



## STMPE610

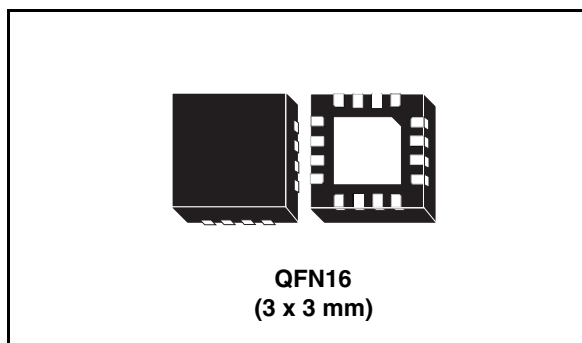
S-Touch<sup>®</sup>: advanced touchscreen controller  
with 6-bit port expander

### Features

- 6 GPIOs
- 1.8 - 3.3 V operating voltage
- Integrated 4-wire touchscreen controller
- Interrupt output pin
- Wakeup feature on each I/O
- SPI and I<sup>2</sup>C interface
- Up to 2 devices sharing the same bus in I<sup>2</sup>C mode (1 address line)
- 6-input 12-bit ADC
- 128-depth buffer touchscreen controller
- Touchscreen movement detection algorithm
- 25 kV air-gap ESD protection (system level)
- 4 kV HBM ESD protection (device level)

### Applications

- Portable media players
- Game consoles
- Mobile and smartphones
- GPS



### Description

The STMPE610 is a GPIO (general purpose input/output) port expander able to interface a main digital ASIC via the two-line bidirectional bus (I<sup>2</sup>C). A separate GPIO expander is often used in mobile multimedia platforms to solve the problems of the limited amount of GPIOs typically available on the digital engine.

The STMPE610 offers great flexibility, as each I/O can be configured as input, output or specific functions. The device has been designed with very low quiescent current and includes a wakeup feature for each I/O, to optimize the power consumption of the device.

A 4-wire touchscreen controller is built into the STMPE610. The touchscreen controller is enhanced with a movement tracking algorithm to avoid excessive data, 128 x 32 bit buffer and a programmable active window feature.

Table 1. Device summary

Order code	Package	Packaging
STMPE610QTR	QFN16	Tape and reel

# Contents

<b>1</b>	<b>STMPE610 functional overview</b>	<b>4</b>
<b>2</b>	<b>Pin configuration and functions</b>	<b>5</b>
2.1	Pin functions	6
<b>3</b>	<b>I2C and SPI interface</b>	<b>7</b>
3.1	Interface selection	7
<b>4</b>	<b>I2C interface</b>	<b>8</b>
4.1	I2C features	9
4.2	Data input	10
4.3	Read operation	11
4.4	Write operations	12
<b>5</b>	<b>SPI interface</b>	<b>13</b>
5.1	SPI protocol definition	13
5.1.1	Register read	13
5.1.2	Register write	14
5.1.3	Termination of data transfer	14
5.2	SPI timing modes	14
5.2.1	SPI timing definition	14
<b>6</b>	<b>STMPE610 registers</b>	<b>16</b>
<b>7</b>	<b>System and identification registers</b>	<b>18</b>
<b>8</b>	<b>Interrupt system</b>	<b>20</b>
<b>9</b>	<b>Analog-to-digital converter</b>	<b>24</b>
<b>10</b>	<b>Touchscreen controller</b>	<b>28</b>
10.1	Driver and switch control unit	28
10.2	Touch detect delay	33

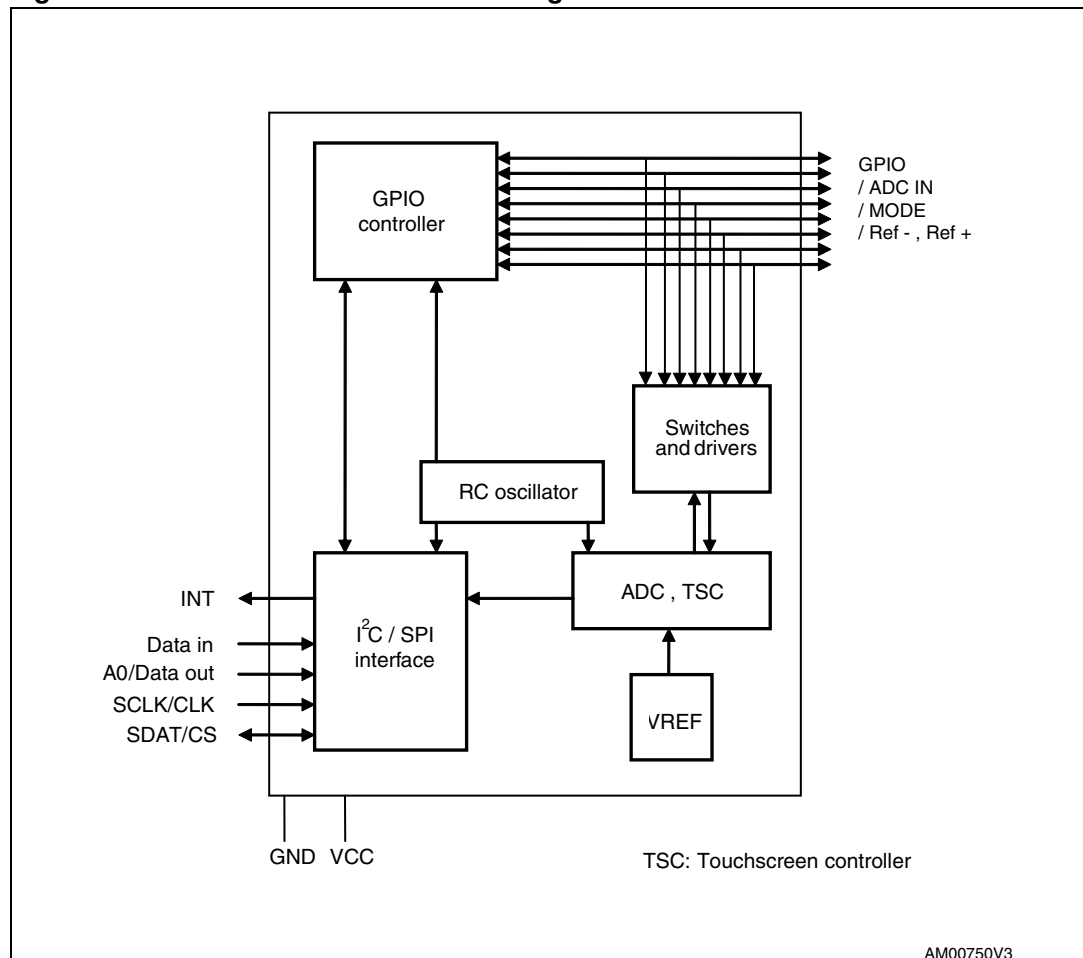
<b>11</b>	<b>Touchscreen controller programming sequence .....</b>	<b>40</b>
<b>12</b>	<b>GPIO controller .....</b>	<b>42</b>
	12.0.1 Power-up reset (POR) .....	44
<b>13</b>	<b>Maximum rating .....</b>	<b>45</b>
	13.1 Recommended operating conditions .....	45
<b>14</b>	<b>Electrical specifications .....</b>	<b>47</b>
<b>15</b>	<b>Package mechanical data .....</b>	<b>49</b>
<b>16</b>	<b>Revision history .....</b>	<b>55</b>

# 1 STMPE610 functional overview

The STMPE610 consists of the following blocks:

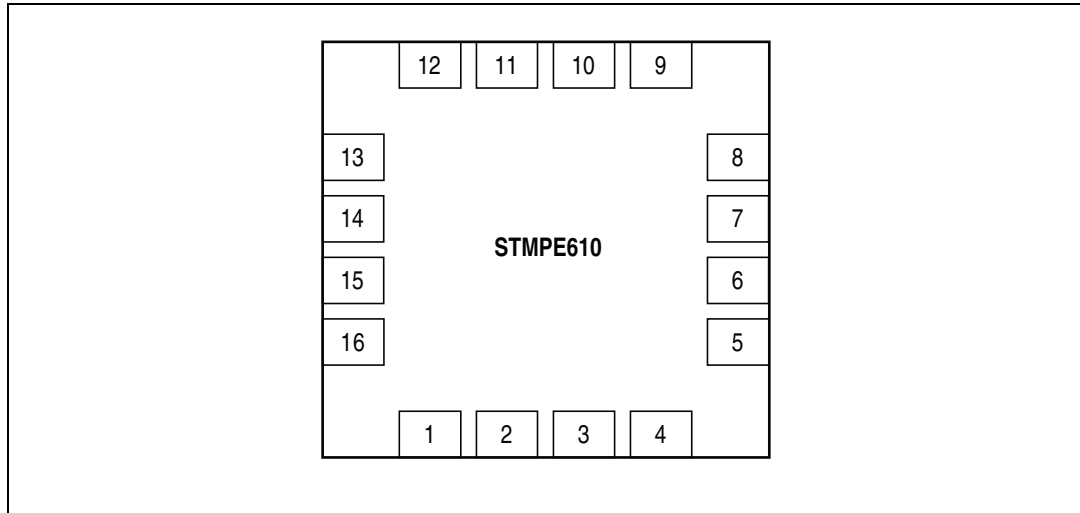
- I<sup>2</sup>C and SPI interface
- Analog-to-digital converter (ADC)
- Touchscreen controller (TSC)
- Driver and switch control unit
- GPIO controller

**Figure 1. STMPE610 functional block diagram**



## 2 Pin configuration and functions

**Figure 2. STMPE610 pin configuration (top through view)**



**Table 2. Pin assignments**

Pin	Name	Function
1	Y-	Y-/GPIO-7
2	INT	Interrupt output ( $V_{CC}$ domain, open drain)
3	A0/Data Out	I <sup>2</sup> C address in Reset, Data out in SPI mode ( $V_{CC}$ domain)
4	SCLK	I <sup>2</sup> C/SPI clock ( $V_{CC}$ domain)
5	SDAT	I <sup>2</sup> C data/SPI CS ( $V_{CC}$ domain)
6	$V_{CC}$	1.8 –3.3 V supply voltage
7	Data in	SPI Data In ( $V_{CC}$ domain)
8	NC	—
9	Mode	MODE In RESET state, MODE selects the type of serial interface "0" - I <sup>2</sup> C "1" - SPI
10	GND	Ground
11	IN2	IN2/GPIO-2
12	IN3	IN3/GPIO-3
13	X+	X+/GPIO-4
14	Vio	Supply for touchscreen driver and GPIO
15	Y+	Y+/GPIO-5
16	X-	X-/GPIO-6

## 2.1 Pin functions

The STMPE610 is designed to provide maximum features and flexibility in a very small pin-count package. Most of the pins are multi-functional. The following table shows how to select the pin's function.

**Table 3. IN2, IN3 pin configuration**

Pin / control register	GPIO_AF = 1	GPIO_AF = 0	
	ADC control 1 bit 1 = don't care	ADC control 1 bit 1 = 0	ADC control 1 bit 1 = 1
IN2	GPIO-2	ADC	External reference +
IN3	GPIO-3	ADC	External reference -

**Table 4. X, Y pin configuration**

Pin / control register	GPIO_AF = 1	GPIO_AF = 0	
	TSC control 1 bit 0 = don't care	TSC control 1 bit 0 = 0	TSC control 1 bit 0 = 1
X+	GPIO-4	ADC	TSC X+
Y+	GPIO-5	ADC	TSC Y+
X-	GPIO-6	ADC	TSC X-
Y-	GPIO-7	ADC	TSC Y-

### 3 I<sup>2</sup>C and SPI interface

#### 3.1 Interface selection

The STMPE610 interfaces with the host CPU via a I<sup>2</sup>C or SPI interface. The pin IN\_1 allows the selection of interface protocol at reset state.

Figure 3. STMPE610 interface

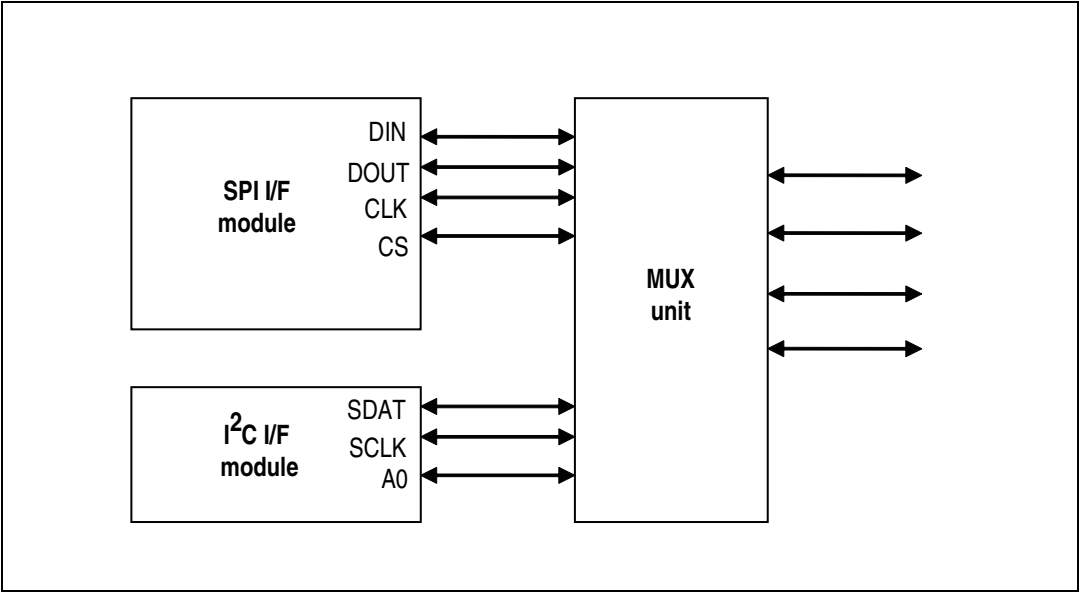


Table 5. Interface selection pins

Pin	I <sup>2</sup> C function	SPI function	Reset state
3	Address 0	Data out	CPHA for SPI
4	Clock	Clock	—
5	SDATA	CS	CPOL_N for SPI
7	—	Data in	—
9	MODE	I <sup>2</sup> C set to '0'	Set to '1' for SPI

# 4 I<sup>2</sup>C interface

The addressing scheme of STMPE610 is designed to allow up to 2 devices to be connected to the same I<sup>2</sup>C bus.

