

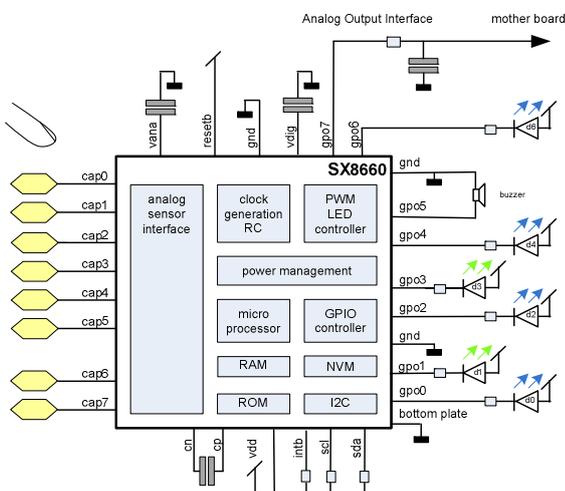
GENERAL DESCRIPTION

The SX8660 is an ultra low power, fully integrated 8-channel solution for capacitive touch-button applications. Unlike many capacitive touch solutions, the SX8660 features dedicated capacitive sense inputs (that requires no external components) in addition to 8 general purpose I/O ports (GPIO). Each of the 8 on-chip GPIO/LED driver is equipped with independent PWM source for enhanced visual effect such as dimming, blinking and breathing.

The SX8660 includes a capacitive 10 bit ADC analog interface with automatic compensation up to 100pF. The high resolution capacitive sensing supports a wide variety of touch pad sizes and shapes and allows capacitive buttons to be created using thick overlay materials (up to 5mm) for an extremely robust and ESD immune system design.

The SX8660 incorporates a versatile firmware that was specially designed to simplify capacitive touch solution design and offers reduced time-to-market. Integrated multi-time programmable memory provides the ultimate flexibility to modify key firmware parameters (gain, threshold, scan period, auto offset compensation) in the field without the need for new firmware development.

The SX8660 supports the 400 kHz I²C serial bus data protocol and includes a field programmable slave address. The tiny 4mm x 4mm footprint makes it an ideal solution for portable, battery powered applications where power and density are at a premium.

TYPICAL APPLICATION CIRCUIT

KEY PRODUCT FEATURES

- ◆ Complete 8 Sensors Capacitive Touch-Button Solution
 - Up to 8 LED Drivers for individual Visual Feedback with Auto Lightening
 - Configurable Single, Repeat, Continuous Fading Mode
 - 256 steps PWM Linear and Logarithmic control
- ◆ High Resolution Capacitive Sensing
 - Up to 100pF of Offset Cap. Compensation at Full Sensitivity
 - Capable of Sensing thru Overlay Materials(<5mm thick)
- ◆ Up to 2 Analog Output Interfaces (AOI-A and AOI-B)
 - Enable button detection thru host's ADC
- ◆ Support of buzzer for audible feedback
- ◆ User-selectable Button Reporting Configuration
 - Report First or Report Strongest
- ◆ Extremely Low Power
 - 8uA (typ) in Sleep Mode
 - 70uA (typ) in Doze Mode (195ms)
 - 200uA (typ) in Active Mode mode (30ms)
- ◆ Programmable Scanning Period from 15ms to several seconds
- ◆ Auto Offset Compensation
 - Eliminates false triggers due to environmental factors (temperature, humidity)
 - Initiated on power-up and configurable intervals
- ◆ Multi-Time In-Field Programmable Firmware Parameters for Ultimate Flexibility
 - On-chip user programmable memory for fast, self contained start-up
- ◆ No External Components per Sensor Input
- ◆ Internal Clock Requires No External Components
- ◆ Differential Sensor Sampling for Reduced EMI
- ◆ Optional 400 KHz I²C Interface w/ Programmable Address
- ◆ -40°C to +85°C Operation

APPLICATIONS

- ◆ LCD TVs, Monitors
- ◆ White Goods
- ◆ Notebook/Netbook/Portable/Handheld computers
- ◆ Consumer Products, Instrumentation, Automotive
- ◆ Mechanical Button Replacement

ORDERING INFORMATION

| Part Number | Temperature Range | Package |
|------------------------------|-------------------|---------------------|
| SX8660I06AULTRT ¹ | -40°C to +85°C | Lead Free MLPQ-UT28 |

¹ 3000 Units/reel

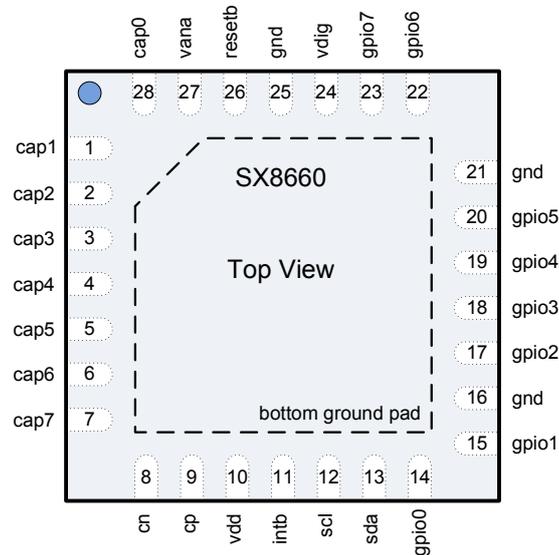
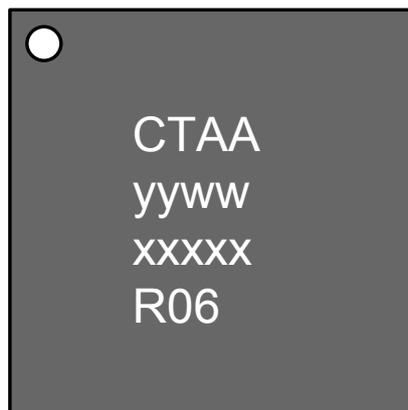
* This device is RoHS/WEEE compliant and Halogen Free

Table of Contents

| | |
|--|-----------|
| GENERAL DESCRIPTION..... | 1 |
| TYPICAL APPLICATION CIRCUIT | 1 |
| KEY PRODUCT FEATURES..... | 1 |
| APPLICATIONS..... | 1 |
| ORDERING INFORMATION..... | 1 |
| 1 GENERAL DESCRIPTION..... | 4 |
| 1.1 Pin Diagram | 4 |
| 1.2 Marking information | 4 |
| 1.3 Pin Description | 5 |
| 1.4 Simplified Block Diagram | 6 |
| 1.5 Acronyms | 6 |
| 2 ELECTRICAL CHARACTERISTICS | 7 |
| 2.1 Absolute Maximum Ratings | 7 |
| 2.2 Recommended Operating Conditions | 7 |
| 2.3 Thermal Characteristics | 7 |
| 2.4 Electrical Specifications | 8 |
| 3 FUNCTIONAL DESCRIPTION..... | 10 |
| 3.1 Quickstart Application | 10 |
| 3.2 Introduction | 11 |
| 3.2.1 General | 11 |
| 3.2.2 GPIOs | 11 |
| 3.2.3 Analog Output Interface A and B (SPO mode) | 11 |
| 3.2.4 Buzzer (SPO mode) | 11 |
| 3.2.5 Parameters | 11 |
| 3.2.6 Configuration | 12 |
| 3.3 Scan Period | 12 |
| 3.4 Operation modes | 12 |
| 3.5 Sensors on the PCB | 13 |
| 3.6 Button Information | 14 |
| 3.7 Buzzer | 14 |
| 3.8 Analog Output Interface | 15 |
| 3.9 Analog Sensing Interface | 17 |
| 3.10 Offset Compensation | 19 |
| 3.11 Processing | 20 |
| 3.12 Configuration | 20 |
| 3.13 Power Management | 22 |
| 3.14 Clock Circuitry | 22 |
| 3.15 I2C interface | 22 |
| 3.16 Interrupt | 23 |
| 3.16.1 Power up | 23 |
| 3.16.2 Assertion | 23 |
| 3.16.3 Clearing | 23 |
| 3.16.4 Example | 24 |
| 3.17 Reset | 24 |

ADVANCED COMMUNICATIONS & SENSING DATASHEET

| | | |
|-------------|--|-----------|
| 3.17.1 | Power up | 24 |
| 3.17.2 | RESETB | 25 |
| 3.17.3 | Software Reset | 25 |
| 3.18 | General Purpose Input and Outputs | 26 |
| 3.18.1 | GPI | 26 |
| 3.18.2 | GPP mode | 27 |
| 3.18.3 | GPO | 28 |
| 3.18.4 | GPO Fading | 29 |
| 3.18.5 | Intensity index vs PWM pulse width | 31 |
| 4 | PIN DESCRIPTIONS | 32 |
| 4.1 | Introduction | 32 |
| 4.2 | ASI pins | 32 |
| 4.3 | Host interface pins | 33 |
| 4.4 | Power management pins | 37 |
| 4.5 | General purpose IO pins | 38 |
| 5 | DETAILED CONFIGURATION DESCRIPTIONS | 39 |
| 5.1 | Introduction | 39 |
| 5.2 | General Parameters | 42 |
| 5.3 | Capacitive Sensors Parameters | 43 |
| 5.4 | Button Parameters | 47 |
| 5.5 | Analog Output Interface Parameters | 52 |
| 5.6 | Buzzer Parameters | 54 |
| 5.7 | Mapping Parameters | 55 |
| 5.8 | GPIO Parameters | 57 |
| 6 | I2C INTERFACE..... | 62 |
| 6.1 | I2C Write | 62 |
| 6.2 | I2C read | 63 |
| 6.3 | I2C Registers Overview | 64 |
| 6.4 | Status Registers | 65 |
| 6.5 | Control Registers | 67 |
| 6.6 | SPM Gateway Registers | 69 |
| 6.6.1 | SPM Write Sequence | 70 |
| 6.6.2 | SPM Read Sequence | 71 |
| 6.7 | NVM burn | 72 |
| 7 | APPLICATION INFORMATION | 73 |
| 8 | PACKAGING INFORMATION | 74 |
| 8.1 | Package Outline Drawing | 74 |
| 8.2 | Land Pattern | 74 |

1 GENERAL DESCRIPTION
1.1 Pin Diagram

Figure 1 Pinout Diagram
1.2 Marking information


yyww = Date Code
 xxxxx = Semtech lot number
 R06 = Semtech Code

Figure 2 Marking Information

1.3 Pin Description

| Number | Name | Type | Description |
|--------------|--------|----------------------|--|
| 1 | CAP1 | Analog | Capacitive Sensor 1 |
| 2 | CAP2 | Analog | Capacitive Sensor 2 |
| 3 | CAP3 | Analog | Capacitive Sensor 3 |
| 4 | CAP4 | Analog | Capacitive Sensor 4 |
| 5 | CAP5 | Analog | Capacitive Sensor 5 |
| 6 | CAP6 | Analog | Capacitive Sensor 6 |
| 7 | CAP7 | Analog | Capacitive Sensor 7 |
| 8 | CN | Analog | Integration Capacitor, negative terminal (1nF between CN and CP) |
| 9 | CP | Analog | Integration Capacitor, positive terminal (1nF between CN and CP) |
| 10 | VDD | Power | Main input power supply |
| 11 | INTB | Digital Output | Interrupt, active LOW, requires pull up resistor (in host or external) |
| 12 | SCL | Digital Input | I2C Clock, requires pull up resistor (in host or external) |
| 13 | SDA | Digital Input/Output | I2C Data, requires pull up resistor (in host or external) |
| 14 | GPIO0 | Digital Input/Output | General Purpose Input/Output 0 |
| 15 | GPIO1 | Digital Input/Output | General Purpose Input/Output 1 |
| 16 | GND | Ground | Ground |
| 17 | GPIO2 | Digital Input/Output | General Purpose Input/Output 2 |
| 18 | GPIO3 | Digital Input/Output | General Purpose Input/Output 3 |
| 19 | GPIO4 | Digital Input/Output | General Purpose Input/Output 4 |
| 20 | GPIO5 | Digital Input/Output | General Purpose Input/Output 5 |
| 21 | GND | Ground | Ground |
| 22 | GPIO6 | Digital Input/Output | General Purpose Input/Output 6 |
| 23 | GPIO7 | Digital Input/Output | General Purpose Input/Output 7 |
| 24 | VDIG | Analog | Digital Core Decoupling, connect to a 100nF decoupling capacitor |
| 25 | GND | Ground | Ground |
| 26 | RESETB | Digital Input | Active Low Reset. Connect to VDD if not used. |
| 27 | VANA | Analog | Analog Core Decoupling, connect to a 100nF decoupling capacitor |
| 28 | CAP0 | Analog | Capacitive Sensor 0 |
| bottom plate | GND | Ground | Exposed pad connect to ground |

Table 1 Pin description

1.4 Simplified Block Diagram

The simplified block diagram of the SX8660 is illustrated in Figure 3.

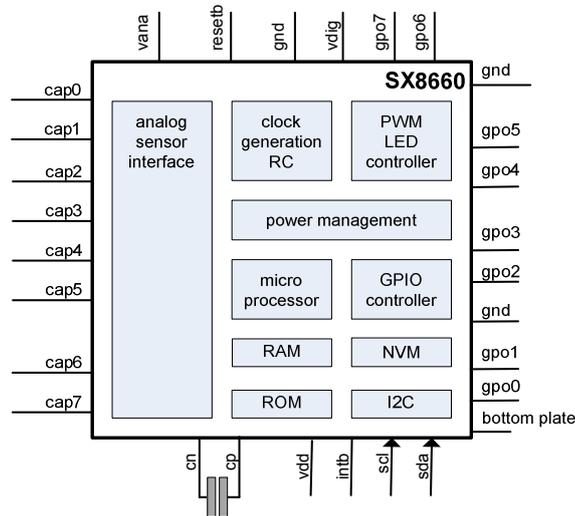


Figure 3 Simplified block diagram of the SX8660

1.5 Acronyms

| | |
|-----|----------------------------|
| AOI | Analog Output Interface |
| ASI | Analog Sensor Interface |
| DCV | Digital Compensation Value |
| GPI | General Purpose Input |
| GPO | General Purpose Output |
| GPP | General Purpose PWM |
| MTP | Multiple Time Programmable |
| NVM | Non Volatile Memory |
| PWM | Pulse Width Modulation |
| QSM | Quick Start Memory |
| SPM | Shadow Parameter Memory |
| SPO | Special Purpose Output |

2 ELECTRICAL CHARACTERISTICS

2.1 Absolute Maximum Ratings

Stresses above the values listed in “Absolute Maximum Ratings” may cause permanent damage to the device.

This is a stress rating only and functional operation of the device at these, or any other conditions beyond the “Recommended Operating Conditions”, is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

| Parameter | Symbol | Min. | Max. | Unit |
|---|--------------------|-------|------|------|
| Supply Voltage | VDD | -0.5 | 3.9 | V |
| Input voltage (non-supply pins) | V _{IN} | -0.5 | 3.9 | V |
| Input current (non-supply pins) | I _{IN} | | 10 | mA |
| Operating Junction Temperature | T _{JCT} | | 125 | °C |
| Reflow temperature | T _{RE} | | 260 | °C |
| Storage temperature | T _{STOR} | -50 | 150 | °C |
| ESD HBM (Human Body model) ⁽ⁱ⁾ | ESD _{HBM} | 3 | | kV |
| Latchup ⁽ⁱⁱ⁾ | I _{LU} | ± 100 | | mA |

Table 2 Absolute Maximum Ratings

(i) Tested to JEDEC standard JESD22-A114

(ii) Tested to JEDEC standard JESD78

2.2 Recommended Operating Conditions

| Parameter | Symbol | Min. | Max. | Unit |
|---|---------------------|------|------|------|
| Supply Voltage | VDD | 2.7V | 3.6 | V |
| Supply Voltage Drop ^(iii, iv, v) | VDD _{drop} | | 100 | mV |
| Supply Voltage for NVM programming | VDD | 3.0V | 3.6 | V |
| Ambient Temperature Range | T _A | -40 | 85 | °C |

Table 3 Recommended Operating Conditions

(iii) Performance for 2.6V < VDD < 2.7V might be degraded.

(iv) Operation is not guaranteed below 2.6V. Should VDD briefly drop below this minimum value, then the SX8660 may require;

- a hardware reset issued by the host using the RESETB pin
- a software reset issued by the host using the I2C interface

(v) In the event the host processor is reset or undergoes a power OFF/ON cycle, it is recommended that the host also resets the SX8660 and assures that parameters are re-written into the SPM (should these differ to the parameters held in NVM).

2.3 Thermal Characteristics

| Parameter | Symbol | Min. | Max. | Unit |
|--|-----------------|------|------|------|
| Thermal Resistance - Junction to Ambient ^(vi) | θ _{JA} | | 25 | °C/W |

Table 4 Thermal Characteristics

(vi) Static airflow

2.4 Electrical Specifications

All values are valid within the operating conditions unless otherwise specified.

| Parameter | Symbol | Conditions | Min. | Typ. | Max. | Unit |
|--|-----------------|---|--------------------|------|-----------------|------------|
| Current consumption | | | | | | |
| Active mode, average | $I_{OP,active}$ | 30ms scan period, 8 sensors enabled, minimum sensitivity | | 200 | 275 | uA |
| Doze mode, average | $I_{OP,Doze}$ | 195ms scan period, 8 sensors enabled, minimum sensitivity | | 70 | 100 | uA |
| Sleep | $I_{OP,sleep}$ | I2C and GPI listening, sensors disabled | | 8 | 17 | uA |
| GPIO set as Input, ResetB, SCL, SDA | | | | | | |
| Input logic high | V_{IH} | | $0.7 \cdot V_{DD}$ | | $V_{DD} + 0.3V$ | V |
| Input logic low | V_{IL} | VSS applied to GND pins | $V_{SS} - 0.3V$ | | 0.8 | V |
| Input leakage current | I_L | CMOS input | | | ± 1 | uA |
| Pull up resistor | R_{PU} | when enabled | | 660 | | k Ω |
| Pull down resistor | R_{PD} | when enabled | | 660 | | k Ω |
| GPIO set as Output, IntB, SDA | | | | | | |
| Output logic high | V_{OH} | $I_{OH} < 4mA$ | $V_{DD} - 0.4$ | | | V |
| Output logic low | V_{OL} | $I_{OL,GPIO} < 12mA$ $I_{OL,SDA,INTB} < 4mA$ | | | 0.4 | V |
| Start-up | | | | | | |
| Power up time | t_{por} | time between rising edge VDD and rising INTB | | | 165 | ms |
| RESETB | | | | | | |
| ResetB pulse width | t_{res} | | 50 | | | ns |
| External components | | | | | | |
| capacitor between VDIG, GND | C_{vdig} | type 0402, tolerance +/-50% | | 100 | | nF |
| capacitor between VANA, GND | C_{vana} | type 0402, tolerance +/-50% | | 100 | | nF |
| capacitor between CP, CN | C_{int} | type 0402, tolerance +/-10% | | 1 | | nF |
| capacitor between VDD, GND | C_{vdd} | type 0402, tolerance +/-50% | | 100 | | nF |

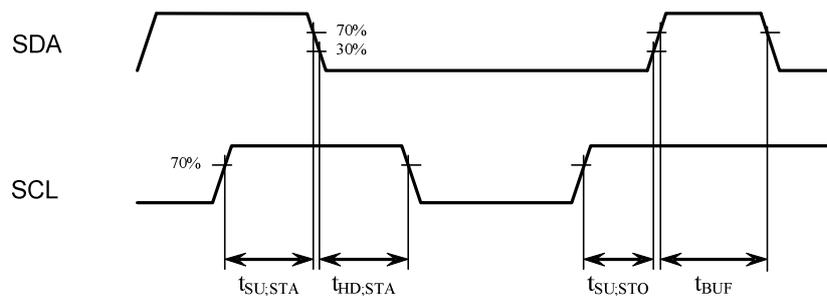
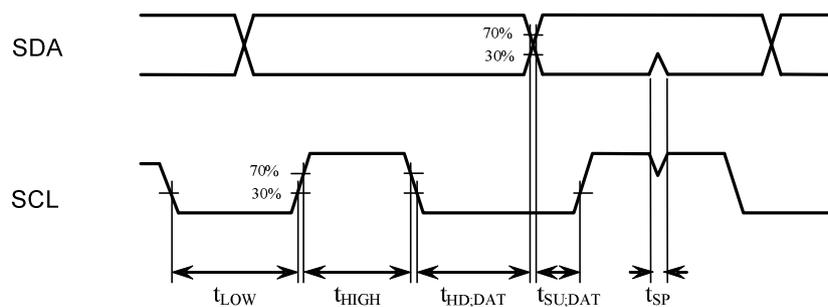
Table 5 Electrical Specifications

| Parameter | Symbol | Conditions | Min. | Typ. | Max. | Unit |
|---|--------------|------------|------|------|------|------|
| I2C Timing Specifications ⁽ⁱ⁾ | | | | | | |
| SCL clock frequency | f_{SCL} | | | | 400 | KHz |
| SCL low period | t_{LOW} | | 1.3 | | | us |
| SCL high period | t_{HIGH} | | 0.6 | | | us |
| Data setup time | $t_{SU,DAT}$ | | 100 | | | ns |
| Data hold time | $t_{HD,DAT}$ | | 0 | | | ns |
| Repeated start setup time | $t_{SU,STA}$ | | 0.6 | | | us |
| Start condition hold time | $t_{HD,STA}$ | | 0.6 | | | us |
| Stop condition setup time | $t_{SU,STO}$ | | 0.6 | | | us |
| Bus free time between stop and start | t_{BUF} | | 500 | | | us |
| Input glitch suppression | t_{SP} | | | | 50 | ns |

Table 6 I2C Timing Specification
Notes:

(i) All timing specifications, Figure 4 and Figure 5, refer to voltage levels (V_{IL} , V_{IH} , V_{OL}) defined in Table 5.

The interface complies with slave F/S mode as described by NXP: "I2C-bus specification, Rev. 03 - 19 June 2007"


Figure 4 I2C Start and Stop timing

Figure 5 I2C Data timing

3 FUNCTIONAL DESCRIPTION

3.1 Quickstart Application

The SX8660 is preconfigured (Quickstart Application) for an application with eight buttons, one Analog Output Interface, one Buzzer output and five LED drivers using various modes of PWM fading. The different fading modes are discussed in later chapters. Table 7 summarises the default configuration of the eight GPIOs available on the SX8660 Quickstart application.

| GPIO | Function | Fading Mode | Comments |
|------|----------|------------------------|--|
| 0 | LED | Single Fading Mode | Autolight, mapped on Btn0 |
| 1 | LED | Single Fading Mode | Autolight, mapped on Btn1 |
| 2 | LED | Continuous Fading Mode | Autolight, mapped on Btn2 |
| 3 | LED | Continuous Fading Mode | Autolight, mapped on Btn3 |
| 4 | LED | Repeat Fading Mode | Autolight, mapped on Btn4 |
| 5 | BUZZER | Not applicable | Buzzer output, 30ms buzzing period, 4KHz for approx. 15ms, 8KHz for approx. 15ms. |
| 6 | LED | Repeat Fading Mode | Autolight, mapped on Btn6 |
| 7 | AOI-A | Not applicable | Analog Output Interface A (VDD=3.3V) Btn0: 0.3V Btn1: 0.6V Btn2: 0.9V Btn3: 1.2V Btn4: 1.5V Btn5: 1.8V Btn6: 2.1V Btn7: 2.4V |

Table 7 Quickstart Application GPIO configuration

Implementing a schematic based on Figure 6 will be immediately operational after powering without programming the SX8660 (even without host).

