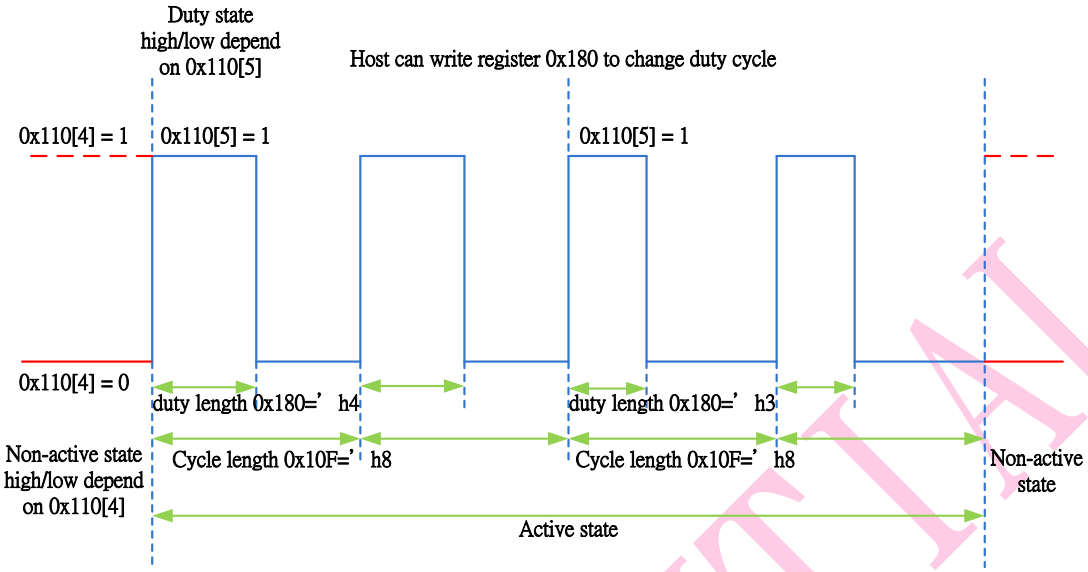
Example:

To set GPIOA0 as push-pull PWM，50% duty cycle and 1.5 MHz frequency:

1. Configure GPIOA0 pin as PWM mode
   A. Write GPIOMD0[0x100] bit[0] as 1
   B. Write PWMOM0[0x10C] bit[0] as 0
2. Configure PWM duty cycle and frequency
   A. Write PWMBCK[0x10E] as 0x00
   B. Write PWMMCL[0x10F] as 0x04
   C. Write MANUHI0[0x180] as 0x02

PWM parameter introduction

The PWM waveform is depicted as following:



Active/non-active state depend on 0x110[3:0].
If event happened, PWM will enter active state, else stay in non-active state.
Ex: 0x110[3:0]=' h5, active state (button 5 on), non-active state (button 5 off).

3.3.2.7 RESET Scan

Once the abnormal TK scan behavior is found, it is recommended to reset scan.

Complete Reset Scan procedure is as follows:

1. SCANEN[0xA0][0] = 0 (disable scan function)
2. SCANSET[0x89][1] = 0 (disable threshold comparison)
3. BLCTRL[0x93][4] = 0 (baseline adjustment disable)
4. BLCTRL[0x93][4] = 1 (baseline adjustment initiate)
5. SCANSET[0x89][1] = 1 (enable threshold comparison)
6. SCANEN[0xA0][0] = 1 (scan function initiate)

Address	Name	Bit	Type	Description	Default
0x0A0	SCANEN	**Scan enable/low power mode enable**			0x00
		7-5	RSV	Reserved	
		4	RW	Stop charging CEXT during Discharge phase 0:Do not stop 1:Stop	
		3	RW	Low power mode enable 0: Disable 1: Enable	
		2	RW	Single TK mode Designing multiple TKs on a relatively compact system may introduce interference between neighboring TKs, this control register limits only the first touched TK to be triggered to avoid malfunction. 0: Multiple TKs may be triggered at the same time 1: Single TK can be triggered at the same time	
		1	RW	Multiple TK triggered enable Once SCANEN[2]='0', multiple TK mode is allowed, however, in the cases of water flooding and noisy environment, multiple TK may be false triggered. This control register tries to eliminate this noise interference. This register is valid only when SCANEN[2]='0'. 0: Once multiple TK are triggered, the corresponding output pins drive signals simultaneously. ◦ 1: Once multiple TK are triggered, no any output pin drives signal, this scenario is defined as malfunction and ignored.	
		0	RW	Scan enable This register controls whether or not to execute scan function. Other scan control registers will only be active after this register is initiated. 0: Disable 1: Enable	

Address	Name	Bit	Type	Description	Default
0x089	SCANSET	**Scan configuration**			0x1A
		7-4	RW	TK de-bounce 0000: 1 scan cycle 0001: 2 scan cycles ---------- 1110: 15 scan cycles 1111: Reserved	
		3-2	RW	Trigger mode selection TK trigger has 3 different modes: Touch trigger (finger on TK)，Touch end trigger (finger off TK) and both to trigger. Once the defined trigger mode happens, the pending flag is set, working together with the interrupt enable, then the TK trigger will issue interrupt. 00: Touch trigger 01: Touch end trigger 10: Both trigger 11: Reserved	
		1	RW	Threshold Comparison 0: Disable 1: Enable	
		0	RW	TK pins status during non-scan phase 0: Ground 1: Floating (This setting is valid only when shielding function is disabled.)	

Address	Name	Bit	Type	Description	Default
0x093	BLCTRL	**Scan baseline adjustment**			0x00
		7-5	RSV	Reserved	
		4	RW	Scan baseline adjustment enable 0: Disable 1: Enable	
		3-0	RW	Initial baseline adjustment cycle count 0000: 1 scan cycle 0001: 2scan cycles ---------- 1110: 15 scan cycles 1111: Reserved	

3.4. Advanced Functions

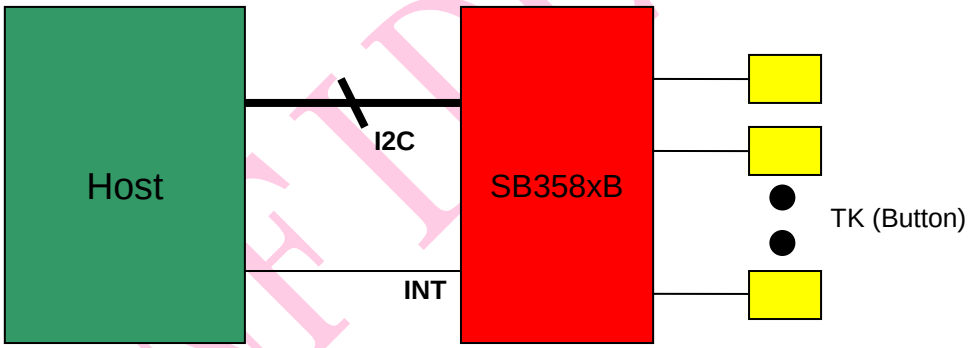
3.4.1 Interrupt

SB358x will scan the enabled TK pins in order, the scan result generates 2-byte raw count to respond the TK capacitive change due to touch behavior. Once the raw count number is larger than pre-defined threshold value, SB358x drives INT# low to inform SOC there is touch event.

Once SOC receives SB interrupt notice, it will identify certain TK has been triggered by readying registers BUTPF0[0x0A4] and BUTPF1[0x0A5]. Moreover, registers BUTSTA0[0x0A1] and BUTSTA1[0x0A2] has recorded the TK status of each corresponding TK that can be used to judge each TK capacitance increase (finger on TK) or TK capacitance decrease (finger off TK).

It is recommended to clear TK flag status by writing "1" to registers BUTPF0[0x0A4] and BUTPF1[0x0A5]. Otherwise, TK status can be mis-judged. There are 3 TK trigger modes: Touch trigger (finger on TK), Touch end trigger (finger off TK) and Both trigger. Register SCANSET[0x089] bit [3:2] is for selecting initiation mode. Register SCANSET[0x089] [7:4] is for debounce configuration.

Please refer to Chapter 3 for details on registers.



Example:

1. Initiate TK8~0 as Touch Keys
   A. Write BTNSCANEN0[0x09D] as 0xFF
   B. Write BTNSCANEN1[0x09E] as 0x01
2. Enable Touch trigger mode and Touch end trigger mode, threshold comparison, and disable de-bounce.
   A. Write SCANSET[0x089]为 0x0B
3. Initiate touch scan
   A. Write SCANEN[0x0A0] bit 0 as 1

3.4.2 Baseline Adjustment

The TK capacitance may vary with ambient temperature and humidity change, and different cover materials, if the pre-defined threshold is a fixed number, the TK performance won't be consistence and guaranteed under the environmental changes. To solve this issue, SB358x adopts baseline adjustment scheme that continuously reads the external TK capacitance changes and come out a baseline based on corresponding raw counts, the pre-defined threshold refers to this baseline and adjust its value accordingly. Baseline adjustment effectively eliminates the environmental effect to TK performance.

Register BLCTRL[0x093]bit 4='1' enables baseline adjustment, register BLCTRL[0x093][0:3] defines the scan cycle count to decide the baseline during SB3589x initialization phase. More scan cycle counts generate more reliable baseline, however, it takes longer scan time.

As shown below, the baseline adjustment operation is basically divided into five operating region, each operation region needs to be set through the corresponding register, the relevant registers ([0x8A] ~ [0x91]) is defined in the following table:

Address	Name	Bit	Type	Description	Default
0x08A	BL_ZX1	**Baseline Calculation Coefficient zx1**			0x00
		7-0	RW	**Calculation Coefficient for region 1**	
0x08B	BL_Z0X2	**Baseline Calculation Coefficient z0x2**			0x7C
		7-0	RW	**Calculation Coefficient is** BL_Z0X2/128	
0x08C	BL_Z1X2	**Baseline Calculation Coefficient z1x2**			0x04
		7-0	RW	**Calculation Coefficient** is BL_Z1X2/128	
0x08D	BL_Z0X3	**Baseline Calculation Coefficient z0x3**			0x78
		7-0	RW	**Calculation Coefficient** is BL_Z0X3/128	
0x08E	BL_Z1X3	**Baseline Calculation Coefficient z1x3**			0x08
		7-0	RW	**Calculation Coefficient** is BL_Z1X3/128	
0x08F	BL_Z0X4	**Baseline Calculation Coefficient z0x4**			0x78
		7-0	RW	**Calculation Coefficient** is BL_Z0X4/128	
0x090	BL_Z1X4	**Baseline Calculation Coefficient z1x4**			0x08
		7-0	RW	**Calculation Coefficient** is BL_Z1X4/128	
0x091	BL_ZX5	**Baseline Calculation Coefficient zx5**			0x20

Register ([0x8A] ~ [0x91]) definition and baseline adjustment calculation example:
BL_ZX1[0x8A]: region 1 **Calculation Coefficient**
Update baseline = original baseline + BL_ZX1[0x8A].