Figure 4. STMPE610 I<sup>2</sup>C interface

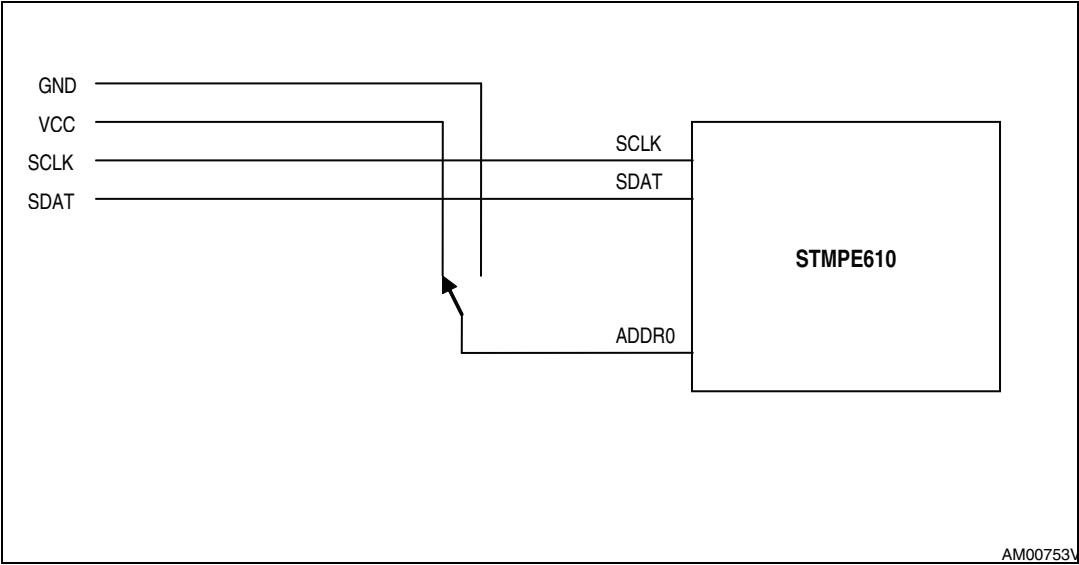
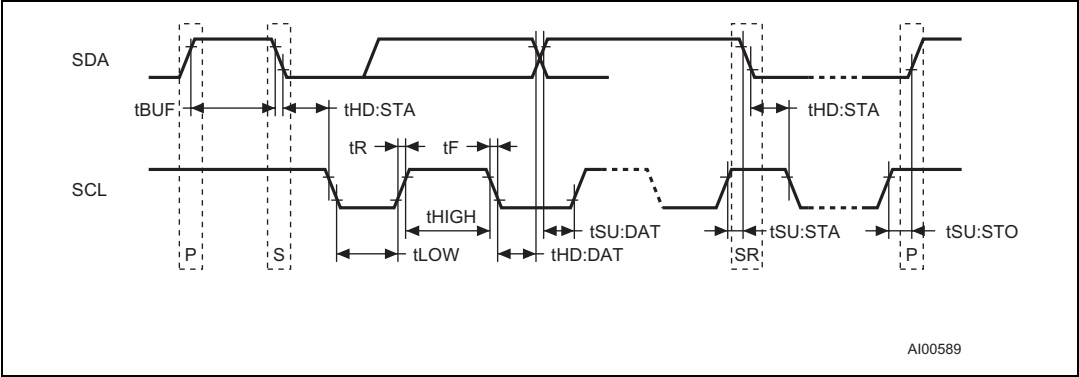


Table 6. I<sup>2</sup>C address

ADDR0	Address
0	0 x 82
1	0 x 88

For the bus master to communicate to the slave device, the bus master must initiate a Start condition and be followed by the slave device address. Accompanying the slave device address, is a read/write bit (R/W). The bit is set to 1 for read and 0 for write operation. If a match occurs on the slave device address, the corresponding device gives an acknowledge on the SDA during the 9<sup>th</sup> bit time. If there is no match, it deselects itself from the bus by not responding to the transaction.

Figure 5. I<sup>2</sup>C timing diagram





**Table 7. I<sup>2</sup>C timing**

Symbol	Parameter	Min	Typ	Max	Unit
f <sub>SCL</sub>	SCL clock frequency	0	—	400	kHz
t <sub>LOW</sub>	Clock low period	1.3	—	—	μs
t <sub>HIGH</sub>	Clock high period	600	—	—	ns
t <sub>F</sub>	SDA and SCL fall time	—	—	300	ns
t <sub>HD:STA</sub>	START condition hold time (after this period the first clock is generated)	600	—	—	ns
t <sub>SU:STA</sub>	START condition setup time (only relevant for a repeated start period)	600	—	—	ns
t <sub>SU:DAT</sub>	Data setup time	100	—	—	ns
t <sub>HD:DAT</sub>	Data hold time	0	—	—	μs
t <sub>SU:STO</sub>	STOP condition setup time	600	—	—	ns
t <sub>BUF</sub>	Time the bus must be free before a new transmission can start	1.3	—	—	μs

## 4.1 I<sup>2</sup>C features

The features that are supported by the I<sup>2</sup>C interface are listed below:

- I<sup>2</sup>C slave device
- Operates at 1.8 V
- Compliant to Philips I<sup>2</sup>C specification version 2.1
- Supports standard (up to 100 Kbps) and fast (up to 400 Kbps) modes

### Start condition

A Start condition is identified by a falling edge of SDA while SCLK is stable at high state. A Start condition must precede any data/command transfer. The device continuously monitors for a Start condition and does not respond to any transaction unless one is encountered.

### Stop condition

A Stop condition is identified by a rising edge of SDA while SCLK is stable at high state. A Stop condition terminates communication between the slave device and the bus master. A read command that is followed by NoAck can be followed by a Stop condition to force the slave device into idle mode. When the slave device is in idle mode, it is ready to receive the next I<sup>2</sup>C transaction. A Stop condition at the end of a write command stops the write operation to registers.

### Acknowledge bit

The acknowledge bit is used to indicate a successful byte transfer. The bus transmitter releases the SDA after sending eight bits of data. During the ninth bit, the receiver pulls

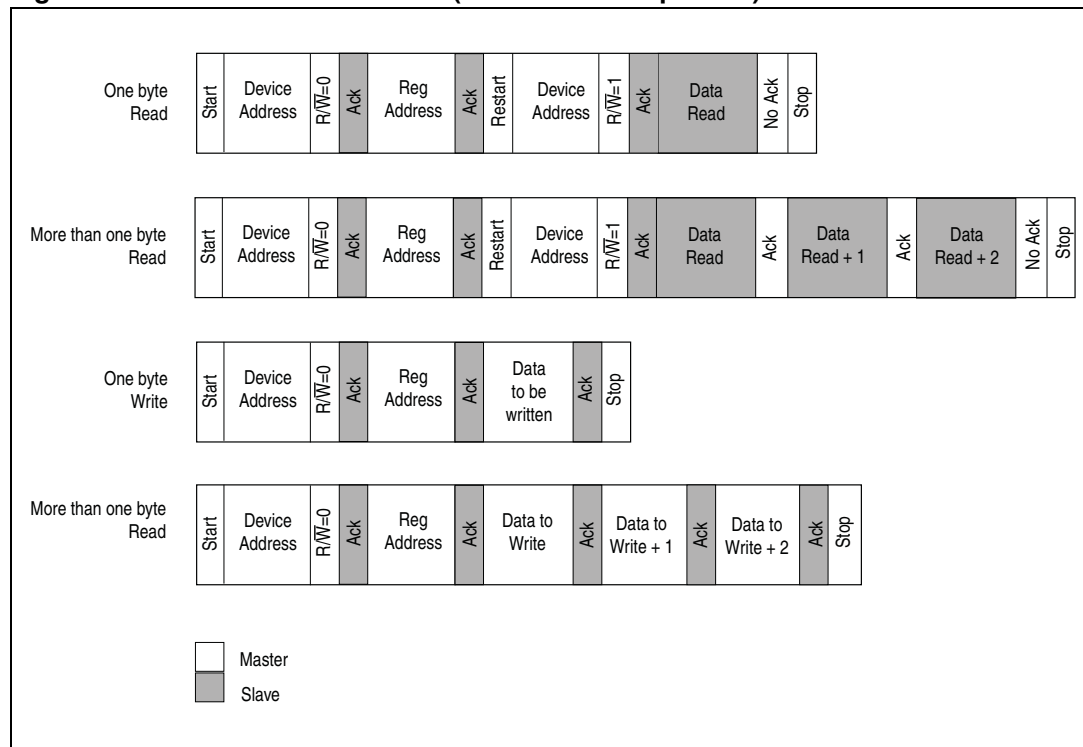
the SDATA low to acknowledge the receipt of the eight bits of data. The receiver may leave the SDATA in high state if it does not acknowledge the receipt of the data.

## 4.2 Data input

The device samples the data input on SDATA on the rising edge of the SCLK. The SDATA signal must be stable during the rising edge of SCLK and the SDATA signal must change only when SCLK is driven low.

**Table 8. Operating modes**

Mode	Byte	Programming sequence
Read	$\geq 1$	Start, Device address, $R/\overline{W} = 0$ , Register address to be read
		Restart, Device address, $R/\overline{W} = 1$ , Data Read, Stop
		If no Stop is issued, the Data Read can be continuously performed. If the register address falls within the range that allows an address auto-increment, then the register address auto-increments internally after every byte of data being read.
Write	$\geq 1$	Start, Device address, $R/\overline{W} = 0$ , Register address to be written, Data Write, Stop
		If no Stop is issued, the Data Write can be continuously performed. If the register address falls within the range that allows address auto-increment, then the register address auto-increments internally after every byte of data being written in. For those register addresses that fall within a non-incremental address range, the address will be kept static throughout the entire write operation. Refer to the memory map table for the address ranges that are auto and non-increment.

**Figure 6. Read and write modes (random and sequential)**

### 4.3 Read operation

A write is first performed to load the register address into the Address Counter but without sending a Stop condition. Then, the bus master sends a reStart condition and repeats the Device Address with the R/W bit set to 1. The slave device acknowledges and outputs the content of the addressed byte. If no additional data is to be read, the bus master must not acknowledge the byte and terminates the transfer with a Stop condition.

If the bus master acknowledges the data byte, then it can continue to perform the data reading. To terminate the stream of data bytes, the bus master must not acknowledge the last output byte, and be followed by a Stop condition. If the address of the register written into the Address Counter falls within the range of addresses that has the auto-increment function, the data being read will be coming from consecutive addresses, which the internal Address Counter automatically increments after each byte output. After the last memory address, the Address Counter 'rolls-over' and the device continues to output data from the memory address of 0x00. Similarly, for the register address that falls within a non-increment range of addresses, the output data byte comes from the same address (which is the address referred by the Address Counter).

#### Acknowledgement in read operation

For the above read command, the slave device waits, after each byte read, for an acknowledgement during the ninth bit time. If the bus master does not drive the SDA to a low state, then the slave device terminates and switches back to its idle mode, waiting for the next command.

## 4.4 Write operations

A write is first performed to load the register address into the Address Counter without sending a Stop condition. After the bus master receives an acknowledgement from the slave device, it may start to send a data byte to the register (referred by the Address Counter). The slave device again acknowledges and the bus master terminates the transfer with a Stop condition.

If the bus master needs to write more data, it can continue the write operation without issuing the Stop condition. Whether the Address Counter autoincrements or not after each data byte write depends on the address of the register written into the Address Counter. After the bus master writes the last data byte and the slave device acknowledges the receipt of the last data, the bus master may terminate the write operation by sending a Stop condition. When the Address Counter reaches the last memory address, it 'rolls-over' to the next data byte write.

## 5 SPI interface

The SPI interface in STMPE610 uses a 4-wire communication connection (DATA IN, DATA OUT, CLK, CS). In the diagram, “Data in” is referred to as MOSI (master out slave in) and “DATA out” is referred to as MISO (master in slave out).

### 5.1 SPI protocol definition

The SPI (serial peripheral interface) follows a byte sized transfer protocol. All transfers begin with an assertion of CS\_n signal (falling edge). The protocol for reading and writing is different and the selection between a read and a write cycle is dependent on the first captured bit on the slave device. A '1' denotes a read operation and a '0' denotes a write operation. The SPI protocol defined in this section is shown in Figure 3.

The following are the main features supported by this SPI implementation.

- Support of 1 MHz maximum clock frequency.
- Support for autoincrement of address for both read and write.
- Full duplex support for read operation.
- Daisy chain configuration support for write operation.
- Robust implementation that can filter glitches of up to 50 ns on the CS\_n and SCL pins.
- Support for all 4 modes of SPI as defined by the CPHA, CPOL bits on SPICON.

#### 5.1.1 Register read

The following steps need to be followed for register read through SPI.