The default sensitivity is set to 0x03 for all sensors (assumed sensor area approximately 1 cm² covered by 2mm thick acrylic overlay material).

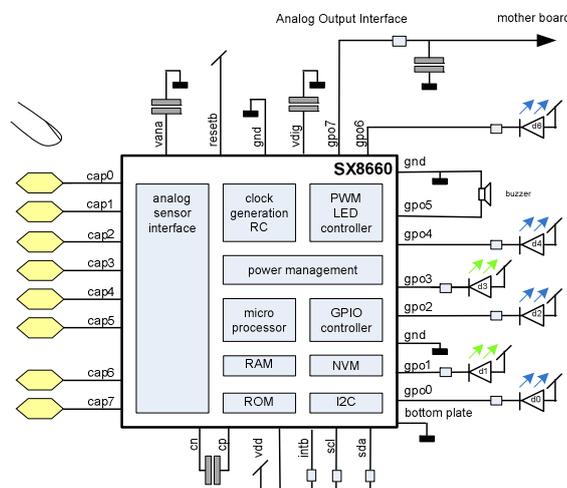


Figure 6 Quickstart Application

3.2 Introduction

3.2.1 General

The SX8660 is intended to be used in applications which require capacitive sensors covered by isolating overlay material. A finger approaching the capacitive sensors will change the charge that can be loaded on the sensors. The SX8660 measures the change of charge and converts that into digital values (ticks). The larger the charge on the sensors, the larger the number of ticks will be. The charge to ticks conversion is done by the SX8660 Analog Sensor Interface (ASI).

The ticks are further processed by the SX8660 and converted in a high level, easy to use information for the user's host.

The information between SX8660 and the user's host is passed through the I2C interface with an additional interrupt signal indicating that the SX8660 has new information. For buttons this information is simply touched or released. The SX8660 can operate without the I2C and interrupt by using the analog output interface (GPIO7 and/or GPIO6) which voltage level indicates the button touched or GPO with the autolight function.

3.2.2 GPIOs

Feedback to the user is using General Purpose Input Output (GPIO) pins. The SX8660 offers up to eight individual configurable GPIO pins. The GPIO can e.g. be set as a LED driver which slowly fade-in when a finger touches a button and slowly fade-out when the button is released. Fading intensity variations can be logarithmic or linear. Interval speed and initial and final light intensity can be selected by the user. The fading is done using a 256 step PWM. The SX8660 has eight individual PWM generators, one for each GPIO pin.

The LED fading-in and fading-out mode is called the GPO (fading) mode.

The LED fading can be initiated automatically by the SX8660 by setting the SX8660 autolighting feature. A simple touch on a sensor and the corresponding LED will fade-in without any host interaction over the I2C.

In case the autolighting feature is disabled then the host will decide to start a LED fading-in period, simply by setting the GPO pin to 'high' using one I2C command. The SX8660 will then slowly fade-in the LED using the PWM autonomously.

In case the host needs to have full control of the LED intensity then the host can set the GPIO in the PWM mode (GPP). The host is then able to set the PWM pulse width freely at the expense of an increased I2C occupation.

The GPIOs can be set further in standard Input mode (GPI) and special purpose output (SPO) for buzzer or analog output interface.

3.2.3 Analog Output Interface A and B (SPO mode)

The Analog Output Interface (AOI) is a PWM output signal between ground and VDD. The duty cycle of the AOI output will change depending on which button is touched. A host controller can then measure the mean voltage delivered on the AOI output as a means of detecting which button is touched at any given time.

The AOI feature allows the SX8660 device to directly replace legacy mechanical button controllers in a quick and effortless manner. The SX8660 supports up to two Analog Output Interfaces, AOI-A and AOI-B, on GPIO7 and GPIO6 respectively. The SX8660 allows buttons to be mapped on either AOI-A or AOI-B. The button mapping as well as the mean voltage level that each button produces on a AOI output can be configured by the user through a set of parameters described in later chapters (see 5.5).

3.2.4 Buzzer (SPO mode)

The SX8660 can drive a buzzer, on GPIO5, to provide audible feedback on button touches. The buzzer duration is set to approximately 30ms per default (5.6).

3.2.5 Parameters

The SX8660 has many low level built-in, fixed algorithms and procedures. To allow a lot of freedom for the user and adapt the SX8660 for different applications these algorithms and procedures can be configured with a large set of parameters which will be described in the following sections. Examples of parameters are which sensors

ADVANCED COMMUNICATIONS & SENSING DATASHEET

are buttons, which GPIO is used for outputs, for the Analog Output Interfaces, the Buzzer or LEDs and which GPIO is mapped to which button.

Sensitivity and detection thresholds of the sensors are part of these parameters. Assuming that overlay material and sensors areas are identical then the sensitivities and thresholds will be the same for each sensor. In case sensors are not of the same size then sensitivities or thresholds might be chosen individually per sensor.

So a smaller size sensor can have a larger sensitivity while a big size sensor may have the lower sensitivity.

3.2.6 Configuration

During a development phase the parameters can be determined and fine tuned by the users and downloaded over the I2C in a dynamic way. The parameter set can be downloaded over the I2C by the host each time the SX8660 boots up. This allows a flexible way of setting the parameters at the expense of I2C occupation.

In case the parameters are frozen they can be programmed in Multiple Time Programmable (MTP) Non Volatile Memory (NVM) on the SX8660. The programming needs to be done once (over the I2C). The SX8660 will then boot up from the NVM and additional parameters from the host are not required anymore.

In case the host desires to overwrite the boot-up NVM parameters (partly or even complete) this can be done by additional I2C communications.

3.3 Scan Period

The basic operation Scan period of the SX8660 sensing interface can be split into three periods over time.

In the first period (Sensing) the SX8660 is sensing all enabled CAP inputs, from CAP0 towards CAP7.

In the second period (Processing) the SX8660 processes the sensor data, verifies and updates the GPIO and the I2C.

In the third period (Timer) the SX8660 is set in a low power mode and waits until a new cycle starts.

Figure 7 shows the different SX8660 periods over time.

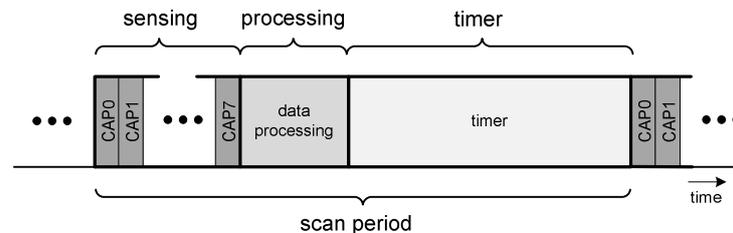


Figure 7 Scan Period

The scan period determines the minimum reaction time of the SX8660. The scan period can be configured by the host from 15ms to values larger than a second.

The reaction time is defined as the interval between a touch on the sensor and the moment that the SX8660 generates the interrupt on the INTB pin. The shorter the scan period the faster the reaction time will be.

Very low power consumptions can be obtained by setting very long scan periods with the expense of having longer reaction times.

All external events like GPIO, I2C and the interrupt are updated in the processing period, so once every scan period. If e.g. a GPI would change state directly after the processing period then this will be reported with a slack of one scan period later in time.

3.4 Operation modes

The SX8660 has 3 operation modes. The main difference is found in the reaction time (corresponding to the scan period) and power consumption.

Active mode offers fast scan periods. The typical reaction time is 30ms. All enabled sensors are scanned and information data is processed within this interval.

Doze mode increases the scan period time which increases the reaction time to 195ms typical and at the same time reduces the operating current.

Sleep mode turns the SX8660 OFF, except for the I2C and GPI peripheral, minimizing operating current while maintaining the power supplies. In Sleep mode the SX8660 does not do any sensor scanning. The Sleep mode will be exited by any I2C access or a GPI interrupt.

The user can specify other scan periods for the Active and Doze mode and decide for other compromises between reaction time and power consumption.

In most applications the reaction time needs to be fast when fingers are present, but can be slow when no person uses the application. In case the SX8660 is not used for a specific time it will go from Active mode into Doze mode and power will be saved. This time-out is determined by the Passive Timer which can be configured by the user or turned OFF if not required.

To leave Doze mode and enter Active mode this can be done by a simple touch on any button.

The host can decide to force the operating mode by issuing commands over the I2C (using register GpioOpMode) and take fully control of the SX8660. The diagram in Figure 8 shows the available operation modes and the possible transitions.

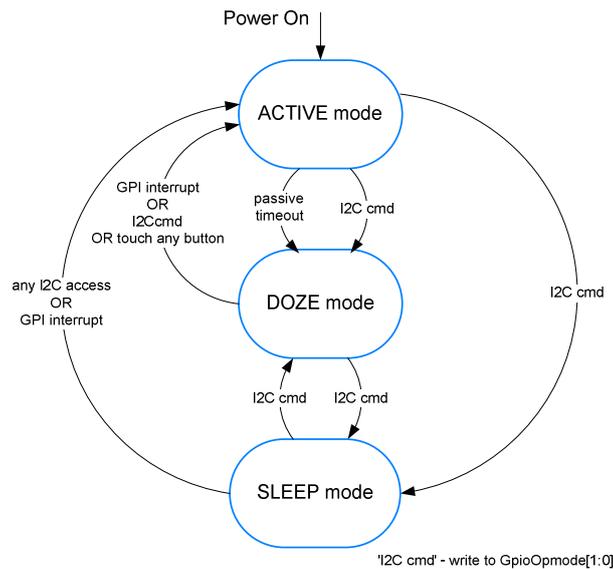


Figure 8 Operation modes

3.5 Sensors on the PCB

The capacitive sensors are relatively simple copper areas on the PCB connected to the eight SX8660 capacitive sensor input pins (CAP0...CAP7). The sensors are covered by isolating overlay material (typically 1mm...3mm). The area of a sensor is typically one square centimetre which corresponds about to the area of a finger touching the overlay material.

The capacitive sensors can be setup as ON/OFF buttons (see example Figure 9) for control applications.

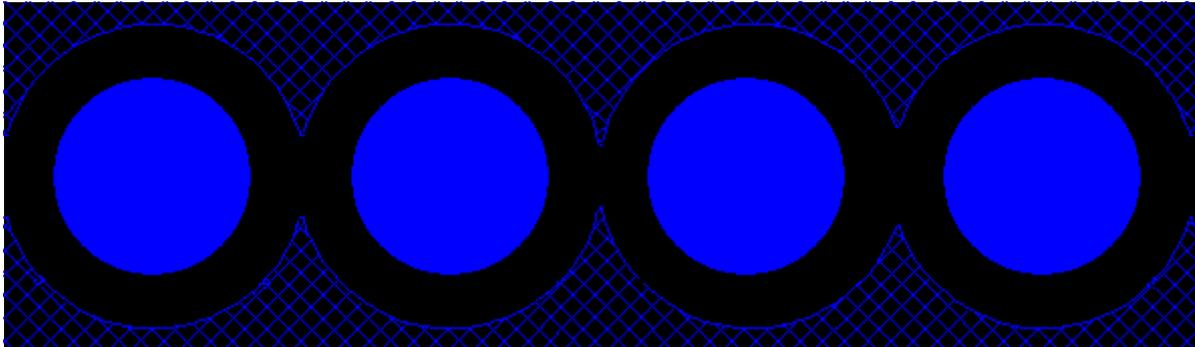


Figure 9 PCB top layer of four sensors for buttons (surrounded by a ground plane)

3.6 Button Information

The buttons have two simple states (see Figure 10): ON (touched by finger) and OFF (released and no finger press).

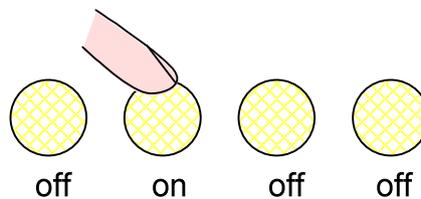


Figure 10 Buttons

A finger is detected as soon as the number of ticks from the ASI reaches a user-defined threshold plus a hysteresis.

A release is detected if the tick from the ASI goes below the threshold minus a hysteresis. The hysteresis around the threshold avoids rapid touch and release signalling during transients.

3.7 Buzzer

The SX8660 has the ability to drive a buzzer (on GPIO5) to provide an audible indication that a button has been touched. The buzzer is driven by a square signal for approximately 30ms (default). During the first phase (15ms) the signal's frequency is default 4KHz while in the second phase (15ms) the signal's frequency default is 8KHz.

The buzzer is activated only once during any button touch and is not repeated for long touches. The user can choose to enable or disable the buzzer by configuration and define the idle level, frequencies and phase durations (see 5.6).

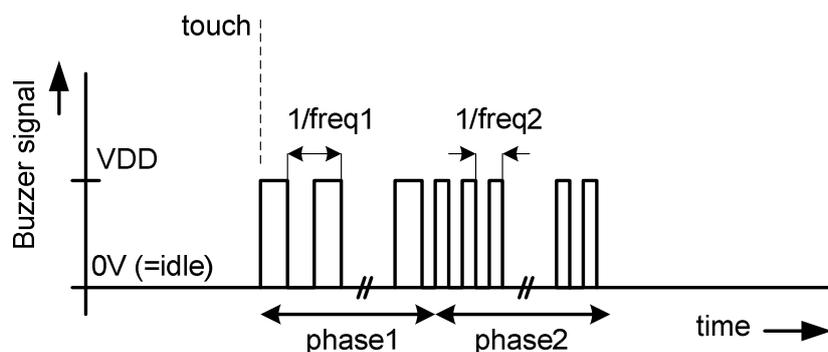


Figure 11 buzzer behavior

3.8 Analog Output Interface

The Analog Output Interface outputs a PWM signal with a varying duty cycle depending on which button is touched. By filtering (with a simple RC filter) the PWM signal results in a DC voltage, different for each button touch. The host controller measures the DC voltage level and determines which buttons has been touched.

In the case of single button touches, each button produces its own voltage level as configured by the user (see 5.5 and Table 8).

Figure 12 show how the AOI will behave when the user touches and releases different buttons. The AOI will switch between the AOI idle level and the level for each button.

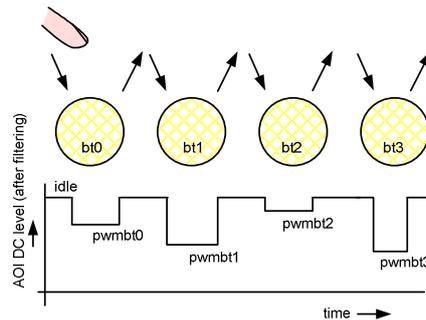


Figure 12 AOI behavior

The PWM blocks used in AOI modes are 8-bits based and clocked at 2MHz typically.

The PWM period can be set to 256 (default) or 64. The 256 period offers a better granularity at a lower frequency, while the 64 period is faster and with fewer steps.

Figure 13 shows the PWM definition of the AOI.

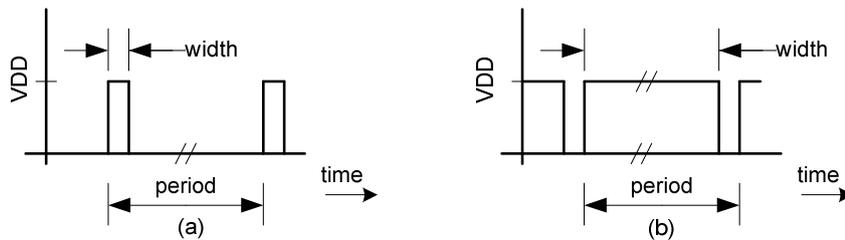


Figure 13 PWM definition, (a) small pulse width, (b) large pulse width

Table 8 describes the AOI level index versus the PWM pulse width. The user can select 256 steps (index) in case the period is set to 255.

In case the period is set to 64 then the index from 0 to 63 applies.

| Index | Width |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| 0 | 0 | 32 | 33 | 64 | 65 | 96 | 97 | 128 | 129 | 160 | 161 | 192 | 193 | 224 | 225 |
| 1 | 2 | 33 | 34 | 65 | 66 | 97 | 98 | 129 | 130 | 161 | 162 | 193 | 194 | 225 | 226 |
| 2 | 3 | 34 | 35 | 66 | 67 | 98 | 99 | 130 | 131 | 162 | 163 | 194 | 195 | 226 | 227 |
| 3 | 4 | 35 | 36 | 67 | 68 | 99 | 100 | 131 | 132 | 163 | 164 | 195 | 196 | 227 | 228 |
| 4 | 5 | 36 | 37 | 68 | 69 | 100 | 101 | 132 | 133 | 164 | 165 | 196 | 197 | 228 | 229 |
| 5 | 6 | 37 | 38 | 69 | 70 | 101 | 102 | 133 | 134 | 165 | 166 | 197 | 198 | 229 | 230 |
| 6 | 7 | 38 | 39 | 70 | 71 | 102 | 103 | 134 | 135 | 166 | 167 | 198 | 199 | 230 | 231 |
| 7 | 8 | 39 | 40 | 71 | 72 | 103 | 104 | 135 | 136 | 167 | 168 | 199 | 200 | 231 | 232 |
| 8 | 9 | 40 | 41 | 72 | 73 | 104 | 105 | 136 | 137 | 168 | 169 | 200 | 201 | 232 | 233 |
| 9 | 10 | 41 | 42 | 73 | 74 | 105 | 106 | 137 | 138 | 169 | 170 | 201 | 202 | 233 | 234 |
| 10 | 11 | 42 | 43 | 74 | 75 | 106 | 107 | 138 | 139 | 170 | 171 | 202 | 203 | 234 | 235 |
| 11 | 12 | 43 | 44 | 75 | 76 | 107 | 108 | 139 | 140 | 171 | 172 | 203 | 204 | 235 | 236 |
| 12 | 13 | 44 | 45 | 76 | 77 | 108 | 109 | 140 | 141 | 172 | 173 | 204 | 205 | 236 | 237 |
| 13 | 14 | 45 | 46 | 77 | 78 | 109 | 110 | 141 | 142 | 173 | 174 | 205 | 206 | 237 | 238 |
| 14 | 15 | 46 | 47 | 78 | 79 | 110 | 111 | 142 | 143 | 174 | 175 | 206 | 207 | 238 | 239 |
| 15 | 16 | 47 | 48 | 79 | 80 | 111 | 112 | 143 | 144 | 175 | 176 | 207 | 208 | 239 | 240 |

| Index | Width |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| 16 | 17 | 48 | 49 | 80 | 81 | 112 | 113 | 144 | 145 | 176 | 177 | 208 | 209 | 240 | 241 |
| 17 | 18 | 49 | 50 | 81 | 82 | 113 | 114 | 145 | 146 | 177 | 178 | 209 | 210 | 241 | 242 |
| 18 | 19 | 50 | 51 | 82 | 83 | 114 | 115 | 146 | 147 | 178 | 179 | 210 | 211 | 242 | 243 |
| 19 | 20 | 51 | 52 | 83 | 84 | 115 | 116 | 147 | 148 | 179 | 180 | 211 | 212 | 243 | 244 |
| 20 | 21 | 52 | 53 | 84 | 85 | 116 | 117 | 148 | 149 | 180 | 181 | 212 | 213 | 244 | 245 |
| 21 | 22 | 53 | 54 | 85 | 86 | 117 | 118 | 149 | 150 | 181 | 182 | 213 | 214 | 245 | 246 |
| 22 | 23 | 54 | 55 | 86 | 87 | 118 | 119 | 150 | 151 | 182 | 183 | 214 | 215 | 246 | 247 |
| 23 | 24 | 55 | 56 | 87 | 88 | 119 | 120 | 151 | 152 | 183 | 184 | 215 | 216 | 247 | 248 |
| 24 | 25 | 56 | 57 | 88 | 89 | 120 | 121 | 152 | 153 | 184 | 185 | 216 | 217 | 248 | 249 |
| 25 | 26 | 57 | 58 | 89 | 90 | 121 | 122 | 153 | 154 | 185 | 186 | 217 | 218 | 249 | 250 |
| 26 | 27 | 58 | 59 | 90 | 91 | 122 | 123 | 154 | 155 | 186 | 187 | 218 | 219 | 250 | 251 |
| 27 | 28 | 59 | 60 | 91 | 92 | 123 | 124 | 155 | 156 | 187 | 188 | 219 | 220 | 251 | 252 |
| 28 | 29 | 60 | 61 | 92 | 93 | 124 | 125 | 156 | 157 | 188 | 189 | 220 | 221 | 252 | 253 |
| 29 | 30 | 61 | 62 | 93 | 94 | 125 | 126 | 157 | 158 | 189 | 190 | 221 | 222 | 253 | 254 |
| 30 | 31 | 62 | 63 | 94 | 95 | 126 | 127 | 158 | 159 | 190 | 191 | 222 | 223 | 254 | 255 |
| 31 | 32 | 63 | 64 | 95 | 96 | 127 | 128 | 159 | 160 | 191 | 192 | 223 | 224 | 255 | 256 |

Table 8 AOI Level index vs. PWM pulse width (normal polarity)

The AOI reports always one button. The AOI can be split over two GPIO pins (AOI-A, AOI-B). The AOI-A interface is mapped on pin GPIO7 and the AOI-B is mapped on pin GPIO6. The user can map any button to either AOI-A or AOI-B or both.

In case buttons are split among both AOI pins, multiple button touches are still resulting in one AOI reporting.

In most applications only one AOI pin will be selected. The two AOI pins allow the user a more coarse detection circuitry at the host. Assuming a 3.3V supply and 8 buttons on one single AOI then the AOI levels could be separated with around 0.3...0.4V.

In case of using the two AOI pins, 4 buttons could be mapped on AOI-A separated with around 0.8V (similar for 4 buttons on AOI-B) which is about the double in case of a single AOI.

In case of a single touch the reported button is the straight forward (as in Figure 12).

If more than one button is touched the reported depends on the selected button reporting mode parameter (5.4).

Three reporting modes exist for the SX8660 (All, First and Strongest).

The All reporting mode is applicable only for the I2C reporting. In All-mode all buttons that are touched are reported in the I2C buttons status bits.

In the First-mode the first touched button will be reported on the AOI and the I2C. All touches that occur afterwards will not be reported as long as the first touch sustains. Only when the first reported button is released will the SX860 report other touches. The button that is reported is the one with the lowest Cap pin index.

Figure 14 shows the First-mode reporting in case of 2 touches.

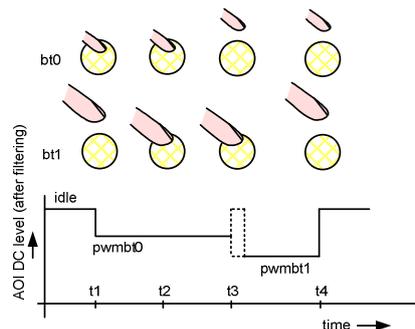


Figure 14 First-mode reporting with 2 touches

At time t1 button0 is touched and reported on the AOI. At time t2 button1 is touched as well but not reported. At time t3 the button0 is released and button1 will be reported immediately (or after one scan period at idle level). At time t4 both buttons are released and the AOI reports the idle level.

In the Strongest-mode the strongest touched button will be reported on the AOI and the I2C. All touches that occur afterwards representing a weaker touch will not be reported. Only a touch which is stronger will be reported by the SX860.

Figure 15 shows the Strongest-mode reporting in case of 2 touches (with bt1 the strongest touch).

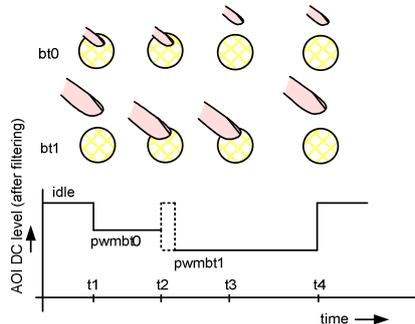


Figure 15 Strongest-mode reporting with 2 touches

At time t1 button0 is touched and reported on the AOI. At time t2 button1 is touched as well. As bt1 is the strongest touch it will be reported on the AOI immediately (or after one scan period at idle level). At time t3 the button0 is released while the AOI continues to report button1. At time t4 both buttons are released and the AOI reports the idle level.

