ADVANCED COMMUNICATIONS & SENSING **DATASHEET**

GENERAL DESCRIPTION

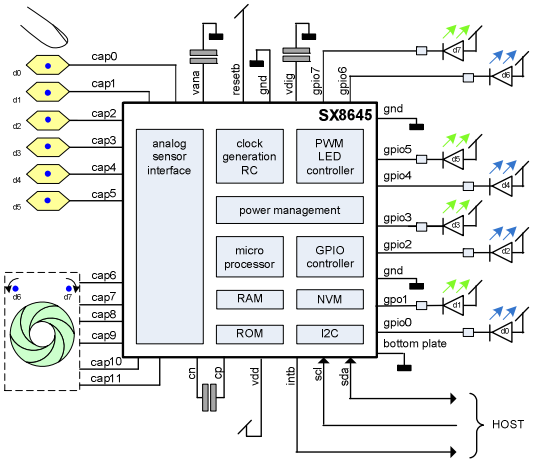
The SX8645 is an ultra low power, fully integrated 12-channel solution for capacitive touch-buttons and wheel applications. Unlike many capacitive touch solutions, the SX8645 features dedicated capacitive sense inputs (that requires no external components) in addition to 8 general purpose I/O ports (GPIO). Each GPIO is typically configured as LED driver with independent PWM source for enhanced lighting control such as intensity and fading.

The SX8645 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 SX8645 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 SX8645 supports the 400 kHz I²C serial bus data protocol and includes a field programmable slave address. The tiny 5mm x 5mm 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 Twelve Sensors Capacitive Touch Controller for Buttons and Wheel
  - Pre-configured for 6 Buttons and a Wheel
  - 8 LED Drivers with Individual Intensity, Fading Control and Autolight Mode
  - 256 steps PWM Linear and Logarithmic control
- High Resolution Capacitive Sensing
  - Up to 100pF of Offset Capacitance Compensation at Full Sensitivity
  - Capable of Sensing through Overlay Materials up to 5mm thick
- Extremely Low Power Optimized for Portable Application
  - 8uA (typ) in Sleep Mode
  - 80uA (typ) in Doze Mode (Scanning Period 195ms)
  - 220uA (typ) in Active Mode (Scanning Period 30ms)
- Programmable Scanning Period from 15ms to 1500ms
- 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
- "Smart" Wake-up Sequence for Easy Activation from Doze
- No External Components per Sensor Input
- Internal Clock Requires No External Components
- Differential Sensor Sampling for Reduced EMI
- 400 KHz Fast-Mode I²C Interface with Interrupt
- -40°C to +85°C Operation

APPLICATIONS

- Notebook/Netbook/Portable/Handheld computers
- Cell phones, PDAs
- Consumer Products, Instrumentation, Automotive
- Mechanical Button Replacement

ORDERING INFORMATION

Part Number	Temperature Range	Package
SX8645I05AWLTRT[1]	-40°C to +85°C	Lead Free MLPQ-W32

[1] 3000 Units/reel

* This device is RoHS/WEEE compliant and Halogen Free

Table of Contents

1 GENERAL DESCRIPTION

1.1 Pin Diagram

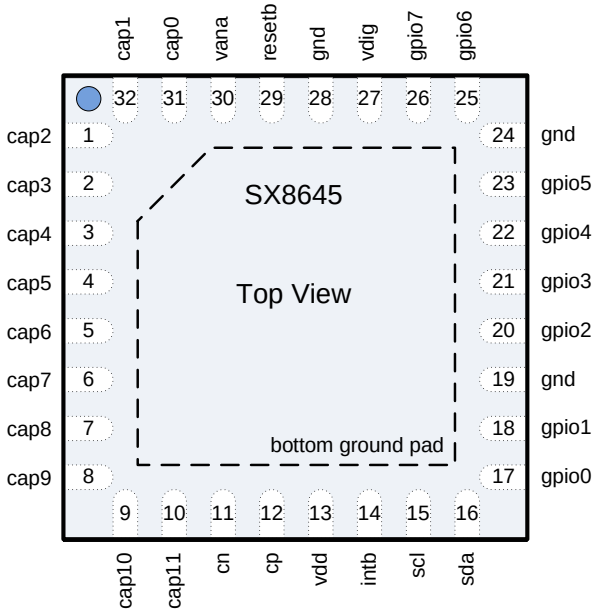
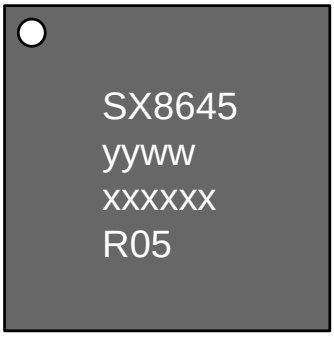


Figure 1 Pinout Diagram

1.2 Marking information



yyww = Date Code
xxxxxx = Semtech lot number
R05 = Semtech Code

Figure 2 Marking Information

1.3 Pin Description

Number	Name	Type	Description
1	CAP2	Analog	Capacitive Sensor 2
2	CAP3	Analog	Capacitive Sensor 3
3	CAP4	Analog	Capacitive Sensor 4
4	CAP5	Analog	Capacitive Sensor 5
5	CAP6	Analog	Capacitive Sensor 6
6	CAP7	Analog	Capacitive Sensor 7
7	CAP8	Analog	Capacitive Sensor 8
8	CAP9	Analog	Capacitive Sensor 9
9	CAP10	Analog	Capacitive Sensor 10
10	CAP11	Analog	Capacitive Sensor 11
11	CN	Analog	Integration Capacitor, negative terminal (1nF between CN and CP)
12	CP	Analog	Integration Capacitor, positive terminal (1nF between CN and CP)
13	VDD	Power	Main input power supply
14	INTB	Digital Output	Interrupt, active LOW, requires pull up resistor (on host or external)
15	SCL	Digital Input	I2C Clock, requires pull up resistor (on host or external)
16	SDA	Digital Input/Output	I2C Data, requires pull up resistor (on host or external)
17	GPIO0	Digital Input/Output	General Purpose Input/Output 0
18	GPIO1	Digital Input/Output	General Purpose Input/Output 1
19	GND	Ground	Ground
20	GPIO2	Digital Input/Output	General Purpose Input/Output 2
21	GPIO3	Digital Input/Output	General Purpose Input/Output 3
22	GPIO4	Digital Input/Output	General Purpose Input/Output 4
23	GPIO5	Digital Input/Output	General Purpose Input/Output 5
24	GND	Ground	Ground
25	GPIO6	Digital Input/Output	General Purpose Input/Output 6
26	GPIO7	Digital Input/Output	General Purpose Input/Output 7
27	VDIG	Analog	Digital Core Decoupling, connect to a 100nF decoupling capacitor
28	GND	Ground	Ground
29	RESETB	Digital Input	Active Low Reset. Connect to VDD if not used.
30	VANA	Analog	Analog Core Decoupling, connect to a 100nF decoupling capacitor
31	CAP0	Analog	Capacitive Sensor 0
32	CAP1	Analog	Capacitive Sensor 1
bottom plate	GND	Ground	Exposed pad connect to ground

Table 1 Pin description

1.4 Simplified Block Diagram

The simplified block diagram of the SX8645 is illustrated in Figure 3.

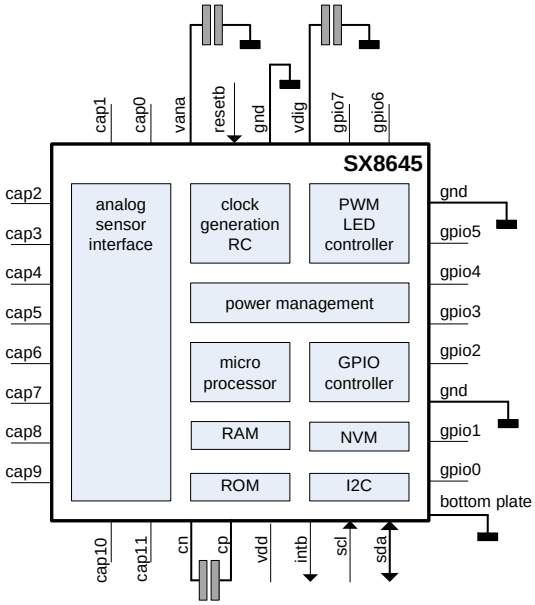


Figure 3 Simplified block diagram of the SX8645

1.5 Acronyms

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

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 SX8645 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 SX8645 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, 12 sensors enabled, minimum sensitivity		220	300	uA
Doze mode, average	$I_{OP,Doze}$	195ms scan period, 12 sensors enabled, minimum sensitivity		80	110	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*VDD		VDD + 0.3V	V
Input logic low	V_{IL}	VSS applied to GND pins	VSS - 0.3V		0.8	V
Input leakage current	L_I	CMOS input			±1	uA
Pull up resistor	R_{PU}	when enabled		660		kΩ
Pull down resistor	R_{PD}	when enabled		660		kΩ
GPIO set as Output, INTB, SDA						
Output logic high	V_{OH}	I_{OH} <4mA	VDD-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			150	ms
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”

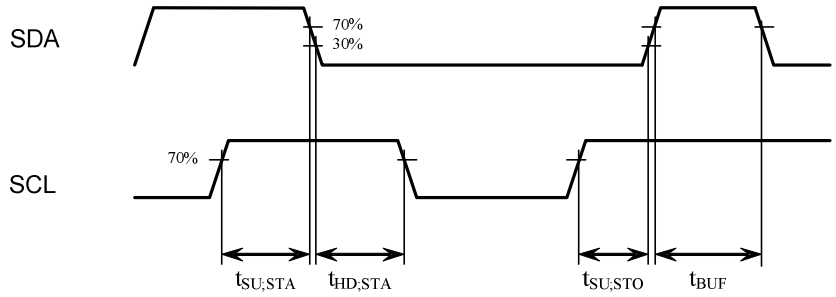


Figure 4 I2C Start and Stop timing

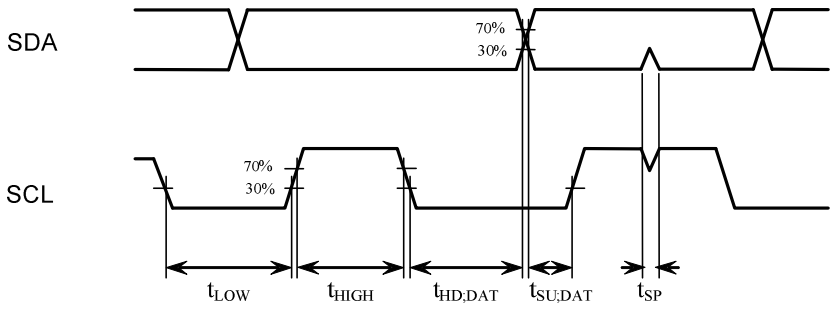


Figure 5 I2C Data timing

3 FUNCTIONAL DESCRIPTION

3.1 Quickstart Application

The SX8645 is preconfigured (Quickstart Application) for an application with 6 buttons, a wheel (consisting of 6 sensors) and 8 LED drivers using logarithmic PWM fading.

Implementing a schematic based on Figure 6 will be immediately operational after powering without programming the SX8645 (even without host).

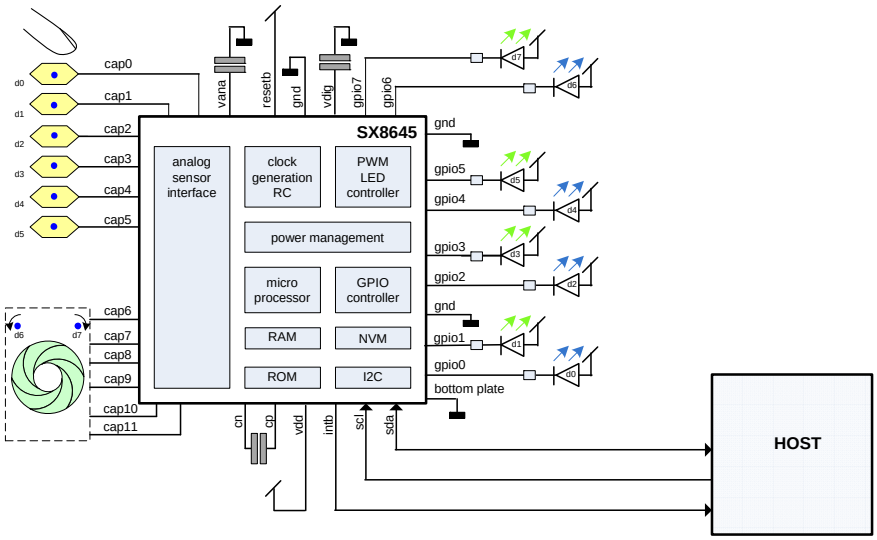


Figure 6 Quickstart Application

Touching the sensor on the CAP0 pin will enable automatically the LED connected to GPIO0. When the CAP0 sensor is released the LED on GPIO0 will slowly fade-out using smooth logarithmic fading.

All other sensors (CAP1 to CAP5) have their own LED associated on a GPIO pin showing a touch or a release.

The sensors on CAP6 to CAP11 are used in a wheel configuration. A moving finger on the wheel will enable the LED on GPIO6 or GPIO7 indicating the finger rotation direction.

The sensor detection and the LED fading described above are operational without any host interaction.

This is made possible using the SX8645 Autolight feature described in the following sections.

3.2 Introduction

3.2.1 General

The SX8645 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 SX8645 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 SX8645 Analog Sensor Interface (ASI).

The ticks are further processed by the SX8645 and converted in a high level, easy to use information for the user's host.