1. Assert CS\_n by driving a '0' on this pin.
2. Drive a '1' on the first SCL launch clock on MOSI to select a read operation.
3. The next 7 bits on MOSI denote the 7-bit register address (MSB first).
4. The next address byte can now be transmitted on the MOSI. If the autoincrement bit is set, the following address transmitted on the MOSI is ignored. Internally, the address is incremented. If the autoincrement bit is not set, then the following byte denotes the address of the register to be read next.
5. Read data is transmitted by the slave device on the MISO (MSB first), starting from the launch clock following the last address bit on the MOSI.
6. Full duplex read operation is achieved by transmitting the next address on MOSI while the data from the previous address is available on MISO.
7. To end the read operation, a dummy address of all 0's is sent on MOSI.

### 5.1.2 Register write

The following steps need to be followed for register write through SPI.

1. Assert CS\_n by driving a '0' on this pin.
2. Drive a '0' on the first SCL launch clock on MOSI to select a write operation.
3. The next 7 bits on MOSI denote the 7-bit register address (MSB first).
4. The next byte on the MOSI denotes data to be written.
5. The following transmissions on MOSI are considered byte-sized data. The register address to which the following data is written depends on whether the autoincrement bit in the SPICON register is set. If this bit has been set previously, the register address is incremented for data writes.

### 5.1.3 Termination of data transfer

A transfer can be terminated before the last launch edge by deasserting the CS\_n signal. If the last launch clock is detected, it is assumed that the data transfer is successful.

## 5.2 SPI timing modes

The SPI timing modes are defined by CPHA and CPOL, CPHA and CPOL are read from the "SDAT" and "A0" pins during power-up reset. The following four modes are defined according to this setting.

**Table 9. SPI timing modes**

CPOL_N (SDAT pin)	CPOL	CPHA (ADDR pin)	Mode
1	0	0	0
1	0	1	1
0	1	0	2
0	1	1	3

The clocking diagrams of these modes are shown in ON reset. The device always operates in mode 0. Once the bits are set in the SPICON register, the mode change takes effect on the next transaction defined by the CS\_n pin being deasserted and asserted.

### 5.2.1 SPI timing definition

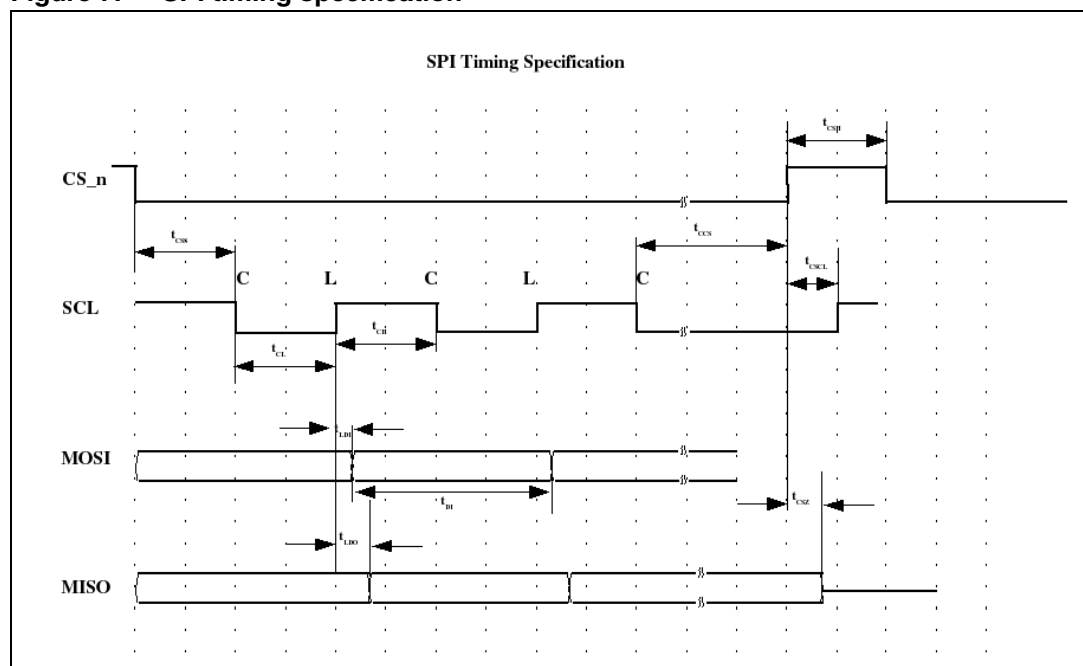
**Table 10. SPI timing specification**

Symbol	Description	Timing			Unit
		Min	Typ	Max	
t <sub>CSS</sub>	CS_n falling to first capture clock	1	—	—	μs
t <sub>CL</sub>	Clock low period	500	—	—	ns
t <sub>CH</sub>	Clock high period	500	—	—	ns

Table 10. SPI timing specification (continued)

Symbol	Description	Timing			Unit
		Min	Typ	Max	
$t_{LDI}$	Launch clock to MOSI data valid	—	—	20	ns
$t_{LDO}$	Launch clock to MISO data valid	—	—	330	$\mu$ s
$t_{DI}$	Data on MOSI valid	1	—	—	$\mu$ s
$t_{CCS}$	Last clock edge to CS_n high	1	—	—	$\mu$ s
$t_{CSH}$	CS_n high period	2	—	—	$\mu$ s
$t_{CSCL}$	CS_n high to first clock edge	300	—	—	ns
$t_{CSZ}$	CS_n high to tri-state on MISO	1	—	—	$\mu$ s

Figure 7. SPI timing specification



## 6 STMPE610 registers

This section lists and describes the registers of the STMPE610 device, starting with a register map and then provides detailed descriptions of register types.

**Table 11. Register summary map table**

Address	Register name	Bit	Type	Reset value	Function
0x00	CHIP_ID	16	R	0x0811	Device identification
0x02	ID_VER	8	R	0x03	Revision number 0x01 for engineering sample 0x03 for final silicon
0x03	SYS_CTRL1	8	R/W	0x00	Reset control
0x04	SYS_CTRL2	8	R/W	0x0F	Clock control
0x08	SPI_CFG	8	R/W	0x01	SPI interface configuration
0x09	INT_CTRL	8	R/W	0x00	Interrupt control register
0x0A	INT_EN	8	R/W	0x00	Interrupt enable register
0x0B	INT_STA	8	R	0x10	interrupt status register
0x0C	GPIO_EN	8	R/W	0x00	GPIO interrupt enable register
0x0D	GPIO_INT_STA	8	R	0x00	GPIO interrupt status register
0x0E	ADC_INT_EN	8	R/W	0x00	ADC interrupt enable register
0x0F	ADC_INT_STA	8	R	0x00	ADC interrupt status register
0x10	GPIO_SET_PIN	8	R/W	0x00	GPIO set pin register
0x11	GPIO_CLR_PIN	8	R/W	0x00	GPIO clear pin register
0x12	GPIO_MP_STA	8	R/W	0x00	GPIO monitor pin state register
0x13	GPIO_DIR	8	R/W	0x00	GPIO direction register
0x14	GPIO_ED	8	R/W	0x00	GPIO edge detect register
0x15	GPIO_RE	8	R/W	0x00	GPIO rising edge register
0x16	GPIO_FE	8	R/W	0x00	GPIO falling edge register
0x17	GPIO_AF	8	R/W	0x00	Alternate function register
0x20	ADC_CTRL1	8	R/W	0x9C	ADC control
0x21	ADC_CTRL2	8	R/W	0x01	ADC control
0x22	ADC_CAPT	8	R/W	0xFF	To initiate ADC data acquisition
0x30	ADC_DATA_CH0	16	R	0x0000	ADC channel 0
0x32	ADC_DATA_CH1	16	R	0x0000	ADC channel 1



Table 11. Register summary map table (continued)

Address	Register name	Bit	Type	Reset value	Function
0x38	ADC_DATA_CH4	16	R	0x0000	ADC channel 4
0x3A	ADC_DATA_CH5	16	R	0x0000	ADC channel 5
0x3C	ADC_DATA_CH6	16	R	0x0000	ADC channel 6
0x3E	ADC_DATA_CH7	16	R	0x0000	ADC channel 7
0x40	TSC_CTRL	8	R/W	0x90	4-wire touchscreen controller setup
0x41	TSC_CFG	8	R/W	0x00	Touchscreen controller configuration
0x42	WDW_TR_X	16	R/W	0x0FFF	Window setup for top right X
0x44	WDW_TR_Y	16	R/W	0x0FFF	Window setup for top right Y
0x46	WDW_BL_X	16	R/W	0x0000	Window setup for bottom left X
0x48	WDW_BL_Y	16	R/W	0x0000	Window setup for bottom left Y
0x4A	FIFO_TH	8	R/W	0x00	FIFO level to generate interrupt
0x4B	FIFO_STA	8	R/W	0x20	Current status of FIFO
0x4C	FIFO_SIZE	8	R	0x00	Current filled level of FIFO
0x4D	TSC_DATA_X	16	R	0x0000	Data port for touchscreen controller data access
0x4F	TSC_DATA_Y	16	R	0x0000	Data port for touchscreen controller data access
0x51	TSC_DATA_Z	8	R	0x0000	Data port for touchscreen controller data access
0x52	TSC_DATA_XYZ	32	R	0x00000000	Data port for touchscreen controller data access
0x56	TSC_FRACT_XYZ	8	RW	0x00	Select the range and accuracy of the pressure measurement
0x57	TSC_DATA	8	R	0x00	Data port for touchscreen controller data access
0x58	TSC_I_DRIVE	8	R/W	0x00	Touchscreen controller drive I
0x59	TSC_SHIELD	8	R/W	0x00	Touchscreen controller shield

## 7 System and identification registers

**Table 12. System and identification registers map**

Address	Register name	Bit	Type	Reset	Function
0x00	CHIP_ID	16	R	0x0811	Device identification
0x02	ID_VER	8	R	0x03	Revision number 0x01 for engineering sample 0x03 for final silicon
0x03	SYS_CTRL1	8	R/W	0x00	Reset control
0x04	SYS_CTRL2	8	R/W	0x0F	Clock control
0x08	SPI_CFG	8	R/W	0x01	SPI interface configuration

### CHIP\_ID

### Device identification

**Address:** 0x00  
**Type:** R  
**Reset:** 0x0811  
**Description:** 16-bit device identification

### ID\_VER

### Revision number

**Address:** 0x02  
**Type:** R  
**Reset:** 0x03  
**Description:** 16-bit revision number

### SYS\_CTRL1

### Reset control

7	6	5	4	3	2	1	0
RESERVED						SOFT_RESET	HIBERNATE

**Address:** 0x03  
**Type:** R/W  
**Reset:** 0x00  
**Description:** The reset control register enables to reset the device

[7:2] RESERVED

[1] SOFT\_RESET: Reset the STMPE610 using the serial communication interface

[0] HIBERNATE: Force the device into hibernation mode.

Forcing the device into hibernation mode by writing '1' to this bit would disable the hot-key feature. If the hot-key feature is required, use the default auto-hibernation mode.

**SYS\_CTRL2****Clock control**

7	6	5	4	3	2	1	0
-	-	-	-	RESERVED	GPIO_OFF	TSC_OFF	ADC_OFF

**Address:** 0x04**Type:** R/W**Reset:** 0x0F**Description:** This register enables to switch off the clock supply

[7:3] RESERVED

[2] GPIO\_OFF: Switch off the clock supply to the GPIO

1: Switches off the clock supply to the GPIO

[1] TSC\_OFF: Switch off the clock supply to the touchscreen controller

1: Switches off the clock supply to the touchscreen controller

[0] ADC\_OFF: Switch off the clock supply to the ADC

1: Switches off the clock supply to the ADC

**SPI\_CFG****SPI interface configuration**

7	6	5	4	3	2	1	0
RESERVED					AUTO_INCR	SPI_CLK_MOD1	SPI_CLK_MOD0

**Address:** 0x08**Type:** R/W**Reset:** 0x01**Description:** SPI interface configuration register

[7:3] RESERVED

[2] AUTO\_INCR:

This bit defines whether the SPI transaction follows an addressing scheme that internally autoincrements or not

[1] SPI\_CLK\_MOD1:

This bit reflects the value of the SCAD/A0 pin during power-up reset

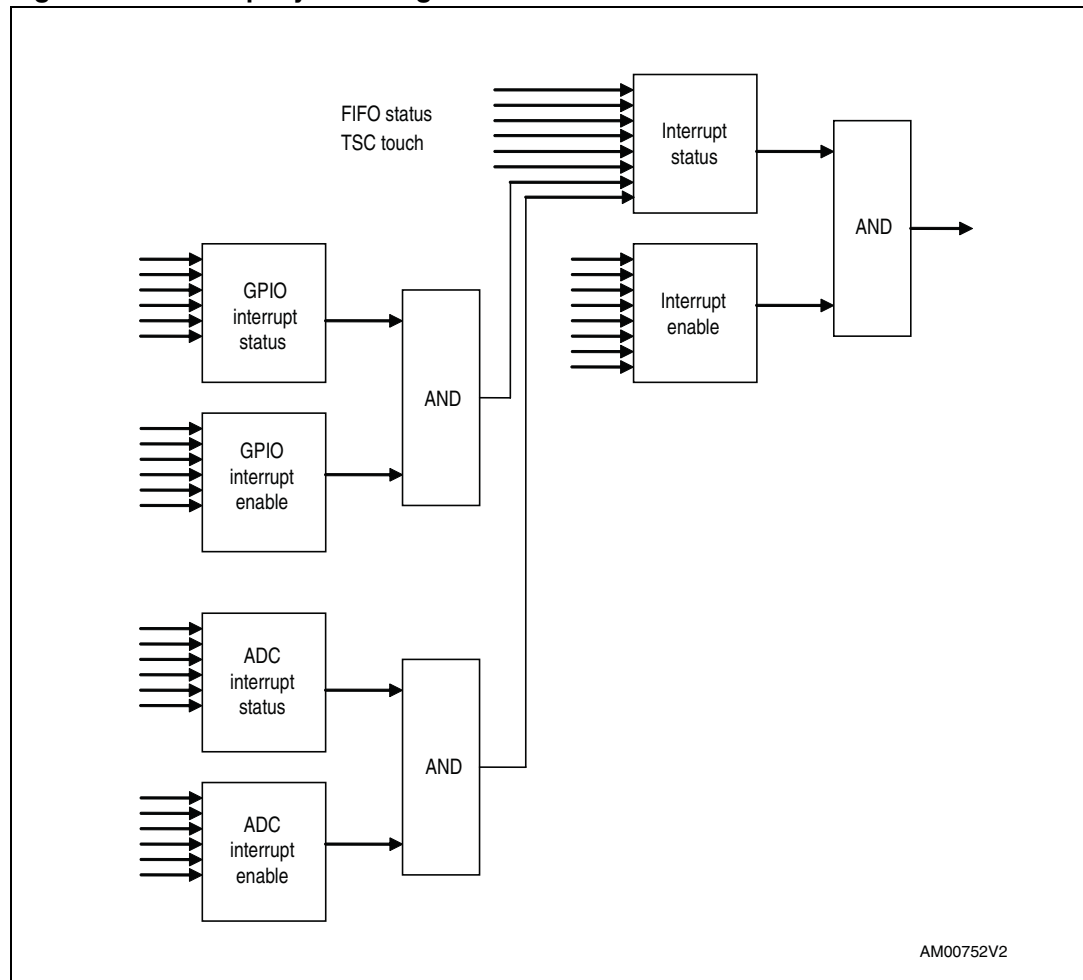
[0] SPI\_CLK\_MOD0:

This bit reflects the value of the SCAD/A0 pin during power-up reset

## 8 Interrupt system

The STMPE610 uses a 2-tier interrupt structure. The ADC interrupts and GPIO interrupts are ganged as a single bit in the “interrupt status register”. The interrupts from the touchscreen controller can be seen directly in the interrupt status register.