In some special cases (when the buzzer is suspected to load heavily the power supply) the user may choose the AOI to go to 0V, to VDD or to the AOI idle level for the duration the buzzer is active.

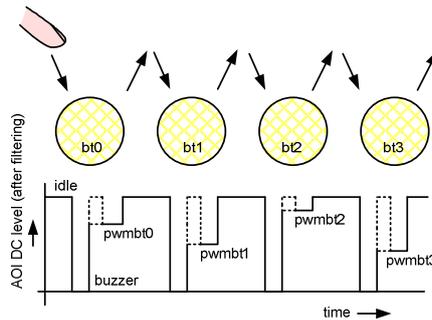


Figure 16 AOI behavior with 0V buzzer state

In Figure 16 the AOI will go to 0V each time the buzzer is active. The AOI returns then to either the idle mode for one scan period or goes immediately to the PWM button level.

3.9 Analog Sensing Interface

The Analog Sensing Interface (ASI) converts the charge on the sensors into ticks which will be further digitally processed. The basic principle of the ASI will be explained in this section.

The ASI consists of a multiplexer selecting the sensor, analog switches, a reference voltage, an ADC sigma delta converter, an offset compensation DAC and an external integration capacitor (see Figure 17).

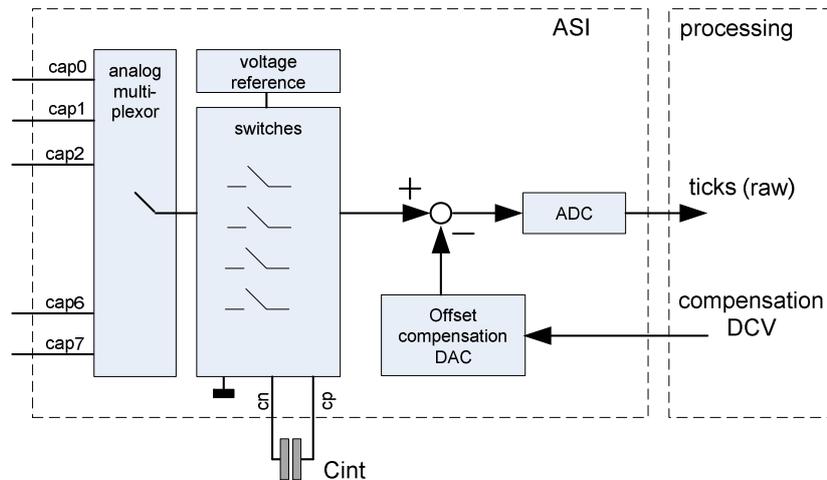


Figure 17 Analog Sensor Interface

To get the ticks representing the charge on a specific sensor the ASI will execute several steps.

The charge on a sensor cap (e.g. CAP0) will be accumulated multiple times on the external integration capacitor, Cint.

This results in an increasing voltage on Cint proportional to the capacitance on CAP0.

At this stage the offset compensation DAC is enabled. The compensation DAC generates a voltage proportional to an estimation of the external capacitance. The estimation is obtained by the offset compensation procedure executed e.g. at power-up.

The difference between the DAC output and the charge on Cint is the desired signal. In the ideal case the difference of charge will be converted to zero ticks if no finger is present and the number of ticks becomes high in case a finger is present.

The difference of charge on Cint and the DAC output will be transferred to the ADC (Sigma Delta Integrator).

After the charge transfer to the ADC the steps above will be repeated.

The larger the number the cycles are repeated the larger the signal out of the ADC with improved SNR. The sensitivity is therefore directly related to the number of cycles.

The SX8660 allows setting the sensitivity for each sensor individually in applications which have a variety of sensors sizes or different overlays or for fine-tuning performances. The optimal sensitivity is depending heavily on the final application. If the sensitivity is too low the ticks will not pass the thresholds and it is not possible to detect fingers. In case the sensitivity is set too large a finger hovering above the sensors will already be detected before the finger really touches the overlay resulting in false detections.

Once the ASI has finished the first sensor, the ticks are stored and the ASI will start measuring the next sensor until all (enabled) sensors pins have been treated.

In case some sensors are disabled then these result in lower power consumption simply because the ASI is active for a shorter period and the following processing period will be shorter.

The ticks from the ASI will then be handled by the digital processing.

The ASI will shut down and wait until new sensing period will start.

3.10 Offset Compensation

The capacitance at the CAP pins is determined by an intrinsic capacitance of the integrated circuit, the PCB traces, ground coupling and the sensor planes. This capacitance is relatively large and might become easily some tens of pF. This parasitic capacitance will vary only slowly over time due to environmental changes.

A finger touch is in the order of one pF. If the finger approaches the sensor this occurs typically fast.

The ASI has the difficult task to detect and distinguish a small, fast changing capacitance, from a large, slow varying capacitance. This would require a very precise, high resolution ADC and complicated, power consuming, digital processing.

The SX8660 features a 16 bit DAC which compensates for the large, slow varying capacitance already in front of the ADC. In other words the ADC converts only the desired small signal. In the ideal world the ADC will put out zero ticks even if the external capacitance is as high as 100pF.

At each power-up of the SX8660 the Digital Compensation Values (DCV) are estimated by the digital processing algorithms. The algorithm will adjust the compensation values such that zero ticks will be generated by the ADC.

Once the correct compensation values are found these will be stored and used to compensate each CAP pin.

If the SX8660 is shut down the compensation values will be lost. At a next power-up the procedure starts all over again. This assures that the SX8660 will operate under any condition. Powering up at e.g. different temperatures will not change the performance of the SX8660 and the host does not have to do anything special.

The DCVs do not need to be updated if the external conditions remain stable.

However if e.g. temperature changes this will influence the external capacitance. The ADC ticks will drift then slowly around zero values basically because of the mismatch of the compensation circuitry and the external capacitance.

In case the average value of the ticks become higher than the positive noise threshold (configurable by user) or lower than the negative threshold (configurable by user) then the SX8660 will initiate a compensation procedure and find a new set of DCVs.

Compensation procedures can as well be initiated by the SX8660 on periodic intervals. Even if the ticks remain within the positive and negative noise thresholds the compensation procedure will then estimate new sets of DCVs.

Finally the host can initiate a compensation procedure by using the I2C interface. This is e.g. required after the host changed the sensitivity of sensors.

3.11 Processing

The first processing step of the raw ticks, coming out of the ASI, is low pass filtering to obtain an estimation of the average capacitance: tick-ave (see Figure 18). This slowly varying average is important in the detection of slowly changing environmental changes.

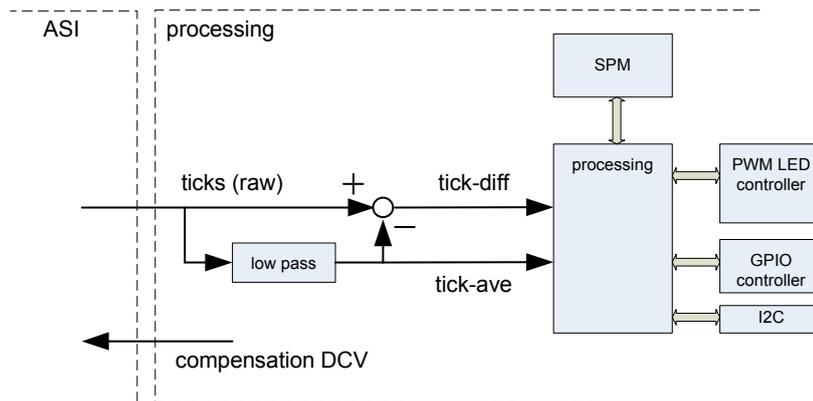


Figure 18 Processing

The difference of the tick average and the raw ticks, tick-diff, is a good estimation of rapid changing input capacitances.

The tick-diff, tick-ave and the configuration parameters in the SPM are then processed and determines the sensor information, I2C registers status and PWM control.

3.12 Configuration

Figure 19 shows the building blocks used for configuring the SX8660.

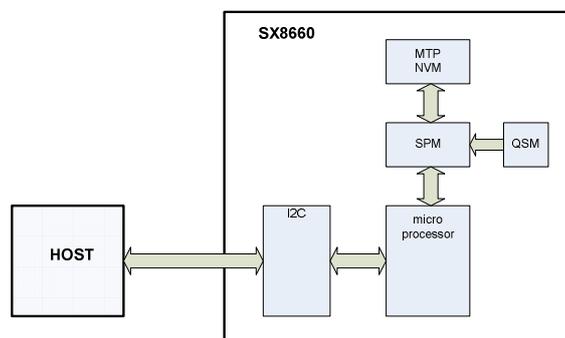


Figure 19 Configuration

The default configuration parameters of the SX8660 are stored in the Quick Start Memory (QSM). This configuration data is setup to a very common application for the SX8660 with 8 buttons. Without any programming or host interaction the SX8660 will start up in the Quick Start Application.

The QSM settings are fixed and cannot be changed by the user.

In case the application needs different settings than the QSM settings then the SX8660 can be setup and/or programmed over the I2C interface.

The configuration parameters of the SX8660 can be stored in the Multiple Time Programmable (MTP) Non Volatile Memory (NVM). The NVM contains all those parameters that are defined and stable for the application. Examples are the number of sensors enabled, sensitivity, active and Doze scan period. The details of these parameters are described in the next chapters.

At power up the SX8660 checks if the NVM contains valid data. In that case the configuration parameter source becomes the NVM. If the NVM is empty or non-valid then the configuration source becomes the QSM. In the next step the SX8660 copies the configuration parameter source into the Shadow Parameter Memory (SPM). The SX8660 is operational and uses the configuration parameters of the SPM.

During power down or reset event the SPM loses all content. It will automatically be reloaded following power up or at the end of the reset event.

The host will interface with the SX8660 through the I2C bus and the analog output interface.

The I2C of the SX8660 consists of 16 registers. Some of these I2C registers are used to read the status and information of the button and GPI. Other I2C registers allow the host to take control of the SX8660. The host can e.g. decide to change the operation mode from active mode to Doze mode or go into sleep (according Figure 8).

Two additional modes allow the host to have an access to the SPM or indirect access to the NVM. These modes are required during development, can be used in real time or in-field programming.

Figure 20 shows the Host SPM mode. In this mode the host can decide to overwrite the SPM. This is useful during the development phases of the application where the configuration parameters are not yet fully defined and as well during the operation of the application if some parameters need small deviations from the QSM or NVM content.

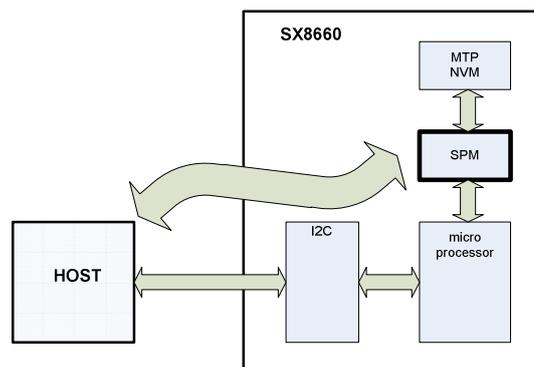


Figure 20 Host SPM mode

The content of the SPM remains valid as long as the SX8660 is powered. After a power down the host needs to re-write the SPM at the next power-up.

Figure 21 shows the Host NVM mode. In this mode the host will be able to write the NVM.

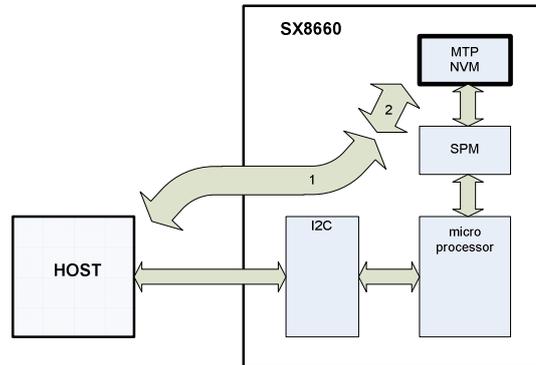


Figure 21 Host NVM mode

The writing of the host towards the NVM is not done directly but done in 2 steps (Figure 21).

In the first step the host writes to the SPM (as in Figure 20). In the second step the host signals the SX8660 to copy the SPM content into the NVM.

Initially the NVM memory is empty and it is required to determine a valid parameter set for the application. This can be done during the development phase using dedicated evaluation hardware representing the final application. This development phase uses probably initially the host SPM mode which allows faster iterations.

Once the parameter set is determined this can be written to the NVM over the I2C using the 2 steps approach by the host or a dedicated programmer for large volumes production (as described in the paragraphs 6.6 and 6.7).

3.13 Power Management

The SX8660 uses on-chip voltage regulators which are controlled by the on-chip microprocessor. The regulators need to be stabilized with an external capacitor between VANA and ground and between VDIG and ground (see Table 5). Both regulators are designed to only drive the SX8660 internal circuitry and must not be loaded externally.

3.14 Clock Circuitry

The SX8660 has its own internal clock generation circuitry that does not require any external components. The clock circuitry is optimized for low power operation and is controlled by the on-chip microprocessor. The typical operating frequency of the oscillating core is 16.7MHz from which all other lower frequencies are derived.

3.15 I2C interface

The I2C interface allows the communication between the host and the SX8660.

The I2C slave implemented on the SX8660 is compliant with the standard (100kb/s) and fast mode (400kb/s). The default SX8660 I2C address equals 0b010 1011.

A different I2C address can be programmed by the user in the NVM.

3.16 Interrupt

3.16.1 Power up

During power up the INTB is kept low. Once the power up sequence is terminated the INTB is cleared autonomously. The SX8660 is then ready for operation. The AOI levels are updated at the latest one scan period after the rising edge of INTB.

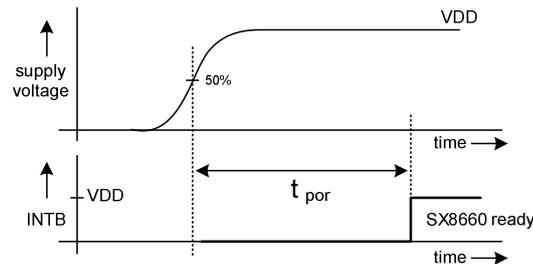


Figure 22 Power Up vs. INTB

During the power on period the SX8660 stabilizes the internal regulators, RC clocks and the firmware initializes all registers.

During the power up the SX8660 is not accessible and I2C communications are forbidden. As soon as the INTB rises the SX8660 will be ready for I2C communication. The value of INTB before power up depends on the INTB pull up resistor supply voltage.

3.16.2 Assertion

INTB is updated in Active or Doze mode once every scan period.

The INTB will be asserted at the following events:

- if a Button event occurred (touch or release if enabled). I2C register CapStatLsb show the detailed status of the Buttons,
- if a GPI edge occurred (rising or falling if enabled). I2C register GpiStat shows the detailed status of the GPI pins,
- when actually entering Active or Doze mode via a host request (may be delayed by 1 scan period). I2C register CompOpmode shows the current operation mode,
- once compensation procedure is completed either through automatic trigger or via host request (may be delayed by 1 scan period),
- once SPM write is effective (may be delayed by 1 scan period),
- once NVM burn procedure is completed (may be delayed by 1 scan period),
- during reset (power up, hardware RESETB, software reset).

3.16.3 Clearing

The clearing of the INTB is done as soon as the host performs a read to any of the SX8660 I2C registers.

3.16.4 Example

A typical example of the assertion and clearing of the INTB and the I2C communication is shown in Figure 23.

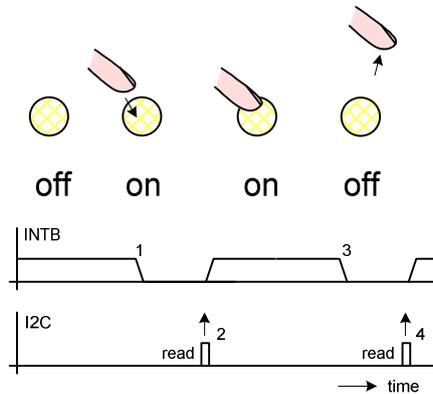


Figure 23 Interrupt and I2C

When a button is touched the SX8660 will assert the interrupt (1). The host will read the SX8660 status information over the I2C (2) and this clears the interrupt.

If the finger releases the button the interrupt will be asserted (3), the host reads the status (4) which clears the interrupt.

In case the host will not react to an interrupt then this will result in a missing touch.

3.17 Reset

The reset can be performed by 3 sources:

- power up,
- RESETB pin,
- software reset.

3.17.1 Power up

During power up the INTB is kept low. Once the power up sequence is terminated the INTB is released autonomously. The SX8660 is then ready for operation.

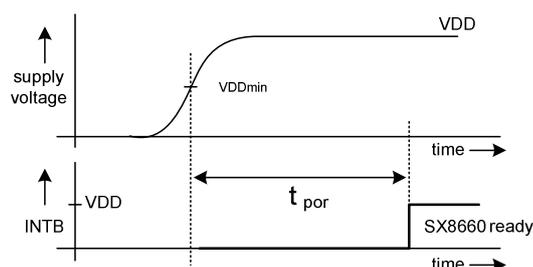


Figure 24 Power Up vs. INTB

During the power on period the SX8660 stabilizes the internal regulators, RC clocks and the firmware initializes all registers.

During the power up the SX8660 is not accessible and I2C communications are forbidden. As soon as the INTB rises the SX8660 will be ready for I2C communication.

3.17.2 RESETB

When RESETB is driven low the SX8660 will reset and start the power up sequence as soon as RESETB is driven high or pulled high.

In case the user does not require a hardware reset control pin then the RESETB pin can be connected to VDD.

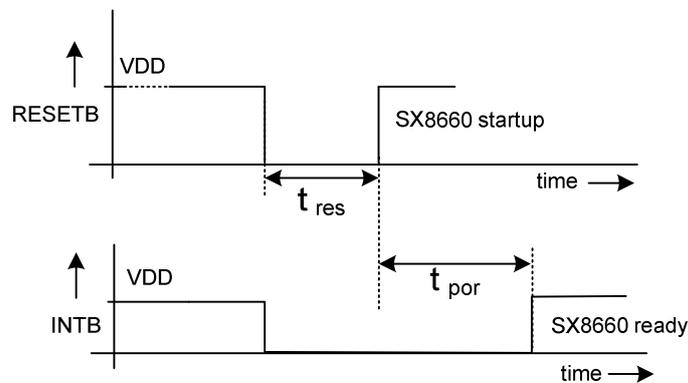


Figure 25 Hardware Reset

3.17.3 Software Reset

To perform a software reset the host needs to write 0xDE followed by 0x00 at the SoftReset register at address 0xB1.

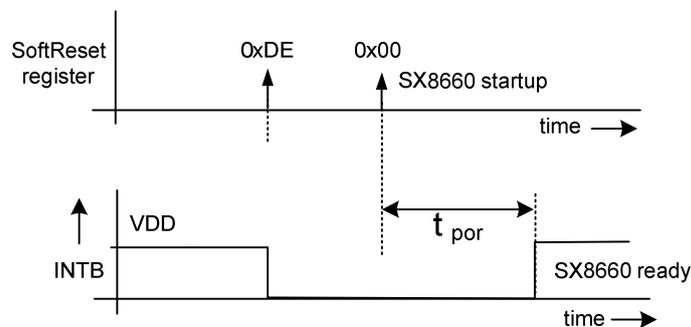


Figure 26 Software Reset

3.18 General Purpose Input and Outputs

The SX8660 offers eight General Purpose Input and Outputs (GPIO) pins which can be configured in any of these modes:

- GPI (General Purpose Input)
- GPP (General Purpose PWM)
- GPO (General Purpose Output)
- SPO (Special Purpose Output)

Each of these modes is described in more details in the following sections.

The polarity of the GPP and GPO pins is defined as in figure below, driving an LED as example. It has to be set accordingly in SPM parameter GpioPolarity.

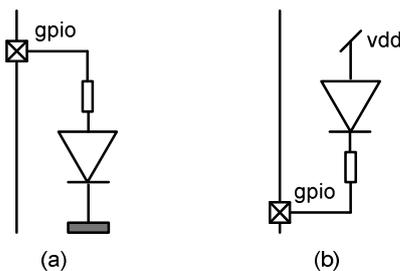


Figure 27 polarity = 1 (a), polarity = 0 (b)

The PWM blocks used in GPP and GPO modes are 8-bits based and clocked at 2MHz typ. hence offering 256 selectable pulse width values with a granularity of 0.5us typ.

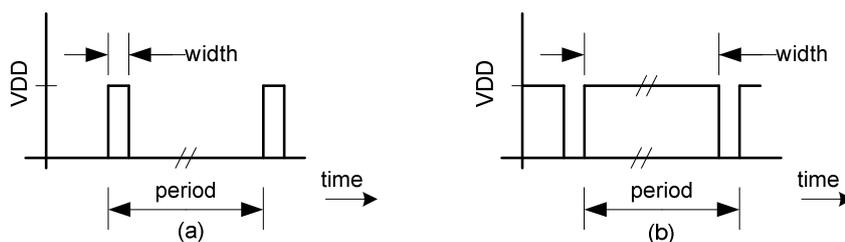


Figure 28 PWM definition, (a) small pulse width, (b) large pulse width

3.18.1 GPI

GPIOs configured as GPI will operate as digital inputs with standard low and high logic levels.

Optional pull-up/down and debounce can be enabled. Each GPI is individually edge programmable for INTB generation which will also exit Sleep/Doze mode if relevant.

SPM/I2C parameters applicable in GPI mode are listed in table below. Please refer to the relevant SPM/I2C parameters sections for more details.

| | | GPI |
|------------|----------------|-----|
| SPM | GpioMode | X |
| | GpioPullUpDown | X |
| | GpioInterrupt | X |
| | GpioDebounce | X |
| I2C | IrqSrc[4] | X |
| | GpiStat | X |

Table 9 SPM/I2C Parameters Applicable in GPI Mode

3.18.2 GPP mode

GPIOs configured as GPP will operate as PWM outputs directly controlled by the host. A typical application is LED dimming.

Typical GPP operation is illustrated in figure below.

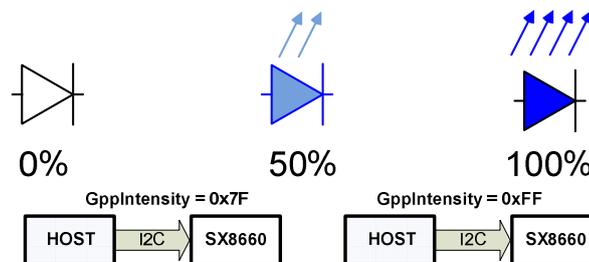


Figure 29 LED control in GPP mode

SPM/I2C parameters applicable in GPP mode are listed in table below. Please refer to the relevant SPM/I2C parameters sections for more details.

| | | GPP |
|------------|------------------|----------------|
| SPM | GpioMode | X |
| | GpioOutPwrUp | X ¹ |
| | GpioPolarity | X |
| | GpioIntensityOn | X ¹ |
| | GpioIntensityOff | X ¹ |
| | GpioFunction | X |
| I2C | GppPinId | X |
| | GppIntensity | X ¹ |

¹ At power up, GppIntensity of each GPP pin is initialized with GpioIntensityOn or GpioIntensityOff depending on GpioOutPwrUp corresponding bits value.