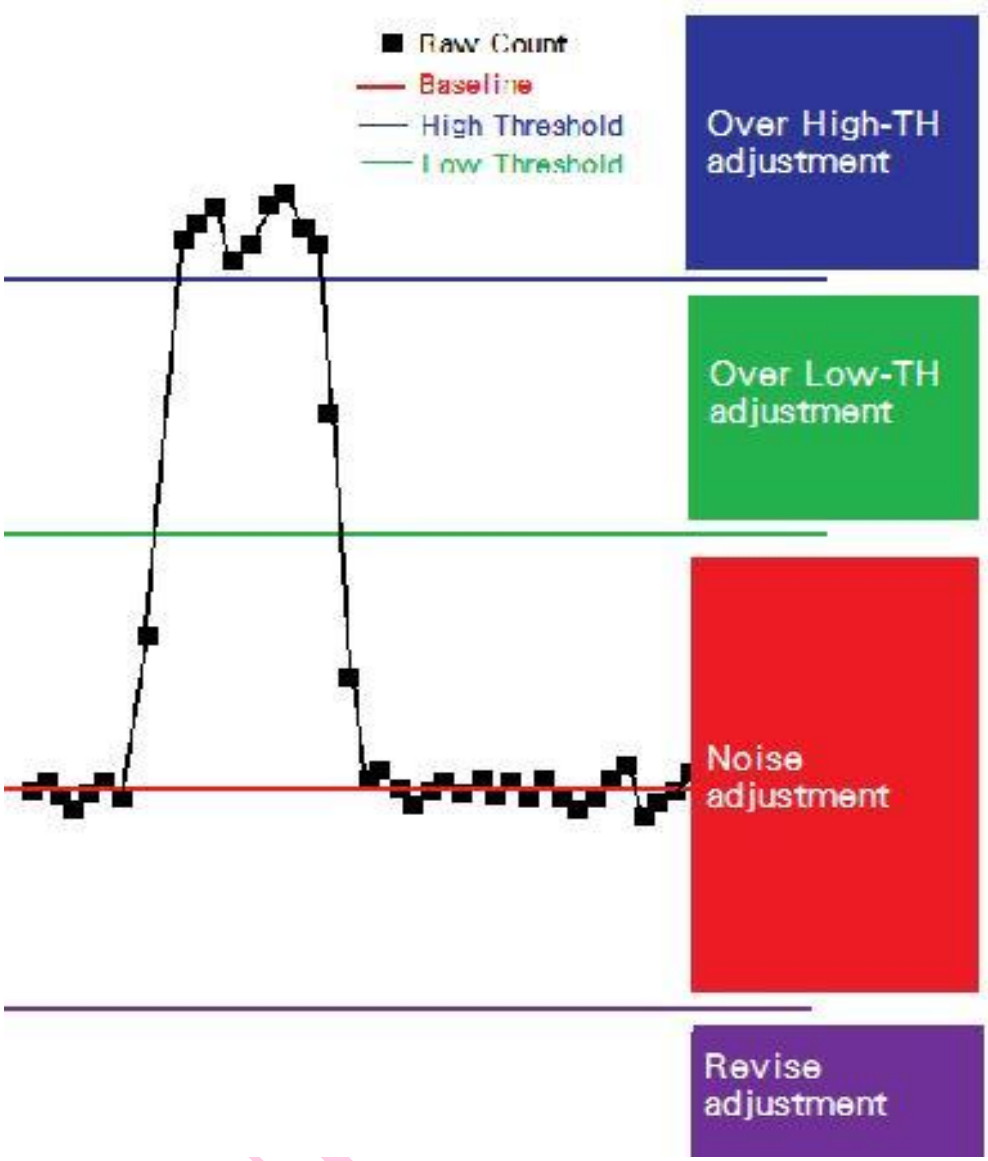
BL_Z0X2[0x8B], BLZ1X2[0x8C]: region 2 **Calculation Coefficient**
Update baseline = original baseline * BL_Z0X2[0x8B]/128 + updated raw count * BL_Z1X2[0x8C]/128

BL_Z0X3[0x8D], BLZ1X3[0x8E]: region 3 **Calculation Coefficient**
Update baseline = original baseline * BL_Z0X3[0x8D]/128 + updated raw count * BL_Z1X3[0x8E]/128

BL_Z0X4[0x8F], BLZ1X4[0x90]: region 4 **Calculation Coefficient**
Update baseline = original baseline * BL_Z0X4[0x8F]/128 + updated raw count * BL_Z1X4[0x90]/128

BL_ZX5[0x91]: region 5 **Calculation Coefficient**
Update baseline = original baseline - BL_ZX5[0x90] * Delta/128.

Note: Delta means the corresponding TK Delta registers settings: DELTA0[0x54]~DELTA8[0x65].



3.4.3 Low Power Mode

SB358xB low power mode enables system to go into hibernation status and requires external initiation of TK to reinstate the operation mode thus enhance power saving efficiency. Low power mode operation procedure as following:

SB358xB requires external TK initiation to wake up the system. Therefore, low power mode is to put SB IC into hibernation status periodically. During the hibernation status, , SB will terminate most of the power supply then automatically return to normal operation mode to judge if there is any external initiation. If there is no TK initiation event during this reinstated normal operation mode, system will back to hibernation mode after the pre-set TK scan is completed.

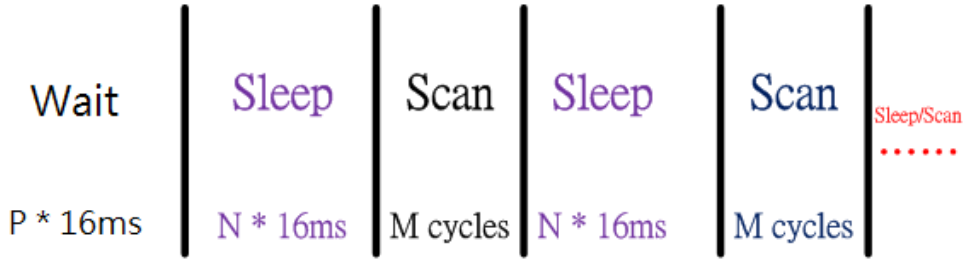
Under the low power mode, there are some timers need to be setup through the register settings to achieve system performance. The timers and relevant registers are:

1. The time period that SB358x will enter sleep mode. (Register SLPTMR0[0x097] and SLPTMR1[0x098][1:0]), this timer setting unit is 16ms.
2. The time period for each sleeping cycle, SB358x will go back to normal operation mode once this timer is expired. (Register WAKETMR[0x0A8]), this timer setting unit is 16ms.. Therefore, the actual time is WAKETMR[0x0A8] *16ms.
3. The time period between SB358x return back to normal operation mode and start TK scan operation. (Register WT2SCAN[0x099])
4. The TK scan time for each periodic return back normal operation mode. (Register SLPTMR1[0x098][7:2]), the maximum number is 63.
5. Initiate low power mode (Register SCANEN[0x0A0][3])

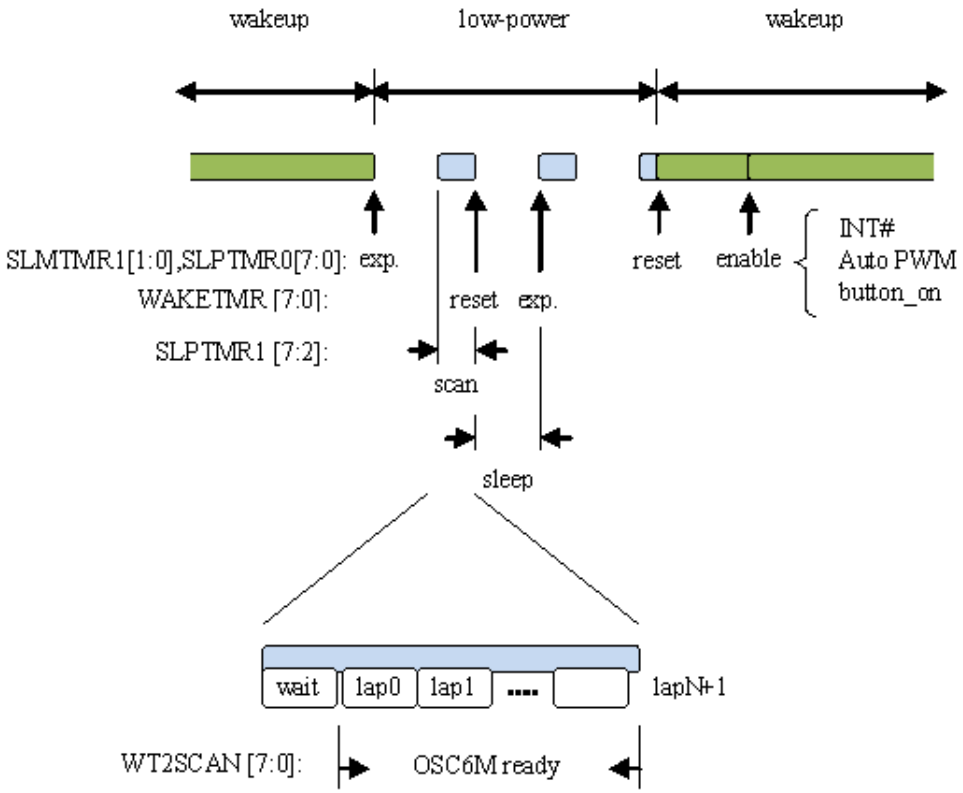
Note:

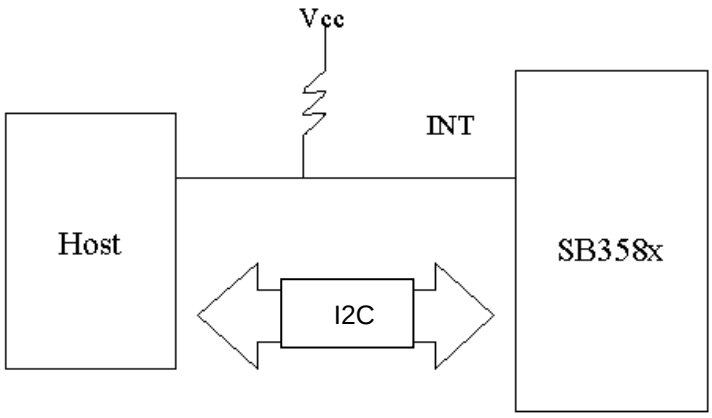
Under low power mode, SB IC will return to normal operation after periodical TK scan. It is also possible to wake up SB IC through INT pins and return to normal operation (Register SLPWAKE[0x0A7][4]).

Under normal operation, INT pin is to inform SB IC whether there is any TK initiation event (Register TNINTEN0[0x09A] & BTNINTEN1[0x09B]).



Low power mode timing





INT :

1. If SB358x detects some event, then SB358x drives it low to the host
2. If the host wants to resume SB358x, host should drive it low to SB358x

Example:

Initiate low power mode, wake up every 1.6 seconds, 3 scans for every wake up.

A. Set hibernation time 1600ms
   Write WAKETMR[0xA8] as 0x64. ex. WAKETMR = 0x64;

B. Set scan for 3 times under low power mode
   Write SLPTMR1 [0x098][7:2]为 3.
   ex. SLPTMR1 |= 0x0C;
   SLPTMR1 &= ~0xF0;

C. Set interrupt
   Write SLPWAKE[0x0A7]为 0x11.

D. Initiate low power mode
   Write SCANEN[0x0A0]为 0x09. ex. SCANEN = 0x09

3.4.4 Anti-high humidity

Under high humidity environment, user may face a possible scenario:

Equipment with 9 touch buttons, there is only one button will response at a time.