**Figure 8. Interrupt system diagram**



**INT\_CTRL****Interrupt control register**

7	6	5	4	3	2	1	0
RESERVED					INT_POLARITY	INT_TYPE	GLOBAL_INT

**Address:** 0x09

**Type:** R/W

**Reset:** 0x00

**Description:** The interrupt control register is used to enable the interruption from a system-related interrupt source to the host.

[7:3] RESERVED

[2] INT\_POLARITY: This bit sets the INT pin polarity

1: Active high/rising edge

0: Active low/falling edge

[1] INT\_TYPE: This bit sets the type of interrupt signal required by the host

1: Edge interrupt

0: Level interrupt

[0] GLOBAL\_INT: This is master enable for the interrupt system

1: Global interrupt

0: Stops all interrupts

**INT\_EN****Interrupt enable register**

7	6	5	4	3	2	1	0
GPIO	ADC	RESERVED	FIFO_EMPTY	FIFO_FULL	FIFO_OFLOW	FIFO_TH	TOUCH_DET

**Address:** 0x0A

**Type:** R/W

**Reset:** 0x00

**Description:** The interrupt enable register is used to enable the interruption from a system related interrupt source to the host.

[7] GPIO: Any enabled GPIO interrupts

[6] ADC: Any enabled ADC interrupts

[5] RESERVED

[4] FIFO\_EMPTY: FIFO is empty

[3] FIFO\_FULL: FIFO is full

[2] FIFO\_OFLOW: FIFO is overflowed

[1] FIFO\_TH: FIFO is equal or above threshold value

[0] TOUCH\_DET: Touch is detected

INT\_STA

Interrupt status register

7	6	5	4	3	2	1	0
GPIO	ADC	RESERVED	FIFO_EMPTY	FIFO_FULL	FIFO_OFLOW	FIFO_TH	TOUCH_DET

- Address:** 0x0B
- Type:** R
- Reset:** 0x10
- Description:** The interrupt status register monitors the status of the interruption from a particular interrupt source to the host. Regardless of whether the INT\_EN bits are enabled, the INT\_STA bits are still updated. Writing '1' to this register clears the corresponding bits. Writing '0' has no effect.

[7] GPIO: Any enabled GPIO interrupts

[6] ADC: Any enabled ADC interrupts

[5] RESERVED

[4] FIFO\_EMPTY: FIFO is empty

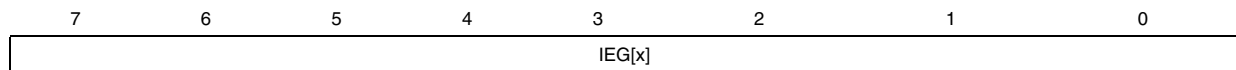
[3] FIFO\_FULL: FIFO is full

[2] FIFO\_OFLOW: FIFO is overflowed

[1] FIFO\_TH: FIFO is equal or above threshold value.

This bit is set when FIFO level equals to threshold value. It will only be asserted again if FIFO level drops to < threshold value, and increased back to threshold value.

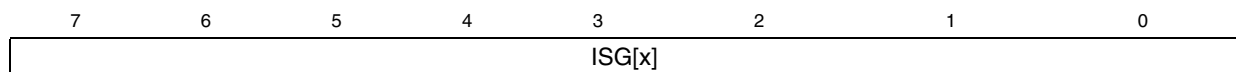
[0] TOUCH\_DET: Touch is detected

**GPIO\_INT\_EN****GPIO interrupt enable register****Address:** 0x0C**Type:** R/W**Reset:** 0x10

**Description:** The interrupt status register monitors the status of the interruption from a particular interrupt source to the host. Regardless of whether the IER bits are enabled, the ISR bits are still updated. Writing '1' to this register clears the corresponding bits. Writing '0' has no effect.

[7:0] IEG[x]: Interrupt enable GPIO mask (where x = 7 to 0)

1: Writing '1' to the IE[x] bit enables the interruption to the host

**GPIO\_INT\_STA****GPIO interrupt status register****Address:** 0x0D**Type:** R/W**Reset:** 0x00

**Description:** The GPIO interrupt status register monitors the status of the interruption from a particular GPIO pin interrupt source to the host. Regardless of whether or not the GPIO\_STA bits are enabled, the GPIO\_STA bits are still updated. The ISG[7:0] bits are the interrupt status bits corresponding to the GPIO[7:0] pins. Writing '1' to this register clears the corresponding bits. Writing '0' has no effect.

[7:0] ISG[x]: GPIO interrupt status (where x = 7 to 0)

**Read:**

Interrupt status of the GPIO[x]. Reading the register will clear any bits that have been set to '1'

**Write:**

Writing to this register has no effect

## 9 Analog-to-digital converter

An 8-input, 12-bit analog-to-digital converter (ADC) is integrated in the STMPE610. The ADC can be used as a generic analog-to-digital converter, or as a touchscreen controller capable of controlling a 4-wire resistive touchscreen.

**Table 13. ADC controller register summary table**

Address	Register name	Size	Description
0x20	ADC_CTRL1	8	ADC control
0x21	ADC_CTRL2	8	ADC control
0x22	ADCCapture	8	To initiate ADC data acquisition
0x30	ADC_DATA_CH0	8	ADC channel 0 (IN3/GPIO-3)
0x32	ADC_DATA_CH1	8	ADC channel 1 (IN2/GPIO-2)
0x38	ADC_DATA_CH4	8	ADC channel 4 (TSC)
0x3A	ADC_DATA_CH5	8	ADC channel 5 (TSC)
0x3C	ADC_DATA_CH6	8	ADC channel 6 (TSC)
0x3E	ADC_DATA_CH7	8	ADC channel 7 (TSC)



**ADC\_CTRL1****ADC control 1**

7	6	5	4	3	2	1	0
RESERVED	SAMPLE_TIME2	SAMPLE_TIME1	SAMPLE_TIME0	MOD_12B	RESERVED	REF_SEL	RESERVED

**Address:** 0x20**Type:** R/W**Reset:** 0x9C**Description:** ADC control register

[7] RESERVED

[6:4] SAMPLE\_TIME<sub>n</sub>: ADC conversion time in number of clock

000: 36

001: 44

010: 56

011: 64

100: 80

101: 96

110: 124

111: Not valid

[3] MOD\_12B: Selects 10 or 12-bit ADC operation

1: 12 bit ADC

0: 10 bit ADC

[2] RESERVED

[1] REF\_SEL: Selects between internal or external reference for the ADC

1: External reference

0: Internal reference

[0] RESERVED

**ADC\_CTRL2****ADC control 2**

7	6	5	4	3	2	1	0
RESERVED						ADC_FREQ_1	ADC_FREQ_0

**Address:** 0x21**Type:** R/W**Reset:** 0x01**Description:** ADC control.

[7] RESERVED

[6] RESERVED

[5] RESERVED

[4] RESERVED

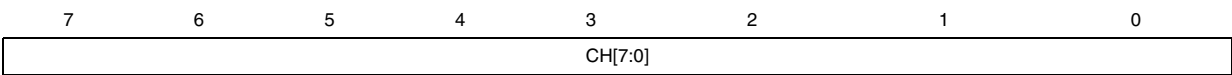
[3] RESERVED

[2] RESERVED

- [1:0] ADC\_FREQ: Selects the clock speed of ADC
  - 00: 1.625 MHz typ.
  - 01: 3.25 MHz typ.
  - 10: 6.5 MHz typ.
  - 11: 6.5 MHz typ.

ADC\_CAPT

ADC channel data capture



**Address:** 0x22

**Type:** R/W

**Reset:** 0xFF

**Description:** To initiate ADC data acquisition

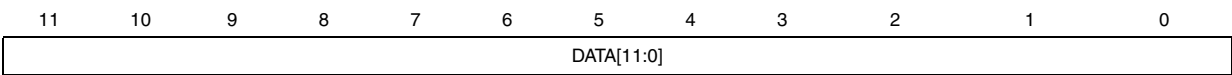
[7:0] CH[7:0]: ADC channel data capture

Write '1' to initiate data acquisition for the corresponding channel. Writing '0' has no effect.

Reads '1' if conversion is completed. Reads '0' if conversion is in progress.

ADC\_DATA\_CHn

ADC channel data registers



**Address:** Add address

**Type:** R/W

**Reset:** 0x0000

**Description:** ADC data register 0-7 (DATA\_CHn=0 -7)

[11:0] DATA[11:0]: ADC channel data

If TSC is enabled, CH3-0 is used for TSC and all readings to these channels give 0x0000

The ADC in STMPE610 operates on an internal RC clock with a typical frequency of 6.5 MHz. The total conversion time in ADC mode depends on the "SampleTime" setting, and the clock division field 'Freq'.

The following table shows the conversion time based on 6.5 MHz, 3.25 MHz and 1.625 MHz clock.

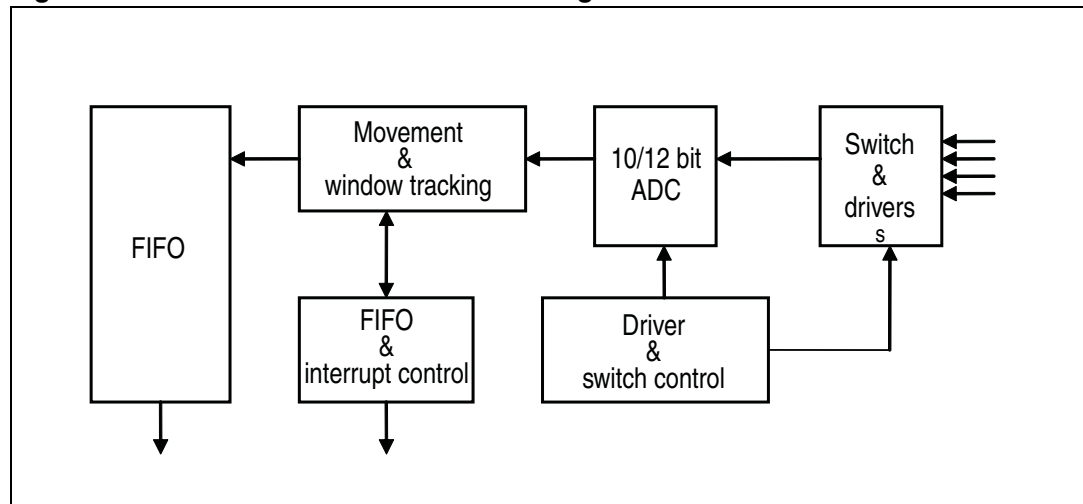
Table 14. ADC conversion time

Sample time setting	Conversion time in ADC clock	6.5 MHz (154 ns)	3.25 MHz (308 ns)	1.625 MHz (615 ns)
000	36	5.5 $\mu$ s (180 kHz)	11 $\mu$ s (90 kHz)	22 $\mu$ s (45 kHz)
001	44	6.8 $\mu$ s (147 kHz)	13.6 $\mu$ s (74 kHz)	27 $\mu$ s (36 kHz)
010	56	8.6 $\mu$ s (116 kHz)	17.2 $\mu$ s (58 kHz)	34.4 $\mu$ s (29 kHz)
011	64	9.9 $\mu$ s (101 kHz)	19.8 $\mu$ s (51 kHz)	39.6 $\mu$ s (25 kHz)
100	80	12.3 $\mu$ s (81.5 kHz)	24.6 $\mu$ s (41 kHz)	49.2 $\mu$ s (20 kHz)
101	96	14.8 $\mu$ s (67.6 kHz)	28.8 $\mu$ s (33 kHz)	59.2 $\mu$ s (17 kHz)
110	124	19.1 $\mu$ s (52.3 kHz)	38.2 $\mu$ s (26 kHz)	56.4 $\mu$ s (13 kHz)

## 10 Touchscreen controller

The STMPE610 is integrated with a hard-wired touchscreen controller for 4-wire resistive type touchscreen. The touchscreen controller is able to operate completely autonomously, and will interrupt the connected CPU only when a pre-defined event occurs.