The information between SX8645 and the user's host is passed through the I2C interface with an additional interrupt signal indicating that the SX8645 has new information. For buttons this information is simply touched or released.

3.2.2 GPIOs

A second path of feedback to the user is using General Purpose Input Output (GPIO) pins. The SX8645 offers 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 steps PWM. The SX8645 has eight individual PWM generators, one for each GPIO pin.
The LED fading can be initiated automatically by the SX8645 by setting the SX8645 Autolight feature. A simple touch on a sensor and the corresponding LED will fade-in without any host interaction over the I2C.
In case the Autolight feature is disabled then the host will decide to start a LED fading-in period, simply by setting the GP0 pin to 'high' using one I2C command. The SX8645 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 GPP mode. 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 the digital standard Input mode (GPI).

3.2.3 Parameters

The SX8645 has many low level built-in, fixed algorithms and procedures. To allow a lot of freedom for the user and adapt the SX8645 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 are buttons or which sensors are parts of a wheel, which GPIO is used for outputs 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.

3.2.4 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 SX8645 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 SX8645. The programming needs to be done once (over the I2C). The SX8645 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 SX8645 sensing interface can be split into three periods over time.

In the first period (Sensing) the SX8645 is sensing all enabled CAP inputs, from CAP0 towards CAP11.

In the second period (Processing) the SX8645 processes the sensor data, verifies and updates the GPIO and I2C status registers.

In the third period (Timer) the SX8645 is set in a low power mode and waits until a new cycle starts.

Figure 7 shows the different SX8645 periods over time.

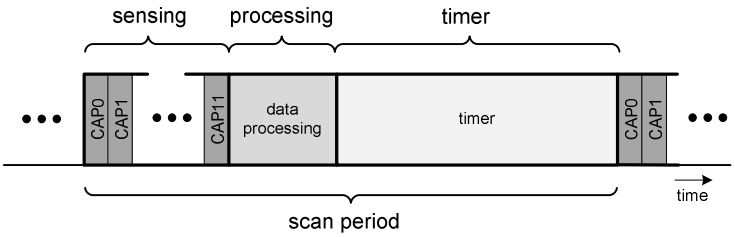


Figure 7 Scan Period

The scan period determines the minimum reaction time of the SX8645. 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 SX8645 generates the interrupt on the INTB pin. The shorter the scan period the faster the reaction time will be.
Very low power consumption can be obtained by setting very long scan periods with the expense of having longer reaction times.

Important: All external events like GPIO, I2C and INTB 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 delay of one scan period later in time.

3.4 Operation modes

The SX8645 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 SX8645 OFF, except for the I2C and GPI peripheral, minimizing operating current while maintaining the power supplies. In Sleep mode the SX8645 does not do any sensor scanning.

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 SX8645 is not used for a specific time it can 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.
For some applications a single button touch might cause undesired wakening up and Active mode would be entered too often.
The SX8645 offers therefore a smart wake-up sequence feature in which the user needs to touch and release a correct sequence of buttons before Active mode will be entered. This is explained in more detail in the Wake-Up Sequence section.

The host can decide to force the operating mode by issuing commands over the I2C (using register CompOpMode) and take fully control of the SX8645.

The diagram in Figure 8 shows the available operation modes and the possible transitions.

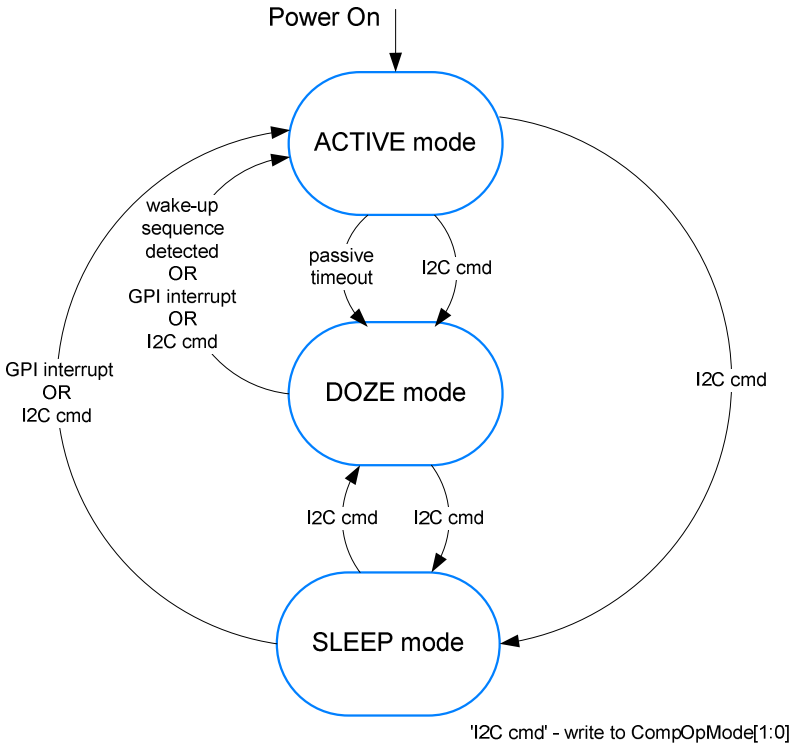


Figure 8 Operation modes

3.5 Sensors on the PCB

The capacitive sensors are relatively simple copper areas on the PCB connected to the twelve SX8645 capacitive sensor input pins (CAP0…CAP11).The sensors are covered by isolating overlay material (typically 1mm...3mm). The area of a sensor is typically one square centimeter 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) or arranged in a wheel configuration (see example Figure 10) for e.g. menu scrolling or volume control applications.

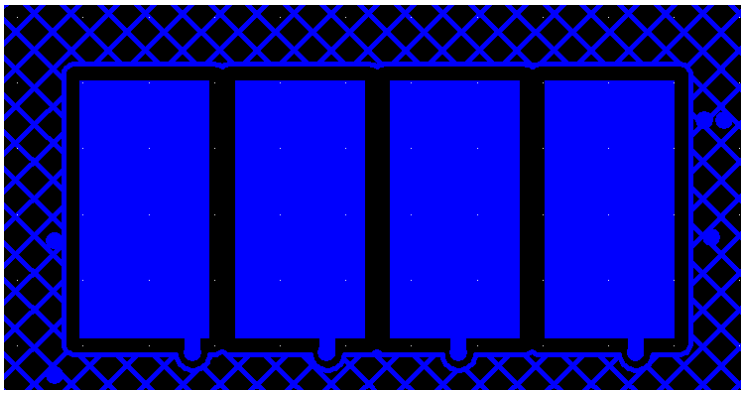


Figure 9 PCB top layer of four sensors for buttons (surrounded by a ground plane)

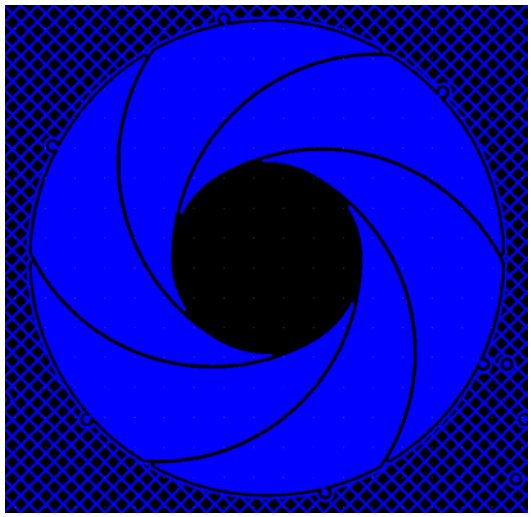


Figure 10 PCB top layer of one wheel using six sensors (surrounded by ground plane)

3.6 Button and Wheel Information

3.6.1 Button Information

The buttons have two simple states (see Figure 11): ON (touched by finger) and OFF (released and no finger press).

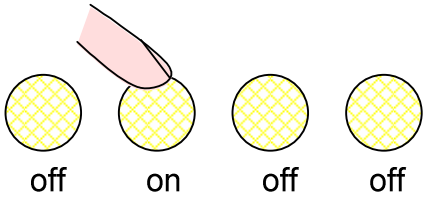


Figure 11 Buttons

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 ticks from the ASI go below the threshold minus a hysteresis. The hysteresis around the threshold avoids rapid touch and release signaling during transients.

3.6.2 Wheel Information

In case sensors are arranged in a wheel configuration the ON, OFF information remains available as if it would be a single sensor button.

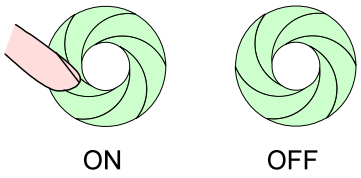


Figure 12 Wheel ON, OFF

Wherever the wheel is touched the information will be set to ON. If no finger is present the wheel information will be OFF.

Due to the 2 dimensional character of the wheel more information can be derived by processing the ticks.
During a touch a finger will influence most of the time the charge on one or two sensors but never all of the sensors at the same time. Some sensor ticks will be larger than others based on the finger position.
The processing algorithms can therefore determine where the finger is positioned on the wheel.
Interpolation between sensors increases the resolution beyond the number of sensors in the wheel.
The interpolation can be done already on the PCB sensor structures (analog, like the wheel in Figure 10) and as well by SX8645 digital processing of the ticks using center of gravity calculations.

The position of the finger on the PCB structures varies between the minimum zero and a user defined maximum (Figure 13).

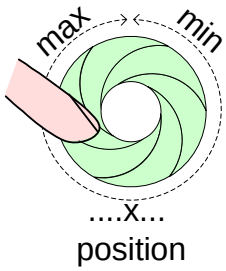


Figure 13 Wheel Position

The position belonging to the minimum and associated to a sensor is defined arbitrarily. The SX8645 defines the minimum position to the sensor with the lowest CAP pin index. E.g. if CAP0 is a button (or disabled) and CAP1 to CAP7 are the sensors of the wheel then the position 'zero' starts at CAP1 and the maximum is found at CAP7.

In addition to the wheel position, the SX8645 allows to detect finger rotation. The rotation occurs if the finger position changes a certain step size between two succeeding scan periods. A very slow moving finger will not be considered as a rotation as the changing position will be minor. The SX8645 allows detecting a rotate clockwise (direction min to max) (see Figure 14) and a rotate counter clockwise (direction max to min) (see Figure 15).

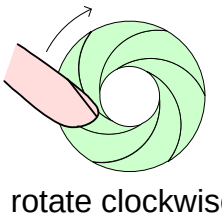


Figure 14 Wheel rotate clockwise

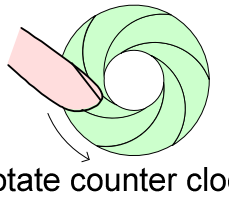


Figure 15 Wheel rotate counter clockwise

3.7 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 16).

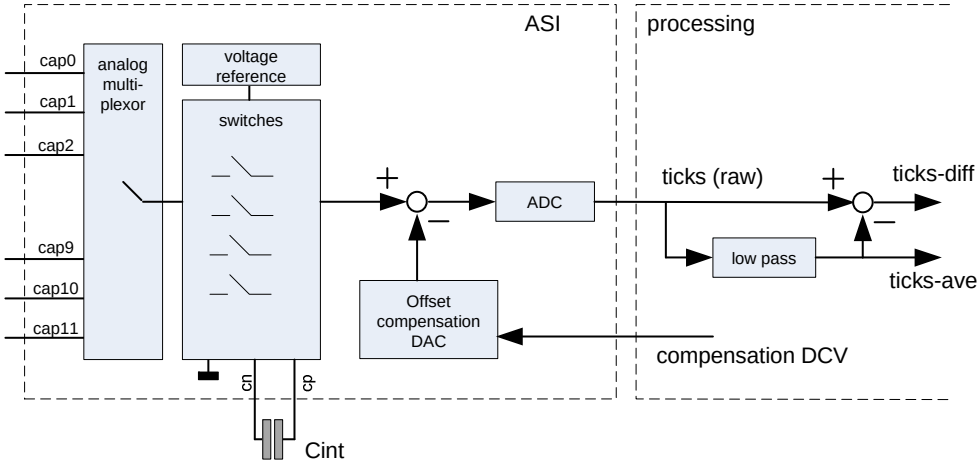


Figure 16 Analog Sensor Interface

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 SX8645 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.

3.8 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 SX8645 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 SX8645 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 SX8645 is shut down the compensation values will be lost. At a next power-up the procedure starts all over again. This assures that the SX8645 will operate under any condition. Powering up at e.g. different temperatures will not change the performance of the SX8645 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 SX8645 will initiate a compensation procedure and find a new set of DCVs.

Compensation procedures can as well be initiated by the SX8645 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 (in Active or Doze mode). This is e.g. required after the host changed the sensitivity of sensors.

3.9 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 17).
This slowly varying average is important in the detection of slowly changing environmental changes.

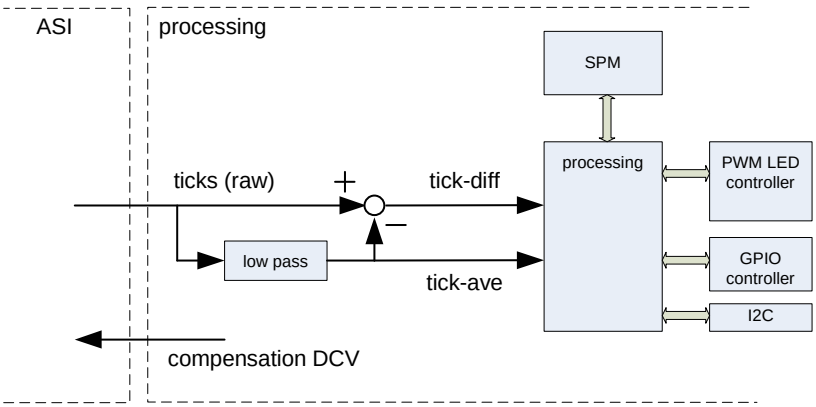


Figure 17 Processing

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.