SOC will judge whether a touch button trigger event happens by checking button status register BUTSTA0[0XA1] & BUTSTA1[0xA2], the trigger mode is Touch end trigger (finger off TK).

1. SB358x reset process is:
   SCANEN[0xA0][0] = 0 (scan disable)
   SCANSET[0x89][1] = 0 (disable threshold comparison)
   BLCTRL[0x93][4] = 0 (baseline adjustment disable)
   BLCTRL[0x93][4] = 1 (baseline adjustment enable)
   SCANSET[0x89][1] = 1 (enable threshold comparison)
   SCANEN[0xA0][0] = 1 (scan enable)

Example:

```
sb358x_chip_reset(void)
{
    //disable scan, baseline adjustment and scan thd comparison
    SCANEN[0xA0][0] = 0;
    SCANSET[0x89][1] = 0;
    BLCTRL[0x93][4] = 0;
    //enable scan, baseline adjustment and scan thd comparison
    BLCTRL[0x93][4] = 1;
    SCANSET[0x89][1] = 1;
    SCANEN[0xA0][0] = 1;
}
```

2. When humidity increases, and there are 3 TK been initiated at the same time, user needs to reset.

Example:

```
If (button_be_touched_num >= 3)
    sb358x_chip_reset();
```

3. When humidity increase, and there is any TK been initiated over 10 seconds, user needs to reset.

Example:

```
if (button_be_touched_time > 10 sec)
    sb358x_chip_reset();
```

4. SOC periodically access buttons status registers, if the initial value of the register contents are changed, it is recommended to reset SB358x.

Example:

```
If (timer_int)      //the timer interrupt event or polling at specific period
    sb358x_chip_reg_val_chk();
```


3.4.5 PWM

3.4.5.1 PWM initiation

Example:

GPIOMD0[0x100] = 0x01; //start GPIOA0 PWM mode

3.4.5.2 Set PWM output mode

PWM output mode can be set as open-drain or push-pull according to different circuit connection requirement.

Example:

PWMOM0[0x10C] = 0x81; //config GPIOA0,A7 as open-drain mode, GPIOA1~A6 as push-pull mode

PWMOM1[0x10D] = 0x00; //config GPIOA8,B0~B2 as push-pull mode

3.4.5.3 Set PWM Clock

Register PWMBCK[0x10E] is for selecting PWM clock. When PWMBCK[0x10E] is 0, PWM clock is 6MHz/(0+1) = 6MHz. When PWMBCK[0x10E] is5, PWM clock is 6MHz / (5+1)= 1MHz。

Example:

PWMBCK[0x10E] = 5; //Set PWM Base Clock to 1MHz

3.4.5.4 Set PWM cycle length and duty cycle

Register PWMMCL[0x10F] is to set cycle length. When PWMMCL[0x10F] is 16, PWM cycle length is 16 PWM clocks.

Pin GPIOA0，GPIOA1…GPIOA7，GPIOA8，GPIOB0，GOIOB1 and GPIOB2 can be set as PWM output. Once these pins been selected as PWM output, Register PWMDUTY[0x180] ~ PWMDUTY[0x18B] will be the corresponding duty cycle control registers. The settings in duty cycle registers stand for the time period when PWM drives high voltage level. For example when set PWMMCL[0x10F] as 16, PWM cycle length is 16 PWM clocks. When PWMDUTY[0x180] is 8, that means GPIOA0 will drive PWM output high with 8 PWM clocks. Therefore, GPIOA0 PWM output duty is 8/16 = 50%.

Example:

PWMMCL[0x10F] =16; //Set PWM Cycle Length to 16 clock cycles

PWMDUTY[0x180] = 8; //Set GPIOA0 Duty Length = 8 clock cycles, Duty=8/16=50%

3.4.5.5 Recommended procedure to update PWM configuration

Example:

```
GPIOMD0[0x100][0] = 0;          //set GPIOA0 mode to GPIO mode first
 …                              //set GPIOA0 PWM related parameter
 …
GPIOMD0[0x100][0] = 1;          //set GPIOA0 mode to PWM mode and start PWM
```

3.4.5.6 Update duty cycle setting after PWM initiation

If the duty cycle control registers (0x0180~0x018B) are modified during PWM operation, the updated duty cycle will be effective until next new PWM cycle.

3.4.6 Improving signal quality of touch button

3.4.6.1 Shielding

In some system designs or applications, enabling the shielding function will increase the TK signal quality and sensitivity. Shielding function control register are SLDEN0 [0xCC] and SLDEN1 [0xCD], corresponding to TK0 ~ 8. If any of TK pins enabled touch key function as well as shielding, these TK pins will drive shielding signal during non-scan cycles; If the TK pins are enabled with shielding function only without TK function, they are simply used as shielding pins.

3.4.6.2 SCANSET[0x89] Bit 0 configuration

In slider, wheel and ITO applications, touch buttons are placed next to each other and hard to do shielding placement and layout within limited PCB area. Under such circumstance, user needs to set Register SCANSET[0x89] bit0 as 1. By doing so, TK pins output will be HiZ (floating) during non-scan cycle. Although this configuration will enhance signal but will also create larger noise at the same time. It is recommended to set SCANSET[0x89] bit 0 as 0 for general applications.

3.4.7 Pseudo Code for Error Detection

3.4.7.1 Electric Fan

For power plug tests, in some cases, the power source level may be instantaneously inconsistent to SOC and SB358x, and possibly cause SOC working normally but reset SB358x. This scenario makes SOC fail to access SB358x.

To correct such abnormal condition, SOC can periodically check on Register BTNSCANEN0[0x9D] and compare with the preset initial data, if the initial value of the register contents are changed, it is recommended to reset SB358x.

Example:

```
if (Specific period timing is met)
{
   if (BTNSCANEN0[0x9D] != initial value)
   {
   // initial SB358xB
   }
}
```

3.4.7.2 TV

Some TV projects requires to pass ESD air test up to 15KV. Different system structure design and ESD operation may cause TK malfunctions.

To solve such abnormal condition, set SOC to periodically check on register BTNSCANEN0[0x9D], BTNSCANEN1[0x9E], BTNSCANEN2[0x9F] and SCANEN[0xA0] and compare with the preset initial data, if the initial value of the register contents are changed, it is recommended to reset SB358x.

Example:

```
if (Specific period timing is met)
{
if ((BTNSCANEN0[0x9D] != initial value) || (BTNSCANEN1[0x9E] != initial value)
||   (BTNSCANEN2[0x9F] != initial value) || (SCANEN[0xA0] != initial value))
   {
   // initial SB358xB
   }
}
```


3.4.7.3 Home Appliance

Home appliances need to test 4KV EFT. Different system power design may cause TK malfunction during testing.

To solve such abnormal condition, set SOC to check on register DELTA0[0x54] and compare with the preset initial data, if the initial value of the register contents are changed, it is recommended to reset SB358x.

Example:

```
if (Specific period timing is met)
{
   if (DELTA0[0x54] != initial value)
   {
   // initial SB358xB
   }
}
```


4. Register Files

4.1 Address mapping table

Function	Type	Address	Description
Scan Result 128 bytes	RW	0x00~0x11	Scan result (9 words)
	RW	0x2A~0x3B	Threshold (9 words)
	RW	0x54~0x65	Delta (9 words) baseline + Delta = threshold
	RW	0x7E~0x7F	Reserved

Function	Type	Address	Description
Scan Control 128 bytes	RW	0x80	I2C setting
	RO	0x81	I2C slave address selection
	RW	0x82	IO internal pull-up enable for H/W trap IO
	RW	0x83~0x85	TK scan timing control
	RW	0x86	Vref de-bounce counter
	RW	0x87	Scan window
	RW	0x88	Discharge timing control
	RW	0x89	Scan configuration
	RW	0x8A~0x91	Baseline calculation coefficient (8 bytes)
	RW	0x92	Scan baseline adjustment time
	RW	0x93	Scan baseline adjustment control
	RO	0x94~0x96	Reserved
	RW	0x97~0x99	Hibernation and wakeup control (3bytes)
	RW	0x9A~0x9B	TK interrupt enable (2 bytes)
	RO	0x9C	Reserved
	RW	0x9D~0x9E	TK enable (2bytes)
	RO	0x9F	Reserved
	RW	0xA0	Scan enable/low power mode enable
	RW	0xA1~0xA2	TK status (2 bytes)
	RO	0xA3	Reserved
	WC	0xA4~0xA5	IRQ pending flag(2 bytes)
	RO	0xA6	Reserved
	WC	0xA7	Wakeup interrupt control
	RW	0xA8	Wakeup timer
	RO	0xA9~0xAA	Reserved
	RW	0xAB	LDO/OSC32K hibernation control
	RO	0xAC~0xB6	Reserved
	RO	0xB7	I2C slave status
	RO	0xB8~0xB9	Reserved
	RO	0xBA	POR status
	RO	0xBB~0xBC	Reserved
	RW	0xBD	Reserved
	RW	0xBE	Reserved
	RW	0xBF	Reserved

	RO	0xC0~0xC2	Calculator medium results (3 bytes)
	RO	0xC3	Scan baseline adjustment counter status
	RO	0xC4	Scan control status
	RO	0xC5	Calculator / wakeup status
	RO	0xC6	HW trap value
	RO	0xC7	Current scan port
	RO	0xC8~0xC9	Current scan result
	RO	0xCA	Current debounce counter

Function	Type	Address	Description
GPIO control 192 bytes	RW	0x100~0x101	GPIO mode selection (2 bytes)
	RO	0x102~0x103	GPIO input mode selection (2 bytes)
	RW	0x104~0x105	GPIO output data (2 bytes)
	RW	0x106~0x109	GPIO input/output enable (4 bytes)
	RO	0x10A~0x10B	Reserved
	RW	0x10C~0x10D	PWM output mode selection (2bytes)
	RW	0x10E	PWM clock
	RW	0x10F	PWM cycle length
	RW	0x110~0x11B	GPIOA0~A8/GPIOB0~B2 PWM configuration (12 bytes)
	RO	0x11C~0x17F	Reserved
	RW	0x180~0x18B	GPIOA0~A8/GPIOB0~B2 PWM duty cycle (12 bytes)
	RO	0x18C~0x19F	Reserved
	RW	0x1A0	GPIO miscellaneous register
	RO	0x1A1~0x1FF	Reserved
	RW	0x100~0x101	GPIO mode selection (2 bytes)
	RO	0x102~0x103	GPIO input data (2 bytes)
	RW	0x104~0x105	GPIO output data (2 bytes)

Function	Type	Address	Description
Reserved 64 bytes	RO	0x1C0~0x1FF	Reserved

4.2 Register definition

4.2.1 Scan result

The scan result is a 16-bit data and stored in the following registers, writing low byte data will be cached, and wait until the high byte data is written, then write the combined 16-bit data into RAM to guarantee the data integrity.