**Figure 9. Touchscreen controller block diagram**



### 10.1 Driver and switch control unit

The driver and switch control unit allows coordination of the ADC and the MUX/switch. With the coordination of this unit, a stream of data is produced at a selected frequency.

The touchscreen drivers can be configured with 2 current ratings: 20 mA or 50 mA. In the case where multiple touch-down on the screen is causing a short, the current from the driver is limited to these values. Tolerance of these current setting is +/- 25%.

#### Movement tracking

The "Tracking Index" in the TSC\_CTRL register specifies a value, which determines the distance between the current touch position and the previous touch position. If the distance is shorter than the tracking index, it is discarded.

The tracking is calculated by summation of the horizontal and vertical movement. Movement is only reported if:

$$(\text{Current X} - \text{Previously Reported X}) + (\text{Current Y} - \text{Previously Reported Y}) > \text{Tracking Index}$$

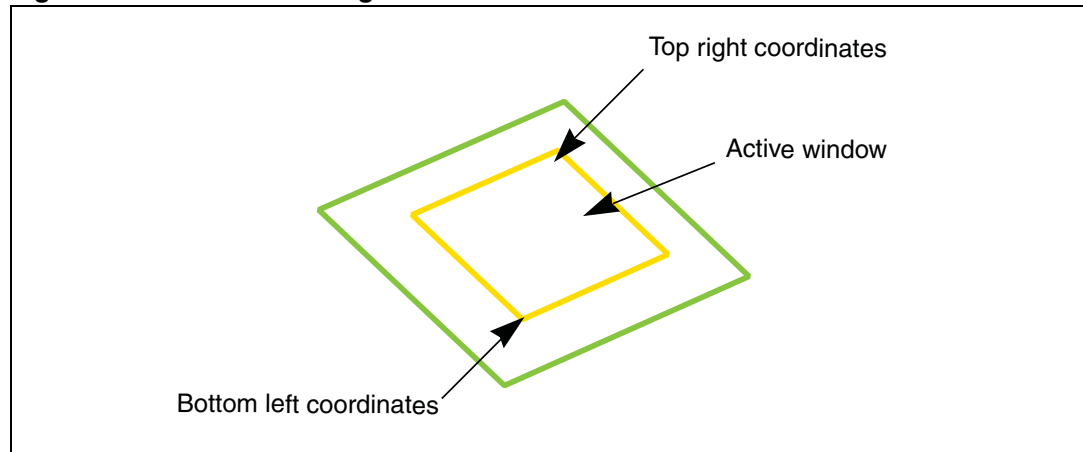
If pressure reporting is enabled (X/Y/Z), an increase in pressure will override the movement tracking and report the new data set, even if X/Y is within the previous tracking index. This is to ensure that a slow touch will not be discarded.

If pressure data is not used, select X/Y mode in touchscreen data acquisition. (Opmode field in TSCControl register).

### Window tracking

The -WDW\_X and WDW\_Y registers allow to pre-set a sub-window in the touchscreen such that any touch position that is outside the sub-window will be discarded.

**Figure 10. Window tracking**



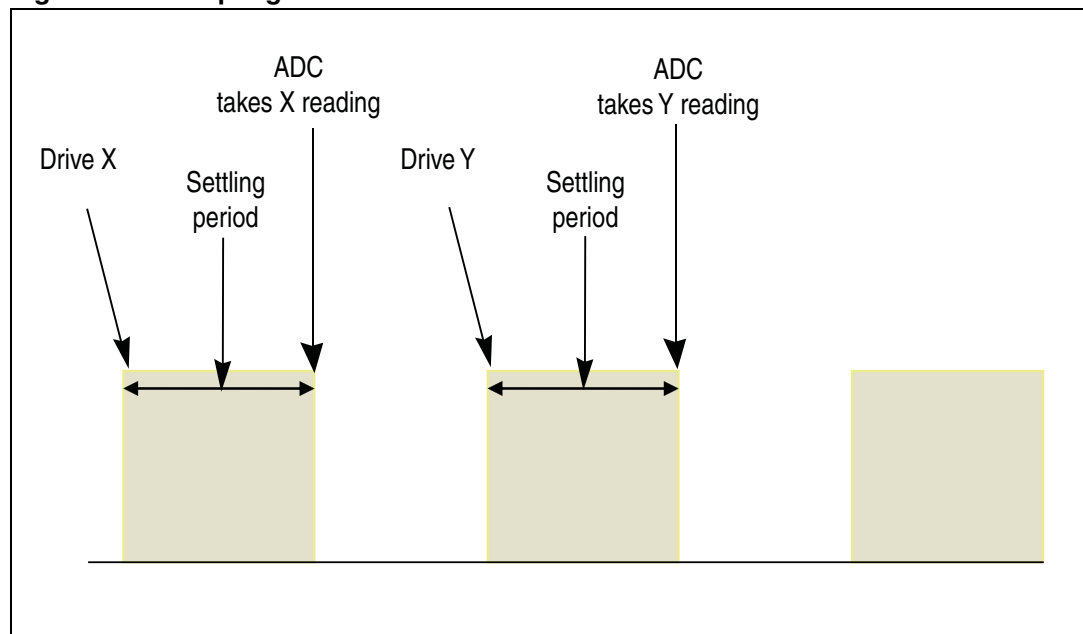
### FIFO

FIFO has a depth of 128 sectors. This is enough for 128 sets of touch data at maximum resolution (2 x 12 bits). FIFO can be programmed to generate an interrupt when it is filled to a pre-determined level.

### Sampling

The STMPE610 touchscreen controller has an internal 180 kHz, 12-bit ADC able to execute autonomous driving/sampling. Each "sample" consists of 4 ADC readings that provide the X and Y locations, as well as the touch pressure.

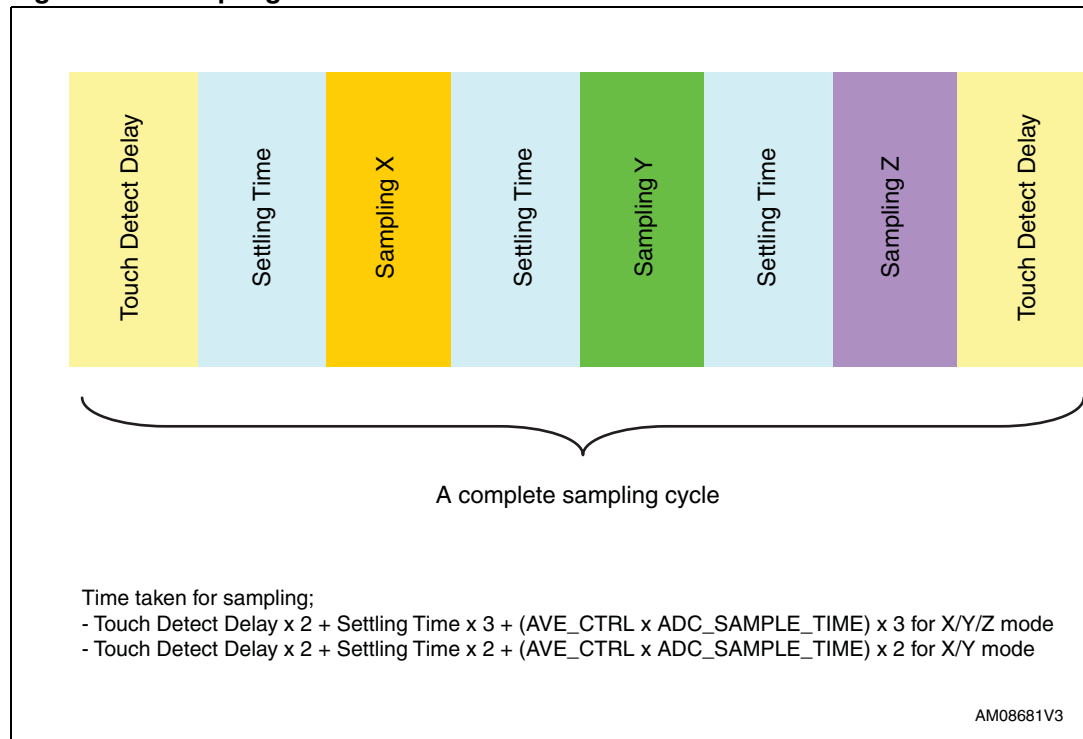
**Figure 11. Sampling**



## Sampling time calculation

The equation for a complete sampling cycle is described below.

**Figure 12. Sampling time calculation**



## Oversampling and averaging function

The STMPE610 touchscreen controller can be configured to oversample by 2/4/8 times and provide the averaged value as final output. This feature helps to reduce the effect of surrounding noise.

**Table 15. Touchscreen controller register summary table**

Address	Register name	Bit	Type	Function
0x40	TSC_CTRL	8	R/W	4-wire touchscreen controller setup
0x41	TSC_CFG	8	R/W	TSC configuration register
0x42	WDW_TR_X	16	R/W	Window setup for top right X
0x44	WDW_TR_Y	16	R/W	Window setup for top right Y
0x46	WDW_TR_X	16	R/W	Window setup for bottom left X
0x48	WDW_TR_Y	16	R/W	Window setup for bottom left Y
0x4A	FIFO_TH	8	R/W	FIFO level to generate interrupt
0x4B	FIFO_CTRL_STA	8	R/W	Current status of FIFO
0x4C	FIFO_SIZE	8	R	Current filled level of FIFO
0x4D	TSC_DATA_X	16	R	Data port for TSC data access
0x4F	TSC_DATA_Y	16	R	Data port for TSC data access

**Table 15. Touchscreen controller register summary table**

Address	Register name	Bit	Type	Function
0x51	TSC_DATA_Z	8	R	Data port for TSC data access
0x52	TSC_DATA_XYZ	32	R	Data port for TSC data access
0x56	TSC_FRACT_Z	8	R/W	TSC_FRACT_Z
0x57	TSC_DATA	8	R	TSC data access port
0x58	TSC_I_DRIVE	8	R/W	TSC_I_DRIVE
0x59	TSC_SHIELD	8	R/W	TSC_SHIELD

**TSC\_CTRL****Touchscreen controller control register**

7	6	5	4	3	2	1	0
TSC_STA	TRACK			OP_MOD			EN

**Address:** 0x40**Type:** R/W**Reset:** 0x90**Description:** 4-wire touchscreen controller (TSC) setup.

- [7] TSC\_STA: TSC status  
 Reads '1' when touch is detected  
 Reads '0' when touch is not detected  
 Writing to this register has no effect

- [6:4] TRACK: Tracking index  
 000: No window tracking  
 001: 4  
 010: 8  
 011: 16  
 100: 32  
 101: 64  
 110: 92  
 111: 127

- [3:1] OP\_MOD: TSC operating mode  
 000: X, Y, Z acquisition  
 001: X, Y only  
 010: X only  
 011: Y only  
 100: Z only  
 This field cannot be written on, when EN = 1

- [0] EN: Enable TSC

**TSC\_CFG****Touchscreen controller configuration register**

7	6	5	4	3	2	1	0
AVE_CTRL_1	AVE_CTRL_0	TOUCH_DET_DELAY_2	TOUCH_DET_DELAY_1	TOUCH_DET_DELAY_0	SETTLING_2	SETTLING_1	SETTLING_0

**Address:** 0x41**Type:** R/W**Buffer:****Reset:****Description:** Touchscreen controller configuration register.

[7:6] AVE\_CTRL\_1/0: Average control

00=1 sample

01=2 samples

10=4 samples

11=8 samples

[5:3] TOUCH\_DET\_DELAY\_2/1/0: Touch detect delay

000 - 10  $\mu$ s001 - 50  $\mu$ s010 = 100  $\mu$ s011 = 500  $\mu$ s

100 = 1 ms

101 = 5 ms

110 = 10 ms

111 = 50 ms

[2:0] SETTLING: Panel driver settling time<sup>(1)</sup>000 = 10  $\mu$ s001 = 100  $\mu$ s010 = 500  $\mu$ s

011 =1 ms

100 = 5 ms

101 = 10 ms

110 = 50 ms

111 =100 ms

1. For large panels (> 6"), a capacitor of 10 nF is recommended at the touchscreen terminals for noise filtering. In this case, settling time of 1 ms or more is recommended.



## 10.2 Touch detect delay

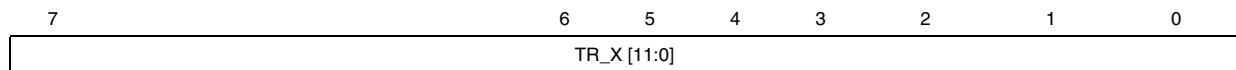
Touch Detect Delay is an additional method used to compensate for the time it takes for the panel voltage to be pulled high during a non-touch condition.

For example, the way it works to detect a touch:

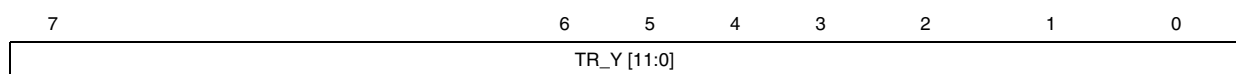
X+ is pulled high and Y+ is driven low. After Touch Detect Delay is expired the level of X+ is read. If no touch, X+ is high. If there is a touch, X+ is low.

If the initial voltage of X+ is low before being pulled high by the internal resistor, especially if a filtering capacitor is connected, this time needs to be compensated. The Touch Delay setting provides time for the voltage to be pulled high in a non-touch condition and avoids a false report of a touch condition.