3.10 Configuration

Figure 18 shows the building blocks used for configuring the SX8645.

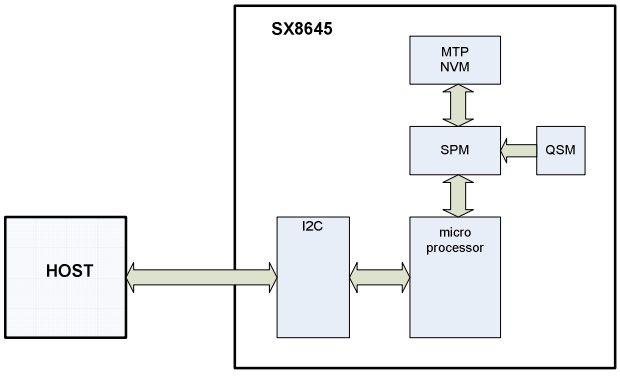


Figure 18 Configuration

The default configuration parameters of the SX8645 are stored in the Quick Start Memory (QSM). This configuration data is setup to a very common application for the SX8645 with 6 buttons and a wheel. Without any programming or host interaction the SX8645 will startup in the Quick Start Application.

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

In case the application needs different settings than the QSM settings then the SX8645 can be setup and/or programmed over the I2C interface.

The configuration parameters of the SX8645 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 SX8645 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 SX8645 copies the configuration parameter source (QSM or NVM) into the Shadow Parameter Memory (SPM). The SX8645 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 (from QSM or NVM) following power up or at the end of the reset event.

The host will interface with the SX8645 through the I2C bus.

The I2C of the SX8645 consists of 16 registers. Some of these I2C registers are used to read the status and information of the button and the wheel. Other I2C registers allow the host to take control of the SX8645. The host can e.g. decide to change the operation mode from Active mode to Doze mode or go into Sleep (according to 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 19 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 to be changed dynamically.

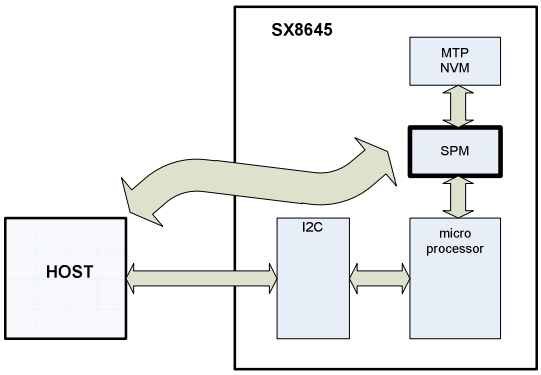


Figure 19 Host SPM mode

The content of the SPM remains valid as long as the SX8645 is powered and no reset is performed. After a power down or reset the host needs to re-write the SPM if relevant for the application.

Figure 20 shows the Host NVM mode. In this mode the host will be able to write the NVM.

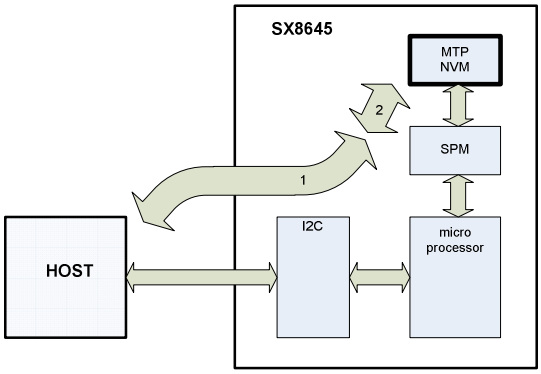


Figure 20 Host NVM mode

The writing of the host towards the NVM is not done directly but done in 2 steps (Figure 20).

In the first step the host writes to the SPM (as in Figure 19). In the second step the host signals the SX8645 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).

3.11 Power Management

The SX8645 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 SX8645 internal circuitry and must not be loaded externally.

3.12 Clock Circuitry

The SX8645 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.

3.13 I2C interface

The I2C interface allows the communication between the host and the SX8645.

The I2C slave implemented on the SX8645 is compliant with the standard (100kb/s) and fast mode (400kb/s)
The default SX8645 I2C address equals 0b010 1011.
A different I2C address can be programmed by the user in the NVM.

3.14 Reset

The reset can be performed by 3 sources:

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

3.14.1 Power up

During power up the INTB is kept low. Once the power up sequence is terminated the INTB is released autonomously. The SX8645 is then ready for operation.

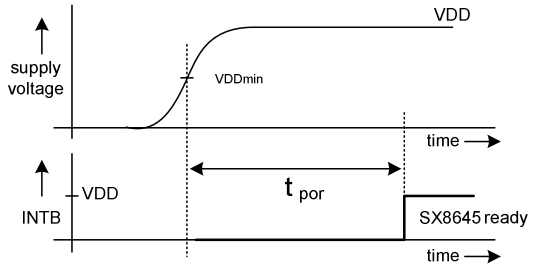


Figure 21 Power Up vs. INTB

During the power on period the SX8645 stabilizes the internal regulators, RC clocks and the firmware initializes all registers.

During the power up the SX8645 is not accessible and I2C communications are forbidden.
As soon as the INTB rises the SX8645 will be ready for I2C communication.

3.14.2 RESETB

When RESETB is driven low the SX8645 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.

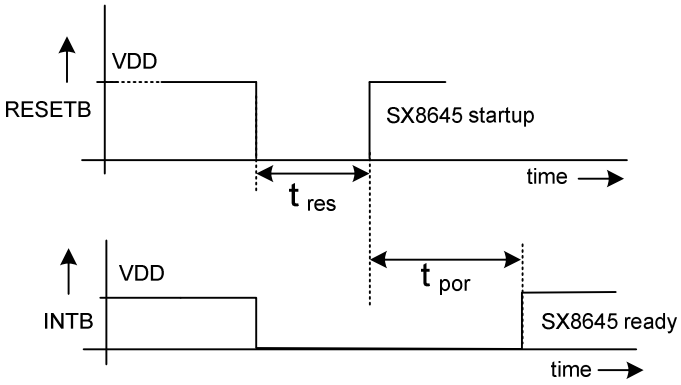


Figure 22 Hardware Reset

3.14.3 Software Reset

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

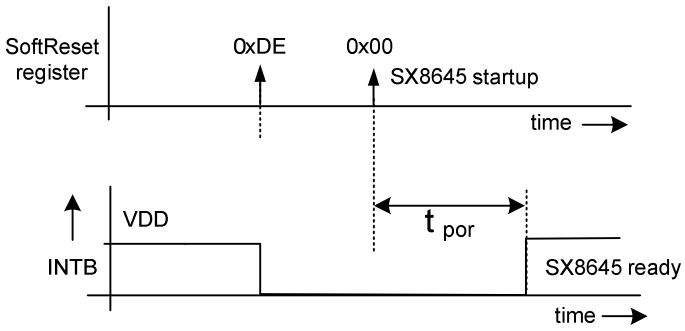


Figure 23 Software Reset

3.15 Interrupt

3.15.1 Power up

During power up the INTB is kept low. Once the power up sequence is terminated the INTB is released autonomously. The SX8645 is then ready for operation.

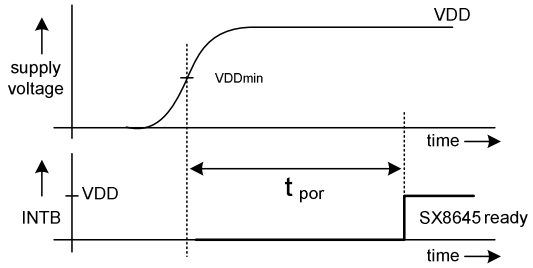


Figure 24 Power Up vs. INTB

During the power on period the SX8645 stabilizes the internal regulators, RC clocks and the firmware initializes all registers.

During the power up the SX8645 is not accessible and I2C communications are forbidden.
As soon as the INTB rises the SX8645 will be ready for I2C communication.

3.15.2 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 registers CapStatMsb and CapStatLsb show the detailed status of the Buttons,
- if a Wheel event occurred (touch, release, rotate clockwise, rotate counter clockwise or position change). I2C registers CapStatMsb, WhlPosMsb and WhlPosLsb show the detailed status of the Wheel,
- 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 either through automatic wakeup or via 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).

3.15.3 Clearing

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

The clearing of the INTB is done as soon as the host performs a read to the IrqSrc I2C register or reset is completed

3.15.4 Example

A typical example of the assertion and clearing of the INTB and the I2C communication is shown in Figure 25.

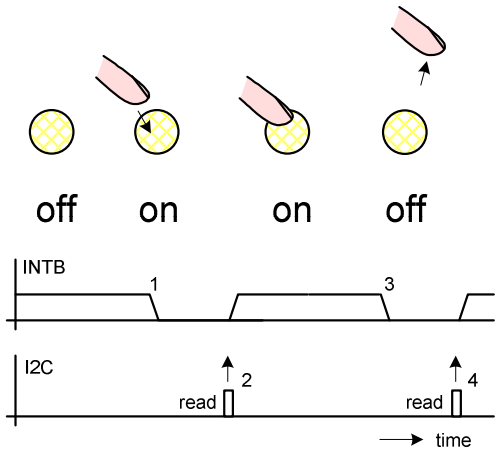


Figure 25 Interrupt and I2C

When a button is touched the SX8645 will assert the interrupt (1). The host will read the IrqSrc information over the I2C and this clears the interrupt (2).

If the finger releases the button the interrupt will be asserted (3). The host reading the IrqSrc information will clear the interrupt (4).

In case the host does not react to an interrupt this results in a missing touch.

3.16 General Purpose Input and Outputs

3.16.1 Introduction and Definitions

The SX8645 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)

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.

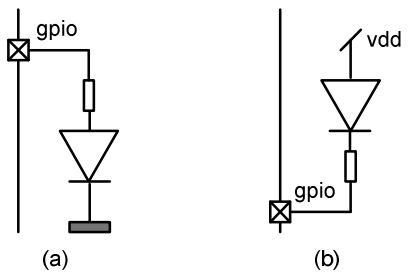


Figure 26 Polarity definition, (a) normal, (b) inverted

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 128us typ.

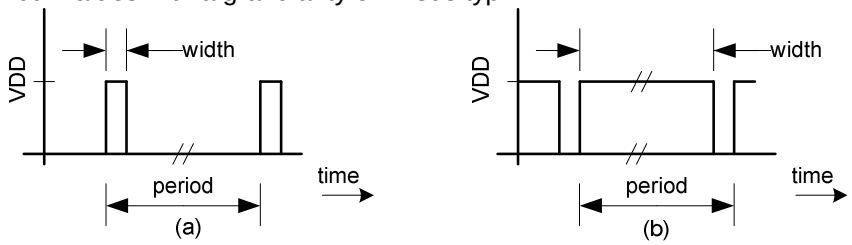


Figure 27 PWM definition, (a) small pulse width, (b) large pulse width

3.16.2 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 7 SPM/I2C Parameters Applicable in GPI Mode

3.16.3 GPP

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.

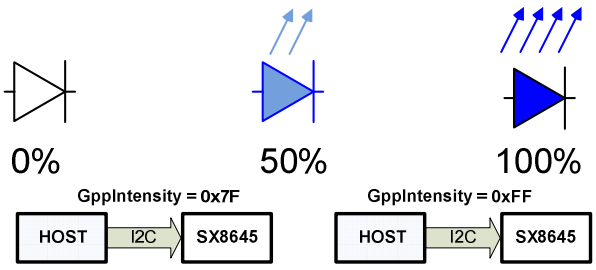


Figure 28 LED control in GPP mode

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 8 SPM/I2C Parameters Applicable in GPP Mode

3.16.4 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.

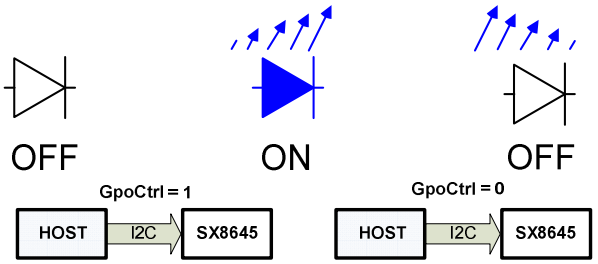


Figure 29 LED Control in GPO mode, Autolight OFF

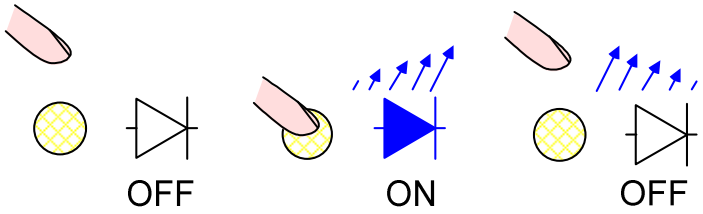


Figure 30 LED Control in GPO mode, Autolight ON (mapped to Button)

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.