Address	Name	Content			Default
		Bit	Type	Description	
0x000	SCANB0L	**Port 0 Scan result low byte**			0
		7-0	RW		
0x001	SCANB0H	**Port 0 Scan result high byte**			0
		7-0	RW		
0x002	SCANB1L	**Port 1 Scan result low byte**			0
		7-0	RW		
0x003	SCANB1H	**Port 1 Scan result high byte**			0
		7-0	RW		
0x004	SCANB2L	**Port 2 Scan result low byte**			0
		7-0	RW		
0x005	SCANB2H	**Port 2 Scan result high byte**			0
		7-0	RW		
0x006	SCANB3L	**Port 3 Scan result low byte**			0
		7-0	RW		
0x007	SCANB3H	**Port 3 Scan result high byte**			0
		7-0	RW		
0x008	SCANB4L	**Port 4 Scan result low byte**			0
		7-0	RW		
0x009	SCANB4H	**Port 4 Scan result high byte**			0
		7-0	RW		
0x00A	SCANB5L	**Port 5 Scan result low byte**			0
		7-0	RW		
0x00B	SCANB5H	**Port 5 Scan result high byte**			0
		7-0	RW		
0x00C	SCANB6L	**Port 6 Scan result low byte**			0
		7-0	RW		
0x00D	SCANB6H	**Port 6 Scan result high byte**			0
		7-0	RW		
0x00E	SCANB7L	**Port 7 Scan result low byte**			0
		7-0	RW		
0x00F	SCANB7H	**Port 7 Scan result high byte**			0
		7-0	RW		
0x010	SCANB8L	**Port 8 Scan result low byte**			0
		7-0	RW		
0x011	SCANB8H	**Port 8 Scan result high byte**			0
		7-0	RW		
0x02A	SCTH0HL	**Port 0 Threshold low byte**			0
		7-0	RW		
0x02B	SCTH0HH	**Port 0 Threshold high byte**			0
		7-0	RW		
0x02C	SCTH1HL	**Port 1 Threshold low byte**			0
		7-0	RW		
0x02D	SCTH1HH	**Port 1 Threshold high byte**			0
		7-0	RW		
0x02E	SCTH2HL	**Port 2 Threshold low byte**			0
		7-0	RW		
0x02F	SCTH2HH	**Port 2 Threshold high byte**			0

		7-0	RW		
0x030	SCTH3HL	**Port 3 Threshold low byte**			0
		7-0	RW		
0x031	SCTH3HH	**Port 3 Threshold high byte**			0
		7-0	RW		
0x032	SCTH4HL	**Port 4 Threshold low byte**			0
		7-0	RW		
0x033	SCTH4HH	**Port 4 Threshold high byte**			0
		7-0	RW		
0x034	SCTH5HL	**Port 5 Threshold low byte**			0
		7-0	RW		
0x035	SCTH5HH	**Port 5 Threshold high byte**			0
		7-0	RW		
0x036	SCTH6HL	**Port 6 Threshold low byte**			0
		7-0	RW		
0x037	SCTH6HH	**Port 6 Threshold high byte**			0
		7-0	RW		
0x038	SCTH7HL	**Port 7 Threshold low byte**			0
		7-0	RW		
0x039	SCTH7HH	**Port 7 Threshold high byte**			0
		7-0	RW		
0x03A	SCTH8HL	**Port 8 Threshold low byte**			0
		7-0	RW		
0x03B	SCTH8HH	**Port 8 Threshold high byte**			0
		7-0	RW		
0x054	DELTA0	**Port 0 Delta low byte**			0xC8
		7-0	RW		
0x055	DELTA0	**Port 0 Delta high byte**			0x00
		7-0	RW		
0x056	DELTA1	**Port 1 Delta low byte**			0xC8
		7-0	RW		
0x057	DELTA1	**Port 1 Delta high byte**			0x00
		7-0	RW		
0x058	DELTA2	**Port 2 Delta low byte**			0xC8
		7-0	RW		
0x059	DELTA2	**Port 2 Delta high byte**			0x00
		7-0	RW		
0x05A	DELTA3	**Port 3 Delta low byte**			0xC8
		7-0	RW		
0x05B	DELTA3	**Port 3 Delta high byte**			0x00
		7-0	RW		
0x05C	DELTA4	**Port 4 Delta low byte**			0xC8
		7-0	RW		
0x05D	DELTA4	**Port 4 Delta high byte**			0x00
		7-0	RW		
0x05E	DELTA5	**Port 5 Delta low byte**			0xC8
		7-0	RW		
0x05F	DELTA5	**Port 5 Delta high byte**			0x00
		7-0	RW		
0x060	DELTA6	**Port 6 Delta low byte**			0xC8
		7-0	RW		
0x061	DELTA6	**Port 6 Delta high byte**			0x00
		7-0	RW		
0x062	DELTA7	**Port 7 Delta low byte**			0xC8
		7-0	RW		

0x063	DELTA7	**Port 7 Delta high byte**			0x00
		7-0	RW		
0x064	DELTA8	**Port 8 Delta low byte**			0xC8
		7-0	RW		
0x065	DELTA8	**Port 8 Delta high byte**			0x00
		7-0	RW		
0x07E~ 0x07F		**Reserved**			0
			RO	Reserved	

4.2.2 Scan configuration

Address	Name	Content			Default
		Bit	Type	Description	
0x080	TRNDCYC	**I2C configuration**			0x00
		7-5	RSV	Reserved	
		4	RW	Packet Error Code (PEC) enable	
		3-0	RW	Turn around cycle	
0x081	SMBADDR	**I2C slave bit address selection**			0x81
		7	RO	Retrieve '1'	
		6	RO	Retrieve ' 0'	
		5	RO	Select 7-bit I2C address, it's controlled by GPIOB0 external pull-up or pull-low resistor. 0: 0xA0 (1010_000x) 1: 0x80 (1000_000x)	
		4-1	RO	Retrieve ' 0'	
		0	RO	Read back I2C command 0: Write command 1: Read command	
0x082	PU	**IO internal pull-up resistor 4.7kΩ/47kΩ enable**			0x1F
		7-5	RSV	Reserved	
		4-0	RW	[4]: GPIOB1**(47kΩ)** [3]: SCLK**(4.7kΩ)** [2]: SDAT**(4.7kΩ)** [1]: GPIOB0**(47kΩ)** [0]: INT#**(4.7kΩ)** 0: Disable 1: Enable	
0x083	S0TIMER	**S0 Timing control**			0x01/ 0x02
		7-0	RW	00000000: 1 system clock 00000001: 2 system clock ---------- 11111111: 256system clock	
0x084	S1TIMER	**S1 Timing control**			0x01/ 0x02
		7-0	RW	00000000: 1 system clock 00000001: 2 system clock ---------- 11111111: 256 system clock	
0x085	IDLETIMER	**Scan idle time**			0x00
		7-0	RW	00000000: 0 system clock 00000001: 1 system clock ---------- 11111111: 255 system clock	
0x086	DEBNVREF	**De-bounce time for Vref comparison**			0x31
		7-4	RW	De-bounce time once beyond Vref 0000: Disabled 0001: 1 system clock ---------- 1111: 255 system clock	

		3-0	RW	De-bounce time once below Vref 0000: Disabled 0001: 1 system clock ---------- 1111: 255 system clock	
0x087	SCANWIN	**Scan window**			0x60/ 0xA0
		7-0	RW	Scan window time 00000000: 1x512 system clock 00000001: 2x512 system clock ---------- 11111111: 256x512 system clock	
0x088	PREDISTIM	**Pre-discharge timing control**			0xFF
		7-0	RW	Pre-discharge timing for CEXT 00000000: 1 system clock 00000001: 2 system clock ---------- 11111111: 256 system clock	
0x089	SCANSET	**Scan configuration**			0x1A
		7-4	RW	TK de-bounce time 0000: 1 scan cycle 0001: 2 scan cycle ---------- 1110: 15 scan cycle 1111: Reserved	
		3-2	RW	Trigger mode selection TK trigger has 3 different modes: Touch trigger (finger on TK)，Touch end trigger (finger off TK) and both to trigger. Once the defined trigger mode happens, the pending flag is set, working together with the interrupt enable, then the TK trigger will issue interrupt. 00: Touch trigger 01: Touch end trigger 10: Both trigger 11: Reserved	
		1	RW	Threshold Comparison 0: Disable 1: Enable	
		0	RW	TK pins status during non-scan phase 0: Ground 1: Floating (This setting is valid only when shielding function is disabled.)	
0x08A	BL_ZX1	**Baseline calculation coefficient zx1**			0x00
		RW	RW	Region 1 calculation coefficient	
0x08B	BL_Z0X2	**Baseline calculation coefficient z0x2**			0x7C
		RW	RW	Calculation coefficient is BL_Z0X2/128	
0x08C	BL_Z1X2	**Baseline calculation coefficient z1x2**			0x04
		RW	RW	Calculation coefficient is BL_Z1X2/128	
0x08D	BL_Z0X3	**Baseline calculation coefficient z0x3**			0x78
		RW	RW	Calculation coefficient is BL_Z0X3/128	
0x08E	BL_Z1X3	**Baseline calculation coefficient z1x3**			0x08
		RW	RW	Calculation coefficient is BL_Z1X3/128	
0x08F	BL_Z0X4	**Baseline calculation coefficient z0x4**			0x78
		RW	RW	Calculation coefficient is BL_Z0X4/128	
0x090	BL_Z1X4	**Baseline calculation coefficient z1x4**			0x08
		RW	RW	Calculation coefficient is BL_Z1X4/128	
0x091	BL_ZX5	**Baseline calculation coefficient zx5**			0x20
		7-0	RW	Calculation coefficient is BL_ZX5/128	

0x092	BLPERIOD	**Scan baseline adjusting time**			0x03
		7-0	RW	After this defined scan cycles, using updated data to do the scan baseline adjustment 00000000: 1 scan cycle 00000001: 2 scan cycle ---------- 11111111: 256 scan cycle	
0x093	BLCTRL	**Scan baseline adjustment control**			0x15
		7-5	RSV	Reserved	
		4	RW	Scan baseline adjustment enable 0: Disable 1: Enable	
		3-0	RW	Scan baseline initial counting 0000: 1scan cycle 0001: 2 scan cycle ---------- 1110: 15 scan cycle 1111: Reserved	
0x094 - 0x096		**Reserved**			0x00
		7-0	RSV	Reserved	
0x097	SLPTMR0	**Sleep timer LSB**			0x00
		7-0	RW	Sleep timer LSB SB358x will enter sleep mode once this sleep timer expires. Register 0x098[1:0] are timer MSB. 0000000000: 1 x16ms 0000000001: 2 x16ms ---------- 1111111111: 1024 x16ms	
0x098	SLPTMR1	**Sleep timer MSB** **Timer for periodical scan in low power mode**			0x07
		7-2	RW	The timer for periodical scan in low power mode, please refer to Chapter 3, Part B: Low Power Mode, for detail operations. 000000: 1scan cycle 000001: 2 scan cycle ---------- 111111: 128 scan cycle	
		1-0	RW	Sleep timer MSB(This timer works together with register 0x097)	
0x099	WT2SCAN	**OSC 6MHz stable time**			0x02
		7-0	RW	During low power mode, SB358x enters into sleep mode, switch system clock from OSC6MHz to OSC32KHz, and then stop OSC6MHz to save power. Before periodical scan, SB358x needs time to resume OSC6MHz and switch system clock back to OSC6MHz. This register defines OSC6MHz stable time for this operation. 00000000: 1 x 61us 00000001: 2 x 61us ---------- 11111111: 256 x 61us	
0x09A	BTNINTEN0	**TK [7:0] interrupt enable**			0x00
		7-0	RW	0x09A[7:0] corresponds to pins TK[7:0]. Once initiated, TK will issue signal through INT pins. 0: Disable 1: Enable	
0x09B	BTNINTEN1	**TK 8 interrupt enable**			0x00