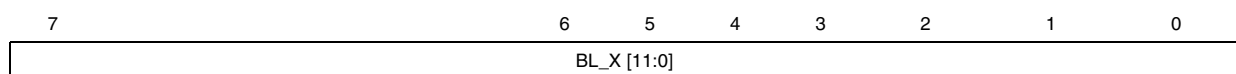
Normally the Touch Detect Delay needs to be long enough to allow the voltage to rise to V+ in a non-touch condition and this will depend on the presence of external filtering capacitors. For more details on recommendation of Touch Detect delay register setting, refer to STMPE811 Application Note (AN2825 ST document).

**WDW\_TR\_X****Window setup for top right X****Address:** 0x42**Type:** R/W**Reset:** 0x0FFF

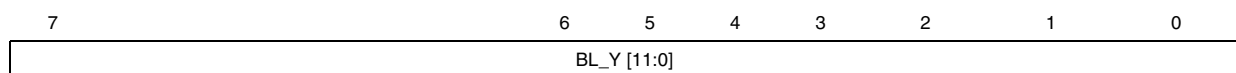
**Description:** Window setup for top right X coordinates  
 [11:0] TR\_X: bit 11:0 of top right X coordinates

**WDW\_TR\_Y****Window setup for top right Y****Address:** 0x44**Type:** R/W**Reset:** 0x0FFF

**Description:** Window setup for top right Y coordinates  
 [11:0] TR\_Y: bit 11:0 of top right Y coordinates

**WDW\_BL\_X****Window setup for bottom left X****Address:** 0x46**Type:** R/W**Reset:** 0x0000

**Description:** Window setup for bottom left X coordinates  
 [11:0] BL\_X: bit 11:0 of bottom left X coordinates

**WDW\_BL\_Y****Window setup for bottom left Y****Address:** 0x48**Type:** R/W**Reset:** 0x0000

**Description:** Window setup for bottom left Y coordinates  
 [11:0] **BL\_X**: bit 11:0 of bottom left Y coordinates

## FIFO\_TH FIFO threshold

7	6	5	4	3	2	1	0
FIFO_TH							

**Address:** 0x4A

**Type:** R/W

**Reset:** 0x00

**Description:** Triggers an interrupt upon reaching or exceeding the threshold value. This field must not be set as zero.

[7:0] FIFO\_TH: Touchscreen controller FIFO threshold

## FIFO\_CTRL\_STA FIFO threshold

7	6	5	4	3	2	1	0
FIFO_OFLOW	FIFO_FULL	FIFO_EMPTY	FIFO_TH_TRIG	RESERVED		FIFO_RESET	

**Address:** 0x4B

**Type:** R/W

**Reset:** 0x20

**Description:** Current status of FIFO..

[7] FIFO\_OFLOW:  
Reads 1 if FIFO is overflow

[6] FIFO\_FULL:  
Reads 1 if FIFO is full

[5] FIFO\_EMPTY:  
Reads 1 if FIFO is empty

[4] FIFO\_TH\_TRIG:  
0 = Current FIFO size is still below the threshold value  
1 = Current FIFO size is at or beyond the threshold value

[3:1] RESERVED

[0] FIFO\_RESET:  
Write '0' : FIFO put out of reset mode  
Write '1' : Resets FIFO. All data in FIFO will be cleared.  
When TSC is enabled, FIFO resets automatically.

**FIFO\_SIZE****FIFO size**

7	6	5	4	3	2	1	0
RESERVED		FIFO_SIZE					

**Address:** 0x4C**Type:** R**Reset:** 0x00

**Description:** Current number of samples available  
 [7:0] FIFO\_SIZE: Number of samples available

**TSC\_DATA\_X****TSC\_DATA\_X**

11	10	9	8	7	6	5	4	3	2	1	0
DATAY[11:0]											

**Address:** 0x4D**Type:** R**Reset:** 0x0000

**Description:** Bit 11:0 of Y dataTSC\_DATA\_Y  
 [11:0] DATAY[11:0]: Bit 11:0 of Y data

**TSC\_DATA\_Y**

11	10	9	8	7	6	5	4	3	2	1	0
DATAY[11:0]											

**Address:** 0x4F**Type:** R**Reset:** 0x0000

**Description:** Bit 11:0 of Y data  
 [11:0] DATAY[11:0]: bit 11:0 of Y data

**TSC\_DATA\_Z****TSC\_DATA\_Z**

7	6	5	4	3	2	1	0
DATAZ[7:0]							

**Address:** 0x51**Type:** R**Reset:** 0x0000

**Description:** Bit 7:0 of Z data  
 [7:0] DATAZ[7:0]: bit 7:0 of Z data

**TSC\_DATA****Touchscreen controller DATA**

7	6	5	4	3	2	1	0
DATA							

**Address:** 0x57 (auto-increment), 0xD7 (non-auto-increment)

**Type:** R

**Reset:** 0x00

**Description:** Data port for TSC data access

[11:0] DATA: data bytes from TSC FIFO

The data format from the TSC\_DATA register depends on the setting of "OpMode" field in TSC\_CTRL register. The samples acquired are accessed in "packed samples". The size of each "packed sample" depends on which mode the touchscreen controller is operating in.

The TSC\_DATA register can be accessed in 2 modes:

- Autoincrement
- Non autoincrement

To access the 128-sets buffer, the non autoincrement mode should be used.

**Table 16. Touchscreen controller DATA register**

TSC_CTRL in operation mode	Number of bytes to read from TSC_DATA	Byte0	Byte1	Byte2	Byte3
000	4	[11:4] of X	[3:0] of X [11:8] of Y	[7:0] of Y	[7:0] of Z
001	3	[11:4] of X	[3:0] of X [11:8] of Y	[7:0] of Y	-
010	2	[11:4] of X	[3:0] of X	-	-
011	2	[11:4] of Y	[3:0] of Y	-	-
100	1	[7:0] of Z	-	-	-

**TSC\_FRACTION\_Z****Touchscreen controller FRACTION\_Z**

7	6	5	4	3	2	1	0
RESERVED					FRACTION_Z		

**Address:** 0x56**Type:** R**Reset:** 0x00**Description:** This register allows to select the range and accuracy of the pressure measurement

[7:3] RESERVED

[2:0] FRACTION\_Z:

000: Fractional part is 0, whole part is 8

001: Fractional part is 1, whole part is 7

010: Fractional part is 2, whole part is 6

011: Fractional part is 3, whole part is 5

100: Fractional part is 4, whole part is 4

101: Fractional part is 5, whole part is 3

110: Fractional part is 6, whole part is 2

111: Fractional part is 7, whole part is 1

**TSC\_I\_DRIVE****Touchscreen controller drive I**

7	6	5	4	3	2	1	0
RESERVED						DRIVE	

**Address:** 0x58**Type:** R/W**Reset:** 0x00**Description:** This register sets the current limit value of the touchscreen drivers

[7:1] RESERVED

[0] DRIVE: maximum current on the touchscreen controller (TSC) driving channel

0: 20 mA typical, 35 mA max

1: 50 mA typical, 80 mA max

TSC\_SHIELD

Touchscreen controller shield

7	6	5	4	3	2	1	0
RESERVED				X+	X-	Y+	Y-

**Address:** 0x59

**Type:** R

**Reset:** 0x00

**Description:** Writing each bit would ground the corresponding touchscreen wire

[7:4] RESERVED

[3:0] SHIELD[3:0]:  
Write 1 to GND X+, X-, Y+, Y- lines

## 11 Touchscreen controller programming sequence

The following are the steps to configure the touchscreen controller (TSC):

- a) Disable the clock gating for the touchscreen controller and ADC in the SYS\_CFG2 register.
- b) Configure the touchscreen operating mode and the window tracking index.
- c) A touch detection status may also be enabled through enabling the corresponding interrupt flag. With this interrupt, the user is informed through an interrupt when the touch is detected as well as lifted.
- d) Configure the TSC\_CFG register to specify the “panel voltage settling time”, touch detection delays and the averaging method used.
- e) A windowing feature may also be enabled through TSCWdwTRX, TSCWdwTRY, TSCWdwBLX and TSCWdwBLY registers. By default, the windowing covers the entire touch panel.
- f) Configure the TSC\_FIFO\_TH register to specify the threshold value to cause an interrupt. The corresponding interrupt bit in the interrupt module must also be enabled. This interrupt bit should be masked off during data fetching from the FIFO in order to prevent an unnecessary trigger of this interrupt. Upon completion of the data fetching, this bit can be re-enabled.
- g) By default, the FIFO\_RESET bit in the TSC\_FIFO\_CTRL\_STA register holds the FIFO in Reset mode. Upon enabling the touchscreen controller (through the EN bit in TSC\_CTRL), this FIFO reset is automatically deasserted. The FIFO status may be observed from the TSC\_FIFO\_CTRL\_STA register or alternatively through the interrupt.
- h) Once the data is filled beyond the FIFO threshold value, an interrupt is triggered (assuming the corresponding interrupt is being enabled). The user is required to continuously read out the data set until the current FIFO size is below the threshold, then, the user may clear the interrupt flag. As long as the current FIFO size exceeds the threshold value, an interrupt from the touchscreen controller is sent to the interrupt module. Therefore, even if the interrupt flag is cleared, the interrupt flag will automatically be asserted, as long as the FIFO size exceeds the threshold value.
- i) The current FIFO size can be obtained from the TSC\_FIFO\_Sz register. This information may assist the user in how many data sets are to be read out from the FIFO, if the user intends to read all in one shot. The user may also read a data set by a data set.
- j) The TSC\_DATA\_X register holds the X-coordinates. This register can be used in all touchscreen operating modes.
- k) The TSC\_DATA\_Y register holds the Y-coordinates. TSC\_DATA\_Y register holds the Y-coordinates.
- l) The TSC\_DATA\_Z register holds the Z value. TSC\_DATA\_Z register holds the Z-coordinates.
- m) The TSCDATA\_XYZ register holds the X, Y and Z values. These values are packed into 4 bytes. This register can only be used when the touchscreen operating mode is 000 and 001. This register is to facilitate less byte read.
- n) For the TSC\_FRACT\_Z register, the user may configure it based on the touchscreen panel resistance. This allows the user to specify the resolution of the



Z value. With the Z value obtained from the register, the user simply needs to multiply the Z value with the touchscreen panel resistance to obtain the touch resistance.

- o) The TSC\_DATA register allows facilitation of another reading format with minimum I<sup>2</sup>C transaction overhead by using the non autoincrement mode (or equivalent mode in SPI). The data format is the same as TSC\_DATA\_XYZ, with the exception that all the data fetched are from the same address.
- p) Enable the EN bit of the TSC\_CTRL register to start the touch detection and data acquisition.
- q) During the auto-hibernate mode, a touch detection can cause a wake-up to the device only when the TSC is enabled and the touch detect status interrupt mask is enabled.
- r) In order to prevent confusion, it is recommended that the user not mix the data fetching format (TSC\_DATA\_X, TSC\_DATA\_Y, TSC\_DATA\_Z, TSC\_DATA\_XYZ and TSC\_DATA) between one reading and the next.
- s) It is also recommended that the user should perform a FIFO reset and TSC disabling when the ADC or TSC setting are reconfigured.

## 12 GPIO controller

A total of 6 GPIOs are available in the STMPE610 port expander device. Most of the GPIOs share physical pins with some alternate functions. The GPIO controller contains the registers that allow the host system to configure each of the pins into either a GPIO, or one of the alternate functions. Unused GPIOs should be configured as outputs to minimize power consumption.

A group of registers are used to control the exact function of each of the 6 GPIOs. The registers and their respective addresses are listed in the following table.

**Table 17. GPIO control registers**

Address	Register name	Size (bit)	Function
0x10	GPIO_SET_PIN	8	Set pin register
0x11	GPIO_CLR_PIN	8	Clear pin state
0x12	GPIO_MP_STA	8	Monitor pin state
0x13	GPIO_DIR	8	Set pin direction
0x14	GPIO_ED	8	Edge detect status
0x15	GPIO_RE	8	Rising edge detection enable
0x16	GPIO_FE	8	Falling edge detection enable
0x17	GPIO_ALT_FUNCT	8	Alternate function register

All GPIO registers are named as GPIO-x, where x represents the functional group.

7	6	5	4	3	2	1	0
GPIO-7	GPIO-6	GPIO-5	GPIO-4	GPIO-3	GPIO-2	RESERVED	RESERVED

### GPIO\_SET\_PIN

### GPIO set pin register

**Address:** 0x10

**Type:** R/W

**Reset:** 0x00

**Description:** GPIO set pin register.

Writing 1 to this bit causes the corresponding GPIO to go to 1 state.

Writing 0 has no effect.

**GPIO\_CLR\_PIN****Clear pin state register****Address:** 0x11**Type:** R/W**Reset:** 0x00**Description:** GPIO clear pin state register.

Writing '1' to this bit causes the corresponding GPIO to go to 0 state.

Writing '0' has no effect.

**GPIO\_MP\_STA****GPIO monitor pin state register****Address:** 0x12**Type:** R/W**Reset:** 0x00**Description:** GPIO monitor pin state.

Reading this bit yields the current state of the bit. Writing has no effect.

**GPIO\_DIR****GPIO set pin direction****Address:** 0x13**Type:** R/W**Reset:** 0x00**Description:** GPIO set pin direction register.Writing '0' sets the corresponding GPIO to input state, and '1' sets it to output state.  
All bits are '0' on reset.**GPIO\_ED\_STA****GPIO edge detect status****Address:** 0x14**Type:** R/W**Reset:** 0x00**Description:** GPIO edge detect status register. An edge transition has been detected.