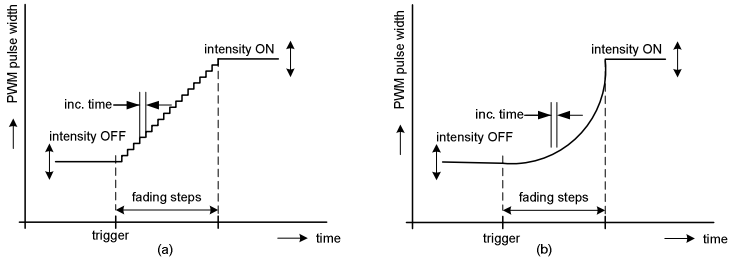


Figure 31 GPO ON transition (LED fade in), normal polarity, (a) linear, (b) logarithmic

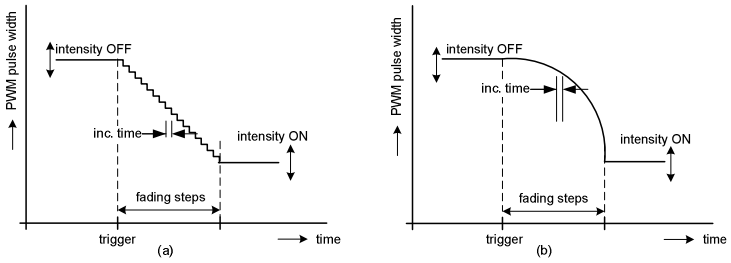


Figure 32 GPO ON transition (LED fade in), inverted polarity, (a) linear, (b) logarithmic

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 33 and Figure 34).

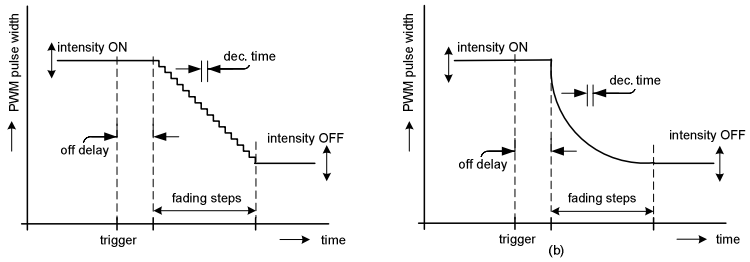


Figure 33 GPO OFF transition (LED fade out), normal polarity, (a) linear, (b) logarithmic

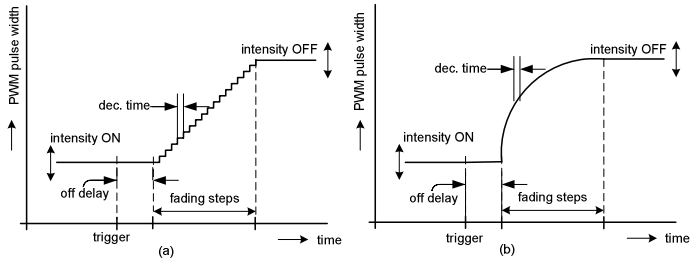


Figure 34 GPO OFF transition (LED fade out), inverted polarity, (a) linear, (b) logarithmic

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.

SPM/I2C parameters applicable in GPO mode are listed in table below.

		GPO
SPM	GpioMode	X
	GpioOutPwrUp	X[1]
	GpioAutoligth	X
	GpioPolarity	X
	GpioIntensityOn	X
	GpioIntensityOff	X
	GpioFunction	X
	GpioIncFactor	X
	GpioDecFactor	X
	GpioIncTime	X
	GpioDecTime	X
	GpioOffDelay	X
I2C	GpoCtrl	X[2]

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

Table 9 SPM/I2C Parameters Applicable in GPO Mode

3.16.5 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	Index	Lin/Log	Index	Lin/Log	Index	Lin/Log	Index	Lin/Log	Index	Lin/Log	Index	Lin/Log	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 10 Intensity index vs. PWM pulse width (normal polarity)

Index	Lin/Log	Index	Lin/Log	Index	Lin/Log	Index	Lin/Log	Index	Lin/Log	Index	Lin/Log	Index	Lin/Log	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 11 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).

3.17 Smart Wake Up

The SX8645 offers a smart wake up mechanism which allows waking-up from the Doze low power mode to the Active mode in a secure/controlled way and not by any unintentional sensor activation.

Until the full correct wake-up sequence is entered, the SX8645 will remain in Doze mode. Any wrong key implies the whole sequence to be entered again.

A sequence of up to 6 keys can be defined. Each key must be followed by a release to be validated.

The smart wake-up mechanism can also be disabled which implies that Doze mode can hence only be exited from GPI or I2C command.

4 PIN DESCRIPTIONS

4.1 Introduction

This chapter describes briefly the pins of the SX8645, 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, ..., CAP11

The capacitance sensor pins (CAP0, CAP1, ..., CAP11) 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 35 shows the simplified diagram of the CAP0, CAP1, ..., CAP11 pins.

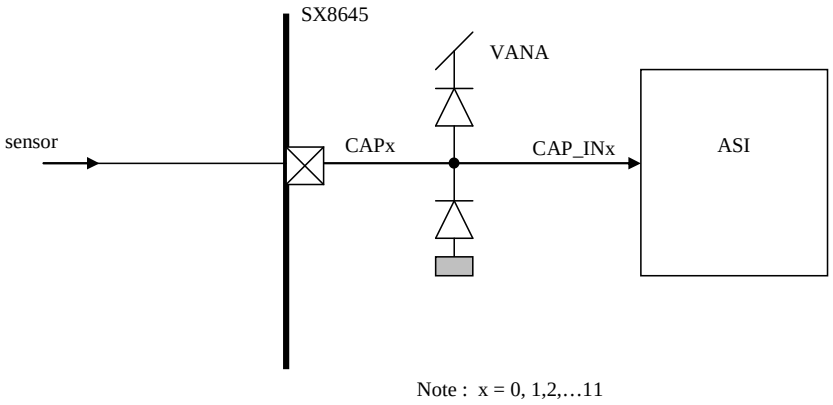


Figure 35 Simplified diagram of CAP0, CAP1, ..., CAP11

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 SX8645.

The CN and CP are protected to VANA and GROUND.

Figure 36 shows the simplified diagram of the CN and CP pins.

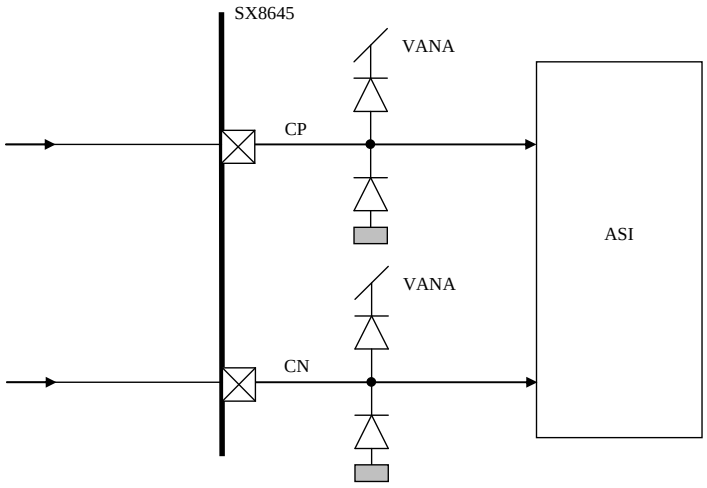


Figure 36 Simplified diagram of CN and CP

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 37 shows a simplified diagram of the INTB pin.

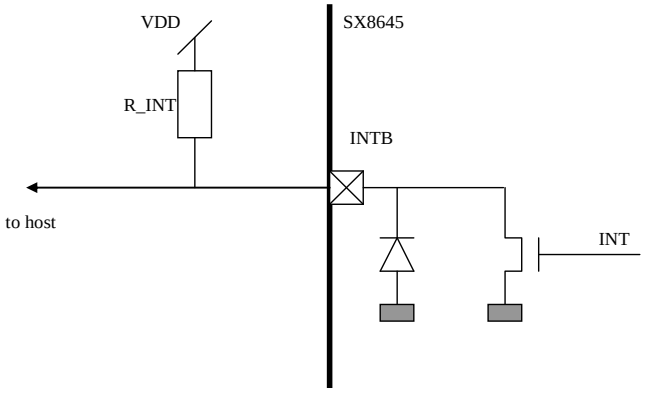


Figure 37 Simplified diagram of INTB

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 38 shows the simplified diagram of the SCL pin.

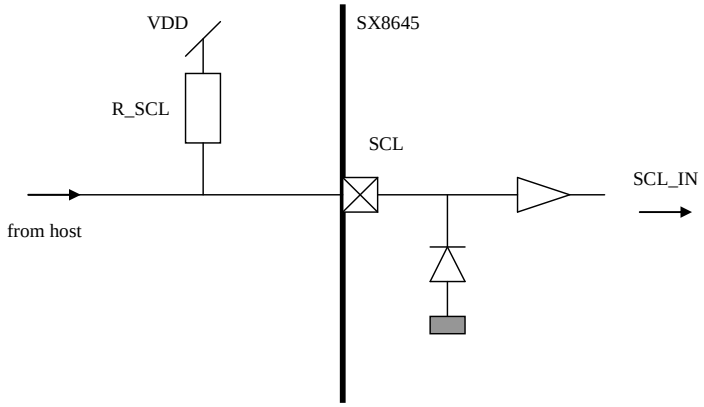


Figure 38 Simplified diagram of SCL

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 39 shows the simplified diagram of the SDA pin.

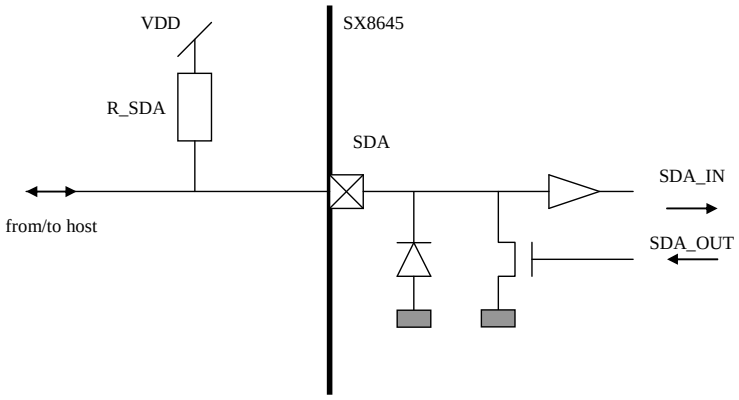


Figure 39 Simplified diagram of SDA

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 40 shows the simplified diagram of the RESETB pin controlled by the host.

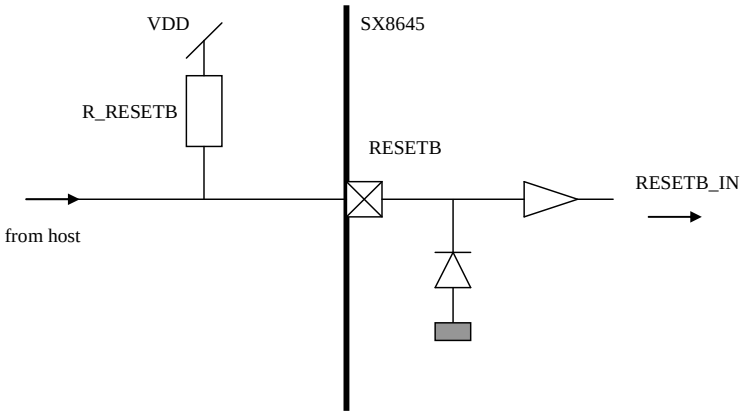


Figure 40 Simplified diagram of RESETB controlled by host

Figure 41 shows the RESETB without host control.

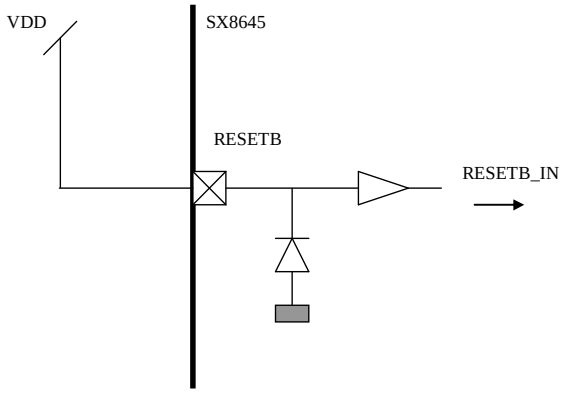


Figure 41 Simplified diagram of RESETB without host control

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 SX8645.
VDD has protection to GROUND.

Figure 42 shows a simplified diagram of the VDD pin.

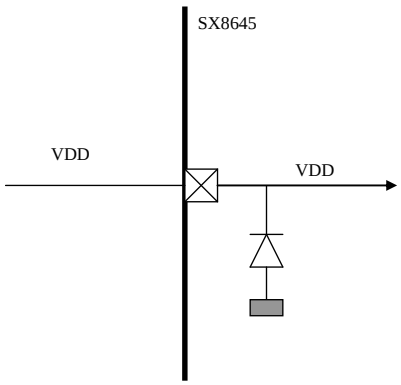


Figure 42 Simplified diagram of VDD

GND

The SX8645 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 43 shows a simplified diagram of the GND pin.

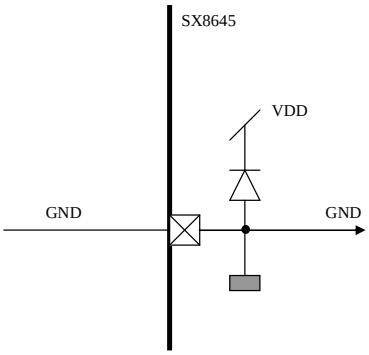


Figure 43 Simplified diagram of GND

VANA, VDIG

The SX8645 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 44 shows a simplified diagram of the VANA and VDIG pin.