Table 10 SPM/I2C Parameters Applicable in GPP Mode

3.18.3 GPO

GPIOs configured as GPO will operate as digital outputs which can generate both standard low/high logic levels and PWM low/high duty cycles levels. Typical application is LED ON/OFF control.

Transitions between ON and OFF states can be triggered either automatically in Autolight mode or manually by the host. This is illustrated in figures below.

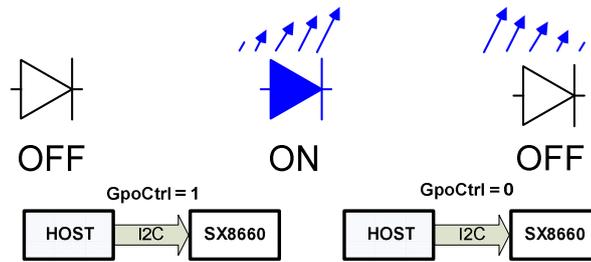


Figure 30 LED Control in GPO mode, Autolight OFF

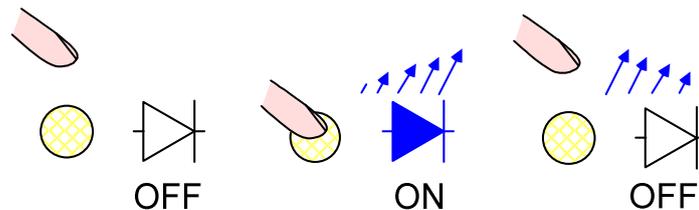


Figure 31 LED Control in GPO mode, Autolight ON (mapped to Button)

Additionally these transitions can be configured to be done with or without fading following a logarithmic or linear function. This is illustrated in figures below.

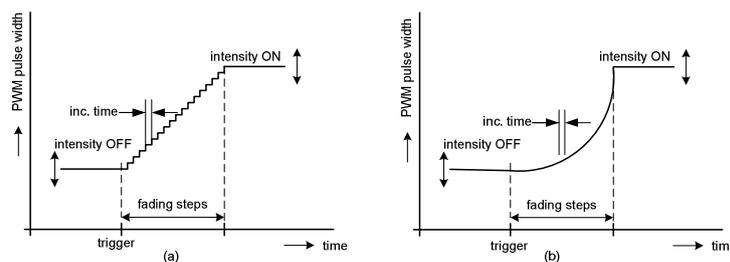


Figure 32 GPO ON transition (LED fade in), normal polarity, (a) linear, (b) logarithmic

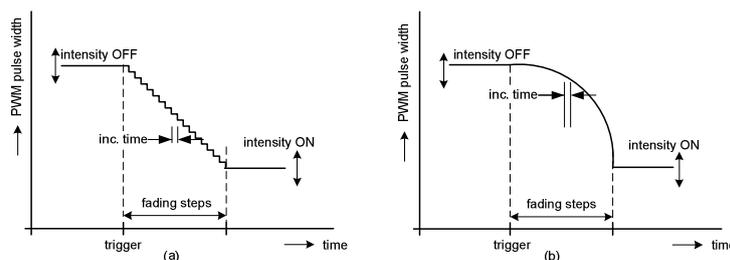


Figure 33 GPO ON transition (LED fade in), inverted polarity, (a) linear, (b) logarithmic

The fading out (e.g. after a button is released) is identical to the fading in but an additional off delay can be added before the fading starts (Figure 34 and Figure 35).

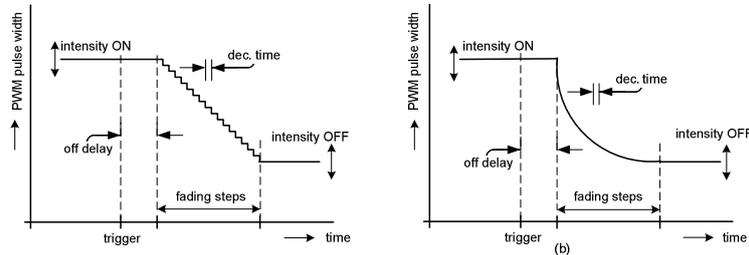


Figure 34 GPO OFF transition (LED fade out), normal polarity, (a) linear, (b) logarithmic

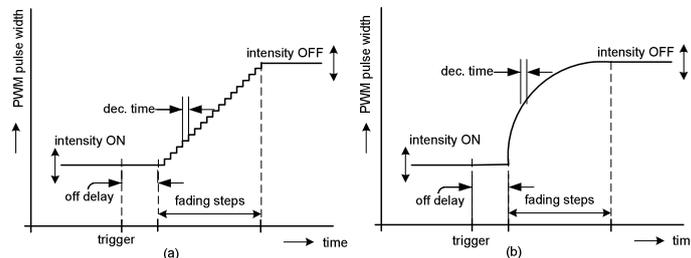


Figure 35 GPO OFF transition (LED fade out), inverted polarity, (a) linear, (b) logarithmic

Please note that standard high/low logic signals are just a specific case of GPO mode and can also be generated simply by setting inc/dec time to 0 (ie OFF) and programming intensity OFF/ON to 0x00 and 0xFF.

3.18.4 GPO Fading

The SX8660 supports three different fading modes, namely i) Single, ii) Continuous and iii) Repeat. These fading modes can be configured for each GPIO individually. Please see 5.8 “GPIO Parameters” for more information on how to configure this feature.

i) Single Fading Mode:

The GPO pin fades in when the associated button is touched and it fades out when it is released. This is shown in Figure 36

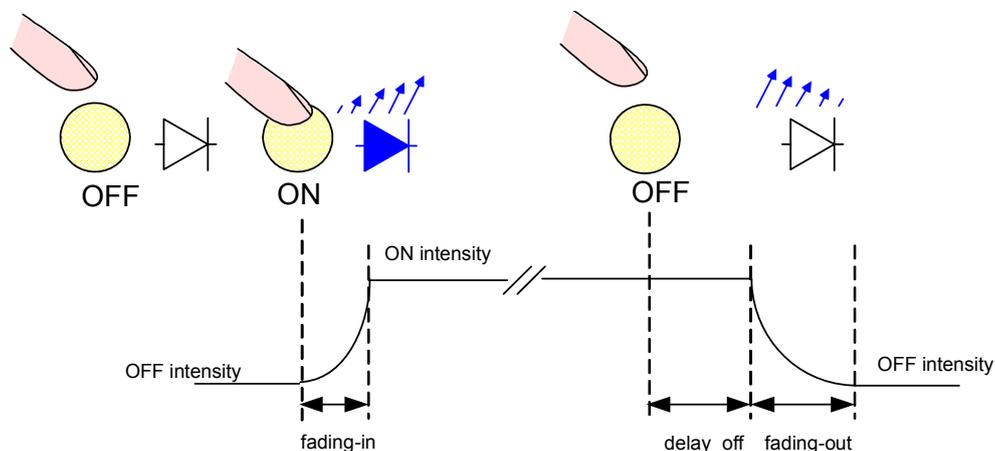


Figure 36 Single Fading Mode

ii) Continuous Fading Mode:

The GPO pin fades in and fades out repeatedly when the associated button is touched. When the button is released the fading in/out either stops immediately after completing a cycle or carries on for a configurable number of times (see register GpioOffDelay). This is shown in Figure 37.

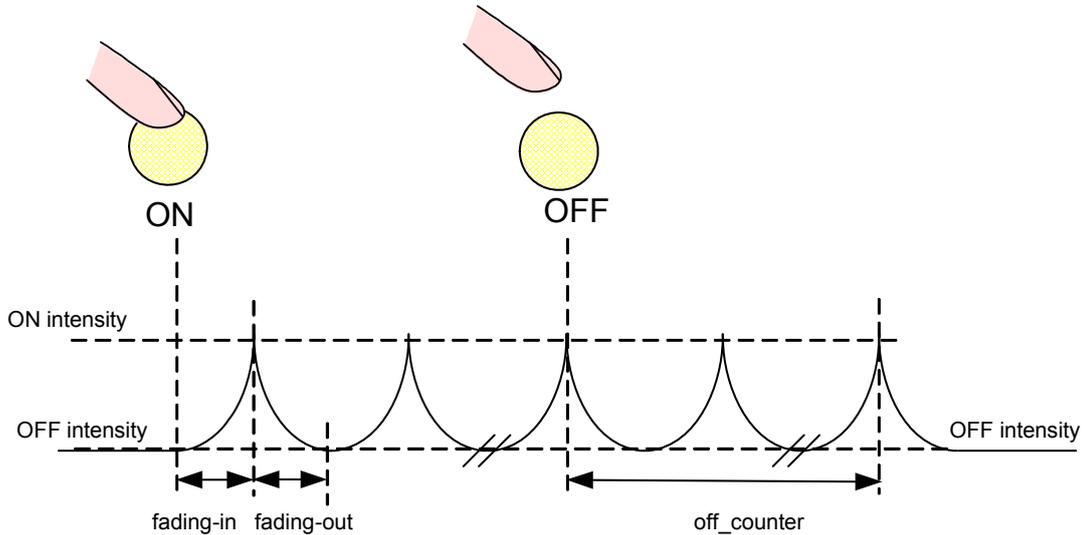


Figure 37 Continuous Fading Mode

iii) Repeat Fading Mode

The GPO pin fades in and fades out for a configurable number of times (see register GpioOffDelay) when the associated button is touched. When the counter reaches its maximum value, the fading in/out stops regardless of whether the button has been released (case 1) or not (case 2). This is shown in Figure 38.

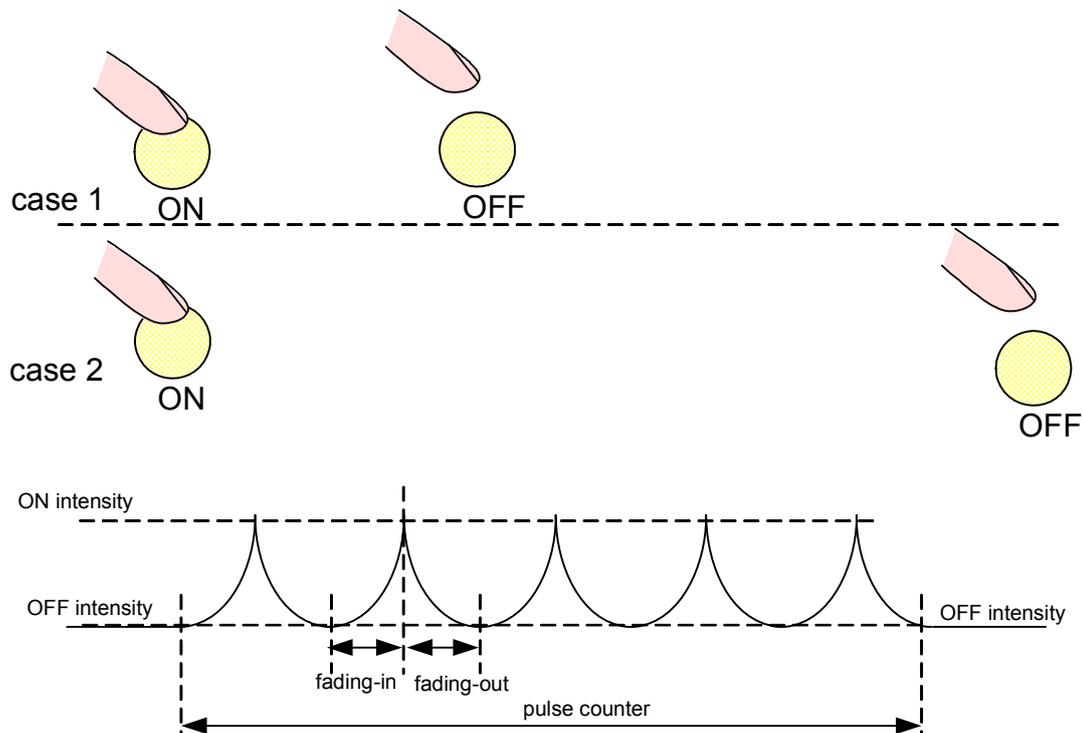


Figure 38 Repeat Fading Mode

3.18.5 Intensity index vs PWM pulse width

Tables below are used to convert all intensity indexes parameters GpioIntensityOff, GpioIntensityOn and GppIntensity but also to generate fading in GPO mode

During fading in(out), the index is automatically incremented(decremented) at every Inc(Dec)Time x Inc(Dec)Factor until it reaches the programmed GpioIntensityOn(Off) value.

| Index | Lin/Log |
|-------|---------|-------|---------|-------|---------|-------|---------|-------|---------|-------|---------|-------|---------|-------|---------|
| 0 | 0/0 | 32 | 33/5 | 64 | 65/12 | 96 | 97/26 | 128 | 129/48 | 160 | 161/81 | 192 | 193/125 | 224 | 225/184 |
| 1 | 2/0 | 33 | 34/5 | 65 | 66/13 | 97 | 98/27 | 129 | 130/49 | 161 | 162/82 | 193 | 194/127 | 225 | 226/186 |
| 2 | 3/0 | 34 | 35/5 | 66 | 67/13 | 98 | 99/27 | 130 | 131/50 | 162 | 163/83 | 194 | 195/129 | 226 | 227/188 |
| 3 | 4/0 | 35 | 36/5 | 67 | 68/13 | 99 | 100/28 | 131 | 132/51 | 163 | 164/84 | 195 | 196/130 | 227 | 228/190 |
| 4 | 5/0 | 36 | 37/5 | 68 | 69/14 | 100 | 101/29 | 132 | 133/52 | 164 | 165/86 | 196 | 197/132 | 228 | 229/192 |
| 5 | 6/2 | 37 | 38/6 | 69 | 70/14 | 101 | 102/29 | 133 | 134/53 | 165 | 166/87 | 197 | 198/133 | 229 | 230/194 |
| 6 | 7/2 | 38 | 39/6 | 70 | 71/14 | 102 | 103/30 | 134 | 135/54 | 166 | 167/88 | 198 | 199/135 | 230 | 231/197 |
| 7 | 8/2 | 39 | 40/6 | 71 | 72/15 | 103 | 104/30 | 135 | 136/55 | 167 | 168/89 | 199 | 200/137 | 231 | 232/199 |
| 8 | 9/2 | 40 | 41/6 | 72 | 73/15 | 104 | 105/31 | 136 | 137/55 | 168 | 169/91 | 200 | 201/139 | 232 | 233/201 |
| 9 | 10/2 | 41 | 42/6 | 73 | 74/15 | 105 | 106/32 | 137 | 138/56 | 169 | 170/92 | 201 | 202/140 | 233 | 234/203 |
| 10 | 11/2 | 42 | 43/7 | 74 | 75/16 | 106 | 107/32 | 138 | 139/57 | 170 | 171/93 | 202 | 203/142 | 234 | 235/205 |
| 11 | 12/2 | 43 | 44/7 | 75 | 76/16 | 107 | 108/33 | 139 | 140/58 | 171 | 172/95 | 203 | 204/144 | 235 | 236/208 |
| 12 | 13/2 | 44 | 45/7 | 76 | 77/16 | 108 | 109/33 | 140 | 141/59 | 172 | 173/96 | 204 | 205/146 | 236 | 237/210 |
| 13 | 14/2 | 45 | 46/7 | 77 | 78/17 | 109 | 110/34 | 141 | 142/60 | 173 | 174/97 | 205 | 206/147 | 237 | 238/212 |
| 14 | 15/3 | 46 | 47/7 | 78 | 79/17 | 110 | 111/35 | 142 | 143/61 | 174 | 175/99 | 206 | 207/149 | 238 | 239/215 |
| 15 | 16/3 | 47 | 48/8 | 79 | 80/18 | 111 | 112/35 | 143 | 144/62 | 175 | 176/100 | 207 | 208/151 | 239 | 240/217 |
| 16 | 17/3 | 48 | 49/8 | 80 | 81/18 | 112 | 113/36 | 144 | 145/63 | 176 | 177/101 | 208 | 209/153 | 240 | 241/219 |
| 17 | 18/3 | 49 | 50/8 | 81 | 82/19 | 113 | 114/37 | 145 | 146/64 | 177 | 178/103 | 209 | 210/155 | 241 | 242/221 |
| 18 | 19/3 | 50 | 51/8 | 82 | 83/19 | 114 | 115/38 | 146 | 147/65 | 178 | 179/104 | 210 | 211/156 | 242 | 243/224 |
| 19 | 20/3 | 51 | 52/9 | 83 | 84/20 | 115 | 116/38 | 147 | 148/66 | 179 | 180/106 | 211 | 212/158 | 243 | 244/226 |
| 20 | 21/3 | 52 | 53/9 | 84 | 85/20 | 116 | 117/39 | 148 | 149/67 | 180 | 181/107 | 212 | 213/160 | 244 | 245/229 |
| 21 | 22/3 | 53 | 54/9 | 85 | 86/21 | 117 | 118/40 | 149 | 150/68 | 181 | 182/109 | 213 | 214/162 | 245 | 246/231 |
| 22 | 23/3 | 54 | 55/9 | 86 | 87/21 | 118 | 119/40 | 150 | 151/69 | 182 | 183/110 | 214 | 215/164 | 246 | 247/233 |
| 23 | 24/4 | 55 | 56/10 | 87 | 88/22 | 119 | 120/41 | 151 | 152/71 | 183 | 184/111 | 215 | 216/166 | 247 | 248/236 |
| 24 | 25/4 | 56 | 57/10 | 88 | 89/22 | 120 | 121/42 | 152 | 153/72 | 184 | 185/113 | 216 | 217/168 | 248 | 249/238 |
| 25 | 26/4 | 57 | 58/10 | 89 | 90/23 | 121 | 122/43 | 153 | 154/73 | 185 | 186/114 | 217 | 218/170 | 249 | 250/241 |
| 26 | 27/4 | 58 | 59/10 | 90 | 91/23 | 122 | 123/44 | 154 | 155/74 | 186 | 187/116 | 218 | 219/172 | 250 | 251/243 |
| 27 | 28/4 | 59 | 60/11 | 91 | 92/24 | 123 | 124/44 | 155 | 156/75 | 187 | 188/117 | 219 | 220/174 | 251 | 252/246 |
| 28 | 29/4 | 60 | 61/11 | 92 | 93/24 | 124 | 125/45 | 156 | 157/76 | 188 | 189/119 | 220 | 221/176 | 252 | 253/248 |
| 29 | 30/4 | 61 | 62/11 | 93 | 94/25 | 125 | 126/46 | 157 | 158/77 | 189 | 190/121 | 221 | 222/178 | 253 | 254/251 |
| 30 | 31/4 | 62 | 63/12 | 94 | 95/25 | 126 | 127/47 | 158 | 159/78 | 190 | 191/122 | 222 | 223/180 | 254 | 255/253 |
| 31 | 32/5 | 63 | 64/12 | 95 | 96/26 | 127 | 128/48 | 159 | 160/80 | 191 | 192/124 | 223 | 224/182 | 255 | 256/256 |

Table 11 Intensity index vs. PWM pulse width (normal polarity)

| Index | Lin/Log |
|-------|---------|-------|---------|-------|---------|-------|---------|-------|---------|-------|---------|-------|---------|-------|---------|
| 0 | 256/256 | 32 | 224/251 | 64 | 192/244 | 96 | 160/230 | 128 | 128/208 | 160 | 96/175 | 192 | 64/131 | 224 | 32/72 |
| 1 | 255/256 | 33 | 223/251 | 65 | 191/243 | 97 | 159/229 | 129 | 127/207 | 161 | 95/174 | 193 | 63/129 | 225 | 31/70 |
| 2 | 254/256 | 34 | 222/251 | 66 | 190/243 | 98 | 158/229 | 130 | 126/206 | 162 | 94/173 | 194 | 62/127 | 226 | 30/68 |
| 3 | 253/256 | 35 | 221/251 | 67 | 189/243 | 99 | 157/228 | 131 | 125/205 | 163 | 93/172 | 195 | 61/126 | 227 | 29/66 |
| 4 | 252/256 | 36 | 220/251 | 68 | 188/242 | 100 | 156/227 | 132 | 124/204 | 164 | 92/170 | 196 | 60/124 | 228 | 28/64 |
| 5 | 251/254 | 37 | 219/250 | 69 | 187/242 | 101 | 155/227 | 133 | 123/203 | 165 | 91/169 | 197 | 59/123 | 229 | 27/62 |
| 6 | 250/254 | 38 | 218/250 | 70 | 186/242 | 102 | 154/226 | 134 | 122/202 | 166 | 90/168 | 198 | 58/121 | 230 | 26/59 |
| 7 | 249/254 | 39 | 217/250 | 71 | 185/241 | 103 | 153/226 | 135 | 121/201 | 167 | 89/167 | 199 | 57/119 | 231 | 25/57 |
| 8 | 248/254 | 40 | 216/250 | 72 | 184/241 | 104 | 152/225 | 136 | 120/201 | 168 | 88/165 | 200 | 56/117 | 232 | 24/55 |
| 9 | 247/254 | 41 | 215/250 | 73 | 183/241 | 105 | 151/224 | 137 | 119/200 | 169 | 87/164 | 201 | 55/116 | 233 | 23/53 |
| 10 | 246/254 | 42 | 214/249 | 74 | 182/240 | 106 | 150/224 | 138 | 118/199 | 170 | 86/163 | 202 | 54/114 | 234 | 22/50 |
| 11 | 245/254 | 43 | 213/249 | 75 | 181/240 | 107 | 149/223 | 139 | 117/198 | 171 | 85/161 | 203 | 53/112 | 235 | 21/48 |
| 12 | 244/254 | 44 | 212/249 | 76 | 180/240 | 108 | 148/223 | 140 | 116/197 | 172 | 84/160 | 204 | 52/110 | 236 | 20/46 |
| 13 | 243/254 | 45 | 211/249 | 77 | 179/239 | 109 | 147/222 | 141 | 115/196 | 173 | 83/159 | 205 | 51/109 | 237 | 19/44 |
| 14 | 242/253 | 46 | 210/249 | 78 | 178/239 | 110 | 146/221 | 142 | 114/195 | 174 | 82/157 | 206 | 50/107 | 238 | 18/41 |
| 15 | 241/253 | 47 | 209/248 | 79 | 177/238 | 111 | 145/221 | 143 | 113/194 | 175 | 81/156 | 207 | 49/105 | 239 | 17/39 |
| 16 | 240/253 | 48 | 208/248 | 80 | 176/238 | 112 | 144/220 | 144 | 112/193 | 176 | 80/155 | 208 | 48/103 | 240 | 16/37 |
| 17 | 239/253 | 49 | 207/248 | 81 | 175/237 | 113 | 143/219 | 145 | 111/192 | 177 | 79/153 | 209 | 47/101 | 241 | 15/35 |
| 18 | 238/253 | 50 | 206/248 | 82 | 174/237 | 114 | 142/218 | 146 | 110/191 | 178 | 78/152 | 210 | 46/100 | 242 | 14/32 |
| 19 | 237/253 | 51 | 205/247 | 83 | 173/236 | 115 | 141/218 | 147 | 109/190 | 179 | 77/150 | 211 | 45/98 | 243 | 13/30 |
| 20 | 236/253 | 52 | 204/247 | 84 | 172/236 | 116 | 140/217 | 148 | 108/189 | 180 | 76/149 | 212 | 44/96 | 244 | 12/27 |
| 21 | 235/253 | 53 | 203/247 | 85 | 171/235 | 117 | 139/216 | 149 | 107/188 | 181 | 75/147 | 213 | 43/94 | 245 | 11/25 |
| 22 | 234/253 | 54 | 202/247 | 86 | 170/235 | 118 | 138/216 | 150 | 106/187 | 182 | 74/146 | 214 | 42/92 | 246 | 10/23 |
| 23 | 233/252 | 55 | 201/246 | 87 | 169/234 | 119 | 137/215 | 151 | 105/185 | 183 | 73/145 | 215 | 41/90 | 247 | 9/20 |
| 24 | 232/252 | 56 | 200/246 | 88 | 168/234 | 120 | 136/214 | 152 | 104/184 | 184 | 72/143 | 216 | 40/88 | 248 | 8/18 |
| 25 | 231/252 | 57 | 199/246 | 89 | 167/233 | 121 | 135/213 | 153 | 103/183 | 185 | 71/142 | 217 | 39/86 | 249 | 7/15 |
| 26 | 230/252 | 58 | 198/246 | 90 | 166/233 | 122 | 134/212 | 154 | 102/182 | 186 | 70/140 | 218 | 38/84 | 250 | 6/13 |
| 27 | 229/252 | 59 | 197/245 | 91 | 165/232 | 123 | 133/212 | 155 | 101/181 | 187 | 69/139 | 219 | 37/82 | 251 | 5/10 |
| 28 | 228/252 | 60 | 196/245 | 92 | 164/232 | 124 | 132/211 | 156 | 100/180 | 188 | 68/137 | 220 | 36/80 | 252 | 4/8 |
| 29 | 227/252 | 61 | 195/245 | 93 | 163/231 | 125 | 131/210 | 157 | 99/179 | 189 | 67/135 | 221 | 35/78 | 253 | 3/5 |
| 30 | 226/252 | 62 | 194/244 | 94 | 162/231 | 126 | 130/209 | 158 | 98/178 | 190 | 66/134 | 222 | 34/76 | 254 | 2/3 |
| 31 | 225/251 | 63 | 193/244 | 95 | 161/230 | 127 | 129/208 | 159 | 97/176 | 191 | 65/132 | 223 | 33/74 | 255 | 0/0 |

Table 12 Intensity index vs. PWM pulse width (inverted polarity)

Recommended/default settings are inverted polarity (to take advantage from high sink current capability) and logarithmic mode (due to the non-linear response of the human eye).