**GPIO\_RE****Rising edge register****Address:** 0x15**Type:** R/W**Reset:** 0x00

**Description:** GPIO rising edge detection enable register.  
 Setting this bit to '1' would enable the detection of the rising edge transition.  
 The detection would be reflected in the GPIO edge detect status register.

**GPIO\_FE****Falling edge detection enable register****Address:** 0x16**Type:** R/W**Reset:** 0x00

**Description:** Setting this bit to '1' would enable the detection of the falling edge transition.  
 The detection would be reflected in the GPIO edge detect status register.

**GPIO\_ALT\_FUNC****Alternate function register****Address:** 0x17**Type:** R/W**Reset:** 0x0F

**Description:** Alternate function register. "0" sets the corresponding pin to function as touchscreen/ADC, and '1' sets it into GPIO mode.

On power-up reset, all GPIOs are set as input.

**Power supply**

The STMPE610 GPIO operates from a separate supply pin ( $V_{IO}$ ). This dedicated supply pin provides a level-shifting feature to the STMPE610. The GPIO remains valid until  $V_{IO}$  is removed.

The host system may choose to turn off  $V_{CC}$  supply while keeping  $V_{IO}$  supplied. However it is not allowed to turn off supply to  $V_{IO}$ , while keeping the  $V_{CC}$  supplied.

The touchscreen is always powered by  $V_{IO}$ . For better resolution and noise immunity,  $V_{IO}$  above 2.8 V is advised.

**12.0.1 Power-up reset (POR)**

The STMPE610 is equipped with an internal POR circuit that holds the device in reset state, until the  $V_{IO}$  supply input is valid. The internal POR is tied to the  $V_{IO}$  supply pin.

## 13 Maximum rating

Stressing the device above the ratings listed in the “Absolute maximum ratings” table may cause permanent damage to the device. These are stress ratings only, and operation of the device at these or any other conditions above those indicated in the operating sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

**Table 18. Absolute maximum ratings**

Symbol	Parameter	Value	Unit
$V_{CC}$	Supply voltage	4.5	V
$V_{IO}$	GPIO supply voltage	4.5	V
ESD	ESD protection on each GPIO pin (air discharge)	4	kV
T	Operating temperature	-40 - 85	°C/W
$T_{STG}$	Storage temperature	-65 - 155	°C/W
$T_J$	Thermal resistance junction-ambient	96	°C/W

### 13.1 Recommended operating conditions

**Table 19. Power consumption**

Symbol	Parameter	Test condition	Value			Unit
			Min	Typ	Max	
$V_{CC}$	I/O supply voltage	$V_{IO} \geq V_{CC}$	1.65	—	3.6	V
$V_{IO}$	Core supply voltage		1.65	—	3.6	V
$I_{CC-active}$	Core supply current	Touchscreen controller at 100 Hz sampling $V_{CC} = 1.8 - 3.3$ V	—	0.5	1.0	uA
$I_{IO-active}$	I/O supply current	Touchscreen controller at 100 Hz sampling $V_{IO} = 1.8$ V	—	0.8	1.2	mA
$I_{IO-active}$	I/O supply current	Touchscreen controller at 100 Hz sampling $V_{IO} = 3.3$ V	—	2.0	2.8	mA
$I_{CC-hibernate}$	Core supply current	Hibernate state, no I2C/SPI activity $V_{CC} = 1.8$ V	—	0.5	1	uA

Table 19. Power consumption (continued)

Symbol	Parameter	Test condition	Value			Unit
			Min	Typ	Max	
$I_{IO-}$ hibernate	I/O supply current	Hibernate state, no I2C/SPI activity $V_{IO} = 1.8 - 3.3 \text{ V}$	—	0.5	1	$\mu\text{A}$
		Hibernate state, no I2C/SPI activity $V_{IO} = 3.3 \text{ V}$	—	1.0	3.0	$\mu\text{A}$

## 14 Electrical specifications

**Table 20. DC electrical characteristics (-40 °C to 85 °C, all GPIOs comply to JEDEC standard JESD-8-7)**

Symbol	Parameter	Test condition	Value			Unit
			Min	Typ	Max	
$V_{IL}$	Input voltage low state	$V_{IO} = 1.8 - 3.3 \text{ V}$	-0.3 V	—	$0.20 V_{IO}$	V
$V_{IH}$	Input voltage high state	$V_{IO} = 1.8 - 3.3 \text{ V}$	$0.80 V_{IO}$	—	$V_{IO} + 0.3 \text{ V}$	V
$V_{OL}$	Output voltage low state	$V_{IO} = 1.8 \text{ V}$ , $I_{OL} = 4 \text{ mA}$	-0.3 V	—	$0.15 V_{IO}$	V
$V_{OH}$	Output voltage high state	$V_{IO} = 3.3 \text{ V}$ , $I_{OL} = 8 \text{ mA}$	$0.85 V_{IO}$	—	—	V
$V_{OL}$ (I <sup>2</sup> C/SPI)	Output voltage low state	$V_{CC} = 1.8 \text{ V}$ , $I_{OL} = 4 \text{ mA}$	-0.3 V	—	$0.15 V_{CC}$	V
$V_{OH}$ (I <sup>2</sup> C/SPI)	Output voltage high state	$V_{CC} = 3.3 \text{ V}$ , $I_{OL} = 8 \text{ mA}$	$0.85 V_{CC}$	—	$V_{CC} + 0.3 \text{ V}$	V

**Table 21. AC electrical characteristics (-40 °C to 85 °C)**

Symbol	Parameter	Test condition	Value			Unit
			Min	Typ	Max	
$CLKI2C_{max}$	I <sup>2</sup> C maximum SCLK	$V_{CC} = 1.8 - 3.3 \text{ V}$	400	—	—	kHz
$CLKSPI_{max}$	SPI maximum clock	$V_{CC} = 1.8 \text{ V}$	800	—	—	kHz
		$V_{CC} = 3.3 \text{ V}$	1000	—	—	kHz

**Table 22. ADC specification (-40 °C to 85 °C)**

Parameter	Test condition	Value			Unit
		Min	Typ	Max	
Full-scale input span		0	—	$V_{ref}$	V
Absolute input range		—	—	$V_{CC} + 0.2$	V
Input capacitance		—	25	—	pF
Leakage current		—	0.1	—	μA
Resolution		—	12	—	bits
No missing codes		11		—	bits
Integral linearity error		—	±4	±6	bits
Offset error		—	±5	±7	LSB
Gain error		—	±14	±18	LSB
Noise	Including internal $V_{ref}$	—	70	—	μVrms
Power supply rejection ratio		—	50	—	dB
Throughput rate		—	180	—	ksps

**Table 23. Switch drivers specification**

Parameter	Test condition	Value			Unit
		Min	Typ	Max	
ON resistance X+, Y+		—	5.5	—	Ω
ON resistance X-, Y-		—	7.3	—	Ω
Drive current	Duration 100 ms	—	—	50	mA

**Table 24. Voltage reference specification**

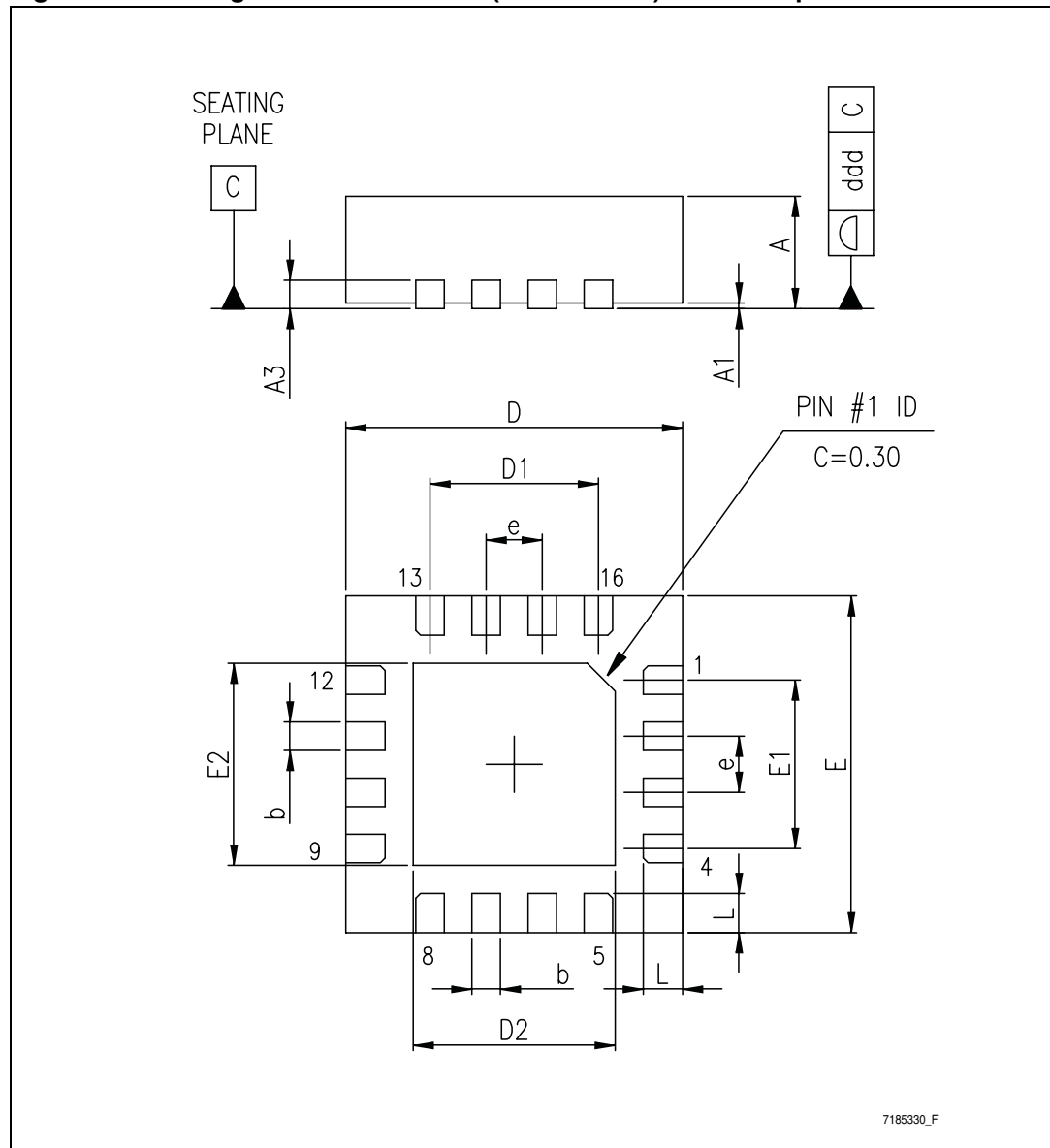
Parameter	Test condition	Value			Unit
		Min	Typ	Max	
Internal reference voltage		2.45	2.50	2.55	V
Internal reference drift		—	25	—	Ppm/C
Output impedance	Internal reference ON	—	300	—	Ω
	Internal reference OFF	—	1	—	GΩ



## 15 Package mechanical data

In order to meet environmental requirements, ST offers these devices in different grades of ECOPACK® packages, depending on their level of environmental compliance. ECOPACK® specifications, grade definitions and product status are available at: [www.st.com](http://www.st.com). ECOPACK® is an ST trademark.

**Figure 13. Package outline for QFN16 (3 x 3 x 1 mm) - 0.50 mm pitch**



1. Drawing not to scale.

Table 25. Package mechanical data for QFN16 (3 x 3 x 1 mm) - 0.50 mm pitch

Symbol	Millimeters		
	Min	Typ	Max
A	0.80	0.90	1.00
A1	—	0.02	0.05
A3	—	0.20	—
b	0.18	0.25	0.30
D	—	3.00	—
D2	1.55	1.70	1.80
E	—	3.00	—
E2	1.55	1.70	1.80
e	—	0.50	—
K	—	0.20	—
L	0.30	0.40	0.50
r	0.09	—	—

Figure 14. Recommended footprint for QFN16 (3 x 3 x 1 mm) - 0.50 mm pitch

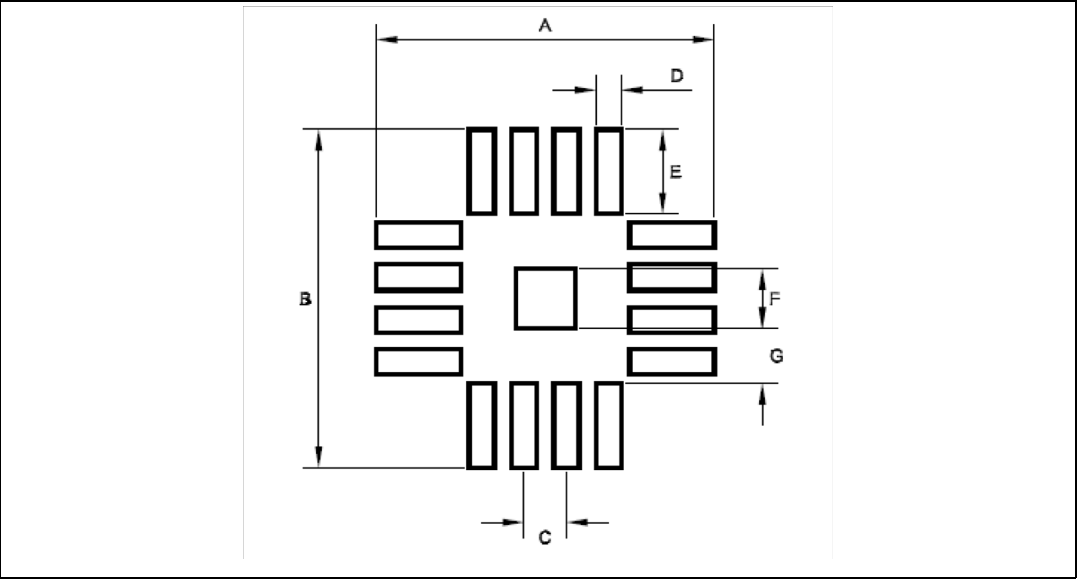


Table 26. Footprint dimensions

Symbol	Millimeters		
	Min	Typ	Max
A	—	3.8	—
B	—	3.8	—
C	—	0.5	—
D	—	0.3	—
E	—	0.8	—
F	—	1.5	—
G	—	0.35	—