		7-1	RSV	Reserved	
		0	RW	TK 8 interrupt enable 0: Disable 1: Enable	
0x09C		**Reserved**			0x00
		7-0	RSV	Reserved	
0x09D	BTNSCANEN0	**TK [7:0] scan enable**			0xFC/ 0x00
		7-0	RW	0x09D[7:0] corresponds to TK pins TK[7:0]. Once initiated, GPIO functions of pins MP[7:0] will disable automatically. Note: this register will only be valid after 0x0A0[0] is initiated. 0: Disable 1: Enable	
0x09E	BTNSCANEN1	**TK 8 scan enable**			0x00
		7-6	RSV	Reserved	
		0	RW	Control for TK8 scan enable Note: this register will only be valid after 0x0A0[0] is initiated. 0: Disable 1: Enable	
0x09F		**Reserved**			0x00
			RSV	Reserved	
0x0A0	SCANEN	**Scan enable/low power mode enable**			0x09/ 0x00
		7-5	RSV	Reserved	
		4	RW	Stop charging CEXT during discharge phase 0: Non- stop 1:Stop	
		3	RW	Low power mode enable 0: Disable 1: Enable	
		2	RW	Single TK initiation limitation If multiple TKs are design on one small system, TKs interfere one another rather severely. User may also touch the adjacent TK. This register is to limit the initiation of the first key been touched. 0: Allow multiple TKs to be initiated 1: Only allow first key touched be initiated	
		1	RW	Multiple TK initiation limitation Under certain water proof design or high noise environment, multiple TK initiation may represent mal-action. This register is to defeat noise. This register is only valid when SCANEN[2]=’0’. 0: Allow multiple TK initiation. When multiple TKs are initiated, issue signal on the corresponding pins. 1: Forbid multiple TK initiation. When multiple TKs are initiated, define as system faulty action and do not issue any signal.	
		0	RW	Scan enable This register controls whether SB will execute TK scan function. Other scan registers will only be valid when this register is initiated. 0: Disable 1: Enable	
0x0A1	BUTSTA0	**TK[7:0] scan status**			0x00

		7-0	RO	SOC will judge whether any TK has been initiated through this register. 0x0A1[7:0] correspond to TK pins TK[7:0]. 0: TK is not initiated 1: TK is initiated	
0x0A2	BUTSTA1	**TK8 scan status**			0x00
		7-1	RSV	Reserved	
		0	RO	SOC will judge whether TK8 has been initiated 0: TK is not initiated 1: TK is initiated	
0x0A3		**Reserved**			0x00
			RSV	Reserved	
0x0A4	BUTPF0	**TK[7:0] Pending Flags**			0x00
		7-0	WCE	0x0A4[7:0] corresponds to TK[7:0] Once TK is triggered, SB358x will issue INT to inform SOC, meanwhile, the triggered TK corresponding pending flag will be set. Whenever SOC completes the TK event process, it's required to clear the enabled pending flag.	
0x0A5	BUTPF1	**TK8** Pending Flag			0x00
		7-1	RSV	Reserved	
		0	WCE	TK8 pending flag	
0x0A6		**Reserved**			0x00
			RSV	Reserved	
0x0A7	**SLPWAKE**	**Hibernation and wake up control**			0x50
		7	RW	Wakeup timer control 0: In sleeping phase of power down mode, when wakeup timer expires, SB358x will resume back to normal operation mode to scan if TK event happens. 1: Disable wakeup timer wakeup control.	
		6	RW	Enter low power mode automatically when there is no TK initiation 0: Disable 1: Enable	
		5	RW	Reserved	
		4	RW	Under low power mode, SOC to wake up SB through INT pin. 0: Disable 1: Enable	
		3	RW	Reserved	
		2	RW	Under low power mode, GPIO outputs high-Z signal 0: Disable 1: Enable	
		1	RO	TK IRQ status IRQ status represents any of 0x0A4 and 0x0A5 pending flags is set 0: No interrupt happens 1: Interrupt happens	
		0	RW	Interrupt enable When this register is initiated, once pending flag (0x0A4 and 0x0A5) is set, interrupt (INT) happens. 0: Disable 1: Enable	
0x0A8	WAKETMR	**Wakeup timer**			0x10

		7-0	RW	Wakeup timer When low power mode enter hibernation, wakeup timer will control each hibernation cycle time. This timer works together with register 0x0A7[7]. 00000000: 1 x16ms 00000001: 2 x16ms ---------- 11111111: 256 x16ms	
0x0A9 - 0x0AA		**Reserved**			
			RW	Reserved	
0x0AB	CH_PHASE 2	**LDO/OSC32K standby control**			0x20 /0x00
		7-6	RW	Reserved	
		5	RW	LDO standby enable Disable LDO during sleeping mode 0: Disable 1: Enable	
		4-0	RW	Reserved	
0x0AC - 0x0CA		**Reserved**			
			RSV	Reserved	
0x0CB	REXTCON	**Internal REXT control**			0x09
		7-4	RSV	Reserved	
		3-2	RW	Internal REXT configuration 00: 16kΩ 01: 20kΩ 10: 24kΩ 11: 28kΩ	
		1-0	RW	Reserved	
0x0CC	SLDEN0	**TK[7:0] Shielding function enable**			0x00
		7-0	RW	0x0CC[7:0] correspond to TK[7:0] shielding enable control 0: Disable 1: Enable	
0x0CD	SLDEN1	**TK8 Shielding function enable**			0x00
		7-1	RSV	Reserved	
		0	RW	TK8 shielding enable 0: Disable 1: Enable	
0x0CE	SLDCTR	**Shielding control**			0x00
		7-6	RSV	Reserved	
		5	RW	Non-scan TK will NOT output shielding signal during discharge (S2) phase. 0: Disable 1: Enable	
		4	RW	Non-scan TK will NOT output shielding signal during charge (S0) phase. 0: Disable 1: Enable	
		3	RW	Reserved	
		2	RW	Once the TK pin is enabled to be shielding pin, TK outputs shielding signal type selection during charge transfer (S1) phase. 0: Ground 1: Floating	

		1	RW	When TK shielding function is enabled during non-scan cycle, if TK output signal type is selected to be charge signal during Charge phase (S0), this register defines the signal is from VCC or LDO25 power source. This setting should work together with bit0. 0: VCC 1: LDO25	
		0	RW	When TK shielding function is enabled during non-scan cycle, this register defines TK output signal type during Charge phase (S0), this output high is selected to be from VCC or LDO25 power source. 0: Charge signal 1: Ground	
0x0CF	INTEN	**INT pin function selection**			0x01 /0x00
		7-1	RSV	Reserved	
		0	RW	INT pin function selection 0: INT 1: GPIO	
0x0D0 - 0x0FB		**Reserved**			
			RSV	Reserved	
0x0FC	PLID	**Product ID**			‘S’
		7-0	RO	“S”: sensor button controller	
0x0FD	CTID	**Type ID**			0x35
		7-0	RO	Hex number “35”	
0x0FE	CPID	**Package ID**			0x85
		7-0	RO	Hex number “85”	
0x0FF	HWREVID	**Revision ID**			0x00
		7-0	RO	00: A0 01: A1 02: A2 ….. (VIA1 option)	

4.3 GPIO

Address	Name	Content			Default
		Bit	Type	Description	
0x100	GPIOMODE0	**GPIO mode selection 0**			0x03 /0x00
		7-0	RW	GPIO A7~A0 mode selection 0: GPIO mode 1: PWM mode	
0x101	GPIOMODE1	**GPIO mode selection 1**			0x0F /0x00
		7-4	RSV	Reserved	
		3-1	RW	GPIO B2~B0 mode selection 0: GPIO mode 1: PWM mode	
		0	RW	GPIO A8 mode selection 0: GPIO mode 1: PWM mode	
0x102	GPIODI0	**GPIO data input 0**			0x00
		7-0	RO	GPIO A7~A0data input	
0x103	GPIODI1	**GPIO data input 1**			0x00
		7-4	RSV	Reserved	
		3-1	RO	GPIO B2~B0data input	
		0	RO	GPIO A8 data input	
0x104	GPIODO0	**GPIO data output 0**			0x00
		7-0	RW	GPIO A7~A0 data output	
0x105	GPIODO1	**GPIO data output 1**			0x00
		7-4	RSV	Reserved	
		3~1	RW	GPIO B2~B0 data output	
		0	RW	GPIO A8 data output	
0x106	GPIOIE0	**GPIO input enable 0**			0x00
		7-0	RW	GPIO A7~A0 input enable 0: Disable 1: Enable	
0x107	GPIOIE1	**GPIO input enable 1**			0x00
		7-4	RSV	Reserved	
		3-1	RW	GPIO B2~B0 input enable 0: Disable 1: Enable	
		0	RW	GPIO A8 input enable 0: Disable 1: Enable	
0x108	GPIOOE0	**GPIO output enable 0**			0x00
		7-0	RW	GPIO A7~A0 output enable 0: Disable 1: Enable	
0x109	GPIOOE1	**GPIO output enable 1**			0x00
		7-4	RSV	Reserved	
		3-1	RW	GPIO B2~B0 output enable 0: Disable 1: Enable	
		0	RW	GPIO A8 output enable 0: Disable 1: Enable	
0x10A~ 0x10B		**Reserved**			0xFF
			RSV	Reserved	
0x10C	PWMOM0	**PWM output mode 0**			0x00
		7-0	RW	GPIO A7~A0 PWM output selection 0: Push-pull mode 1: Open-drain mode	

0x10D	PWMOM1	**PWM output mode 1**			0x00
		7-4	RSV	Reserved	
		3-1	RW	GPIO B2~B0 PWM output selection 0: Push-pull mode 1: Open-drain mode	
		0	RW	GPIO A8 PWM output selection 0: Push-pull mode 1: Open-drain mode	
0x10E	PWMBCK	**PWM clock selection**			0x00
		7-0	RW	00000000: 6MHz/1 00000001: 6MHz/2 ---------- 11111111: 6MHz/256	
0x10F	PWMMCL	**PWM cycle length**			0x00
		7-0	RW	PWM cycle length 00000000: Reserved 00000001: 1 PWM clock ---------- 11111111: 255 PWM clock	
0x110	GPIO0 SET	**GPIOA0 PWM configuration (please refer to Part A3.6 PWM parameter diagram)**			0x13 /0x23
		7~6	RSV	Reserved	
		5	RW	This register defines PWM signal level in active duty state 0: Low level 1: High level	
		4	RW	This register defines PWM signal level in non-active duty state 0: Low level 1: High level	
		3-0	RW	TK trigger indicator 0000: TK0 triggered 0001: TK1 triggered 0010: TK2 triggered 0011: TK3 triggered 0100: TK4 triggered 0101: TK5 triggered 0110: TK6 triggered 0111: TK7 triggered 1000: TK8 triggered 1001: Reserved ---------- 1100: Reserved 1101: Any TK triggered 1110: No TK triggered 1111: Reserved	
0x111	GPIO1 SET	**GPIOA1 PWM configuration**			0x12 /0x22
		7~6	RSV	Reserved	
		5	RW	This register defines PWM signal level in active duty state 0: Low level 1: High level	
		4	RW	This register defines PWM signal level in non-active duty state 0: Low level 1: High level	