4 PIN DESCRIPTIONS

4.1 Introduction

This chapter describes briefly the pins of the SX8660, the way the pins are protected, if the pins are analog, digital, require pull up or pull down resistors and show control signals if these are available.

4.2 ASI pins

CAP0, CAP1,...,CAP7

The capacitance sensor pins (CAP0, CAP1,...,CAP7) are connected directly to the ASI circuitry which converts the sensed capacitance into digital values.

The capacitance sensor pins which are not used should be left open.

The enabled CAP pins need be connected directly to the sensors without significant resistance (typical below some ohms, connection vias are allowed).

The capacitance sensor pins are protected to VANA and GROUND.

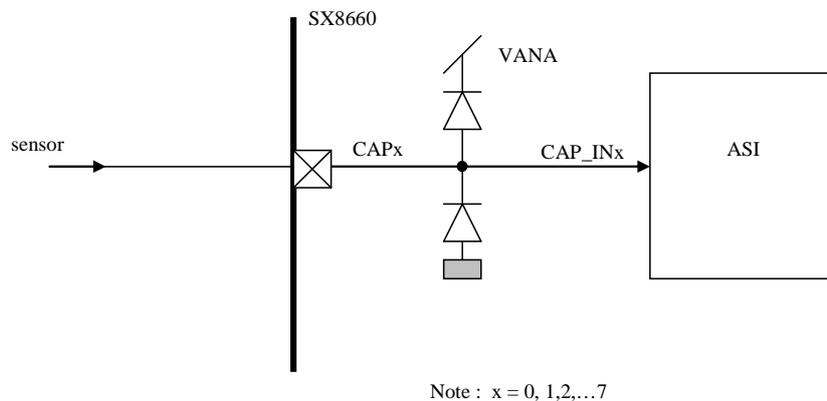


Figure 39 shows the simplified diagram of the CAP0, CAP1,...CAP7 pins.

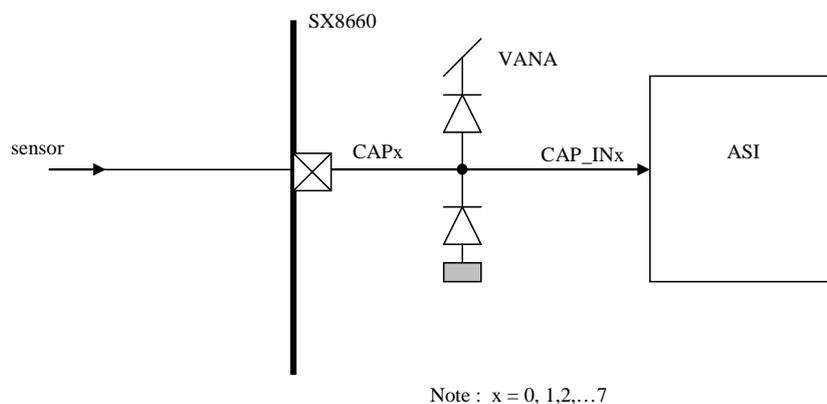


Figure 39 Simplified diagram of CAP0, CAP1,...,CAP7

CN, CP

The CN and the CP pins are connected to the ASI circuitry. A 1nF sampling capacitor between CP and CN needs to be placed as close as possible to the SX8660.

The CN and CP are protected to VANA and GROUND.

Figure 40 shows the simplified diagram of the CN and CP pins.

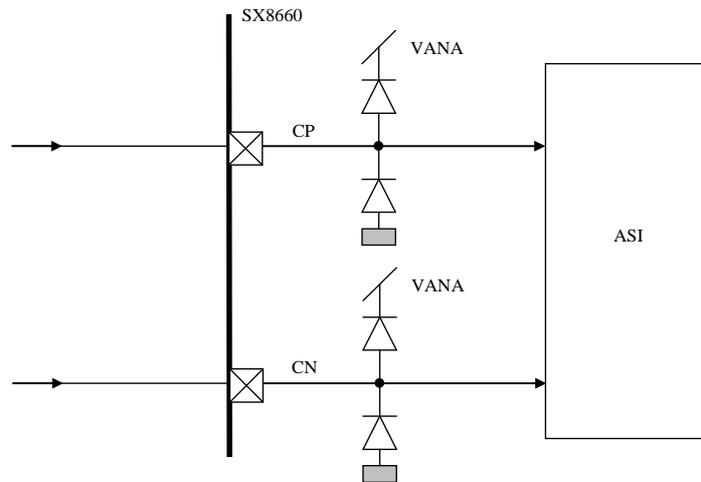


Figure 40 Simplified diagram of CN and CP

4.3 Host interface pins

The host interface consists of the interrupt pin INTB, a reset pin RESETB and the standard I2C pins: SCL and SDA.

INTB

The INTB pin is an open drain output that requires an external pull-up resistor (1..10 kOhm). The INTB pin is protected to VDD using dedicated devices. The INTB pin has diode protected to GROUND.

Figure 41 shows a simplified diagram of the INTB pin.

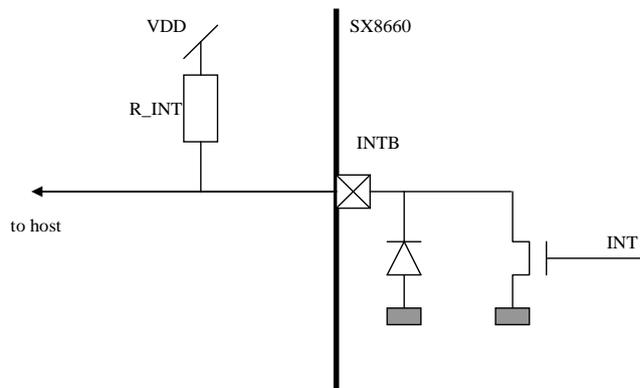


Figure 41 Simplified diagram of INTB

SCL

The SCL pin is a high impedance input pin. The SCL pin is protected to VDD, using dedicated devices, in order to conform to standard I2C slave specifications. The SCL pin has diode protected to GROUND. An external pull-up resistor (1..10 kOhm) is required on this pin.

Figure 42 shows the simplified diagram of the SCL pin.

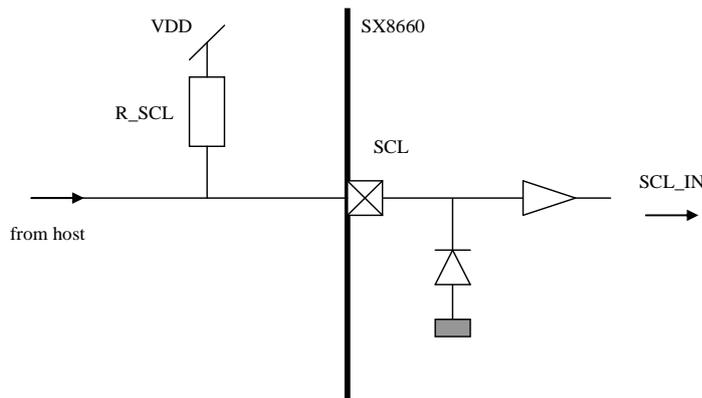


Figure 42 Simplified diagram of SCL

SDA

SDA is an IO pin that can be used as an open drain output pin with external pull-up resistor or as a high impedance input pin. The SDA IO pin is protected to VDD, using dedicated devices, in order to conform to standard I2C slave specifications. The SDA pin has diode protected to GROUND.

An external pull-up resistor (1..10 kOhm) is required on this pin.

Figure 43 shows the simplified diagram of the SDA pin.

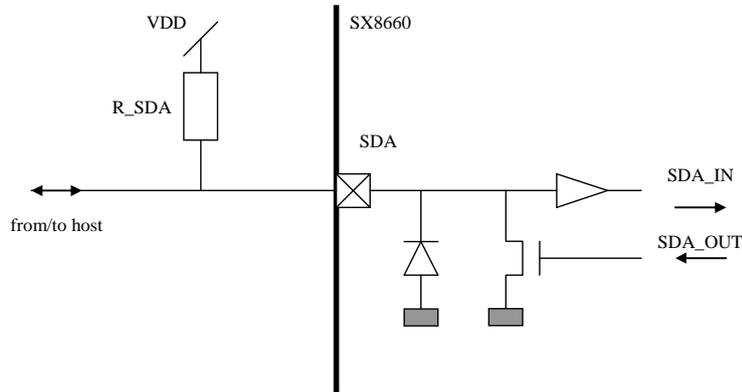


Figure 43 Simplified diagram of SDA

RESETB

The RESETB pin is a high impedance input pin. The RESETB pin is protected to VDD using dedicated devices. The RESETB pin has diode protected to GROUND.

Figure 44 shows the simplified diagram of the RESETB pin controlled by the host.

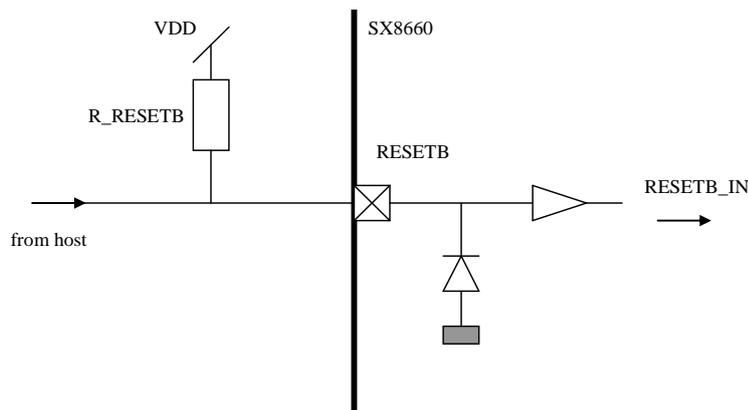


Figure 44 Simplified diagram of RESETB controlled by host

Figure 45 shows the RESETB without host control.

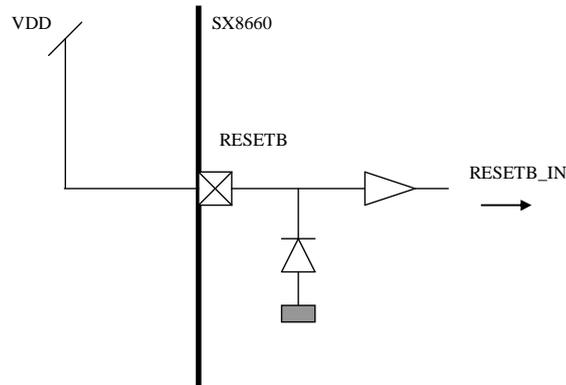


Figure 45 Simplified diagram of RESETB without host control

4.4 Power management pins

The power management pins consist of the Power, Ground and Regulator pins.

VDD

VDD is a power pin and is the main power supply for the SX8660.
 VDD has protection to GROUND.

Figure 46 shows a simplified diagram of the VDD pin.

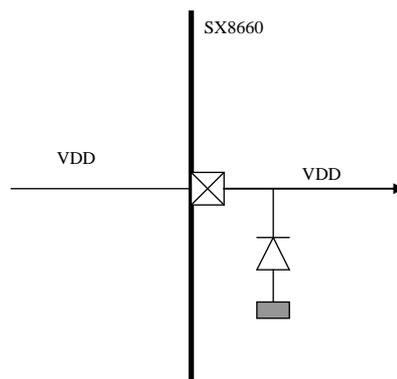


Figure 46 Simplified diagram of VDD

GND

The SX8660 has four ground pins all named GND. These pins and the package center pad need to be connected to ground potential.

The GND has protection to VDD.

Figure 47 shows a simplified diagram of the GND pin.

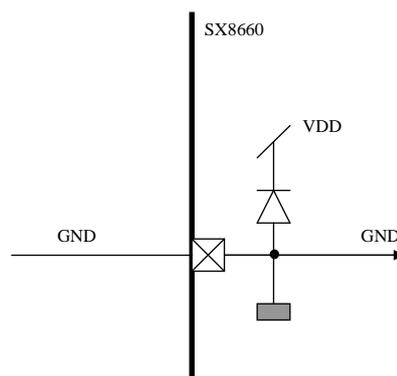


Figure 47 Simplified diagram of GND

VANA, VDIG

The SX8660 has on-chip regulators for internal use (pins VANA and VDIG).

VANA and VDIG have protection to VDD and to GND.

The output of the regulators needs to be de-coupled with a small 100nF capacitor to ground.

Figure 48 shows a simplified diagram of the VANA and VDIG pin.

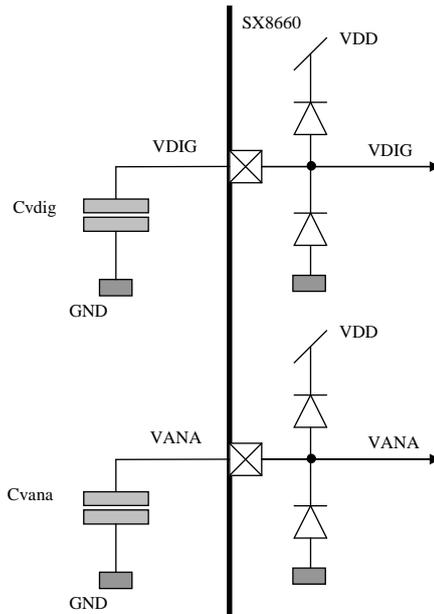


Figure 48 Simplified diagram of VANA and VDIG

4.5 General purpose IO pins

The SX8660 has 8 General purpose input/output (GPIO) pins.

All the GPIO pins have protection to VDD and GND.

The GPIO pins can be configured as GPI, GPO, SPO or GPP.

Figure 49 shows a simplified diagram of the GPIO pins.

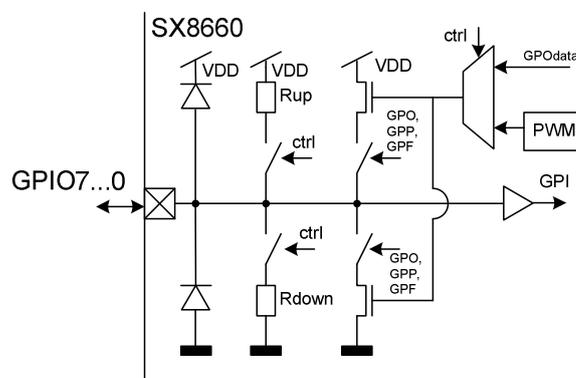


Figure 49 Simplified diagram of GPIO pins

5 DETAILED CONFIGURATION DESCRIPTIONS

5.1 Introduction

The SX8660 configuration parameters are taken from the QSM or the NVM and loaded into the SPM as explained in the chapter 'functional description'.

This chapter describes the details of the configuration parameters of the SX8660.

The SPM is split by functionality into 5 configuration sections:

- General section: operating modes,
- Capacitive Sensors section: related to lower level capacitive sensing,
- Button: related to the conversion from sensor data towards button information,
- Mapping: related to mapping of button information towards GPIO pins,
- GPIO: related to the setup of the GPIO pins.

The total address space of the SPM and the NVM is 128 bytes, from address 0x00 to address 0x7F.

Two types of memory addresses, data are accessible to the user.

- 'application data': Application dependent data that need to be configured by the user.
- 'reserved': Data that need to be maintained by the user to the QSM default values (i.e. when NVM is burned).

The Table 13 and Table 14 resume the complete SPM address space and show the 'application data' and 'reserved' addresses, the functional split and the default values (loaded from the QSM).

| Address | Name | default QSM value | Address | Name | default QSM value | | |
|---------|--------------------|-------------------|---------|---------------------|-------------------------------|------------------------|------|
| 0x00 | Reserved | 0xxx | 0x20 | Button | BtnCfg | 0x70 | |
| 0x01 | Reserved | 0xxx | 0x21 | | BtnAvgThresh | 0x50 | |
| 0x02 | Reserved | 0x40 | 0x22 | | BtnCompNegThresh | 0x50 | |
| 0x03 | Reserved | 0xxx | 0x23 | | BtnCompNegCntMax | 0x01 | |
| 0x04 | General | I2CAddress | 0x2B | | 0x24 | BtnHysteresis | 0x0A |
| 0x05 | | ActiveScanPeriod | 0x02 | | 0x25 | BtnStuckAtTimeout | 0x00 |
| 0x06 | | DozeScanPeriod | 0x0D | | 0x26 | BtnStrongestHysteresis | 0x80 |
| 0x07 | | PassiveTimer | 0x00 | | 0x27 | BtnLongPressTimer | 0x00 |
| 0x08 | | Reserved | 0x00 | | 0x28 | Reserved | 0x00 |
| 0x09 | Capacitive Sensors | CapModeMisc | 0x01 | | 0x29 | Reserved | 0x00 |
| 0x0A | | Reserved | 0x00 | 0x2A | Reserved | 0xFF | |
| 0x0B | | CapMode7_4 | 0x55 | 0x2B | Analog Output Interface (AOI) | AoiCfg | 0x01 |
| 0x0C | | CapMode3_0 | 0x55 | 0x2C | | AoiBtnMapMsb | 0x55 |
| 0x0D | | CapSensitivity0_1 | 0x30 | 0x2D | | AoiBtnMapLsb | 0x55 |
| 0x0E | | CapSensitivity2_3 | 0x00 | 0x2E | | AoiLevelBtn0 | 0x17 |
| 0x0F | | CapSensitivity4_5 | 0x00 | 0x2F | | AoiLevelBtn1 | 0x2E |
| 0x10 | | CapSensitivity6_7 | 0x00 | 0x30 | | AoiLevelBtn2 | 0x45 |
| 0x11 | | Reserved | 0x00 | 0x31 | | AoiLevelBtn3 | 0x5D |
| 0x12 | | Reserved | 0x00 | 0x32 | | AoiLevelBtn4 | 0x74 |
| 0x13 | | CapThresh0 | 0xA0 | 0x33 | | AoiLevelBtn5 | 0x8B |
| 0x14 | | CapThresh1 | 0xA0 | 0x34 | | AoiLevelBtn6 | 0xA3 |
| 0x15 | | CapThresh2 | 0xA0 | 0x35 | AoiLevelBtn7 | 0xBA | |
| 0x16 | | CapThresh3 | 0xA0 | 0x36 | AoiLevelIdle | 0xFF | |
| 0x17 | | CapThresh4 | 0xA0 | 0x37 | Buzzer | BuzzerCfg | 0xA4 |
| 0x18 | | CapThresh5 | 0xA0 | 0x38 | | BuzzerFreqPhase1 | 0x40 |
| 0x19 | | CapThresh6 | 0xA0 | 0x39 | | BuzzerFreqPhase2 | 0x20 |
| 0x1A | | CapThresh7 | 0xA0 | 0x3A | Reserved | 0x00 | |
| 0x1B | | Reserved | 0x00 | 0x3B | Mapping | MapAutoLight0 | 0x76 |
| 0x1C | Reserved | 0x00 | 0x3C | MapAutoLight1 | | 0x54 | |
| 0x1D | Reserved | 0x00 | 0x3D | MapAutoLight2 | | 0x32 | |
| 0x1E | Reserved | 0x00 | 0x3E | MapAutoLight3 | | 0x10 | |
| 0x1F | CapPerComp | 0x00 | 0x3F | MapAutoLightGrp0Msb | | 0x00 | |

Table 13 SPM address map: 0x00...0x3F

Note

- '0xxx': write protected data

| Address | Name | default QSM value | Address | Name | default QSM value |
|---------|----------------|---------------------|----------|---------------------|-------------------|
| 0x40 | Map | MapAutoLightGrp0Lsb | 0x40 | Gpio | GpioDecTime7_6 |
| 0x41 | | MapAutoLightGrp1Msb | 0x41 | | GpioDecTime5_4 |
| 0x42 | | MapAutoLightGrp1Lsb | 0x42 | | GpioDecTime3_2 |
| 0x43 | Gpio | GpioMode7_4 | 0x43 | | GpioDecTime1_0 |
| 0x44 | | GpioMode3_0 | 0x44 | | GpioOffDelay7_6 |
| 0x45 | | GpioIntensityOn0 | 0x45 | | GpioOffDelay5_4 |
| 0x46 | | GpioIntensityOn1 | 0x46 | | GpioOffDelay3_2 |
| 0x47 | | GpioIntensityOn2 | 0x47 | | GpioOffDelay1_0 |
| 0x48 | | GpioIntensityOn3 | 0x48 | | GpioPullUpDown7_4 |
| 0x49 | | GpioIntensityOn4 | 0x49 | | GpioPullUpDown3_0 |
| 0x4A | | GpioIntensityOn5 | 0x4A | | GpioInterrupt7_4 |
| 0x4B | | GpioIntensityOn6 | 0x4B | | GpioInterrupt3_0 |
| 0x4C | | GpioIntensityOn7 | 0x4C | | GpioDebounce |
| 0x4D | | GpioIntensityOff0 | 0x4D | | GpioFadingMode7_4 |
| 0x4E | | GpioIntensityOff1 | 0x4E | | GpioFadingMode3_0 |
| 0x4F | | GpioIntensityOff2 | 0x4F | | Reserved |
| 0x50 | | GpioIntensityOff3 | 0x50 | Reserved | |
| 0x51 | | GpioIntensityOff4 | 0x51 | Reserved | |
| 0x52 | | GpioIntensityOff5 | 0x52 | Reserved | |
| 0x53 | | GpioIntensityOff6 | 0x53 | Reserved | |
| 0x54 | | GpioIntensityOff7 | 0x54 | Reserved | |
| 0x55 | | Reserved | 0x55 | Reserved | |
| 0x56 | | GpioOutPwrUp | 0x56 | Reserved | |
| 0x57 | | GpioAutoLight | 0x57 | Reserved | |
| 0x58 | | GpoPolarity | 0x58 | Reserved | |
| 0x59 | | GpioFunction | 0x59 | Reserved | |
| 0x5A | | GpioIncFactor | 0x5A | Reserved | |
| 0x5B | GpioDecFactor | 0x5B | Reserved | | |
| 0x5C | GpioIncTime7_6 | 0x5C | Reserved | | |
| 0x5D | GpioIncTime5_4 | 0x5D | Reserved | | |
| 0x5E | GpioIncTime3_2 | 0x5E | Reserved | | |
| 0x5F | GpioIncTime1_0 | 0x5F | Reserved | | |
| | | | 0x7F | SpmCrc ¹ | |

Table 14 SPM address map: 0x40...0x7F

 Note¹

- SpmCrc: CRC depending on SPM content, updated in Active or Doze mode.