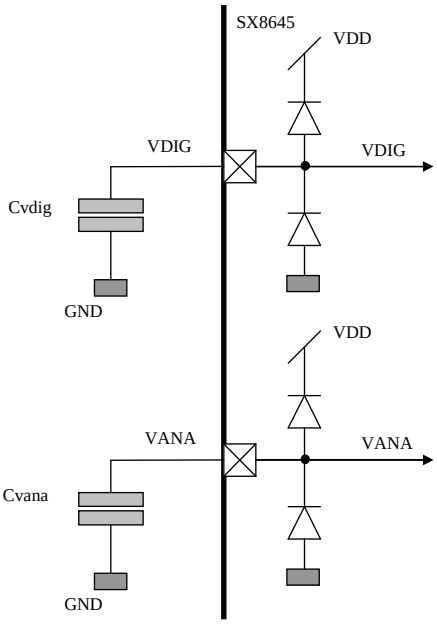


Figure 44 Simplified diagram of VANA and VDIG

4.5 General purpose IO pins

The SX8645 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 or GPP.

Figure 45 shows a simplified diagram of the GPIO pins.

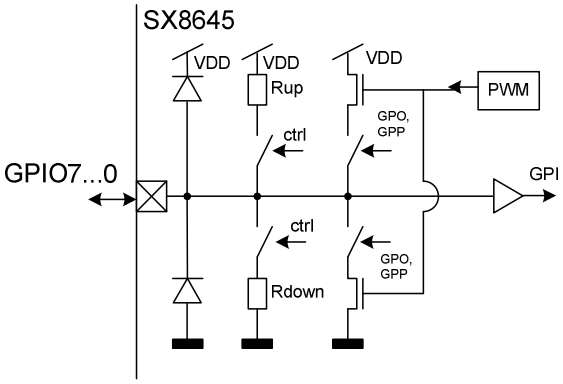


Figure 45 Simplified diagram of GPIO pins

5 DETAILED CONFIGURATION DESCRIPTIONS

5.1 Introduction

The SX8645 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 SX8645.

.

The SPM is split by functionality into 6 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,
- Wheel: related to the conversion from sensor data towards wheel information,
- Mapping: related to mapping of button and wheel information towards wake-up and 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 12 and Table 13 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
0x00	Reserved		0xxx
0x01	Reserved		0xxx
0x02	Reserved		0x18
0x03	Reserved		0xxx
0x04	General	I2CAddress	0x2B
0x05		ActiveScanPeriod	0x02
0x06		DozeScanPeriod	0x0D
0x07		PassiveTimer	0x00
0x08	Reserved		0x00
0x09	Capacitive Sensors	CapModeMisc	0x01
0x0A		CapMode11_8	0xFF
0x0B		CapMode7_4	0xF5
0x0C		CapMode3_0	0x55
0x0D		CapSensitivity0_1	0x00
0x0E		CapSensitivity2_3	0x00
0x0F		CapSensitivity4_5	0x00
0x10		CapSensitivity6_7	0x00
0x11		CapSensitivity8_9	0x00
0x12		CapSensitivity10_11	0x00
0x13		CapThresh0	0xA0
0x14		CapThresh1	0xA0
0x15		CapThresh2	0xA0
0x16		CapThresh3	0xA0
0x17		CapThresh4	0xA0
0x18		CapThresh5	0xA0
0x19		CapThresh6	0xA0
0x1A		CapThresh7	0xA0
0x1B		CapThresh8	0xA0
0x1C		CapThresh9	0xA0
0x1D		CapThresh10	0xA0
0x1E		CapThresh11	0xA0
0x1F		CapPerComp	0x00

Address	Name		default QSM value
0x20	Reserved		0x00
0x21	Button	BtnCfg	0x30
0x22		BtnAvgThresh	0x50
0x23		BtnCompNegThresh	0x50
0x24		BtnCompNegCntMax	0x01
0x25		BtnHysteresis	0x0A
0x26		BtnStuckAtTimeout	0x00
0x27	Wheel	WhlCfg	0x00
0x28		WhlStuckAtTimeout	0x00
0x29		WhlHysteresis	0x03
0x2A		Reserved	0xFF
0x2B		WhlNormMsb	0x01
0x2C		WhlNormLsb	0x40
0x2D		WhlAvgThresh	0x50
0x2E		WhlCompNegThresh	0x50
0x2F		WhlCompNegCntMax	0x01
0x30		WhlRotateThresh	0x02
0x31		WhlOffset	0x00
0x32		Reserved	0x00
0x33	Mapping	MapWakeupSize	0x00
0x34		MapWakeupValue0	0x00
0x35		MapWakeupValue1	0x00
0x36		MapWakeupValue2	0x00
0x37		MapAutoLight0	0xFE
0x38		MapAutoLight1	0x54
0x39		MapAutoLight2	0x32
0x3A		MapAutoLight3	0x10
0x3B		MapAutoLightGrp0Msb	0x00
0x3C		MapAutoLightGrp0Lsb	0x00
0x3D		MapAutoLightGrp1Msb	0x00
0x3E		MapAutoLightGrp1Lsb	0x00
0x3F		MapSegmentHysteresis	0x02

Table 12 SPM address map: 0x00…0x3F

Note

- '0xxx': write protected data

Address	Name		default QSM value
0x40	Gpio	GpioMode7_4	0x00
0x41		GpioMode3_0	0x00
0x42		GpioOutPwrUp	0x00
0x43		GpioAutoLight	0xFF
0x44		GpioPolarity	0x00
0x45		GpioIntensityOn0	0xFF
0x46		GpioIntensityOn1	0xFF
0x47		GpioIntensityOn2	0xFF
0x48		GpioIntensityOn3	0xFF
0x49		GpioIntensityOn4	0xFF
0x4A		GpioIntensityOn5	0xFF
0x4B		GpioIntensityOn6	0xFF
0x4C		GpioIntensityOn7	0xFF
0x4D		GpioIntensityOff0	0x00
0x4E		GpioIntensityOff1	0x00
0x4F		GpioIntensityOff2	0x00
0x50		GpioIntensityOff3	0x00
0x51		GpioIntensityOff4	0x00
0x52		GpioIntensityOff5	0x00
0x53		GpioIntensityOff6	0x00
0x54		GpioIntensityOff7	0x00
0x55		Reserved	0xFF
0x56		GpioFunction	0x00
0x57		GpioIncFactor	0x00
0x58		GpioDecFactor	0x00
0x59		GpioIncTime7_6	0x00
0x5A		GpioIncTime5_4	0x00
0x5B		GpioIncTime3_2	0x00
0x5C		GpioIncTime1_0	0x00
0x5D		GpioDecTime7_6	0x44
0x5E		GpioDecTime5_4	0x44
0x5F		GpioDecTime3_2	0x44

Address	Name		default QSM value
0x60	Gpio	GpioDecTime1_0	0x44
0x61		GpioOffDelay7_6	0x00
0x62		GpioOffDelay5_4	0x00
0x63		GpioOffDelay3_2	0x00
0x64		GpioOffDelay1_0	0x00
0x65		GpioPullUpDown7_4	0x00
0x66		GpioPullUpDown3_0	0x00
0x67		GpioInterrupt7_4	0x00
0x68		GpioInterrupt3_0	0x00
0x69		GpioDebounce	0x00
0x6A	Reserved		0x00
0x6B	Reserved		0x00
0x6C	Reserved		0x00
0x6D	Reserved		0x00
0x6E	Reserved		0x00
0x6F	Reserved		0x50
0x70	Reserved		0x46
0x71	Reserved		0x10
0x72	Reserved		0x45
0x73	Reserved		0x02
0x74	Reserved		0xFF
0x75	Reserved		0xFF
0x76	Reserved		0xFF
0x77	Reserved		0xD5
0x78	Reserved		0x55
0x79	Reserved		0x55
0x7A	Reserved		0x7F
0x7B	Reserved		0x23
0x7C	Reserved		0x22
0x7D	Reserved		0x41
0x7E	Reserved		0xFF
0x7F	SpmCrc*		0xE1

Table 13 SPM address map: 0x40…0x7F

Note*

- 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 and Wheel Information (Figure 8) 0x00: OFF (default) 0x01: 1 second … 0xFF: 255 seconds

Table 14 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	CapMode11_8	7:6	CAP11 Mode	Defines the mode of the CAP pin. 00: Disabled 01: Button 10: Reserved 11: Wheel	Default	Wheel
		5:4	CAP10 Mode			Wheel
		3:2	CAP9 Mode			Wheel
		1:0	CAP8 Mode			Wheel
0x0B	CapMode7_4	7:6	CAP7 Mode			Wheel
		5:4	CAP6 Mode			Wheel
		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 (default) 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			
		3:0	CAP5 Sensitivity			
0x10	CapSensitivity6_7	7:4	CAP6 Sensitivity			
		3:0	CAP7 Sensitivity			
0x11	CapSensitivity8_9	7:4	CAP8 Sensitivity			
		3:0	CAP9 Sensitivity			
0x12	CapSensitivity10_11	7:4	CAP10 Sensitivity			
		3:0	CAP11 Sensitivity			
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			

Capacitive Sensors Parameters				
Address	Name	Bits	Description	
0x1A	CapThresh7	7:0	CAP7 Touch Threshold	
0x1B	CapThresh8	7:0	CAP8 Touch Threshold	
0x1C	CapThresh9	7:0	CAP9 Touch Threshold	
0x1D	CapThresh10	7:0	CAP10 Touch Threshold	
0x1E	CapThresh11	7:0	CAP11 Touch Threshold	
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 15 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.

CapMode11_8, CapMode7_4, CapMode3_0:

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

	minimum	default	maximum
buttons	zero	six	eight
wheel	one (of four sensors)	one (of six sensors)	one (of twelve sensors)

Table 16 Possible CAP pin modes

Buttons and disabled CAP pins can be attributed freely (examples in Figure 46).

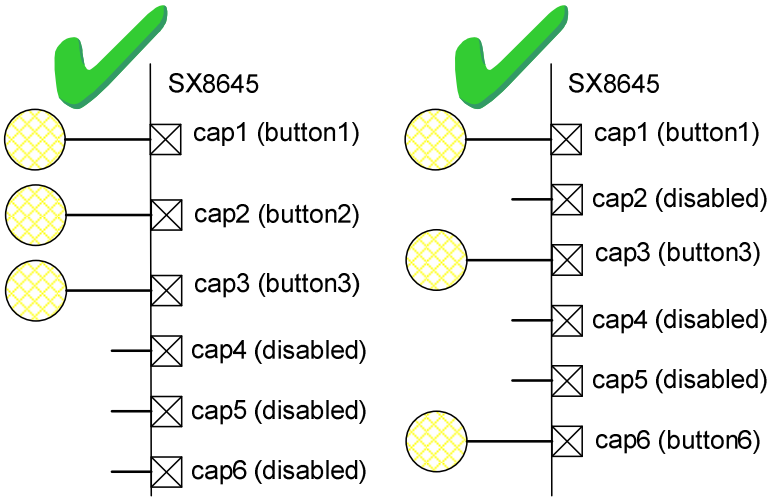


Figure 46 Button examples

Disabled CAP pins inside the wheel sensor attribution sequence are allowed, but CAP buttons inside a wheel are not allowed (see example Figure 47 with CAP3 in a correct and a not allowed configuration).

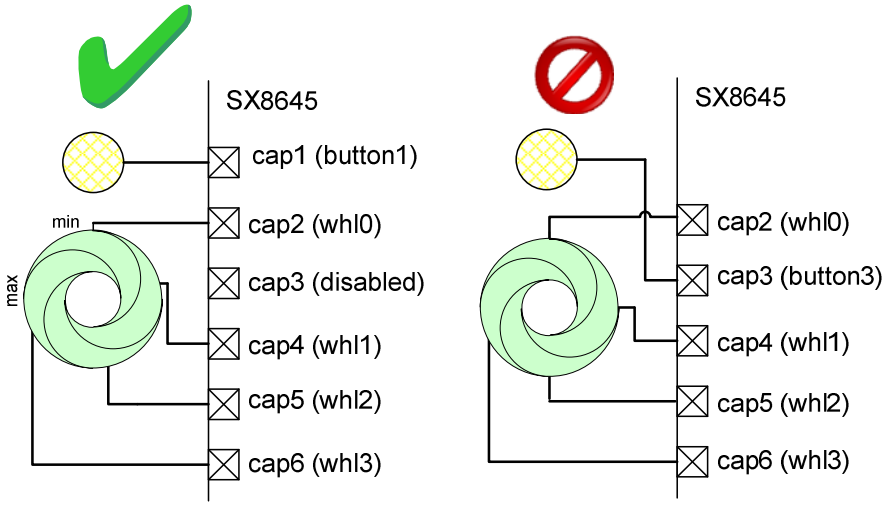


Figure 47 Button and Wheel good/bad configuration examples (I)

The physical order of the wheel sensors on the PCB should correspond to the incremental CAP pin numbers. Crossing wheel PCB sensors and CAP number is not allowed. Figure 48 shows a valid configuration and a wrong configuration where CAP5 andCAP6 are not routed correctly on the PCB.