Figure 15. Carrier tape for QFN16 (3 x 3 x 1 mm) - 0.50 mm pitch

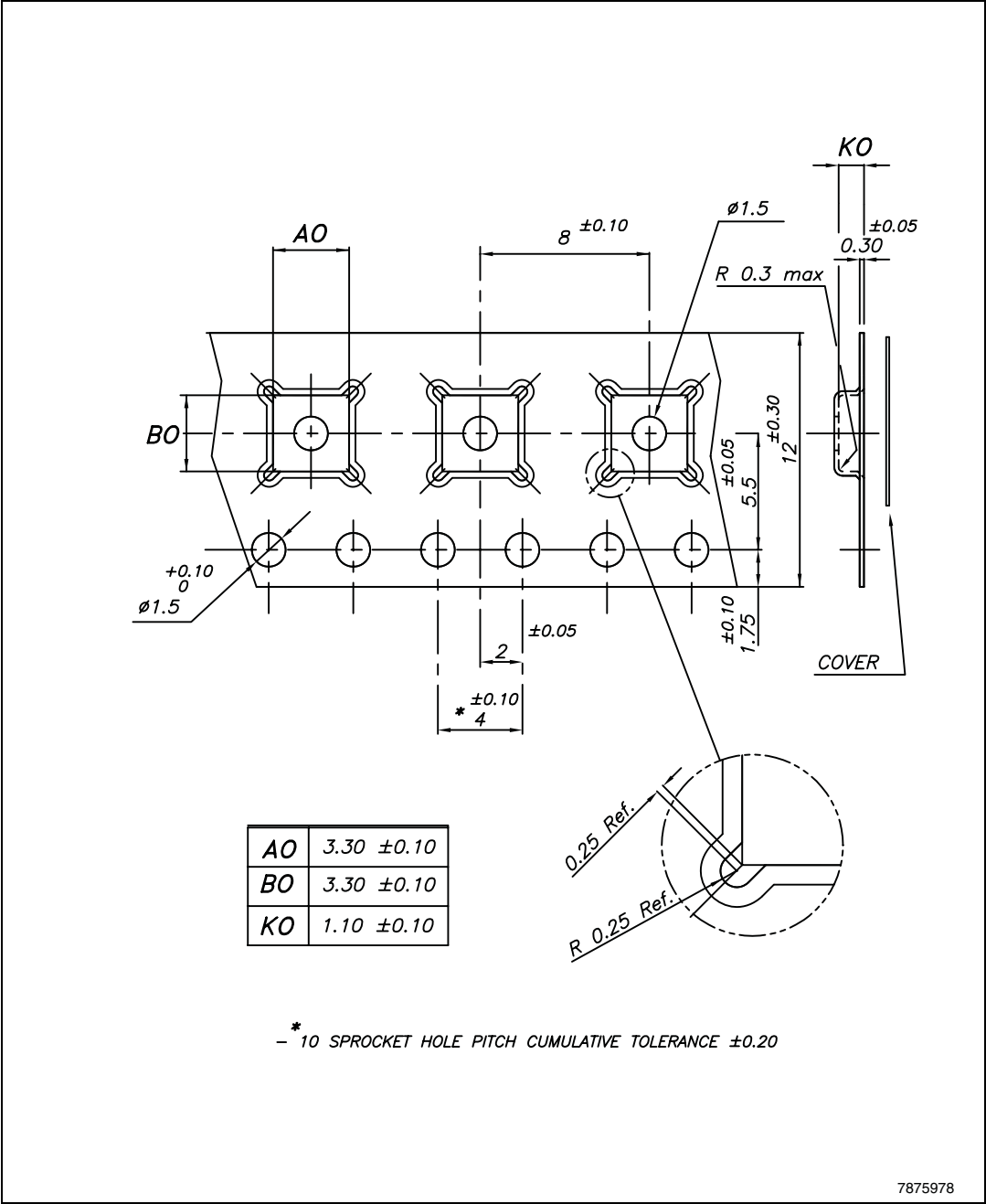


Figure 16. Reel information for QFN16 (3 x 3 x 1 mm) - 0.50 mm pitch

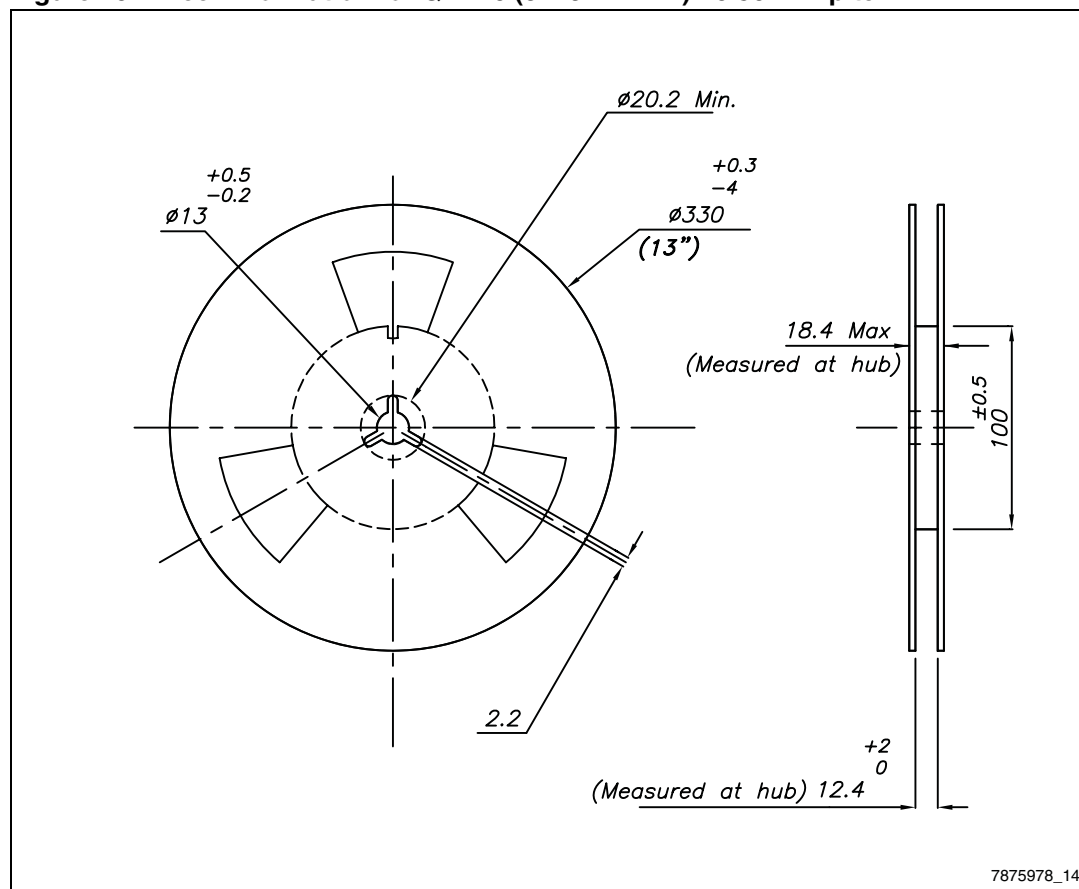
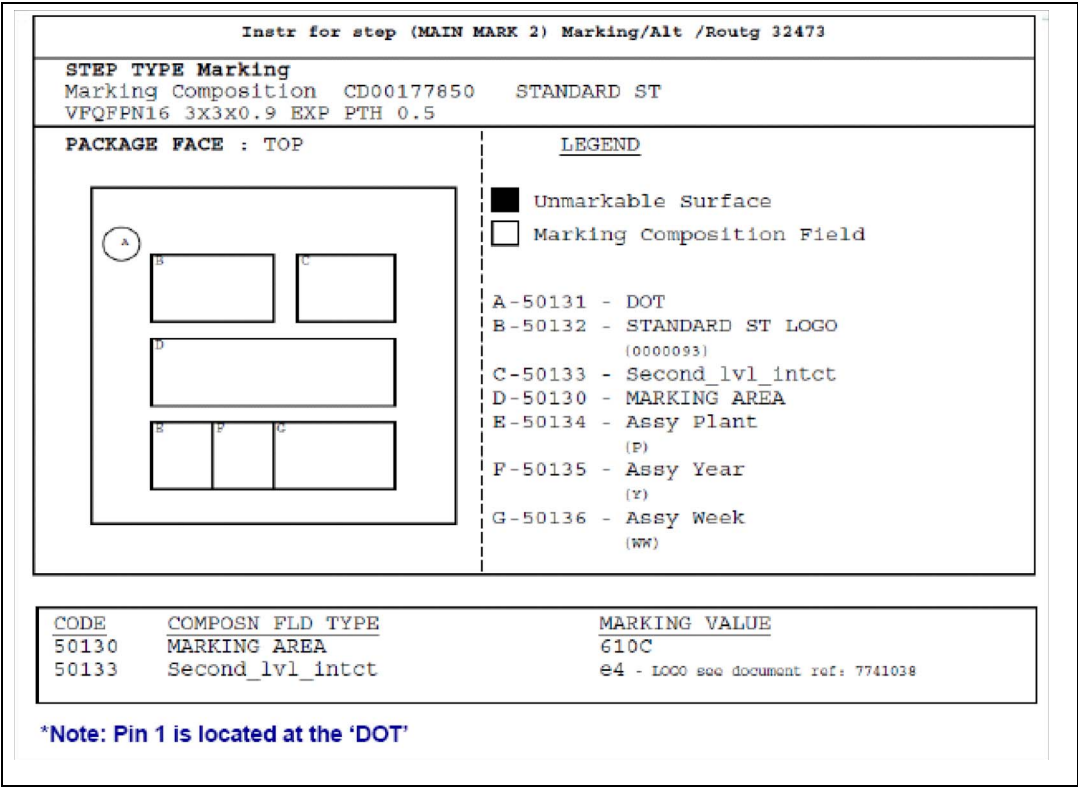


Figure 17. Marking specifications



## 16 Revision history

**Table 27. Document revision history**

Date	Revision	Changes
07-Apr-2009	1	Initial release.
23-Sep-2009	2	Removed “Temperature sensor” from <a href="#">Section 1</a> , <a href="#">Figure 1</a> and <a href="#">Figure 8</a> . Updated: In the SYS_CTRL2 register, the 3rd bit is reserved.
12-Mar-2010	3	Updated: Title of the document and ESD value in <a href="#">Table 18</a> .
09-Sep-2011	4	Added new section: <a href="#">Section 10.2: Touch detect delay</a> Updated V <sub>CC</sub> parameter description: <a href="#">Table 19</a>

**Please Read Carefully:**

Information in this document is provided solely in connection with ST products. STMicroelectronics NV and its subsidiaries ("ST") reserve the right to make changes, corrections, modifications or improvements, to this document, and the products and services described herein at any time, without notice.

All ST products are sold pursuant to ST's terms and conditions of sale.

Purchasers are solely responsible for the choice, selection and use of the ST products and services described herein, and ST assumes no liability whatsoever relating to the choice, selection or use of the ST products and services described herein.

No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted under this document. If any part of this document refers to any third party products or services it shall not be deemed a license grant by ST for the use of such third party products or services, or any intellectual property contained therein or considered as a warranty covering the use in any manner whatsoever of such third party products or services or any intellectual property contained therein.

**UNLESS OTHERWISE SET FORTH IN ST'S TERMS AND CONDITIONS OF SALE ST DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY WITH RESPECT TO THE USE AND/OR SALE OF ST PRODUCTS INCLUDING WITHOUT LIMITATION IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE (AND THEIR EQUIVALENTS UNDER THE LAWS OF ANY JURISDICTION), OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.**

**UNLESS EXPRESSLY APPROVED IN WRITING BY TWO AUTHORIZED ST REPRESENTATIVES, ST PRODUCTS ARE NOT RECOMMENDED, AUTHORIZED OR WARRANTED FOR USE IN MILITARY, AIR CRAFT, SPACE, LIFE SAVING, OR LIFE SUSTAINING APPLICATIONS, NOR IN PRODUCTS OR SYSTEMS WHERE FAILURE OR MALFUNCTION MAY RESULT IN PERSONAL INJURY, DEATH, OR SEVERE PROPERTY OR ENVIRONMENTAL DAMAGE. ST PRODUCTS WHICH ARE NOT SPECIFIED AS "AUTOMOTIVE GRADE" MAY ONLY BE USED IN AUTOMOTIVE APPLICATIONS AT USER'S OWN RISK.**

Resale of ST products with provisions different from the statements and/or technical features set forth in this document shall immediately void any warranty granted by ST for the ST product or service described herein and shall not create or extend in any manner whatsoever, any liability of ST.

ST and the ST logo are trademarks or registered trademarks of ST in various countries.

Information in this document supersedes and replaces all information previously supplied.

The ST logo is a registered trademark of STMicroelectronics. All other names are the property of their respective owners.

© 2011 STMicroelectronics - All rights reserved

STMicroelectronics group of companies

Australia - Belgium - Brazil - Canada - China - Czech Republic - Finland - France - Germany - Hong Kong - India - Israel - Italy - Japan - Malaysia - Malta - Morocco - Philippines - Singapore - Spain - Sweden - Switzerland - United Kingdom - United States of America

[www.st.com](http://www.st.com)