5.2 General Parameters

| General Parameters | | | |
|--------------------|------------------|------|---|
| Address | Name | Bits | Description |
| 0x04 | I2CAddress | 7 | Reserved |
| | | 6:0 | Defines the I2C address (default 0x2B). The I2C address will be active after a reset. |
| 0x05 | ActiveScanPeriod | 7:0 | Active Mode Scan Period (Figure 7) 0x00: Reserved 0x01: 15ms 0x02: 30ms (default) ... 0xFF: 255 x 15ms |
| 0x06 | DozeScanPeriod | 7:0 | Doze Mode Scan Period (Figure 7) 0x00: Reserved 0x01: 15ms ... 0x0D: 195ms (default) ... 0xFF: 255 x 15ms |
| 0x07 | PassiveTimer | 7:0 | Passive Timer on Button Information (Figure 8) 0x00: Off (default) 0x01: 1 second ... 0xFF: 255 seconds |
| 0x08 | Reserved | 7:0 | Reserved |

Table 15 General Parameters

5.3 Capacitive Sensors Parameters

| Capacitive Sensors Parameters | | | | | | |
|-------------------------------|-------------------|------|---------------------------------------|---|---------|--------|
| Address | Name | Bits | Description | | | |
| 0x09 | CapModeMisc | 7:3 | Reserved | | | |
| | | 2 | IndividualSensitivity | Defines common sensitivity for all sensors or individual sensor sensitivity. 0: Common settings (CapSensitivity0_1[7:4]) 1: Individual CAP sensitivity settings (CapSensitivityx_x) | | |
| | | 1:0 | Reserved | Reserved: '01' | | |
| 0x0A | Reserved | 7:0 | Reserved | | | |
| 0x0B | CapMode7_4 | 7:6 | CAP7 Mode | Defines the mode of the CAP pin. 00: Disabled 01: Button 10: Reserved 11: Reserved | Default | Button |
| | | 5:4 | CAP6 Mode | | | Button |
| | | 3:2 | CAP5 Mode | | | Button |
| | | 1:0 | CAP4 Mode | | | Button |
| 0x0C | CapMode3_0 | 7:6 | CAP3 Mode | | | Button |
| | | 5:4 | CAP2 Mode | | | Button |
| | | 3:2 | CAP1 Mode | | | Button |
| | | 1:0 | CAP0 Mode | | | Button |
| 0x0D | CapSensitivity0_1 | 7:4 | CAP0 Sensitivity - Common Sensitivity | Defines the sensitivity. 0x0: Minimum 0x1: 1 ... 0x7: Maximum 0x8..0xF: Reserved | | |
| | | 3:0 | CAP1 Sensitivity | | | |
| 0x0E | CapSensitivity2_3 | 7:4 | CAP2 Sensitivity | | | |
| | | 3:0 | CAP3 Sensitivity | | | |
| 0x0F | CapSensitivity4_5 | 7:4 | CAP4 Sensitivity | CapSensitivity0_1 default: 0x30 CapSensitivity2_3 default: 0x00 CapSensitivity4_5 default: 0x00 CapSensitivity6_7 default: 0x00 | | |
| | | 3:0 | CAP5 Sensitivity | | | |
| 0x10 | CapSensitivity6_7 | 7:4 | CAP6 Sensitivity | | | |
| | | 3:0 | CAP7 Sensitivity | | | |
| 0x11 | Reserved | 7:0 | Reserved | | | |
| 0x12 | Reserved | 7:0 | Reserved | | | |
| 0x13 | CapThresh0 | 7:0 | CAP0 Touch Threshold | Defines the Touch Threshold ticks. 0x00: 0, 0x01: 4, ... 0xA0: 640 (default), ... 0xFF: 1020 | | |
| 0x14 | CapThresh1 | 7:0 | CAP1 Touch Threshold | | | |
| 0x15 | CapThresh2 | 7:0 | CAP2 Touch Threshold | | | |
| 0x16 | CapThresh3 | 7:0 | CAP3 Touch Threshold | | | |
| 0x17 | CapThresh4 | 7:0 | CAP4 Touch Threshold | | | |
| 0x18 | CapThresh5 | 7:0 | CAP5 Touch Threshold | | | |
| 0x19 | CapThresh6 | 7:0 | CAP6 Touch Threshold | | | |
| 0x1A | CapThresh7 | 7:0 | CAP7 Touch Threshold | | | |
| 0x1B | Reserved | 7:0 | Reserved | | | |
| 0x1C | Reserved | 7:0 | Reserved | | | |
| 0x1D | Reserved | 7:0 | Reserved | | | |

| Capacitive Sensors Parameters | | | |
|-------------------------------|------------|------|--|
| Address | Name | Bits | Description |
| 0x1E | Reserved | 7:0 | Reserved |
| 0x1F | CapPerComp | 7:4 | Reserved |
| | | 3:0 | Periodic Offset Compensation Defines the periodic offset compensation. 0x0: OFF (default) 0x1: 1 second 0x2: 2 seconds ... 0x7: 7 seconds 0x8: 16 seconds 0x9: 18 seconds ... 0xE: 28 seconds 0xF: 60 seconds |

Table 16 Capacitive Sensors Parameters
CapModeMisc:

By default the ASI is using a common sensitivity for all capacitive sensors as in the usual case overlay material and sensors sizes are about equal. The register bits CapSensitivity0_1[7:4] determine the sensitivity for all sensors in common sensitivity mode.

In special applications it might be required to have a different, individual, sensitivity for each CAP pin. This can be obtained by setting bit CapModeMisc[2]. The individual sensitivity mode results in longer sensing periods than required in common sensitivity mode.

CapMode7_4, CapMode3_0:

The CAP pins can be set as a button or disabled depending on the application.

| | minimum | default | maximum |
|---------|---------|---------|---------|
| buttons | one | eight | eight |

Table 17 Possible CAP pin modes

Buttons and disabled CAP pins can be attributed freely (examples in Figure 50).

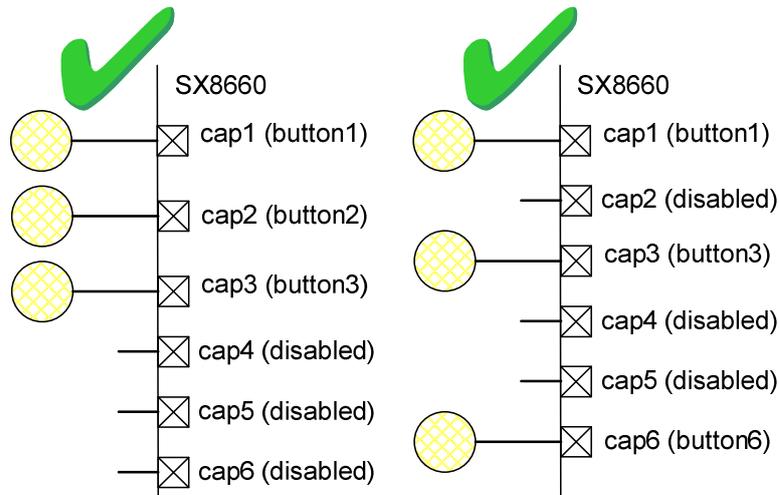


Figure 50 Button examples

CapSensitivity0_1, CapSensitivity2_3, CapSensitivity4_5, CapSensitivity6_7:

The sensitivity of the sensors can be set between 8 values. The higher the sensitivity is set the larger the value of the ticks will be.

The minimum sensitivity can be used for thin overlay materials and large sensors, while the maximum sensitivity is required for thicker overlay and smaller sensors.

The required sensitivity needs to be determined during a product development phase. Too low sensitivity settings result in missing touches. Too high sensitivity settings will result in fault detection of fingers hovering above the sensors.

The sensitivity is identical for all sensors in common sensitivity mode using the bits CapSensitivity0_1[7:4] and can be set individually using register CapModeMisc[2].

The maximum number of ticks that can be obtained depends on the selected sensitivity as illustrated in Table 18.

| Sensitivity | Approximate Maximum Tick Level |
|-------------|--------------------------------|
| 0 | 1000 |
| 1 | 2000 |
| 2 | 3000 |
| 3 | 4000 |
| 4 | 5000 |
| 5 | 6000 |
| 6 | 7000 |
| 7 | 8000 |

Table 18 ASI Maximum Tick Levels

CapThresh0, CapThresh1, CapThresh2, CapThresh3, CapThresh4, CapThresh5, CapThresh6, CapThresh7:

For each CAP pin a threshold level can be set individually.
The threshold levels are used by the SX8660 for making touch and release decisions.
The details are explained in the sections for buttons.

CapPerComp:

The SX8660 offers a periodic offset compensation for applications which are subject to substantial environmental changes. The periodic offset compensation is done at a defined interval and only if buttons are released.

5.4 Button Parameters

| Button Parameters | | | |
|-------------------|-------------------|------|---|
| Address | Name | Bits | Description |
| 0x20 | BtnCfg | 7:6 | Defines the buttons events reporting method on I2C, AOI, GPO, Buzzer 00: All* 01: First (default) 10: Strongest 11: Reserved *The AOI-A/AOI-B are forced to AoiLevelIdle. |
| | | 5:4 | Defines the buttons interrupt (for all buttons) 00 : Interrupts masked 01 : Triggered on Touch 10 : Triggered on Release 11 : Triggered on Touch and Release (default) |
| | | 3 | Defines the number of samples at the scan period for determining a release. 00 : no debounce, use incoming sample (default) 01 : 2 samples debounce 10 : 3 samples debounce 11 : 4 samples debounce |
| | | 2:0 | Defines the number of samples at the scan period for determining a touch. 00 : no debounce, use incoming sample (default) 01 : 2 samples debounce 10 : 3 samples debounce 11 : 4 samples debounce |
| 0x21 | BtnAvgThresh | 7:0 | Defines the positive threshold for disabling the processing filter averaging. If ticks are above the threshold, then the averaging is suspended 0x00: 0 0x01: 4 ... 0x50: 320 (default) ... 0xFF: 1020 |
| 0x22 | BtnCompNegThresh | 7:0 | Defines the negative offset compensation threshold. 0x00: 0 0x01: 4 ... 0x50: 320 (default) ... 0xFF: 1020 |
| 0x23 | BtnCompNegCntMax | 7:0 | Defines the number of ticks (below the negative offset compensation threshold) which will initiate an offset compensation. 0x00: Reserved 0x01: 1 sample (default) ... 0xFF-> samples |
| 0x24 | BtnHysteresis | 7:0 | Defines the button hysteresis corresponding to a percentage of the CAP thresholds (defined in Table 19). 0x00: 0% ... 0x0A: 10% (default) ... 0x64: 100% All buttons use the same hysteresis |
| 0x25 | BtnStuckAtTimeout | 7:0 | Defines the stuck at timeout. 0x00: OFF (default) |

| Button Parameters | | | |
|-------------------|------------------------|------|---|
| Address | Name | Bits | Description |
| | | | 0x01: 1 second ... 0xFF: 255 seconds |
| 0x26 | BtnStrongestHysteresis | 7:0 | <p>Defines the hysteresis value for the strongest button filtering engine. This parameter is only valid when BtnCfg has been configured to “report the strongest touch”.</p> <p>The hysteresis element eliminates the jittery output due to environmental noise when two CAP sensors have values very close to each other. The BtnStrongestHysteresis defines how much bigger the signal of the second sensor needs to be compared to the strongest detected sensor, before the second sensor becomes the strongest detected touch.</p> <p>Default value = 0x80 (128)</p> |
| 0x27 | BtnLongPressTimer | 7:0 | <p>Defines the long press timeout on AOI pins (applicable in First Reporting Mode)</p> <p>0x00: OFF (default) 0x01: 1 second ... 0xFF: 255 seconds</p> |
| 0x28 | Reserved | 7:0 | Reserved |
| 0x29 | Reserved | 7:0 | Reserved |

Table 19 Button Parameters

A reliable button operation requires a coherent setting of the registers.

Figure 51 shows an example of a touch and a release. The ticks will vary slightly around the zero idle state.

When the touch occurs the ticks will rise sharply. At the release of the button the ticks will go down rapidly and converge to the idle zero value.

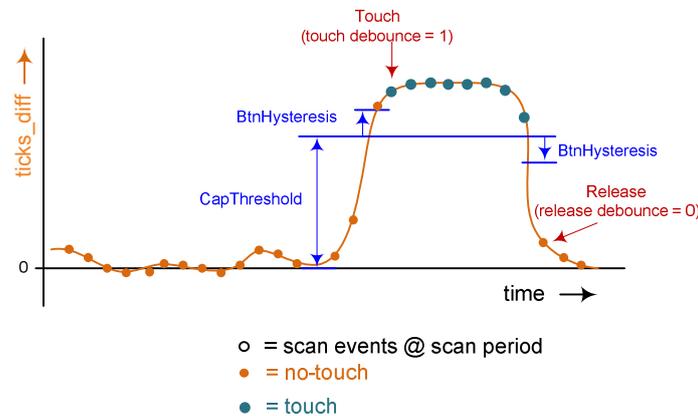


Figure 51 Touch and Release Example

As soon as the ticks become larger than the CAP thresholds (see registers of the previous section) plus the hysteresis (defined in register BtnHysteresis) the debounce counter starts.

In the example of Figure 51 the touch is validated after 2 ticks (BtnCfg [2:0] = 1).

The release is detected immediately (BtnCfg [3] = 0) at the first tick which is below the threshold minus the hysteresis.

BtnCfg

The SX8660 button interface has three modes of operation:

- **Report All:** reports all touches of multiple fingers
- **Report First:** reports only the first detected touch. Subsequent touches are ignored until the first touch is released.
- **Report Strongest:** reports the strongest touch. When the signal from another sensor rises above the first sensor's signal, the second sensor is then reported instead.

The user can select to have the interrupt signal on touching a button, releasing a button or both

In noisy environments it may be required to debounce the touch and release detection decision.

In case the debounce is enabled the SX8660 will count up to the number of debounce samples BtnCfg [1:0], BtnCfg [3:2] before taking a touch or release decision. The sample period is identical to the scan period.

BtnAvgPosThresh

Small environmental and system noise cause the ticks to vary slowly around the zero idle mode value.

In case the ticks get slightly positive this is considered as normal operation. Very large positive tick values indicate a valid touch. The averaging filter is disabled as soon as the average reaches the value defined by BtnAvgPosThresh. This mechanism avoids that a valid touch will be averaged and finally the tick difference becomes zero.

In case three or more sensors reach the BtnAvgPosThresh value simultaneously then the SX8660 will start an offset compensation procedure.

Small environmental and system noise cause the ticks to vary slowly around the zero idle mode value.

In case the ticks get slightly negative this is considered as normal operation. However large negative values will trigger an offset compensation phase and a new set of DCVs will be obtained.

The decision to trigger a compensation phase based on negative ticks is determined by the value in the register `BtnCompNegThresh` and by the number of ticks below the negative thresholds defined in register `BtnCompNegCntMax`. An example is shown in Figure 52.

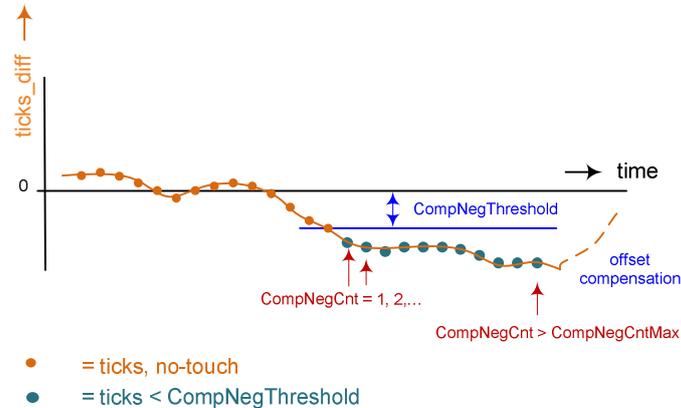


Figure 52 Negative Ticks Offset Compensation Trigger

`BtnCompNegThresh`

Small negative ticks are considered as normal operation and will occur very often.

Larger negative ticks however need to be avoided and a convenient method is to trigger an offset compensation phase. The new set of DCV will assure the idle ticks will be close to zero again.

A trade-off has to be found for the value of this register. A negative threshold too close to zero will trigger a compensation phase very often. A very negative threshold will never trigger.

`BtnCompNegCntMax`

As soon as the ticks get smaller than the Negative Threshold the Negative Counter starts to count.

If the counter goes beyond the Negative Counter Max then the offset compensation phase is triggered.

The recommended value for this register is '1' which means that the offset compensation starts on the first tick below the negative threshold.

`BtnHysteresis`

The hysteresis percentage is identical for all buttons.

A touch is detected if the ticks are getting larger as the value defined by:

$$\text{CapThreshold} + \text{CapThreshold} * \text{hysteresis}.$$

A release is detected if the ticks are getting smaller as the value defined by:

$$\text{CapThreshold} - \text{CapThreshold} * \text{hysteresis}.$$

`BtnStuckAtTimeout`

The stuckat timer can avoid sticky buttons.

If the stuckat timer is set to one second then the touch of a finger will last only for one second and considered released, even if the finger remains on the button for a longer time. After the actual finger release the button can be touched again and will be reported as usual.

In case the stuckat timer is not required it can be set to zero.

ADVANCED COMMUNICATIONS & SENSING DATASHEET
BtnStrongestHysteresis

This parameter defines the hysteresis value for the adjacent button filtering engine. This parameter is only valid when BtnCfg has been configured to report the strongest touch.

When the SX8660 device has been configured to report the strongest touch, a situation may arise where the CAP signals of two sensors are of approximately equal value. Environmental noise can cause the signals of these two sensors to fluctuate as shown in Figure 53 (b).

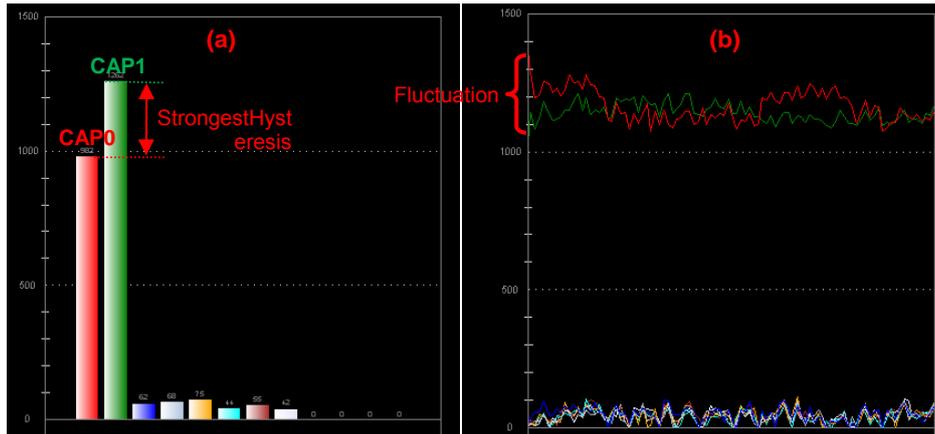


Figure 53 Strongest touch and Hysteresis

As a result of that, the output of the SX8660 device would also change very quickly as each of the two sensors becomes the sensor with the strongest touch value. To eliminate this jitter, the SX8660 device adds a hysteresis element to the calculation of the strongest touch sensor. In that respect, the strongest CAP sensor is calculated as the sensor whose value is greater than the second detected strongest CAP sensor by the Strongest hysteresis amount.

For example, as shown in Figure 53, the strongest CAP sensor is initially CAP0 (Figure 53 (b)). CAP1 becomes the strongest detected touch only if at some point in time the following holds true:

$$\text{CAP1 signal} = \text{CAP0 signal} + \text{StrongestHysteresis}$$

Similarly, if CAP2 is now also touched, it will only become the strongest detected touch if:

$$\text{CAP2 signal} = \text{CAP1 signal} + \text{StrongestHysteresis}.$$

BtnLongPressTimer

This timer defines the time in seconds that the AOI will put out a voltage level corresponding to the button touched. The timer is applicable in the First Reporting Mode. After the timer expires the AOI will return to the idle level even if the button is still touched.

The I2C status and GPO are not affected by this timer (i.e. they will be updated when the button is actually released).

5.5 Analog Output Interface Parameters

| Analog Output Interface (AOI) Parameters | | | |
|--|--------------|--|---|
| Address | Name | Bits | Description |
| 0x2B | AoiCfg | 7:6 | Reserved |
| | | 5:4 | AoiLevelDuringBuzzer (same for A&B) 0x00: AOI button level (default) 0x01: AOI Idle level 0x10: min level (0V) 0x11: max level(VDD) |
| | | 3 | Aoi pwm period (same for A&B) 0: 0xFF (255) (default) 1: 0x3F (63) |
| | | 2:0 | Reserved (default 0x01) |
| 0x2C | AoiBtnMapMsb | 7:0 | Button[7] Button[6] Button[5] Button[4] |
| | | Maps a button touch to one of the two Analog Output Interfaces (AOI-A / AOI-B), or both. | |
| | | 00 : None 01 : AOI-A (GPIO7) 10 : AOI-B (GPIO6) 11 : Both | |
| | | Default Value : 0x5555, i.e. Btn[7..0] mapped on AOI-A | |
| 0x2D | AoiBtnMapLsb | 7:0 | Button[3] Button[2] Button[1] Button[0] |
| | | Default Value : 0x5555, i.e. Btn[7..0] mapped on AOI-A | |
| | | | |
| | | | |
| 0x2E | AoiLevelBtn0 | 7:0 | Defines the level index (cf Table 8) for Buttons and Idle 0x00: 0 0x01: 1 ... 0xFF: 255 Default AoiLevelBtn0 = 0x17 Default AoiLevelBtn1 = 0x2E Default AoiLevelBtn2 = 0x45 Default AoiLevelBtn3 = 0x5D Default AoiLevelBtn4 = 0x74 Default AoiLevelBtn5 = 0x8B Default AoiLevelBtn6 = 0xA3 Default AoiLevelBtn7 = 0xBA Default AoiLevelIdle = 0xFF. The level index should be smaller or equal to Aoi pwm period as defined in AoiCfg[3] |
| 0x2F | AoiLevelBtn1 | 7:0 | |
| 0x30 | AoiLevelBtn2 | 7:0 | |
| 0x31 | AoiLevelBtn3 | 7:0 | |
| 0x32 | AoiLevelBtn4 | 7:0 | |
| 0x33 | AoiLevelBtn5 | 7:0 | |
| 0x34 | AoiLevelBtn6 | 7:0 | |
| 0x35 | AoiLevelBtn7 | 7:0 | |
| 0x36 | AoiLevelIdle | 7:0 | |

AoiBtnMap

This register is used to map the available buttons to SWI-A, AOI-B or both. For example, to map buttons 0 to 3 and buttons 4 to 7 on AOI-B, write the following value to the AoiPwmBtnMap register

```
AoiCfg = 0xAA55;
```

AoiLevelBtn0, AoiLevelBtn1, AoiLevelBtn2, AoiLevelBtn3, AoiLevelBtn4, AoiLevelBtn5, AoiLevelBtn6,
AoiLevelBtn7, AoiLevelIdle

These registers define the level that will be output on AOI-A or AOI-B (depending on button mapping) when the corresponding button is touched or when the corresponding state is active or idle. The duty cycle is defined as a number of steps.