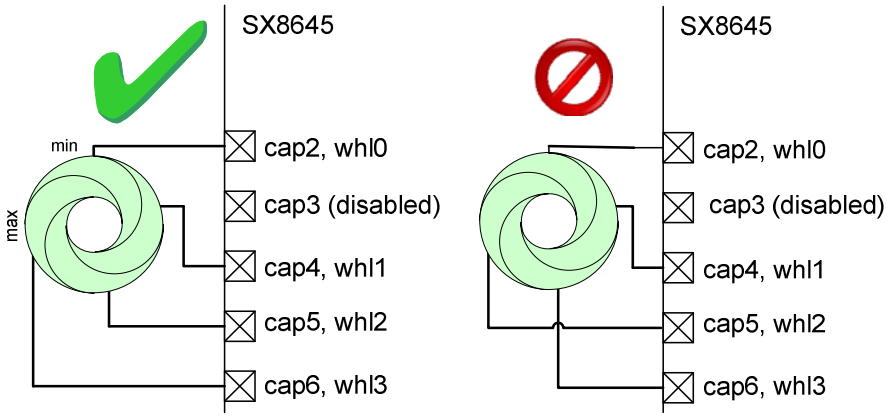


Figure 48 Button and Wheel good/bad configuration examples (II)

The minimum position of the wheel is associated to the CAP pin, attributed to the wheel, with the lowest index (in Figure 48 this is CAP2).
The maximum position of the wheel is associated to the CAP pin, attributed to the wheel, with the highest index (in Figure 48 this is CAP6).

CapSensitivity0_1, CapSensitivity2_3, CapSensitivity4_5, CapSensitivity6_7, CapSensitivity8_9, CapSensitivity10_11:

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 17.

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

Table 17 ASI Maximum Tick Levels

CapThresh0, CapThresh1, CapThresh2, CapThresh3, CapThresh4, CapThresh5, CapThresh6, CapThresh7, CapThresh8, CapThresh9, CapThresh10, CapThresh11:

For each CAP pin a threshold level can be set individually.
The threshold levels are used by the SX8645 for making touch and release decisions on e.g. touch or no-touch.
The details are explained in the sections for buttons and wheel.

CapPerComp:

The SX8645 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 wheel and buttons are released.

5.4 Button Parameters

Button Parameters			
Address	Name	Bits	Description
0x21	BtnCfg	7:6	Defines the buttons events reporting method. 00: Multiple reporting of all touches and releases (default) 01: Single reporting of the first button touch. Next button touches and releases are ignored until release of the first button. 10: Reserved 11: Reserved
		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)
		3:2	Defines the number of samples at the scan period for determining a release 00: OFF, use incoming sample (default) 01: 2 samples debounce 10: 3 samples debounce 11: 4 samples debounce
		1:0	Defines the number of samples at the scan period for determining a touch 00: OFF, use incoming sample (default) 01: 2 samples debounce 10: 3 samples debounce 11: 4 samples debounce
0x22	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
0x23	BtnCompNegThresh	7:0	Defines the negative offset compensation threshold. 0x00: 0 0x01: 4 … 0x50: 320 (default) … 0xFF: 1020
0x24	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
0x25	BtnHysteresis	7:0	Defines the button hysteresis corresponding to a percentage of the CAP thresholds (defined in Table 18). 0x00: 0% … 0x0A: 10% (default) … 0x64: 100% All buttons use the same hysteresis
0x26	BtnStuckAtTimeout	7:0	Defines the stuck at timeout. 0x00: OFF (default)

Button Parameters			
Address	Name	Bits	Description
			0x01: 1 second … 0xFF: 255 seconds

Table 18 Button Configuration Parameters

A reliable button operation requires a coherent setting of the registers.
Figure 49 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.

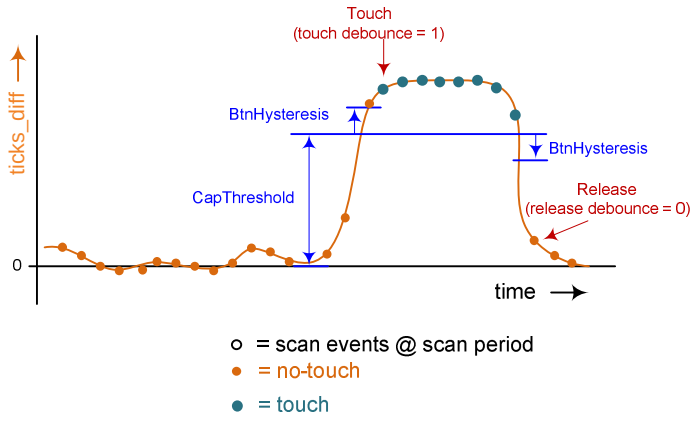


Figure 49 Touch and Release Example

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 49 the touch is validated after 2 samples (BtnCfg [1:0] = 01).
The release is detected immediately (BtnCfg [3:2] = 00) at the first sample which is below the threshold minus the hysteresis.

BtnCfg

The SX8645 can report all touches of multiple fingers or the SX8645 can be set to report only the first detected touch. In the later case all succeeding touches are ignored. The very first touch should be released before a next touch will be detected.

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 SX8645 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.

BtnAvgThresh

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

BtnAvgThresh. 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 BtnAvgThresh value simultaneously then the SX8645 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 50.

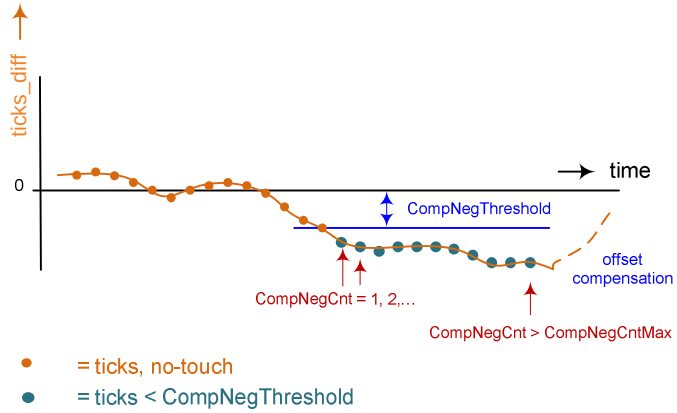


Figure 50 Negative Ticks Offset Compensation Trigger

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:

CapThreshold + CapThreshold * hysteresis.

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

CapThreshold - CapThreshold * 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.

5.5 Wheel Parameters

Wheel Parameters				
Address	Name	Bits	Description	
0x27	WhlCfg	7:4	Reserved	
		3:2	Defines the number of samples at the scan period for determining a release 00: OFF, use incoming sample (default) 01: 2 samples debounce 10: 3 samples debounce 11: 4 samples debounce	
		1:0	Defines the number of samples at the scan period for determining a touch 00: OFF, use incoming sample (default) 01: 2 samples debounce 10: 3 samples debounce 11: 4 samples debounce	
0x28	WhlStuckAtTimeout	7:0	Defines the stuck at timeout. 0x00: OFF (default) 0x01: 1 second … 0xFF: 255 seconds	
0x29	WhlHysteresis	7:0	Defines the Wheel Touch/Release Hysteresis. 0x00: 0 0x01: 4 … 0x03: 12 (default) … 0xFF: 1020	
0x2B	WhlNormMsb	7:0	Wheel Norm Msb	Defines the 16 bits wheel norm (default 0x0140)
0x2C	WhlNormLsb	7:0	Wheel Norm Lsb	
0x2D	WhlAvgThresh	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	
0x2E	WhlCompNegThresh	7:0	Defines the negative offset compensation threshold. 0x00: 0 0x01: 4 … 0x50: 320 (default) … 0xFF: 1020	
0x2F	WhlCompNegCntMax	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: 255 samples	
0x30	WhlRotateThresh	7:0	Defines the threshold for detecting a rotate clockwise or counter clockwise. The threshold is a percentage of the maximum wheel position. 0x00: 0% … 0x02: 2% (default)	

Wheel Parameters			
Address	Name	Bits	Description
			… 0x64: 100% A succeeding position difference, at the scan period, above the threshold is considered as a rotate clockwise or counter clockwise.

Table 19 Wheel Parameters

The pressure represents the finger touch on the sensors of the wheel and it used to determine if a wheel is touched or released.

$$WhlPressure = \sum_{i=0}^{N-1} (ticks\_diff(i) - CapThresh(i))$$

- N is the number of sensors,
- A sensor with ticks smaller than the CapThreshold is not taken into account for calculating the pressure

In case the pressure equals zero the wheel status is released.
In case the pressure is larger as the Wheel Hysteresis the wheel status is touched.

The position of a finger on a wheel is calculated by the centre of gravity algorithm.

$$WhlPos = \frac{WhlNorm}{32} * \frac{\sum_{i=0}^{N-1} i * (ticks\_diff(i) - CapThresh(i))}{\sum_{i=0}^{N-1} (ticks\_diff(i) - CapThresh(i))}$$

- N is the number of sensors,
- A sensor with ticks smaller than the CapThreshold is not taken into account for calculating the position,
- WhlNorm[15:0] is a 16 bit number determined by WhlNormMsb[15:8] and WhlNormLsb[7:0].
- WhlPos is the wheel position (16 bits) which can be read by the host over the I2C registers WhlPosMsb and WhlPosLsb

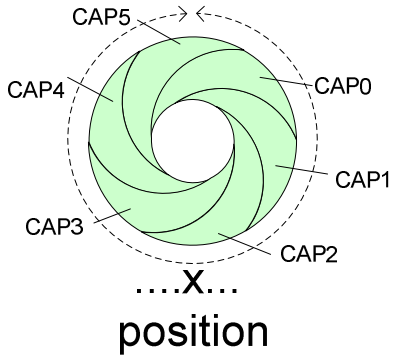


Figure 51 Wheel Position

Figure 51 shows an example of a wheel composed of 6 sensors (CAP0, CAP1… CAP5).
The default wheel norm value 320 (WhlNormMsb = 0x01, WhlNormLsb = 0x40), is taken for the example.
A touch on CAP0 gives the wheel position: 0.
A touch on CAP1 gives the wheel position: 10.
A touch on CAP5 gives the wheel position: 50.

If a touch occurs on CAP0 and CAP1 the centre of gravity algorithm will interpolate.
Assuming the touch is identically distributed on CAP0 and CAP1 then the position will be: 5
Assuming the touch is identically distributed on CAP1 and CAP2 then the position will be: 15
Assuming the touch is identically distributed on CAP4 and CAP5 then the position will be: 55

The minimum position of a wheel equals 0.
The maximum position is obtained if the finger is very slightly on CAP5 and heavily on CAP0.

The maximum position (WhlPosMax) is defined by:

$$WhlPosMax = \frac{WhlNorm}{32} \times N$$

with:
N is the number of sensors in the wheel

Slow varying wheel ticks due to environmental changes are handled as buttons in the previous section.
If the ticks pass below the wheel negative threshold for more than the compensation negative max counter then an offset compensation phase will be triggered.
If the ticks pass above the wheel average positive threshold then the averaging filters will be held.

A finger that moves very slowly over the wheel is not considered as a rotation. The status rotate clockwise and rotate counter clockwise will not be set.
A finger that moves faster on the wheel will change the rotation status.
A rotation is detected if the difference of the position for two succeeding samples at the scanning rate goes beyond the rotation threshold (WhlRotateThresh). A large rotation threshold requires very rapid finger rotations, while a small rotation threshold detects more easily rotations but gets sensitive to noise variations as well.

5.6 Mapping Parameters

Mapping Parameters				
Address	Name	Bits	Description	
0x33	MapWakeupSize	7:3	Reserved	
		2:0	Doze -> Active wake up sequence size. 0: Any sensor event (default) 1: key0 2: key0, key1 … 6: key0, key1,…key5 7: No sensor event, only GPI or I2C cmd can exit Doze mode Each key must be followed by a release to be validated. Any other sensor event before the release is ignored. Any wrong key implies the whole sequence to be entered again.	
0x34	MapWakeupValue0	7:4	key5	Defines the sensor event associated to each key. 0x00: Btn0 (default) 0x01: Btn1 … 0x0B: Btn11 0x0C: Wheel Touch 0x0D: Rotate Counter Clockwise 0x0E: Rotate Clockwise 0x0F: Reserved
		3:0	key4	
0x35	MapWakeupValue1	7:4	key3	
		3:0	key2	
0x36	MapWakeupValue2	7:4	key1	
		3:0	key0	
0x37	MapAutoLight0	7:4	GPIO[7]	Defines the mapping between GPOs (with Autolight ON) and sensor events. 0x00: Btn0 (default) 0x01: Btn1 … 0x0B: Btn11 0x0C: Group0 as defined by MapAutoLightGrp0 0x0D: Group1 as defined by MapAutoLightGrp1 0x0E: Rotate Counter Clockwise 0x0F: Rotate Clockwise Several GPOs can be mapped to the same sensor event and will be controlled simultaneously.
		3:0	GPIO[6]	
0x38	MapAutoLight1	7:4	GPIO[5]	
		3:0	GPIO[4]	
0x39	MapAutoLight2	7:4	GPIO[3]	
		3:0	GPIO[2]	
0x3A	MapAutoLight3	7:4	GPIO[1]	
		3:0	GPIO[0]	
0x3B	MapAutoLightGrp0Msb	7	Reserved	
		6	Segment	Defines Group0 sensor events: 0: OFF (default) 1: ON If any of the enabled sensor events occurs the Group0 event will occur as well. All sensors events within the group can be independently set except wheel event Segment which is exclusive (ie must be the only one enabled to be used)
		5	Rotate Clockwise	
		4	Rotate Counter Clockwise	
		3	Btn11	
		2	Btn10	
		1	Btn9	
		0	Btn8	
0x3C	MapAutoLightGrp0Lsb	7	Btn7	
		6	Btn6	
		5	Btn5	
		4	Btn4	
		3	Btn3	

Mapping Parameters				
Address	Name	Bits	Description	
		2	Btn2	
		1	Btn1	
		0	Btn0	
0x3D	MapAutoLightGrp1Msb	7	Reserved	
		6	Wheel Touch	Defines Group1 sensor events: 0: OFF (default) 1: ON If any of the enabled sensor events occurs the Group0 event will occur as well. All sensors events within the group can be independently set.
		5	Rotate Clockwise	
		4	Rotate Counter Clockwise	
		3	Btn11	
		2	Btn10	
		1	Btn9	
		0	Btn8	
0x3E	MapAutoLightGrp1Lsb	7	Btn7	
		6	Btn6	
		5	Btn5	
		4	Btn4	
		3	Btn3	
		2	Btn2	
		1	Btn1	
		0	Btn0	
0x3F	MapSegmentHysteresis	7:0	Defines the position hysteresis for detecting a segment change. The hysteresis is defined as a percentage of the maximum wheel position. 0x00: 0% … 0x02: 2% (default) … 0x64: 100% This hysteresis applies to all segments of the wheel.	