		3-0	RW	TK trigger indicator 0000: TK 0 triggered 0001: TK 1 triggered 0010: TK 2 triggered 0011: TK 3 triggered 0100: TK 4 triggered 0101: TK 5 triggered 0110: TK 6 triggered 0111: TK 7 triggered 1000: TK 8 triggered 1001: Reserved ---------- 1100: Reserved 1101: Any TK is triggered 1110: No TK is triggered 1111: Reserved	
0x112	GPIO2 SET	**GPIOA2 PWM configuration**			0x1F
		7~6	RSV	Reserved	
		5	RW	This register defines PWM signal level in active duty state 0: Low level 1: High level	
		4	RW	This register defines PWM signal level in non-active duty state 0: Low level 1: High level	
		3-0	RW	TK trigger indicator 0000: TK 0 triggered 0001: TK 1 triggered 0010: TK 2 triggered 0011: TK 3 triggered 0100: TK 4 triggered 0101: TK 5 triggered 0110: TK 6 triggered 0111: TK 7 triggered 1000: TK 8 triggered 1001: Reserved ---------- 1100: Reserved 1101: Any TK is triggered 1110: No TK is triggered 1111: Reserved	
0x113	GPIO3 SET	**GPIOA3 PWM configuration**			0x1F
		7~6	RSV	Reserved	
		5	RW	This register defines PWM signal level in active duty state 0: Low level 1: High level	
		4	RW	This register defines PWM signal level in non-active duty state 0: Low level 1: High level	

		3-0	RW	TK trigger indicator 0000: TK 0 triggered 0001: TK 1 triggered 0010: TK 2 triggered 0011: TK 3 triggered 0100: TK 4 triggered 0101: TK 5 triggered 0110: TK 6 triggered 0111: TK 7 triggered 1000: TK 8 triggered 1001: Reserved ---------- 1100: Reserved 1101: Any TK is triggered 1110: No TK is triggered 1111: Reserved	
0x114	GPIO4 SET	**GPIOA4 PWM configuration**			0x1F
		7~6	RSV	Reserved	
		5	RW	This register defines PWM signal level in active duty state 0: Low level 1: High level	
		4	RW	This register defines PWM signal level in non-active duty state 0: Low level 1: High level	
		3-0	RW	TK trigger indicator 0000: TK 0 triggered 0001: TK 1 triggered 0010: TK 2 triggered 0011: TK 3 triggered 0100: TK 4 triggered 0101: TK 5 triggered 0110: TK 6 triggered 0111: TK 7 triggered 1000: TK 8 triggered 1001: Reserved ---------- 1100: Reserved 1101: Any TK is triggered 1110: No TK is triggered 1111: Reserved	
0x115	GPIO5 SET	**GPIOA5 PWM configuration**			0x1F
		7~6	RSV	Reserved	
		5	RW	This register defines PWM signal level in active duty state 0: Low level 1: High level	
		4	RW	This register defines PWM signal level in non-active duty state 0: Low level 1: High level	

		3-0	RW	TK trigger indicator 0000: TK 0 triggered 0001: TK 1 triggered 0010: TK 2 triggered 0011: TK 3 triggered 0100: TK 4 triggered 0101: TK 5 triggered 0110: TK 6 triggered 0111: TK 7 triggered 1000: TK 8 triggered 1001: Reserved ---------- 1100: Reserved 1101: Any TK is triggered 1110: No TK is triggered 1111: Reserved	
0x116	GPIO6 SET	**GPIOA6 PWM configuration**			0x1F
		7~6	RSV	Reserved	
		5	RW	This register defines PWM signal level in active duty state 0: Low level 1: High level	
		4	RW	This register defines PWM signal level in non-active duty state 0: Low level 1: High level	
		3-0	RW	TK trigger indicator 0000: TK 0 triggered 0001: TK 1 triggered 0010: TK 2 triggered 0011: TK 3 triggered 0100: TK 4 triggered 0101: TK 5 triggered 0110: TK 6 triggered 0111: TK 7 triggered 1000: TK 8 triggered 1001: Reserved ---------- 1100: Reserved 1101: Any TK is triggered 1110: No TK is triggered 1111: Reserved	
0x117	GPIO7 SET	**GPIOA7 PWM configuration**			0x1F
		7~6	RSV	Reserved	
		5	RW	This register defines PWM signal level in active duty state 0: Low level 1: High level	
		4	RW	This register defines PWM signal level in non-active duty state 0: Low level 1: High level	

		3-0	RW	TK trigger indicator 0000: TK 0 triggered 0001: TK 1 triggered 0010: TK 2 triggered 0011: TK 3 triggered 0100: TK 4 triggered 0101: TK 5 triggered 0110: TK 6 triggered 0111: TK 7 triggered 1000: TK 8 triggered 1001: Reserved ---------- 1100: Reserved 1101: Any TK is triggered 1110: No TK is triggered 1111: Reserved	
0x118	GPIO8 SET	**GPIOA8 PWM configuration**			0x14 /0x24
		7~6	RSV	Reserved	
		5	RW	This register defines PWM signal level in active duty state 0: Low level 1: High level	
		4	RW	This register defines PWM signal level in non-active duty state 0: Low level 1: High level	
		3-0	RW	TK trigger indicator 0000: TK 0 triggered 0001: TK 1 triggered 0010: TK 2 triggered 0011: TK 3 triggered 0100: TK 4 triggered 0101: TK 5 triggered 0110: TK 6 triggered 0111: TK 7 triggered 1000: TK 8 triggered 1001: Reserved ---------- 1100: Reserved 1101: Any TK is triggered 1110: No TK is triggered 1111: Reserved	
0x119	GPIOB0 SET	**GPIOB0 PWM configuration**			0x16 /0x26
		7~6	RSV	Reserved	
		5	RW	This register defines PWM signal level in active duty state 0: Low level 1: High level	
		4	RW	This register defines PWM signal level in non-active duty state 0: Low level 1: High level	

		3-0	RW	TK trigger indicator 0000: TK 0 triggered 0001: TK 1 triggered 0010: TK 2 triggered 0011: TK 3 triggered 0100: TK 4 triggered 0101: TK 5 triggered 0110: TK 6 triggered 0111: TK 7 triggered 1000: TK 8 triggered 1001: Reserved ---------- 1100: Reserved 1101: Any TK is triggered 1110: No TK is triggered 1111: Reserved	
0x11A	GPIOB1 SET	**GPIOB1 PWM configuration**			0x17 /0x27
		7~6	RSV	Reserved	
		5	RW	This register defines PWM signal level in active duty state 0: Low level 1: High level	
		4	RW	This register defines PWM signal level in non-active duty state 0: Low level 1: High level	
		3-0	RW	TK trigger indicator 0000: TK 0 triggered 0001: TK 1 triggered 0010: TK 2 triggered 0011: TK 3 triggered 0100: TK 4 triggered 0101: TK 5 triggered 0110: TK 6 triggered 0111: TK 7 triggered 1000: TK 8 triggered 1001: Reserved ---------- 1100: Reserved 1101: Any TK is triggered 1110: No TK is triggered 1111: Reserved	
0x11B	GPIOB2 SET	**GPIOB2 PWM configuration**			0x15 /0x25
		7~6	RSV	Reserved	
		5	RW	This register defines PWM signal level in active duty state 0: Low level 1: High level	
		4	RW	This register defines PWM signal level in non-active duty state 0: Low level 1: High level	

		3-0	RW	TK trigger indicator 0000: TK 0 triggered 0001: TK 1 triggered 0010: TK 2 triggered 0011: TK 3 triggered 0100: TK 4 triggered 0101: TK 5 triggered 0110: TK 6 triggered 0111: TK 7 triggered 1000: TK 8 triggered 1001: Reserved ---------- 1100: Reserved 1101: Any TK is triggered 1110: No TK is triggered 1111: Reserved	
0x11C~ 0x17F		**Reserved**			0xFF
			RSV	Reserved	
0x180	GPIO0 HILEN	**GPIO A0 PWM (please refer to Part A3.6 PWM parameter diagram)**			0x00
		7-0	RW	PWM active pulse length (Duty cycle = PWM active pulse length / PWM cycle length) 00000000: Duty cycle = 0 / PWM cycle length 00000001: Duty cycle = 1 / PWM cycle length ---------- 11111111: Duty cycle = 255 / PWM cycle length	
0x181	GPIO1 HILEN	**GPIO A1 PWM active pulse length**			0x00
		7-0	RW	PWM active pulse length (Duty cycle = PWM active pulse length / PWM cycle length) 00000000: Duty cycle = 0 / PWM cycle length 00000001: Duty cycle = 1 / PWM cycle length ---------- 11111111: Duty cycle = 255 / PWM cycle length	
0x182	GPIO2 HILEN	**GPIO A2 PWM active pulse length**			0x00
		7-0	RW	PWM active pulse length (Duty cycle = PWM active pulse length / PWM cycle length) 00000000: Duty cycle = 0 / PWM cycle length 00000001: Duty cycle = 1 / PWM cycle length ---------- 11111111: Duty cycle = 255 / PWM cycle length	
0x183	GPIO3 HILEN	**GPIO A3 PWM active pulse length**			0x00
		7-0	RW	PWM active pulse length (Duty cycle = PWM active pulse length / PWM cycle length) 00000000: Duty cycle = 0 / PWM cycle length 00000001: Duty cycle = 1 / PWM cycle length ---------- 11111111: Duty cycle = 255 / PWM cycle length	
0x184	GPIO4 HILEN	**GPIO A4 PWM active pulse length**			0x00
		7-0	RW	PWM active pulse length (Duty cycle = PWM active pulse length / PWM cycle length) 00000000: Duty cycle = 0 / PWM cycle length 00000001: Duty cycle = 1 / PWM cycle length ---------- 11111111: Duty cycle = 255 / PWM cycle length	

0x185	GPIO5 HILEN	**GPIO A5 PWM active pulse length**			0x00
		7-0	RW	PWM active pulse length (Duty cycle = PWM active pulse length / PWM cycle length) 00000000: Duty cycle = 0 / PWM cycle length 00000001: Duty cycle = 1 / PWM cycle length ---------- 11111111: Duty cycle = 255 / PWM cycle length	
0x186	GPIO6 HILEN	**GPIO A6 PWM active pulse length**			0x00
		7-0	RW	PWM active pulse length (Duty cycle = PWM active pulse length / PWM cycle length) 00000000: Duty cycle = 0 / PWM cycle length 00000001: Duty cycle = 1 / PWM cycle length ---------- 11111111: Duty cycle = 255 / PWM cycle length	
0x187	GPIO7 HILEN	**GPIO A7 PWM active pulse length**			0x00
		7-0	RW	PWM active pulse length (Duty cycle = PWM active pulse length / PWM cycle length) 00000000: Duty cycle = 0 / PWM cycle length 00000001: Duty cycle = 1 / PWM cycle length ---------- 11111111: Duty cycle = 255 / PWM cycle length	
0x188	GPIO7 HILEN	**GPIO A8 PWM active pulse length**			0x00
		7-0	RW	PWM active pulse length (Duty cycle = PWM active pulse length / PWM cycle length) 00000000: Duty cycle = 0 / PWM cycle length 00000001: Duty cycle = 1 / PWM cycle length ---------- 11111111: Duty cycle = 255 / PWM cycle length	
0x189	GPIO7 HILEN	**GPIO B0 PWM active pulse length**			0x00
		7-0	RW	PWM active pulse length (Duty cycle = PWM active pulse length / PWM cycle length) 00000000: Duty cycle = 0 / PWM cycle length 00000001: Duty cycle = 1 / PWM cycle length ---------- 11111111: Duty cycle = 255 / PWM cycle length	
0x18A	GPIO7 HILEN	**GPIO B1 PWM active pulse length**			0x00
		7-0	RW	PWM active pulse length (Duty cycle = PWM active pulse length / PWM cycle length) 00000000: Duty cycle = 0 / PWM cycle length 00000001: Duty cycle = 1 / PWM cycle length ---------- 11111111: Duty cycle = 255 / PWM cycle length	
0x18B	GPIO7 HILEN	**GPIO B2 PWM active pulse length**			0x00
		7-0	RW	PWM active pulse length (Duty cycle = PWM active pulse length / PWM cycle length) 00000000: Duty cycle = 0 / PWM cycle length 00000001: Duty cycle = 1 / PWM cycle length ---------- 11111111: Duty cycle = 255 / PWM cycle length	
0x18C - 0x19F		**Reserved**			0xFF
			RSV	Reserved	
0x1A0	GPIOMISC	**GPIO miscellaneous register**			0x00
		7-2	RSV	Reserved	
		1	RW	Reserved	
		0	RW	PWM clock status when PWM mode is turned off: 0: Continue operation 1: Stop operation	