The mean voltage of a PWM signal is given by:

$$\text{Mean voltage} \approx (\text{AoiLevelBtnx} / \text{AoiPwmPeriod}) * \text{Maximum Voltage (VDD)}$$

or:

$$\text{AoiLevelBtnx} \approx (\text{Mean voltage} / \text{Maximum Voltage (VDD)}) * \text{AoiPwmPeriod}$$

AoiPwmPeriod is 255 or 63.

Example:

When button 0 is touched the desired AOI voltage is 0.30 Volts. (with AoiPwmPeriod=255)

To calculate the AoiLevelBtnx is as follows:

Assuming a 3.3V VDD:

$$\begin{aligned} \text{AoiLevelBtnx} &\approx (\text{Mean voltage} / \text{Maximum Voltage (VDD)}) * \text{AoiPwmPeriod} \\ &\approx (0.3/3.3) * 255 \approx 23 \end{aligned}$$

Write 0x17 on AoiBtn0DutyCycle.

5.6 Buzzer Parameters

| Buzzer Parameters | | | |
|-------------------|------------------|------|---|
| Address | Name | Bits | Description |
| 0x37 | BuzzerCfg | 7:6 | Phase 1 duration 0x00: ~ 5ms 0x01: ~ 10ms 0x02: ~ 15ms (default) 0x03: ~ 30ms |
| | | 5:4 | Phase 2 duration 0x00: ~ 5ms 0x01: ~ 10ms 0x02: ~ 15ms (default) 0x03: ~ 30ms |
| | | 3 | BuzzerLevelIdle 0x0: min level (0V), (default) 0x1: max level (VDD) |
| | | 2:0 | Buzzer pwm prescaler value Default 0x04 |
| 0x38 | BuzzerFreqPhase1 | 7:0 | Defines the frequency for the first phase of the buzzer $\text{freq} \approx 4\text{MHz} / (2^{\text{prescaler}} * \text{BuzzerFreqPhase1})$ Default 0x40 (4KHz) |
| 0x39 | BuzzerFreqPhase2 | 7:0 | Defines the frequency for the second phase of the buzzer $\text{freq} \approx 4\text{MHz} / (2^{\text{prescaler}} * \text{BuzzerFreqPhase2})$ Default 0x20 (8KHz) |
| 0x3A | Reserved | 7:0 | Reserved |

5.7 Mapping Parameters

| Mapping Parameters | | | | | |
|--------------------|---------------------|------|-------------|---|---|
| Address | Name | Bits | Description | | |
| 0x3B | MapAutoLight0 | 7:4 | GPIO[7] | default 0x7 | Defines the mapping between GPOs (with Autolight ON) and sensor events. 0x00: Btn0 (default) 0x01: Btn1 ... 0x07: Btn7 0x08...0x0B: Reserved 0x0C: Group0 as defined by MapAutoLightGrp0 0x0D: Group1 as defined by MapAutoLightGrp1 0x0E: Reserved 0x0F: Reserved |
| | | 3:0 | GPIO[6] | default 0x6 | |
| 0x3C | MapAutoLight1 | 7:4 | GPIO[5] | default 0x5 | Several GPOs can be mapped to the same sensor event and will be controlled simultaneously. |
| | | 3:0 | GPIO[4] | default 0x4 | |
| 0x3D | MapAutoLight2 | 7:4 | GPIO[3] | default 0x3 | |
| | | 3:0 | GPIO[2] | default 0x2 | |
| 0x3E | MapAutoLight3 | 7:4 | GPIO[1] | default 0x1 | |
| | | 3:0 | GPIO[0] | default 0x0 | |
| 0x3F | MapAutoLightGrp0Msb | 7:0 | Reserved | | |
| 0x40 | MapAutoLightGrp0Lsb | 7 | Btn7 | Defines Group0 sensor events: 0: OFF (default) 1: ON | |
| | | 6 | Btn6 | | |
| | | 5 | Btn5 | If any of the enabled sensor events occurs the Group0 event will occur as well. | |
| | | 4 | Btn4 | | |
| | | 3 | Btn3 | All sensors events within the group can be independently set. | |
| | | 2 | Btn2 | | |
| | | 1 | Btn1 | | |
| | | 0 | Btn0 | | |
| 0x41 | MapAutoLightGrp1Msb | 7:0 | Reserved | | |
| 0x42 | MapAutoLightGrp1Lsb | 7 | Btn7 | Defines Group1 sensor events: 0: OFF (default) 1: ON | |
| | | 6 | Btn6 | | |
| | | 5 | Btn5 | If any of the enabled sensor events occurs the Group0 event will occur as well. | |
| | | 4 | Btn4 | | |
| | | 3 | Btn3 | All sensors events within the group can be independently set. | |
| | | 2 | Btn2 | | |
| | | 1 | Btn1 | | |
| | | 0 | Btn0 | | |

Table 20 Mapping Parameters

MapAutoLight0, MapAutoLight1, MapAutoLight2, MapAutoLight3
 MapAutoLightGrp0Msb, MapAutoLightGrp0Lsb, MapAutoLightGrp1Msb, MapAutoLightGrp1Lsb

These registers define the mapping between the GPO pins (with Autolight ON) and the sensor information which will control its ON/OFF state.

The mapping can be done to a specific sensor event but also on groups (in this case any sensor event in the group will control the GPO).

Table 21 defines for each selectable sensor event, which action will trigger corresponding GPO to switch ON or OFF.

| MapAutoLight | GPO ON | GPO OFF |
|--------------|--------|---------|
| BtnX | Touch | Release |

Table 21 Autolight Mapping, Sensor Information

Examples:

- If GPO[0] should change state accordingly to Btn4 then MapAutoLight3[3:0] should be set to 0x04.
- If GPO[0] should change state accordingly to Btn0 or Btn1 then Group0 can be used as following:
 - MapAutoLight3[3:0] should be set to 0x0C (ie Group0).
 - MapAutoLightGrp0 should be set to 0x0003 (ie Btn0 or Btn1)

5.8 GPIO Parameters

| GPIO Parameters | | | | | | | |
|-----------------|-------------------|------|----------------------------------|---|--|--------------------|-----|
| Address | Name | Bits | Description | | | | |
| 0x43 | GpioMode7_4 | 7:6 | GPIO[7] Mode | Defines the GPIO mode. 00: GPO 01: GPP 10: GPI 11: SPO: AOI-A for GPIO[7], AOI-B for GPIO[6], Buzzer for GPIO[5]), Reserved for GPIO[4..0] | Default: 0xCC00 | AOI-A | |
| | | 5:4 | GPIO[6] Mode | | | GPO | |
| | | 3:2 | GPIO[5] Mode | | | Buzzer | |
| | | 1:0 | GPIO[4] Mode | | | GPO | |
| 0x44 | GpioMode3_0 | 7:6 | GPIO[3] Mode | | Defines the GPIO mode. 00: GPO 01: GPP 10: GPI 11: SPO: AOI-A for GPIO[7], AOI-B for GPIO[6], Buzzer for GPIO[5]), Reserved for GPIO[4..0] | Default: 0xCC00 | GPO |
| | | 5:4 | GPIO[2] Mode | | | | GPO |
| | | 3:2 | GPIO[1] Mode | | | | GPO |
| | | 1:0 | GPIO[0] Mode | | | | GPO |
| 0x45 | GpioIntensityOn0 | 7:0 | ON Intensity Index | Defines the ON intensity index 0x00: 0 0x01: 1 ... 0xFF: 255 (default) | | | |
| 0x46 | GpioIntensityOn1 | 7:0 | | | | | |
| 0x47 | GpioIntensityOn2 | 7:0 | | | | | |
| 0x48 | GpioIntensityOn3 | 7:0 | | | | | |
| 0x49 | GpioIntensityOn4 | 7:0 | | | | | |
| 0x4A | GpioIntensityOn5 | 7:0 | | | | | |
| 0x4B | GpioIntensityOn6 | 7:0 | | | | | |
| 0x4C | GpioIntensityOn7 | 7:0 | | | | | |
| 0x4D | GpioIntensityOff0 | 7:0 | OFF Intensity Index | Defines the OFF intensity index 0x00: 0 (default) 0x01: 1 ... 0xFF: 255 | | | |
| 0x4E | GpioIntensityOff1 | 7:0 | | | | | |
| 0x4F | GpioIntensityOff2 | 7:0 | | | | | |
| 0x50 | GpioIntensityOff3 | 7:0 | | | | | |
| 0x51 | GpioIntensityOff4 | 7:0 | | | | | |
| 0x52 | GpioIntensityOff5 | 7:0 | | | | | |
| 0x53 | GpioIntensityOff6 | 7:0 | | | | | |
| 0x54 | GpioIntensityOff7 | 7:0 | | | | | |
| 0x56 | GpioOutPwrUp | 7:0 | GPIO[7] Output Value at Power Up | Defines the values of GPO and GPP pins after power up ie default values of I2C parameters GpoCtrl and GppIntensity respectively. 0: OFF(GPO) / IntensityOff (GPP) (default) 1: ON (GPO) / IntensityOn (GPP) | | | |
| | | | GPIO[6] Output Value at Power Up | | | | |
| | | | GPIO[5] Output Value at Power Up | | | | |
| | | | GPIO[4] Output Value at Power Up | | | | |
| | | | GPIO[3] Output Value at Power Up | | | | |
| | | | GPIO[2] Output Value at Power Up | | Bits corresponding to GPO pins with Autolight ON should be left to 0. | | |
| | | | GPIO[1] Output Value at Power Up | | | | |
| | | | GPIO[0] Output Value at Power Up | | Before being actually initialized GPIOs are set as inputs with pull up. | | |
| 0x57 | GpioAutoLight | 7:0 | GPIO[7] AutoLight | Enables Autolight in GPO mode 0 : OFF 1 : ON (default) | | | |
| | | | GPIO[6] AutoLight | | | | |
| | | | GPIO[5] AutoLight | | | | |

| GPIO Parameters | | | |
|-----------------|----------------|------|--|
| Address | Name | Bits | Description |
| | | | GPIO[4] AutoLight GPIO[3] AutoLight GPIO[2] AutoLight GPIO[1] AutoLight GPIO[0] AutoLight |
| 0x58 | GpioPolarity | 7:0 | GPIO[7] Output Polarity GPIO[6] Output Polarity GPIO[5] Output Polarity GPIO[4] Output Polarity GPIO[3] Output Polarity GPIO[2] Output Polarity GPIO[1] Output Polarity GPIO[0] Output Polarity |
| | | | Defines the polarity of the GPO and GPP pins. 0: Inverted 1: Normal 0xA0 (default) SPO pins needs Normal Polarity |
| 0x59 | GpioFunction | 7:0 | GPIO[7] Function GPIO[6] Function GPIO[5] Function GPIO[4] Function GPIO[3] Function GPIO[2] Function GPIO[1] Function GPIO[0] Function |
| | | | Defines the intensity index vs PWM pulse width function. 0: Logarithmic 1: Linear 0xA0 (default) |
| 0x5A | GpioIncFactor | | GPIO[7] Fading Increment Factor GPIO[6] Fading Increment Factor GPIO[5] Fading Increment Factor GPIO[4] Fading Increment Factor GPIO[3] Fading Increment Factor GPIO[2] Fading Increment Factor GPIO[1] Fading Increment Factor GPIO[0] Fading Increment Factor |
| | | | Defines the fading increment factor. 0: 1, intensity index incremented every increment time (default) 1: 16, intensity index incremented every 16 increment times |
| 0x5B | GpioDecFactor | | GPIO[7] Fading Decrement Factor GPIO[6] Fading Decrement Factor GPIO[5] Fading Decrement Factor GPIO[4] Fading Decrement Factor GPIO[3] Fading Decrement Factor GPIO[2] Fading Decrement Factor GPIO[1] Fading Decrement Factor GPIO[0] Fading Decrement Factor |
| | | | Defines the fading decrement factor. 0: 1, intensity index decremented every decrement time (default) 1: 16, intensity index decremented every 16 decrement times |
| 0x5C | GpioIncTime7_6 | 7:4 | GPIO[7] Fading Increment Time |
| | | | Defines the fading increment time. |

| GPIO Parameters | | | | | |
|-----------------|-------------------|------|-------------------------------|---|--|
| Address | Name | Bits | Description | | |
| 0x5D | GpioIncTime5_4 | 3:0 | GPIO[6] Fading Increment Time | 0x0: OFF 0x1: 0.5ms 0x2: 1ms ... 0xF: 7.5ms The total fading in time will be: $GpioIncTime * GpioIncFactor *$ (GpioIntensityOn – GpioIntensityOff) default: 0x00122200 | |
| | | 7:4 | GPIO[5] Fading Increment Time | | |
| 0x5E | GpioIncTime3_2 | 3:0 | GPIO[4] Fading Increment Time | | |
| | | 7:4 | GPIO[3] Fading Increment Time | | |
| 0x5F | GpioIncTime1_0 | 3:0 | GPIO[2] Fading Increment Time | | |
| | | 7:4 | GPIO[1] Fading Increment Time | | |
| 0x60 | GpioDecTime7_6 | 3:0 | GPIO[0] Fading Increment Time | | |
| | | 7:4 | GPIO[7] Fading Decrement Time | | |
| 0x61 | GpioDecTime5_4 | 3:0 | GPIO[6] Fading Decrement Time | | Defines the fading decrement time. 0x0: OFF 0x1: 0.5ms 0x2: 1ms ... 0x4: 2.0ms (default) ... 0xF: 7.5ms The total fading out time will be: $GpioDecTime * GpioDecFactor *$ (GpioIntensityOn – GpioIntensityOff) default: 0x42122244 |
| | | 7:4 | GPIO[5] Fading Decrement Time | | |
| 0x62 | GpioDecTime3_2 | 3:0 | GPIO[4] Fading Decrement Time | | |
| | | 7:4 | GPIO[3] Fading Decrement Time | | |
| 0x63 | GpioDecTime1_0 | 3:0 | GPIO[2] Fading Decrement Time | | |
| | | 7:4 | GPIO[1] Fading Decrement Time | | |
| 0x64 | GpioOffDelay7_6 | 3:0 | GPIO[0] Fading Decrement Time | | |
| | | 7:4 | GPIO[7] OFF Delay | | |
| 0x65 | GpioOffDelay5_4 | 3:0 | GPIO[6] OFF Delay | i) Single Fading Mode Defines the delay between release and start of fading out: 0x0: OFF (default) 0x1: 200ms 0x2: 400ms ... 0xF: 3000ms ii) Continuous Fading Mode Defines the number of fading cycles after a release. 0x0: OFF 0x1: one 0x2: two ... 0xF fifteen iii) Repeat Fading Mode Defines the number of fading cycles after a touch. 0x0: one 0x1: two ... 0xF: sixteen | |
| | | 7:4 | GPIO[5] OFF Delay | | |
| 0x66 | GpioOffDelay3_2 | 3:0 | GPIO[4] OFF Delay | | |
| | | 7:4 | GPIO[3] OFF Delay | | |
| 0x67 | GpioOffDelay1_0 | 3:0 | GPIO[2] OFF Delay | | |
| | | 7:4 | GPIO[1] OFF Delay | | |
| 0x68 | GpioPullUpDown7_4 | 7:6 | GPIO[7] Pullup/down | | Enables pullup/down resistors for GPI pins. 00 : None (default) 01 : Pullup 10 : Pulldown 11 : Reserved |
| | | 5:4 | GPIO[6] Pullup/down | | |
| | | 3:2 | GPIO[5] Pullup/down | | |
| | | 1:0 | GPIO[4] Pullup/down | | |
| 0x69 | GpioPullUpDown3_0 | 7:6 | GPIO[3] Pullup/down | | |

| GPIO Parameters | | | |
|-----------------|-------------------|------|-------------------------|
| Address | Name | Bits | Description |
| | | 5:4 | GPIO[2] Pullup/down |
| | | 3:2 | GPIO[1] Pullup/down |
| | | 1:0 | GPIO[0] Pullup/down |
| 0x6A | GpioInterrupt7_4 | 7:6 | GPI[7] Interrupt |
| | | 5:4 | GPI[6] Interrupt |
| | | 3:2 | GPI[5] Interrupt |
| | | 1:0 | GPI[4] Interrupt |
| 0x6B | GpioInterrupt3_0 | 7:6 | GPI[3] Interrupt |
| | | 5:4 | GPI[2] Interrupt |
| | | 3:2 | GPI[1] Interrupt |
| | | 1:0 | GPI[0] Interrupt |
| 0x6A | GpioDebounce | 7:0 | GPI[7] Debounce |
| | | | GPI[6] Debounce |
| | | | GPI[5] Debounce |
| | | | GPI[4] Debounce |
| | | | GPI[3] Debounce |
| | | | GPI[2] Debounce |
| | | | GPI[1] Debounce |
| | | | GPI[0] Debounce |
| 0x6A | GpioFadingMode7_4 | 7:6 | Fading mode for GPIO[7] |
| | | 5:4 | Fading mode for GPIO[6] |
| | | 3:2 | Fading mode for GPIO[5] |
| | | 1:0 | Fading mode for GPIO[4] |
| 0x6B | GpioFadingMode3_0 | 7:6 | Fading mode for GPIO[3] |
| | | 5:4 | Fading mode for GPIO[2] |
| | | 3:2 | Fading mode for GPIO[1] |
| | | 1:0 | Fading mode for GPIO[0] |

Defines the GPI edge which will trigger INTB falling edge and exit Sleep/Doze modes if relevant.
 00 : None (default)
 01 : Rising
 10 : Falling
 11 : Both

Enables the GPI debounce (done on 10 consecutive samples at 1ms).
 0 : OFF (default)
 1 : ON

Defines the Fading mode for GPO[7:0].
 00: Single Fading Mode
 01: Continuous Fading Mode
 10: Repeat Fading Mode
 11: Reserved

default: 0x2A50
 The fading modes are expected to be defined at power up by the QSM or NVM. In case the fading modes need to be changed after power up this can be done when the GPOs are all OFF.

Table 22 resumes the applicable SPM and I2C parameters for each GPIO mode.

| | | GPI | GPP | GPO | SPO |
|------------|------------------|-----|----------------|------------------|----------------|
| SPM | GpioMode | X | X | X | X ⁵ |
| | GpioOutPwrUp | | X ¹ | X ^{2,3} | |
| | GpioAutolight | | | X | |
| | GpioPolarity | | X | X | |
| | GpioIntensityOn | | X ¹ | X | |
| | GpioIntensityOff | | X ¹ | X | |
| | GpioFunction | | X | X | |
| | GpioIncFactor | | | X | |
| | GpioDecFactor | | | X | |
| | GpioIncTime | | | X | |
| | GpioDecTime | | | X | |
| | GpioOffDelay | | | X | |
| | GpioPullUpDown | X | | | |
| | GpioInterrupt | X | | | |
| | GpioDebounce | X | | | |
| I2C | IrqSrc[4] | X | | | |
| | GpiStat | X | | | |
| | GpoCtrl | | | X ⁴ | |
| | GppPinId | | X | | |
| | GppIntensity | | X ¹ | | |

¹ At power up, GppIntensity of each GPP pin is initialized with GpioIntensityOn or GpioIntensityOff depending on GpioOutPwrUp corresponding bits value.

² Only if Autolight is OFF, else must be left to 0 (default value)

³ GpioOutPwrUp must be set to OFF in Repeat and Continuous Fading Modes (with Autolight OFF)

⁴ Only if Autolight is OFF, else ignored

⁵ In SPO mode assure the following settings: GpioOutPwrUp=OFF, GpioAutoLight=ON, GpioPolarity=Normal, GpioFunction=Linear

Table 22 Applicable SPM/I2C Parameters vs. GPIO Mode

6 I2C INTERFACE

The I2C implemented on the SX8660 is compliant with:

- standard (100kb/s), fast mode (400kb/s)
- slave mode
- 7 bit address (default 0x2B). The default address can be changed in the NVM at address 0x04.

The host can use the I2C to read and write data at any time. The effective changes will be applied at the next processing phase (section 3.3).

Three types of registers are considered:

- status (read). These registers give information about the status of the capacitive buttons, GPIOs, operation modes etc...
- control (read/write). These registers control the soft reset, operating modes, GPIOs and offset compensation.
- SPM gateway (read/write). These registers are used for the communication between host and the SPM. The SPM gateway communication is done typically at power up and is not supposed to be changed when the application is running. The SPM needs to be re-stored each time the SX8660 is powered down. The SPM can be stored permanently in the NVM memory of the SX8660. The SPM gateway communication over the I2C at power up is then not required.

The I2C will be able to read and write from a start address and then perform read or writes sequentially, and the address increments automatically.

The supported I2C access formats are described in the next sections.

6.1 I2C Write

The format of the I2C write is given in Figure 54.

After the start condition [S], the slave address (SA) is sent, followed by an eighth bit ('0') indicating a Write. The SX8660 then acknowledges [A] that it is being addressed, and the master sends an 8 bit Data Byte consisting of the SX8660 Register Address (RA). The slave acknowledges [A] and the master sends the appropriate 8 bit Data Byte (WD0). Again the slave acknowledges [A]. In case the master needs to write more data, a succeeding 8 bit Data Byte will follow (WD1), acknowledged by the slave [A]. This sequence will be repeated until the master terminates the transfer with the Stop condition [P].

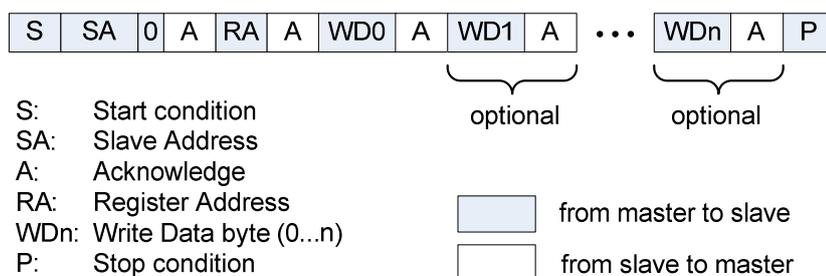


Figure 54 I2C write

The register address is incremented automatically when successive register data (WD1...WDn) is supplied by the master.

6.2 I2C read

The format of the I2C read is given in Figure 55.

After the start condition [S], the slave address (SA) is sent, followed by an eighth bit ('0') indicating a Write. The SX8660 then acknowledges [A] that it is being addressed, and the master responds with an 8 bit data consisting of the Register Address (RA). The slave acknowledges [A] and the master sends the Repeated Start Condition [Sr]. Once again, the slave address (SA) is sent, followed by an eighth bit ('1') indicating a Read.

The SX8660 responds with acknowledge [A] and the Read Data byte (RD0). If the master needs to read more data it will acknowledge [A] and the SX8660 will send the next read byte (RD1). This sequence can be repeated until the master terminates with a NACK [N] followed by a stop [P].