Table 20 Mapping Parameters

MapWakeupSize

The number of keys defining the wakeup sequence can be set from 1 to 6.
If the size is set to 0 then wakeup is done on any sensor event.
if the size is set to 6 then wakeup is done only by GPI or an I2C command.

MapWakeupValue0, MapWakeupValue1, MapWakeupValue2

For the wakeup sequence Btn2 -> Btn5 -> Btn6 ->Btn0 the required register settings are:

- MapWakeupSize set to 0x04,
- key0 = 0x2
- key1 = 0x5

=> MapWakeupValue2 set to 0x52

- key2 = 0x6

- key3 = 0x0
=> MapWakeupValue2 set to 0x06

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
Wheel Touch	Touch	Release
Wheel Rotation Clock Wise	Rotation Clock Wise	Rotation Clock Counter Wise or Release
Wheel Rotation Counter Clock Wise	Rotation Counter Clock Wise	Rotation Clock Wise or Release
Wheel Segment	Segment Touched	Segment Released

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)

When the Wheel Segment event is mapped, the number of GPOs mapped to it determines the number of wheel segments. The GPO with the lowest pin index is mapped on the segment with the smallest positions.
E.g. if two GPOs (e.g.GPO[0] and GPO[1]) are mapped to the Wheel Segment event then the wheel is split in two segments. GPO[0] will turn ON for a touch on the wheel segment [0, WhlPosMax/2] and GPO[1] for a touch on the wheel segment [WhlPosMax/2, WhlPosMax].

5.7 GPIO Parameters

GPIO Parameters				
Address	Name	Bits	Description	
0x40	GpioMode7_4	7:6	GPIO[7] Mode	Defines the GPIO mode. 00: GPO (default) 01: GPP 10: GPI 11: Reserved
		5:4	GPIO[6] Mode	
		3:2	GPIO[5] Mode	
		1:0	GPIO[4] Mode	
0x41	GpioMode3_0	7:6	GPIO[3] Mode	
		5:4	GPIO[2] Mode	
		3:2	GPIO[1] Mode	
		1:0	GPIO[0] Mode	
0x42	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) Bits corresponding to GPO pins with Autolight ON should be left to 0. Before being actually initialized GPIOs are set as inputs with pull up.
			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	
			GPIO[1] Output Value at Power Up	
			GPIO[0] Output Value at Power Up	
0x43	GpioAutoLight	7:0	GPIO[7] AutoLight	Enables Autolight in GPO mode 0 : OFF 1 : ON (default)
			GPIO[6] AutoLight	
			GPIO[5] AutoLight	
			GPIO[4] AutoLight	
			GPIO[3] AutoLight	
			GPIO[2] AutoLight	
			GPIO[1] AutoLight	
			GPIO[0] AutoLight	
0x44	GpioPolarity	7:0	GPIO[7] Output Polarity	Defines the polarity of the GPO and GPP pins. 0: Inverted (default) 1: Normal
			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	
0x45	GpioIntensityOn0	7:0	ON Intensity Index	Defines the ON intensity index 0x00: 0 0x01: 1
0x46	GpioIntensityOn1			
0x47	GpioIntensityOn2			

GPIO Parameters				
Address	Name	Bits	Description	
0x48	GpioIntensityOn3			…
0x49	GpioIntensityOn4			0xFF: 255 (default)
0x4A	GpioIntensityOn5			
0x4B	GpioIntensityOn6			
0x4C	GpioIntensityOn7			
0x4D	GpioIntensityOff0	7:0	OFF Intensity Index	Defines the OFF intensity index 0x00: 0 (default) 0x01: 1 … 0xFF: 255
0x4E	GpioIntensityOff1			
0x4F	GpioIntensityOff2			
0x50	GpioIntensityOff3			
0x51	GpioIntensityOff4			
0x52	GpioIntensityOff5			
0x53	GpioIntensityOff6			
0x54	GpioIntensityOff7			
0x56	GpioFunction	7:0	GPIO[7] Function	Defines the intensity index vs PWM pulse width function. 0: Logarithmic (default) 1: Linear
			GPIO[6] Function	
			GPIO[5] Function	
			GPIO[4] Function	
			GPIO[3] Function	
			GPIO[2] Function	
			GPIO[1] Function	
			GPIO[0] Function	
0x57	GpioIncFactor	7:0	GPIO[7] 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
			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	
0x58	GpioDecFactor	7:0	GPIO[7] 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
			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	
0x59	GpioIncTime7_6	7:4	GPIO[7] Fading Increment Time	Defines the fading increment time.

GPIO Parameters				
Address	Name	Bits	Description	
		3:0	GPIO[6] Fading Increment Time	0x0: OFF (default) 0x1: 0.5ms 0x2: 1ms … 0xF: 7.5ms The total fading in time will be: GpioIncTime*GpioIncFactor* (GpioIntensityOn – GpioIntensityOff)
0x5A	GpioIncTime5_4	7:4	GPIO[5] Fading Increment Time	
		3:0	GPIO[4] Fading Increment Time	
0x5B	GpioIncTime3_2	7:4	GPIO[3] Fading Increment Time	
		3:0	GPIO[2] Fading Increment Time	
0x5C	GpioIncTime1_0	7:4	GPIO[1] Fading Increment Time	
		3:0	GPIO[0] Fading Increment Time	
0x5D	GpioDecTime7_6	7:4	GPIO[7] 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)
		3:0	GPIO[6] Fading Decrement Time	
0x5E	GpioDecTime5_4	7:4	GPIO[5] Fading Decrement Time	
		3:0	GPIO[4] Fading Decrement Time	
0x5F	GpioDecTime3_2	7:4	GPIO[3] Fading Decrement Time	
		3:0	GPIO[2] Fading Decrement Time	
0x60	GpioDecTime1_0	7:4	GPIO[1] Fading Decrement Time	
		3:0	GPIO[0] Fading Decrement Time	
0x61	GpioOffDelay7_6	7:4	GPIO[7] OFF Delay	Defines the delay after GPO OFF trigger before fading out starts. 0x0: OFF (default) 0x1: 200ms 0x2: 400ms … 0xF: 3000ms
		3:0	GPIO[6] OFF Delay	
0x62	GpioOffDelay5_4	7:4	GPIO[5] OFF Delay	
		3:0	GPIO[4] OFF Delay	
0x63	GpioOffDelay3_2	7:4	GPIO[3] OFF Delay	
		3:0	GPIO[2] OFF Delay	
0x64	GpioOffDelay1_0	7:4	GPIO[1] OFF Delay	
		3:0	GPIO[0] OFF Delay	
0x65	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	
0x66	GpioPullUpDown3_0	7:6	GPIO[3] Pullup/down	
		5:4	GPIO[2] Pullup/down	
		3:2	GPIO[1] Pullup/down	
		1:0	GPIO[0] Pullup/down	
0x67	GpioInterrupt7_4	7:6	GPI[7] Interrupt	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
		5:4	GPI[6] Interrupt	
		3:2	GPI[5] Interrupt	
		1:0	GPI[4] Interrupt	
0x68	GpioInterrupt3_0	7:6	GPI[3] Interrupt	
		5:4	GPI[2] Interrupt	
		3:2	GPI[1] Interrupt	

GPIO Parameters				
Address	Name	Bits	Description	
		1:0	GPI[0] Interrupt	
0x69	GpioDebounce	7:0	GPI[7] Debounce	Enables the GPI debounce (done on 10 consecutive samples at 1ms). 0 : OFF (default) 1 : ON
			GPI[6] Debounce	
			GPI[5] Debounce	
			GPI[4] Debounce	
			GPI[3] Debounce	
			GPI[2] Debounce	
			GPI[1] Debounce	
			GPI[0] Debounce	

Table 22 GPIO Parameters

Table 23 resumes the applicable SPM and I2C parameters for each GPIO mode.

		GPI	GPP	GPO
SPM	GpioMode	X	X	X
	GpioOutPwrUp		X[1]	X[2]
	GpioAutoligth			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[3]
	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] Only if Autolight is OFF, else ignored

Table 23 Applicable SPM/I2C Parameters vs. GPIO Mode

6 I2C INTERFACE

The I2C implemented on the SX8645 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, wheel, GPIs, 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 SX8645 is powered down.
The SPM can be stored permanently in the NVM memory of the SX8645. 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 52.
After the start condition [S], the slave address (SA) is sent, followed by an eighth bit (‘0’) indicating a Write. The SX8645 then Acknowledges [A] that it is being addressed, and the Master sends an 8 bit Data Byte consisting of the SX8645 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].

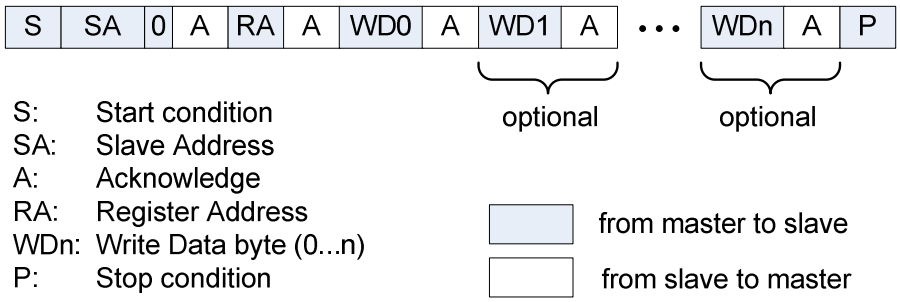


Figure 52 I2C write

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 53.
After the start condition [S], the slave address (SA) is sent, followed by an eighth bit ('0') indicating a Write. The SX8645 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 SX8645 responds with an Acknowledge [A] and the read Data byte (RD0). If the master needs to read more data it will acknowledge [A] and the SX8645 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].

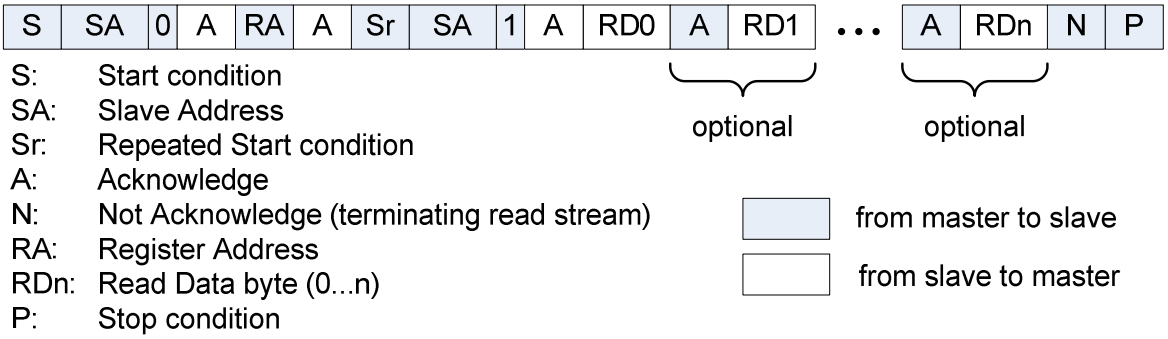


Figure 53 I2C read

6.3 I2C Registers Overview

Address	Name	R/W	Description
0x00	IrqSrc	read	Interrupt Source
0x01	CapStatMsb	read	Wheel/Button Status MSB
0x02	CapStatLsb	read	Button Status LSB
0x03	WhlPosMsb	read	Wheel Position MSB
0x04	WhlPosLsb	read	Wheel Position LSB
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 24 I2C Registers Overview

6.4 Status Registers

Address	Name	Bits	Description	
0x00	IrqSrc	7	Reserved	Interrupt source flags 0: Inactive (default) 1: Active INTB goes low if any of these bits is set. More than one bit can be set. Reading IrqSrc clears it together with INTB.
		6	NVM burn interrupt flag	
		5	SPM write interrupt flag	
		4	GPI interrupt flag	
		3	Wheel interrupt flag	
		2	Buttons interrupt flag	
		1	Compensation interrupt flag	
		0	Operating Mode interrupt flag	

Table 25 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[3] is set if a Wheel event occurred (touch, release, rotation clockwise, rotation counter clockwise or position change) . CapStatMsb, WhlPosMsb and WhlPosLsb show the detailed status of the Wheel.

IrqSrc[2] is set if a Button event occurred (touch or release if enabled). CapStatMsb and 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 either through automatic wakeup or via host request. CompOpmode shows the current operation mode.

Address	Name	Bits	Description	
0x01	CapStatMsb	7	Reserved	
		6	Wheel Rotation Clockwise	Wheel Rotation status 0: No rotation (default) 1: Rotation The status remains high as long as the wheel is touched and no opposite rotation has occurred.
		5	Wheel Rotation Counter Clockwise	
		4	Wheel Touched	Wheel Touch status 0: Released (default) 1: Touched
		3	Button 11 Touched	Button Touch status 0: Released (default) 1: Touched
		2	Button 10 Touched	
		1	Button 9 Touched	
		0	Button 8 Touched	
0x02	CapStatLsb	7	Button 7 Touched	
		6	Button 6 Touched	
		5	Button 5 Touched	
		4	Button 4 Touched	
		3	Button 3 Touched	
		2	Button 2 Touched	
		1	Button 1 Touched	
		0	Button 0 Touched	

Table 26 Wheel, Button status MSB/LSB

Address	Name	Bits	Description	
0x03	WhlPosMsb	7:0	Wheel Position[15:8]	Shows the current (touched) or last (released) wheel position[15:0] unsigned (default 0x00)
0x04	WhlPosLsb	7:0	Wheel Position[7:0]	

Table 27 Wheel position MSB/LSB

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 28 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 29 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

Table 30 I2C compensation, operation modes

* 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.