0x1A1 - 0x1FF		**Reserved**			0xFF
			RSV	Reserved	

5. Electronic Characteristics

5.1 Absolute Maximum Rating

Symbol	Parameter	Condition	Rating	Unit
V_{CC} (Note)	Power Source Voltage	All the values refer to GND.	-0.3 ~ 3.6 -0.5 ~ 5.5	V
V_i (Note)	Input Voltage		-0.3 ~ 3.6 -0.5 ~ 5.5	V
V_o (Note)	Output Voltage		-0.3 ~ 3.6 -0.5 ~ 5.5	V
T_{STG}	Storage Temperature		-65 ~ 150	℃
	ESD Tolerance (IEC/EN61000-4-2)	HBM	8K	V
		EFT	4K	V

(Note) The voltage should be kept steady. Sudden voltage variation should be limited. The exact operating rating depends on 3.0V～5.5V power source

5.2 DC Characteristics

Parameter	Symbol	Min	Typ.	Max	Unit	Condition
Input Low Threshold (3.3V)	V_{t-}	1.14	1.28		V	
Input High Threshold (3.3V)	V_{t+}		1.8	1.95	V	
Input Low Threshold (5V)	V_{t-}	1.86	2.24		V	
Input High Threshold (5V)	V_{t+}		2.87	3.08	V	
Output Low Voltage (3.3V)	V_{OL}		0.4		V	
Output High Voltage (3.3V)	V_{OH}		2.8		V	
Output Low Voltage (5V)	V_{OL}		0.7		V	
Output High Voltage (5V)	V_{OH}		4.5		V	
Input Leakage Current	I_{IL}			1	µA	No pull-up

5.3 Recommend Operation Condition

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
Vcc (Note)	Power Source Voltage	3.0	3.3	3.6	V
Vcc (Note)	Power Source Voltage	4.5	5.0	5.5	V
GND	Ground Voltage	-0.3	0	0.3	V
T_{op}	Operating Temperature	-20		85	℃

(Note) The voltage should be kept steady. Sudden voltage variation should be limited. The exact operating rating depends on 3.0V～5.5V power source

5.4 Operating Current

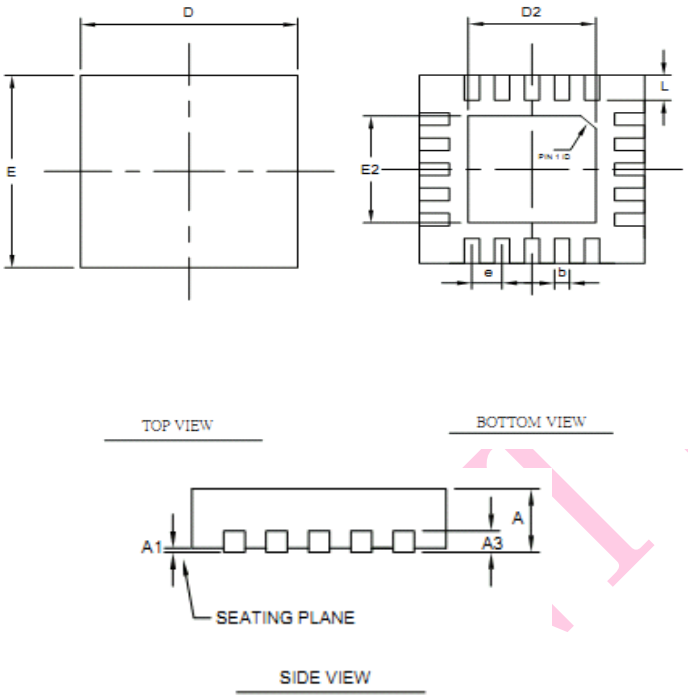
Symbol	Parameter	Limits	Unit
		Typ.	
Icc	Static power consumption in operation state with OSC 6MHz and 3.3V.	0.7	mA
Icc	Static power consumption in operation state with OSC 6MHz and 5V.	0.8	mA
Icc	Power consumption in sleeping mode with OSC 32KHz and 3.3V.	110	μA
Icc	Power consumption in sleeping mode with OSC 32KHz and 5V.	130	μA

6. Packaging Information

6.1Dimensions

6.1.1 QFN

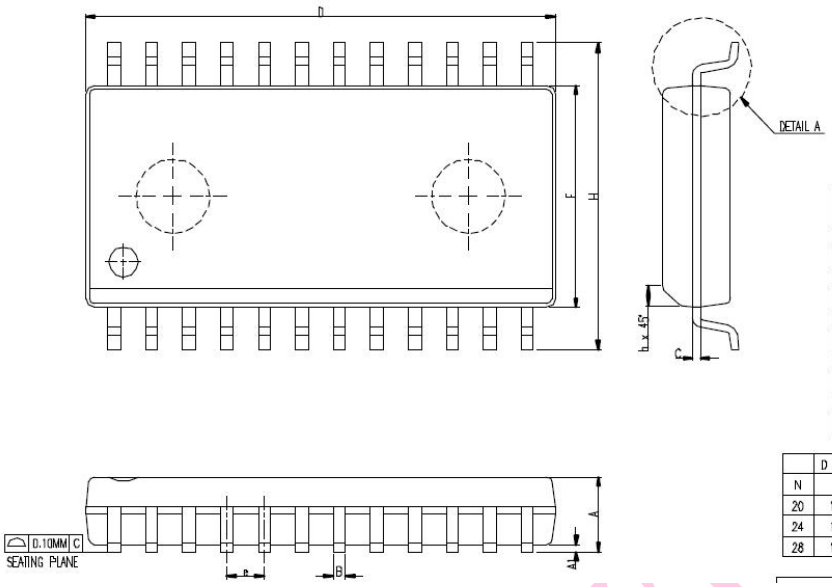
20-QFN



Symbol	Dimensions in Millimeters		
	Min.	Normal	Max.
A	0.70	0.75	0.80
A1	0.00	0.02	0.05
A3	0.203 REF		
b	0.15	0.20	0.25
D	2.90	3.00	3.10
D2	1.60	1.70	1.80
E	2.90	3.00	3.10
E2	1.60	1.70	1.80
e	0.40 BSC		
L	0.30	0.40	0.50

6.1.2 SOP

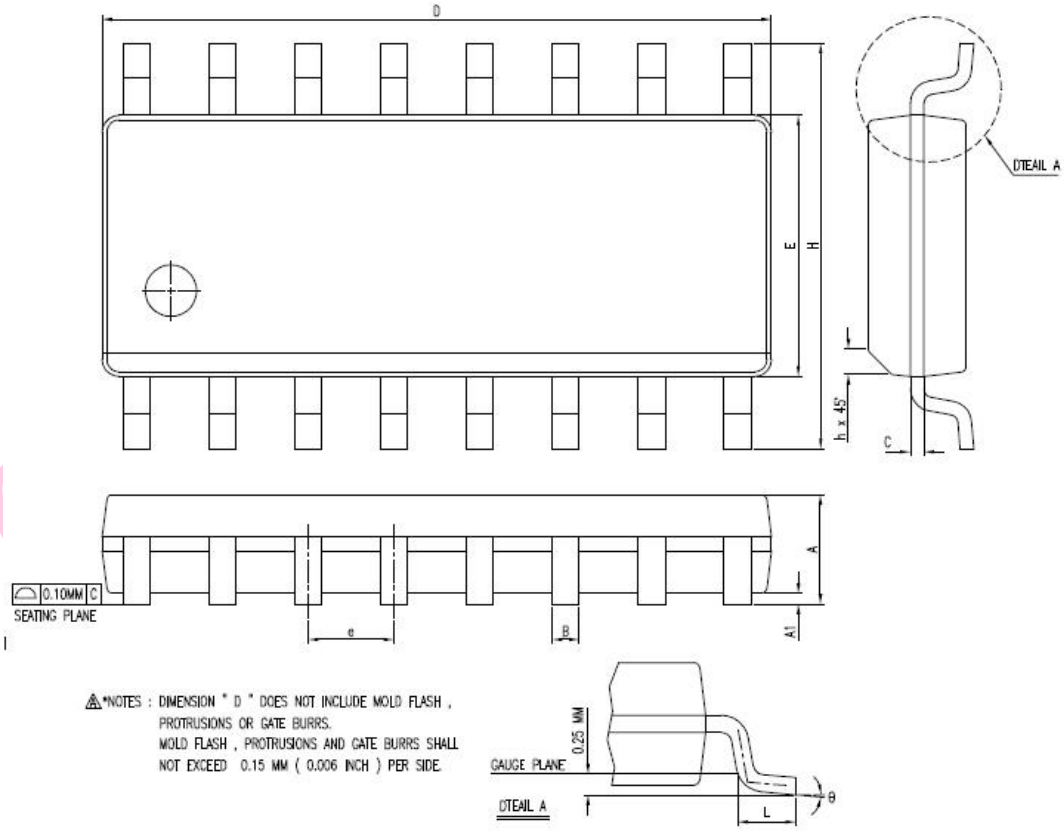
20-SOP



SYMBOL	DIMENSION IN MM		DIMENSION IN INCH	
	MIN.	MAX.	MIN.	MAX.
A	2.35	2.65	0.0926	0.1043
A1	0.10	0.30	0.0040	0.0118
B	0.33	0.51	0.013	0.020
C	0.23	0.32	0.0091	0.0125
e	1.27 BSC		0.050 BSC	
E	7.40	7.60	0.2914	0.2992
H	10.00	10.65	0.394	0.419
L	0.40	1.27	0.016	0.050
h	0.25	0.75	0.010	0.029
θ	0°	8°	0°	8°