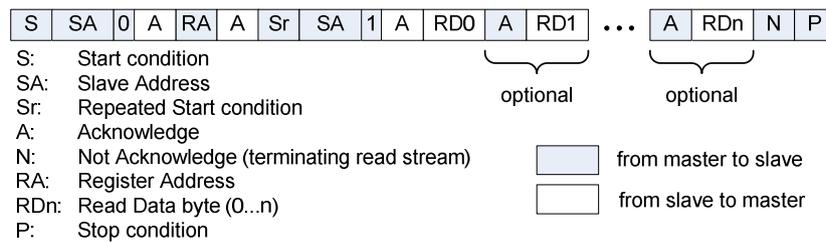


Figure 55 I2C read

6.3 I2C Registers Overview

| Address | Name | R/W | Description |
|---------|--------------|------------|---------------------------------|
| 0x00 | IrqSrc | read | Interrupt Source |
| 0x01 | Reserved | | |
| 0x02 | CapStat | read | Button Status |
| 0x03 | Reserved | | |
| 0x04 | Reserved | | |
| 0x05 | Reserved | | |
| 0x06 | Reserved | | |
| 0x07 | GpiStat | read | GPI Status |
| 0x08 | SpmStat | read | SPM Status |
| 0x09 | CompOpMode | read/write | Compensation and Operating Mode |
| 0x0A | GpoCtrl | read/write | GPO Control |
| 0x0B | GppId | read/write | GPP Pin Selection |
| 0x0C | GppIntensity | read/write | GPP Intensity |
| 0x0D | SpmCfg | read/write | SPM Configuration |
| 0x0E | SpmBaseAddr | read/write | SPM Base Address |
| 0x0F | Reserved | | |
| 0xAC | SpmKeyMsb | read/write | SPM Key MSB |
| 0xAD | SpmkeyLsb | read/write | SPM Key LSB |
| 0xB1 | SoftReset | read/write | Software Reset |

Table 23 I2C Registers Overview

6.4 Status Registers

| Address | Name | Bits | Description |
|---------|--------|------|-------------------------------|
| 0x00 | IrqSrc | 7 | Reserved |
| | | 6 | NVM burn interrupt flag |
| | | 5 | SPM write interrupt flag |
| | | 4 | GPI interrupt flag |
| | | 3 | Reserved |
| | | 2 | Buttons interrupt flag |
| | | 1 | Compensation interrupt flag |
| | | 0 | Operating Mode interrupt flag |

Table 24 Interrupt Source

The delay between the actual event and the flags indicating the interrupt source may be one scan period.

IrqSrc[6] is set once NVM burn procedure is completed.

IrqSrc[5] is set once SPM write is effective.

IrqSrc[4] is set if a GPI edge as programmed in GpioInterrupt occurred. GpiStat shows the detailed status of the GPI pins.

IrqSrc[2] is set if a Button event occurred (touch or release if enabled). CapStatLsb show the detailed status of the Buttons.

IrqSrc[1] is set once compensation procedure is completed either through automatic trigger or via host request.

IrqSrc[0] is set when actually entering Active or Doze mode via host request. CompOpmode shows the current operation mode.

| Address | Name | Bits | Description |
|---------|---------|------|-----------------|
| 0x02 | CapStat | 7 | Status button 7 |
| | | 6 | Status button 6 |
| | | 5 | Status button 5 |
| | | 4 | Status button 4 |
| | | 3 | Status button 3 |
| | | 2 | Status button 2 |
| | | 1 | Status button 1 |
| | | 0 | Status button 0 |

Status of individual buttons
0: Released (default)
1: Touched

Table 25 I2C Cap status

| Address | Name | Bits | Description |
|---------|---------|------|---|
| 0x07 | GpiStat | 7:0 | GPI[7:0] Status Status of each individual GPI pin 0: Low 1: High Bits of non-GPI pins are set to 0. |

Table 26 I2C GPI status

| Address | Name | Bits | Description |
|---------|---------|------|---|
| 0x08 | SpmStat | 7:4 | reserved |
| | | 3 | NvmValid Indicates if the current NVM is valid. 0: No – QSM is used 1: Yes – NVM is used |
| | | 2:0 | NvmCount Indicates the number of times NVM has been burned: 0: None – QSM is used (default) 1: Once – NVM is used if NvmValid = 1, else QSM. 2: Twice – NVM is used if NvmValid = 1, else QSM. 3: Three times – NVM is used if NvmValid = 1, else QSM. 4: More than three times – QSM is used |

Table 27 I2C SPM status

6.5 Control Registers

| Address | Name | Bits | Description |
|---------|------------|------|---|
| 0x09 | CompOpMode | 7:3 | Reserved*, write only '00000' |
| | | 2 | Compensation Indicates/triggers compensation procedure 0: Compensation completed (default) 1: read -> compensation running ; write -> trigger compensation |
| | | 1:0 | Operating Mode Indicates/programs** operating mode 00: Active mode (default) 01: Doze mode 10: Sleep mode 11: Reserved |

* The reading of these reserved bits will return varying values.

** After the operating mode change (Active/Doze) the host should wait for INTB or 300ms before performing any I2C read access.

Table 28 I2C compensation, operation modes

| Address | Name | Bits | Description |
|---------|---------|------|--|
| 0x0A | GpoCtrl | 7:0 | GpoCtrl[7:0] Triggers ON/OFF state of GPOs when Autolight is OFF 0: OFF (ie go to IntensityOff) 1: ON (ie go to IntensityOn) Default is set by SPM parameter GpioOutPwrUp Bits of non-GPO pins are ignored. |

Table 29 I2C GPO Control

| Address | Name | Bits | Description |
|---------|----------|------|--|
| 0x0B | GppPinId | 7:3 | Reserved, write only '00000' |
| | | 2:0 | GPP Pin Identifier Defines the GPP pin to which the GppIntensity is assigned for the following read/write operations 0x0 = GPP0 (default) 0x1 = GPP1 ... 0x7 = GPP7 GPPx refers to pin GPIOx configured as GPP |

Table 30 I2C GPP Pin Identifier

| Address | Name | Bits | Description |
|---------|--------------|------|--|
| 0x0C | GppIntensity | 7:0 | Defines the intensity index of the GPP pin selected in GppPinId 0x00: 0 0x01: 1 ... 0xFF: 255 Reading returns the intensity index of the GPP pin selected in GppPinId. Default value is IntensityOn or IntensityOff depending on GpioOutPwrUp. |

Table 31 I2C GPP Intensity

| Address | Name | Bits | Description |
|---------|-----------|------|--|
| 0xB1 | SoftReset | 7:0 | Writing 0xDE followed by 0x00 will reset the chip. |

Table 32 I2C Soft Reset

6.6 SPM Gateway Registers

The SX8660 I2C interface offers two registers for exchanging the SPM data with the host.

- SpmCfg
- SpmBaseAddr

| Address | Name | Bits | Description |
|---------|--------|------|---|
| 0x0D | SpmCfg | 7:6 | 00: Reserved |
| | | 5:4 | Defines the normal operation or SPM mode 00: I2C in normal operation mode (default) 01: I2C in SPM mode 10: Reserved 11: Reserved |
| | | 3 | Defines r/w direction of SPM 0: SPM write access (default) 1: SPM read access |
| | | 2:0 | 000: Reserved |

Table 33 SPM access configuration

| Address | Name | Bits | Description |
|---------|-------------|------|--|
| 0x0E | SpmBaseAddr | 7:0 | SPM Base Address (modulo 8). The lowest address is 0x00 (default) The highest address is 0x78. |

Table 34 SPM Base Address

The exchange of data, read and write, between the host and the SPM is always done in bursts of eight bytes. The base address of each burst of eight bytes is a modulo 8 number, starting at 0x00 and ending at 0x78.

The registers SpmKeyMsb and SpmKeyLsb are required for NVM programming as described in section 6.7.

| Address | Name | Bits | Description |
|---------|-----------|------|---|
| 0xAC | SpmKeyMsb | 7:0 | SPM to NVM burn Key MSB Unlock requires writing data: 0x62 |

Table 35 SPM Key MSB at I2C register address 0xAC

| Address | Name | Bits | Description |
|---------|-----------|------|---|
| 0xAD | SpmKeyLsb | 7:0 | SPM to NVM burn Key LSB Unlock requires writing data: 0x9D |

Table 36 SPM Key LSB

6.6.1 SPM Write Sequence

The SPM must always be written in blocks of 8 bytes. The sequence is described below:

1. Set the I2C in SPM mode by writing "01" to SpmCfg[5:4] and SPM write access by writing '0' to SpmCfg[3].
2. Write the SPM base address to SpmBaseAddr (The base address needs to be a value modulo 8).
3. Write the eight consecutive bytes to I2C address 0, 1, 2, ...7
4. Terminate by writing "000" to SpmCfg[5:3].

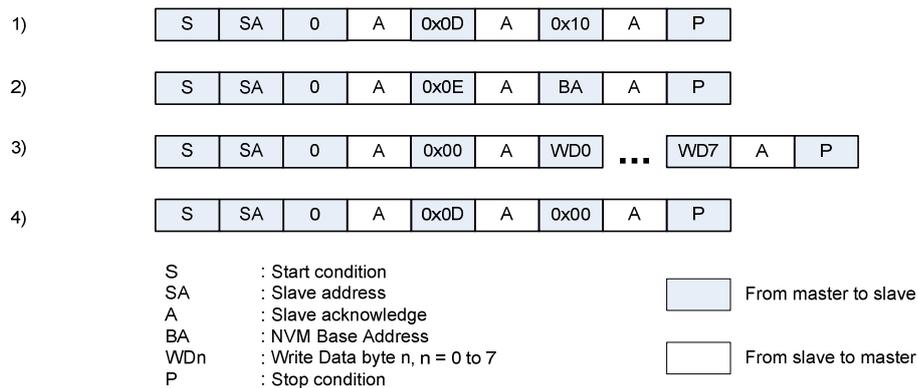


Figure 56: SPM write sequence

The complete SPM can be written by repeating 16 times the cycles shown in Figure 56 using base addresses 0x00, 0x08, 0x10, ...0x70, 0x78.

Once the SPM write sequence is actually applied, the INTB pin will be asserted. The host clears the interrupt by reading any I2C register. At the same time the bit GenStatMsb[6], indicating the SPM write is done, will be cleared.

ADVANCED COMMUNICATIONS & SENSING DATASHEET

6.6.2 SPM Read Sequence

The SPM must always be read in blocks of 8 bytes. The sequence is described below:

1. Set the I2C in SPM mode by writing “01” to SpmCfg[5:4] and SPM read access by writing ‘1’ to SpmCfg[3].
2. Write the SPM base address to SpmBaseAddr (The base address needs to be a value modulo 8).
3. Read the eight consecutive bytes from I2C address 0, 1, 2, ...7
4. Terminate by writing “000” to SpmCfg[5:3].

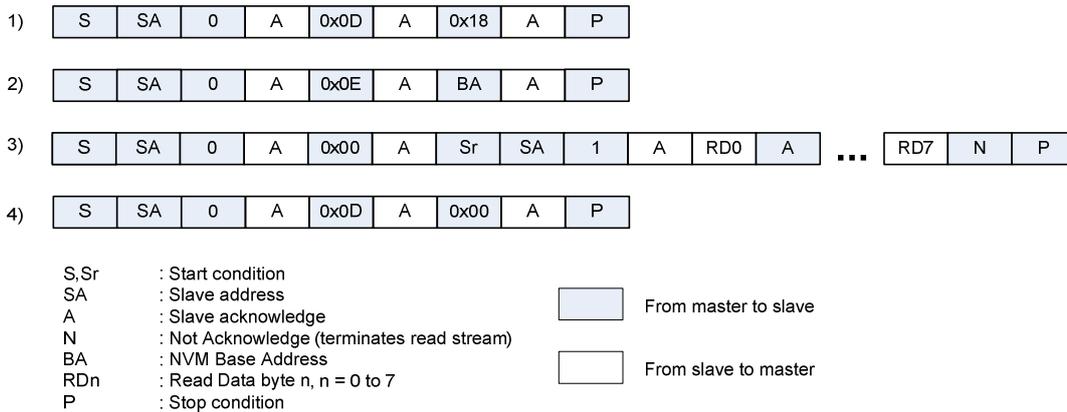


Figure 57: SPM Read Sequence

The complete SPM can be read by repeating 16 times the cycles shown in Figure 57 using base addresses 0x00, 0x08, 0x10, ..., 0x70, 0x78.

Once the SPM read sequence is actually applied, the INTB pin will be asserted. The host clears the interrupt by reading any I2C register. At the same time the bit GenStatMsb[6], indicating the SPM write is done, will be cleared.

6.7 NVM burn

The content of the SPM can be copied permanently (burned) into the NVM to be used as the new default parameters. The burning of the NVM can be done up to three times and must be done only when the SPM is completely written with the desired data.

The number of times the NVM has been burned can be monitored by reading NvmCycle from the I2C register GenStatLsb[7:5].

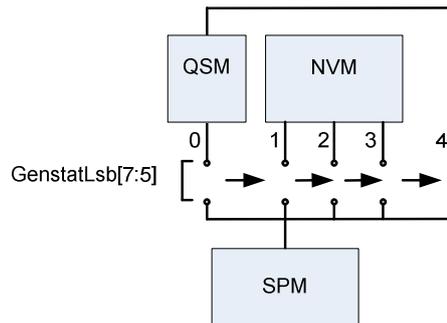


Figure 58 Simplified Diagram NvmCycle

Figure 58 shows the simplified diagram of the NvmCycle counter. The SX8660 is delivered with empty NVM and NvmCycle set to zero. The SPM points to the QSM.

Each NVM burn will increase the NvmCycle. At the fourth NVM burn the SX8660 switches definitely to the QSM.

The burning of the SPM into the NVM is done by executing a special sequence of four I2C commands.

- | | |
|--|------------------------------------|
| 1. Write the data 0x62 to the I2C register I2CKeyMsb. | Terminate the I2C write by a STOP. |
| 2. Write the data 0x9D to the I2C register I2CKeyLsb. | Terminate the I2C write by a STOP. |
| 3. Write the data 0xA5 to the I2C register I2CSpmBaseAddr. | Terminate the I2C write by a STOP. |
| 4. Write the data 0x5A to the I2C register I2CSpmBaseAddr. | Terminate the I2C write by a STOP. |

This is illustrated in Figure 59.

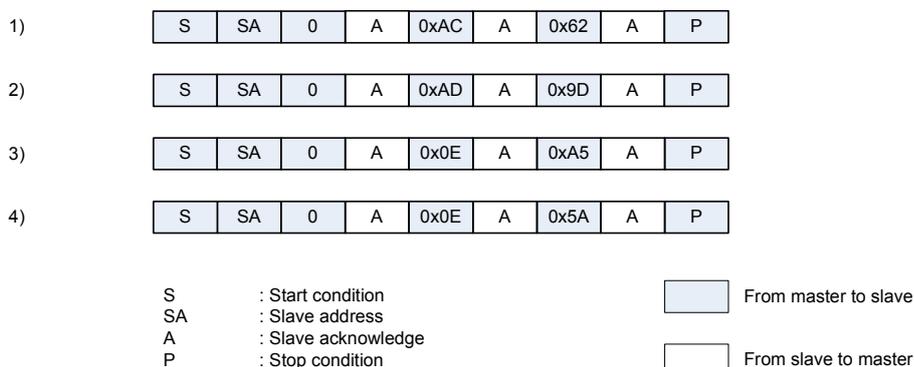


Figure 59: NVM burn procedure

7 APPLICATION INFORMATION

A typical application schematic is shown in Figure 60.

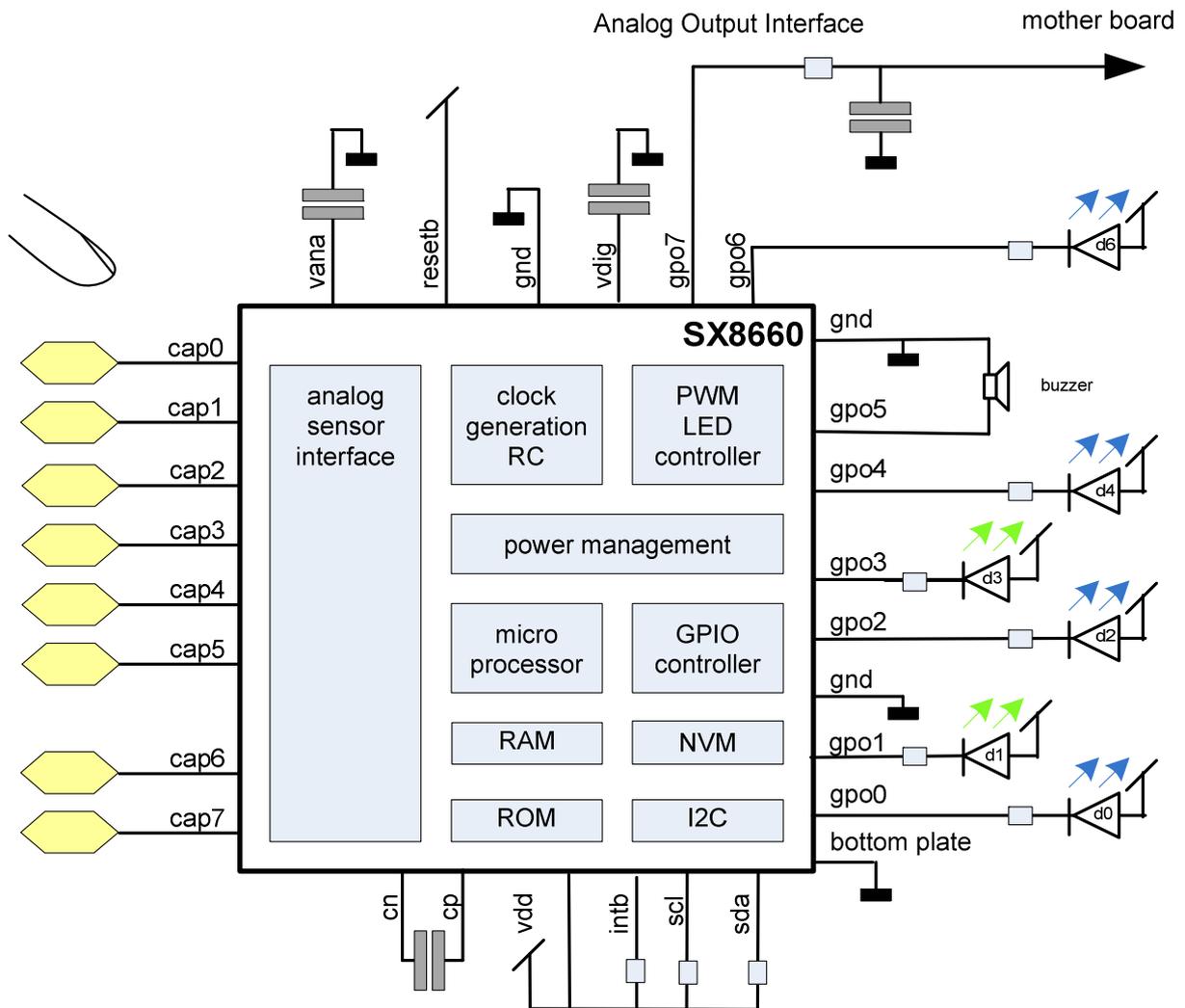


Figure 60 Typical Application

8 PACKAGING INFORMATION

8.1 Package Outline Drawing

SX8660 is assembled in a MLPQ-UT28 package as shown in Figure 61.

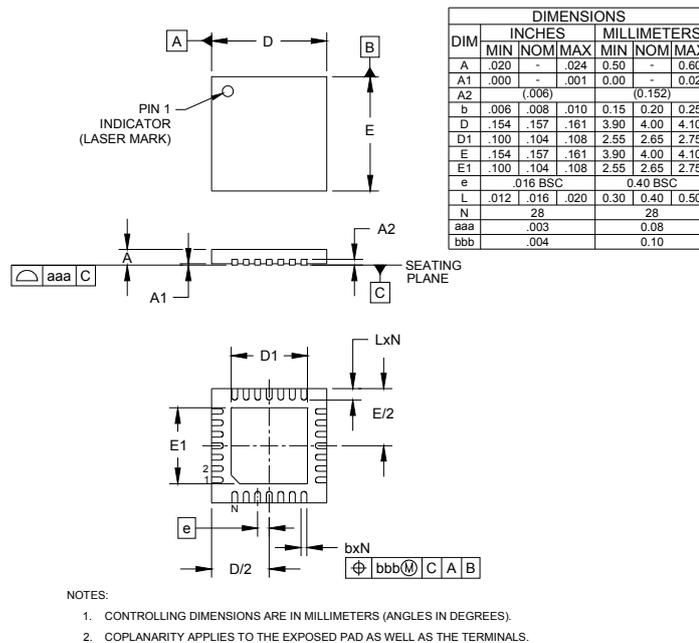
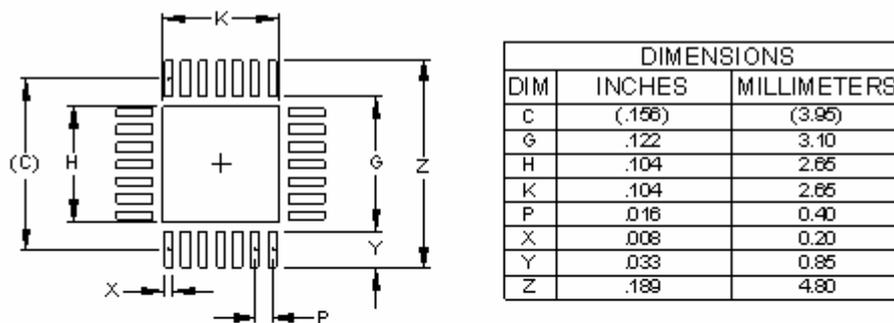


Figure 61 Package outline drawing

8.2 Land Pattern

The land pattern of MLPQ-UT28 package, 4 mm x 4 mm is shown in Figure 62.



- NOTES:
1. CONTROLLING DIMENSIONS ARE IN MILLIMETERS (ANGLES IN DEGREES).
 2. THIS LAND PATTERN IS FOR REFERENCE PURPOSES ONLY. CONSULT YOUR MANUFACTURING GROUP TO ENSURE YOUR COMPANY'S MANUFACTURING GUIDELINES ARE MET.
 3. THERMAL VIAS IN THE LAND PATTERN OF THE EXPOSED PAD SHALL BE CONNECTED TO A SYSTEM GROUND PLANE. FAILURE TO DO SO MAY COMPROMISE THE THERMAL AND/OR FUNCTIONAL PERFORMANCE OF THE DEVICE.
 4. SQUARE PACKAGE DIMENSIONS APPLY IN BOTH X AND Y DIRECTIONS.

Figure 62 Land pattern

© Semtech 2010

All rights reserved. Reproduction in whole or in part is prohibited without the prior written consent of the copyright owner. The information presented in this document does not form part of any quotation or contract, is believed to be accurate and reliable and may be changed without notice. No liability will be accepted by the publisher for any consequence of its use. Publication thereof does not convey nor imply any license under patent or other industrial or intellectual property rights. Semtech assumes no responsibility or liability whatsoever for any failure or unexpected operation resulting from misuse, neglect improper installation, repair or improper handling or unusual physical or electrical stress including, but not limited to, exposure to parameters beyond the specified maximum ratings or operation outside the specified range.

SEMTECH PRODUCTS ARE NOT DESIGNED, INTENDED, AUTHORIZED OR WARRANTED TO BE SUITABLE FOR USE IN LIFE-SUPPORT APPLICATIONS, DEVICES OR SYSTEMS OR OTHER CRITICAL APPLICATIONS. INCLUSION OF SEMTECH PRODUCTS IN SUCH APPLICATIONS IS UNDERSTOOD TO BE UNDERTAKEN SOLELY AT THE CUSTOMER'S OWN RISK. Should a customer purchase or use Semtech products for any such unauthorized application, the customer shall indemnify and hold Semtech and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs damages and attorney fees which could arise.

Notice: All referenced brands, product names, service names and trademarks are the property of their respective owners.

Contact Information

Semtech Corporation
Advanced Communications and Sensing Products Division
200 Flynn Road, Camarillo, CA 93012
Phone: (805) 498-2111 Fax: (805) 498-3804