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 31 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 32 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 33 I2C GPP Intensity

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

Table 34 I2C Soft Reset

6.6 SPM Gateway Registers

The SX8645 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	Enables I2C SPM mode 00: OFF (default) 01: ON 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 35 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 36 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 37 SPM Key MSB

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

Table 38 SPM Key LSB

6.6.1 SPM Write Sequence

The SPM write can be done in any mode (Active, Doze, Sleep). Writing the SPM in Sleep is useful to avoid potential transient behaviors.

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].

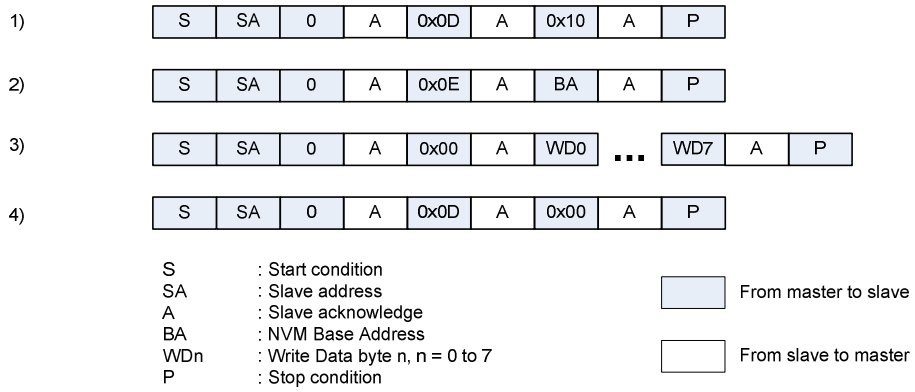


Figure 54: SPM Write Sequence

The complete SPM can be written by repeating 16 times the cycles shown in Figure 54 using base addresses 0x00, 0x08, 0x10, …, 0x70, 0x78. Between each sequence the host should wait for INTB (Active/Doze) or 30ms in Sleep.

In Active or Doze mode, once the SPM write sequence is actually applied, the INTB pin will be asserted and IrqSrc[5] set. In Sleep mode the SPM write can be actually applied with a delay of 30ms.

The host clears the interrupt and IrqSrc[5] by reading the IrqSrc register.

6.6.2 SPM Read Sequence

The SPM read can be done in any mode (Active, Doze, Sleep).

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].

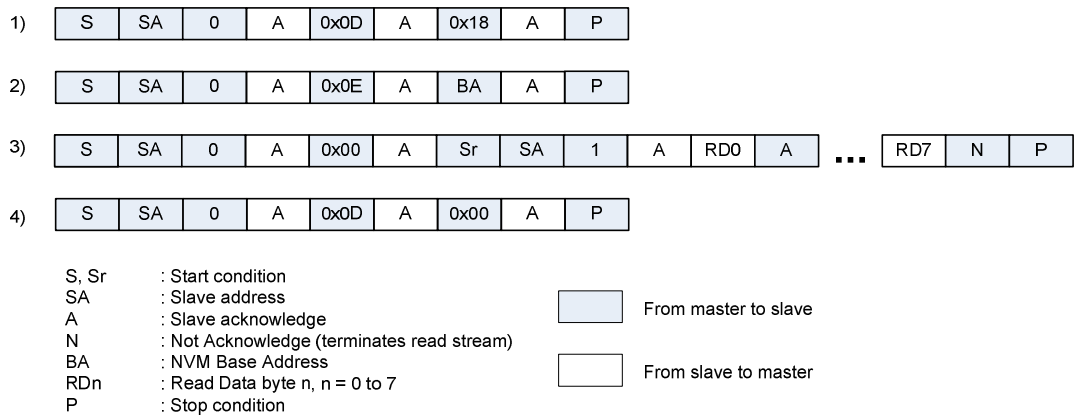


Figure 55: SPM Read Sequence

The complete SPM can be read by repeating 16 times the cycles shown in Figure 55 using base addresses 0x00, 0x08, 0x10, …, 0x70, 0x78.

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 NVM burn must be done in Active or Doze mode.
Once the NVM burn process is terminated IrqSrc[6] will be set and INTB asserted.
After a reset the burned NVM parameters will be copied into the SPM.

The number of times the NVM has been burned can be monitored by reading NvmCount from the I2C register SpmStat[2:0].

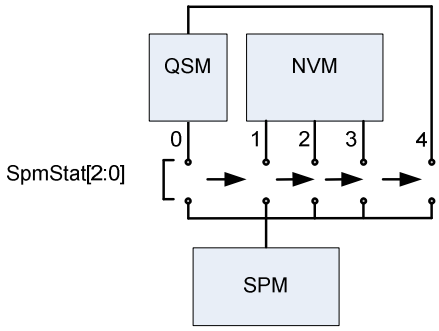


Figure 56 Simplified Diagram NvmCount

Figure 56 shows the simplified diagram of the NVM counter. The SX8645 is delivered with empty NVM and NvmCount set to zero. The SPM points to the QSM.
Each NVM burn will increase the NvmCount. At the fourth NVM burn the SX8645 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 57.

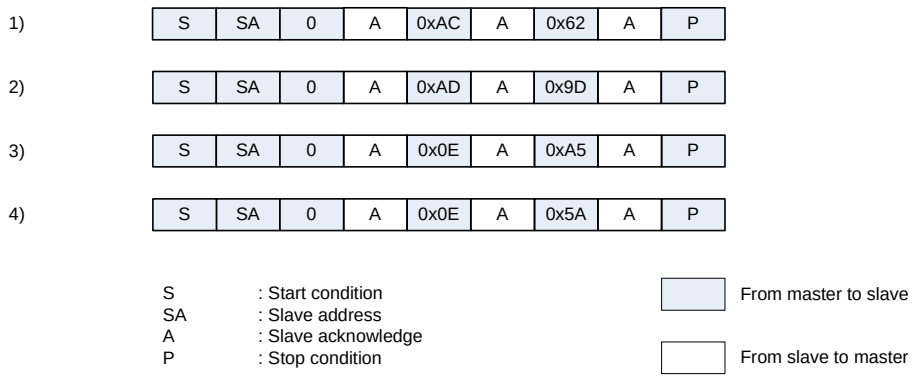


Figure 57: NVM burn procedure

7 APPLICATION INFORMATION

A typical application schematic is shown in Figure 58.

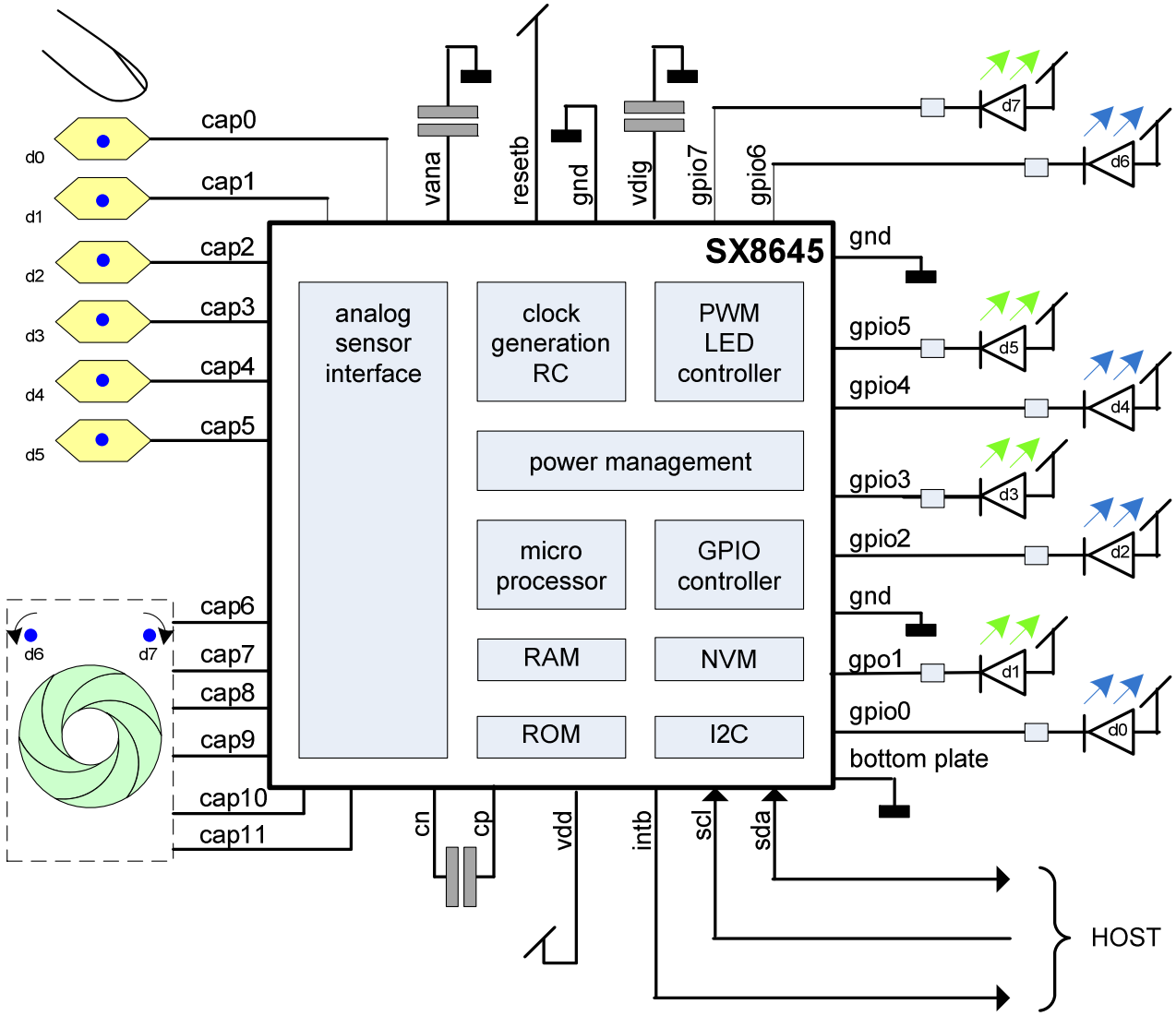
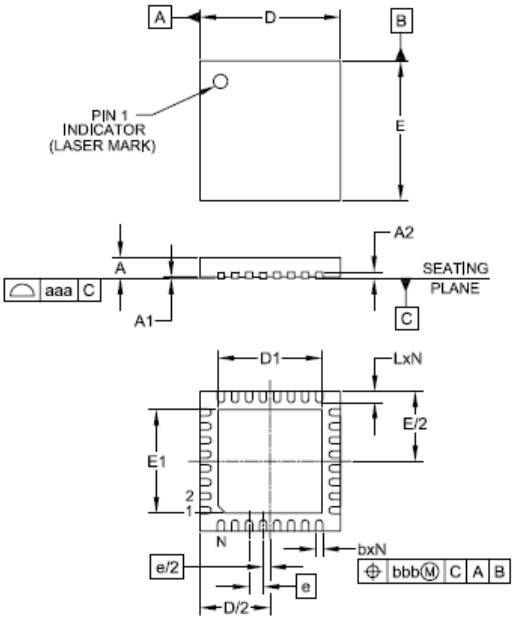


Figure 58 Typical Application

8 PACKAGING INFORMATION

8.1 Package Outline Drawing

SX8645 is assembled in a MLPQ-W32 package as shown in Figure 59.



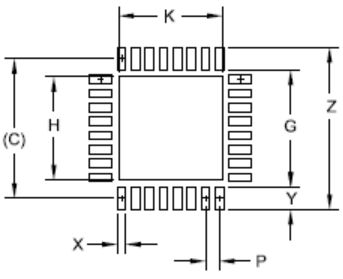
DIM	INCHES			MILLIMETERS		
	MIN	NOM	MAX	MIN	NOM	MAX
A	.028	-	.031	0.70	-	0.80
A1	.000	-	.002	0.00	-	0.05
A2	-	(.008)	-	-	(0.20)	-
b	.007	.010	.012	0.18	0.25	0.30
D	.193	.197	.201	4.90	5.00	5.10
D1	.142	.146	.150	3.60	3.70	3.80
E	.193	.197	.201	4.90	5.00	5.10
E1	.142	.146	.150	3.60	3.70	3.80
e	.020 BSC			0.50 BSC		
L	.012	.016	.020	0.30	0.40	0.50
N	32			32		
aaa	.003			0.08		
bbb	.004			0.10		

NOTES:
1. CONTROLLING DIMENSIONS ARE IN MILLIMETERS (ANGLES IN DEGREES).
2. COPLANARITY APPLIES TO THE EXPOSED PAD AS WELL AS THE TERMINALS.

Figure 59 Package Outline Drawing

8.2 Land Pattern

The land pattern of MLPQ-W32 package, 5 mm x 5 mm is shown in Figure 60.



DIM	INCHES	MILLIMETERS
C	(.197)	(5.00)
G	.165	4.20
H	.146	3.70
K	.146	3.70
P	.020	0.50
X	.012	0.30
Y	.031	0.80
Z	.228	5.80

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 60 Land Pattern

Contact Information

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