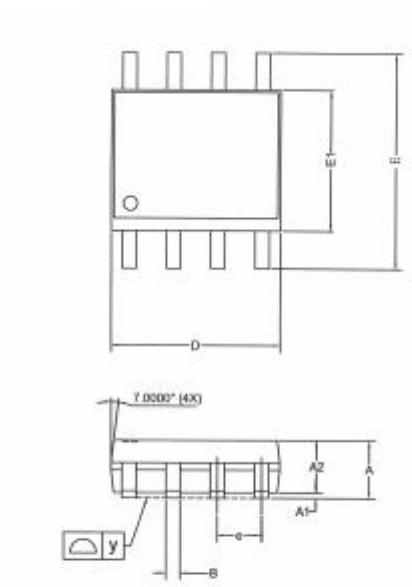
N	D DIMENSION (IN MM) MIN.	D DIMENSION (IN MM) MAX.	D DIMENSION (IN INCH) MIN.	D DIMENSION (IN INCH) MAX.	JEDEC
20	12.60	13.00	0.4961	0.5118	MS-013 (AC)
24	15.20	15.60	0.5985	0.6141	MS-013 (AD)
28	17.70	18.10	0.6969	0.7125	MS-013 (AE)

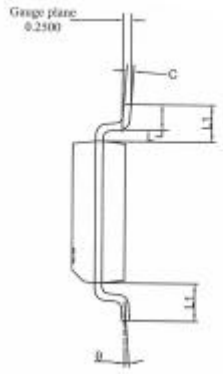
16-NSOP



SYMBOL	DIMENSION IN MM		DIMENSION IN INCH	
	MIN.	MAX.	MIN.	MAX.
A	1.35	1.75	0.0532	0.0688
A1	0.10	0.25	0.0040	0.0098
B	0.33	0.51	0.013	0.020
C	0.19	0.25	0.0075	0.0098
e	1.27 BSC		0.050 BSC	
D	9.80	10.00	0.3859	0.3937
H	5.80	6.20	0.2284	0.2440
E	3.80	4.00	0.1497	0.1574
L	0.40	1.27	0.016	0.050
h	0.25	0.50	0.0099	0.0196
θ	0°	8°	0°	8°
JEDEC	MS-012 (AC)			

8-NSOP





SYMBOLS	DIMENSIONS IN MILLIMETER			DIMENSIONS IN INCH		
	MIN	NOM	MAX	MIN	NOM	MAX
A	1.35	1.60	1.75	0.053	0.063	0.069
A1	0.10	0.15	0.25	0.004	0.006	0.010
A2	1.25	——	——	0.049	——	——
B	0.33	——	0.51	0.013	——	0.020
C	0.17	——	0.25	0.007	——	0.010
D	4.70	4.90	5.10	0.185	0.193	0.200
E1	3.80	3.90	4.00	0.150	0.154	0.157
e	——	1.27BCS	——	——	0.050BCS	——
E	5.80	6.00	6.20	0.228	0.236	0.244
L	0.40	——	1.27	0.016	——	0.050
y	——	——	0.10	——	——	0.004
θ	0°	——	8°	0°	——	8°
L1-L1'	——	——	0.12	——	——	0.005
L1	1.04REF			0.041REF		

NOTE
1. PACKAGE BODY SIZES EXCLUDE MOLD FLASH AND GATE BURRS
2. DIMENSION L IS MEASURED IN GAGE PLANE
3. TOLERANCE 0.10 mm UNLESS OTHERWISE SPECIFIED
4. CONTROLLING DIMENSION IS MILLIMETER. CONVERTED INCH DIMENSIONS ARE NOT NECESSARILY EXACT.
5. FOLLOWED FROM JEDEC MS-012

6.2 Manual Soldering Temperature Limitation

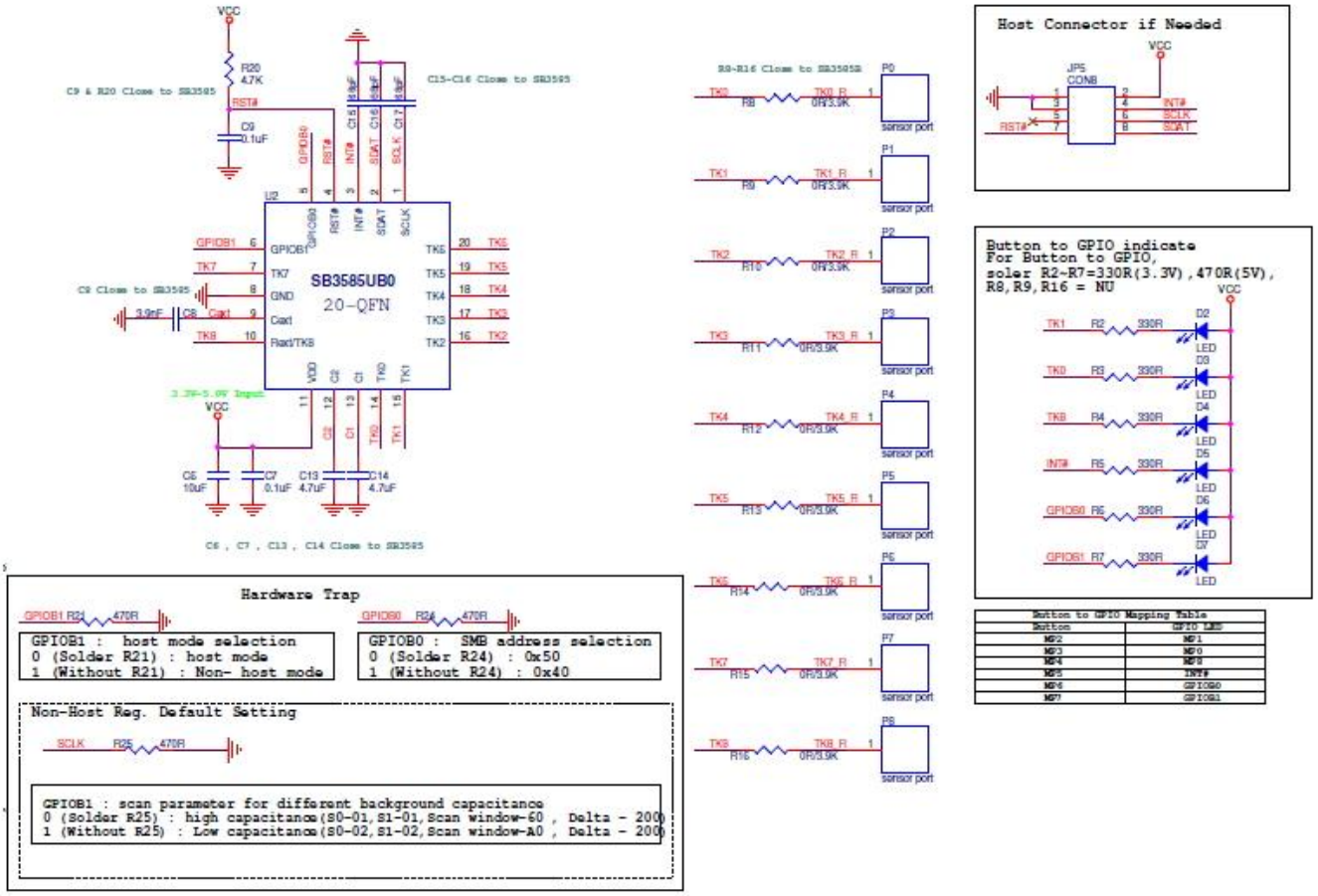
Part Number	Typical Soldering Temperature and duration
SB3585UB0 (QFN)	285℃, 3~5sec
SB3585WB0 (SOP)	350℃, 3~5sec
SB3586WB0 (NSOP)	
SB3584WB0 (NSOP)	
SB3581WB0 (NSOP)	

6.3 Part Numbers Descriptions

Part Number	Package Size	Lead Free Process	Status
SB3585UB0	3mm x 3mm x 0.75mm QFN-20	Lead Free	MP
SB3585WB0	SOP-20 300mil	Lead Free	MP
SB3586WB0	NSOP-16 150mil		
SB3584WB0	NSOP-16 150mil		
SB3581WB0	NSOP-8 150mil		

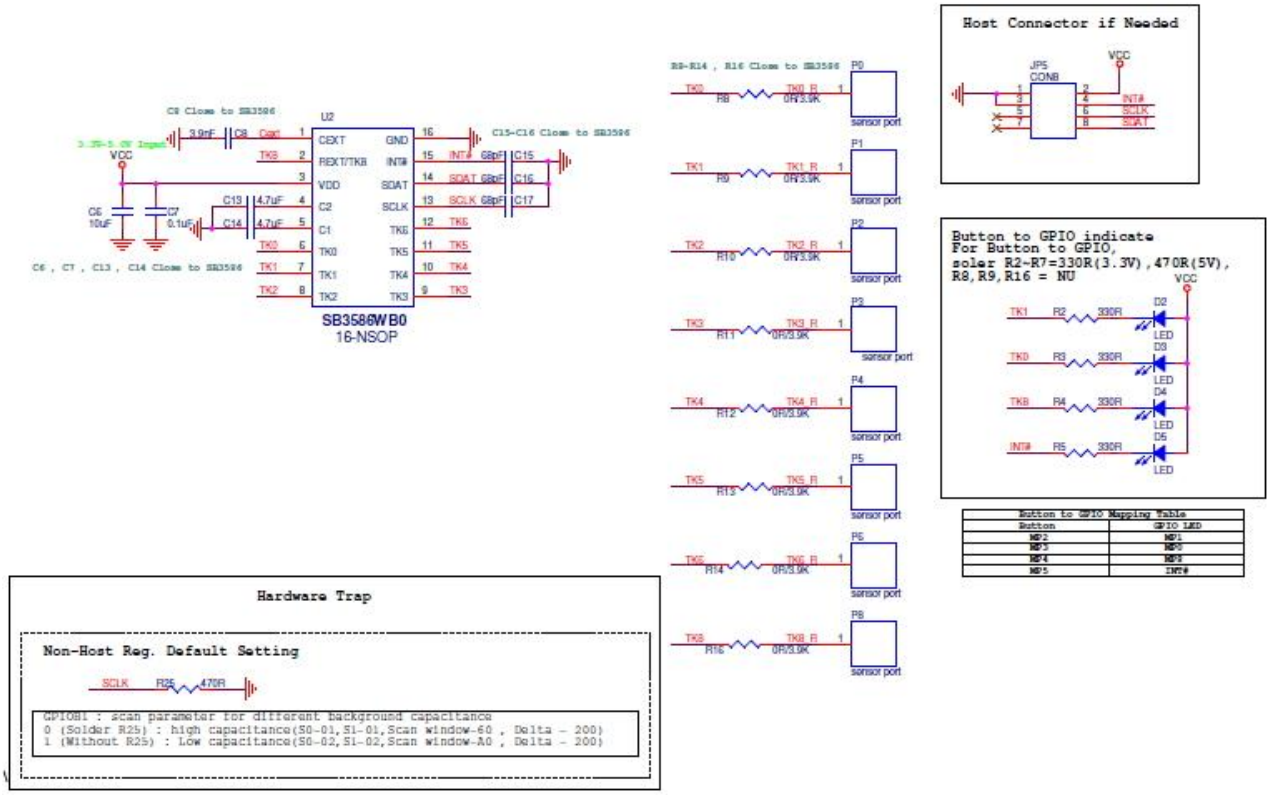
7. Application Circuits

7.1 Reference Design with SB3585UB0



Button to GPIO Mapping Table	
Button	GPIO LED
MP2	MP1
MP3	MP0
MP4	MP8
MP5	INT#
MP6	GPIOB0
MP7	GPIOB1

7.2 Reference design with SB3586WB0



Button to GPIO Mapping Table	
Button	GPIO LED
MP2	MP1
MP3	MP0
MP4	MP8
MP5	INT#

