Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!

Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832
Email & Skype: info@chipsmall.com Web: www.chipsmall.com
Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China

Atmel QTouch Library

User Guide

Supports QTouch® and QMatrix® acquisition for Keys, Sliders and Rotors

Rev. 8207L-AT42-05/12

Table of Contents

1 Preface

This manual contains information that enables customers to implement capacitive touch solutions on ATMEL AVR® microcontrollers and ARM®-based AT91SAM microcontrollers using ATMEL QTouch libraries. This guide is a functional description of the library software, its programming interface and it also describes its use on the supported reference systems.

Use of this software is bound by the Software License Agreement included with the Library. This user guide is applicable for Atmel QTouch® Library 5.0 .

Related documents from ATMEL
Documents related to QTouch capacitive sensing solutions from ATMEL are

- TS2080A/B data sheet.
- QT600 users guide
- Release Notes for ATMEL QTouch libraries.
- A library selection excel workbook that is used for the selection of the appropriate library variant from the package available under in the install directory. The default location is C:\Program Files\Atmel\Atmel_QTouch_Libraries_5.x\
- Capacitive touch sensor design guide http://www.atmel.com/dyn/resources/prod_documents/doc10620.pdf .

If you need Assistance
For assistance with QTouch capacitive sensing software libraries and related issues, contact your local ATMEL sales representative or send an email to *touch@atmel.com for AVR libraries and at91support@atmel.com for SAM libraries.*

2 Introduction

ATMEL QTouch Library is a royalty free software library (available for GCC and IAR compiler tool chains) for developing touch applications on standard AVR and SAM microcontrollers. Customers can link the library into their applications in order to provide touch sensing capability in their projects. The Library can be used to develop single chip solutions for control applications which have touch sensing capabilities, or to develop standalone touch sensing solutions which interface with other host or control devices.

Features of ATMEL QTouch Library include

- Capacitive touch sensing using patented charge-transfer signal acquisition for robust sensing.
- Support for a wide range of 8- and 32-bit AVRs.
- Support for 32-bit ARM microcontrollers.
- Support for 8-bit tiny AVRs having flash of 2K bytes.
- Support both QTouch and QMatrix acquisition methods and autonomous touch for UC3L.
- Support up to 64 touch sense channels for generic libraries and up to 136 channels for UC3L libraries.
- Flexible choice of touch sensing functionality (keys, sliders, wheels) in a variety of combinations.
- Includes Adjacent Key Suppression® (AKS®) technology for the unambiguous detection of key events.
- Support for both IAR and GCC compiler tool chains.
- A comparison of various features and parameters between QTouch Libraries for Generic 8-bit and 32-bit AVRs as well as Device Specific Libraries is provided in the table below.

Feature Comparison between Generic QTouch Libraries and Device Specific Libraries

Parameter/Functionality	Generic Libraries, Tiny_Mega_Xmega	Tiny 2K Libraries	Tiny20 Libraries	Tiny40 Libraries	Generic Libraries, 32 Bit AVR	UC3L Libraries	ATSAM Libraries
Technology	QTouch, QMatrix	QTouch	QTouch-ADC	QTouch-ADC	QTouch, QMatrix	QTouch, QMatrix	QTouch
Rotors/Sliders Support	Yes	No	No	No	Yes	Yes	Yes
Filter Callback	Yes	Yes	No	Yes	Yes	Yes	Yes
Library Status Flags	Yes	Yes	No	Yes(Only Burst Again Flag)	Yes	Yes	Yes
Library Signature	Yes	No	No	No	Yes	Yes	Yes
Calibrate Sensing	Yes	Yes (Only burst_again flag)	No	Yes	Yes	Yes	Yes
Reset Sensing	Yes	Yes	No	Yes	Yes	Yes	Yes
Sensor Deltas	Yes	Yes	No	Yes	Yes	Yes	Yes
Maximum AKS Groups	7	7	1	7	7	7	7

Maximum Channels, QT	16	4	5	12	32	17	32
Maximum Rotors/Sliders, QT	4	0	0	0	8		8
Maximum Channels, QM	64	0	0	0	64	64	0
Maximum Rotors/Sliders, QM	8	0	0	0	8		0
Autonomous Touch	No	No	No	No	No	Yes	No
Sensor Reconfiguration	Yes	Yes	No	No	Yes	Yes	Yes
Frequency Hopping SS Enabled	Always	If _POWER_ OPTIMIZATION = 0	Never	Never	Always	Programmable	Always
Delay Cycles Parameter	QT_DELAY_CLCYES (QT Values: 1 to 255 QM Values: 1,2,3,4,5,10,25,50)	QT_DELAY_CLCYES (Value: 1 to 255)	DEF_CHARGE_SHARE_DELAY (Value: 1 to 255)	DEF_QT_CHARGE_SHARE_DELAY (Value: 1 to 255)	QT_DELAY_CYCLES (QT Values : 1 to 255 QM Values: 1,2,3,4,5,10,25,50)	xx_CHLEN, xx_SELEN (QT/QM Value: 3 to 255)	QT_DELAY_CLCYES (Value: 3 to 255)
Debug Interface Enable Macro	_DEBUG_INTERFACE_	None	NDEBUG	_DEBUG_QTOUCH_STUDIO_	_DEBUG_INTERFACE_	DEF_TOUCH_QDEBUG_ENABLE	_DEBUG_INTERFACE_

This user guide describes the content, design and use of the QTouch Libraries. This should be read in conjunction with all of the applicable documents listed below

- Device datasheet for the selected ATMEL device used for touch sensing.
- Data sheet for the selected evaluation board.
- A library selection guide that is used for the selection of the appropriate library from the released package. Default path:
  C:\ Program Files\Atmel\Atmel_QTouch_Libaries_5.x\Library_Selection_Guide.xls

The intended readers of this document are engineers, who use the QTouch Library on ATMEL microcontrollers to realize capacitive touch sensing solutions.

3 Overview

This chapter gives a brief introduction to each of the chapters that make up this document

1. **Preface**
2. **Introduction**: Provides an introduction to the scope and use of the QTouch Library.
3. **Overview**: This chapter
4. **Abbreviations and Definitions**: Provides a description of the abbreviations and definitions used in this document
5. **Generic QTouch Libraries**: Provides an overview of the QTouch libraries and the different acquisition methods for generic ATMEL devices.
6. **Device Specific Libraries:** Provides an overview of the QTouch libraries and the different acquisition methods for ATMEL devices specific for touch sensing.
7. **Revision History**: Provides a revision history of this document

4 Abbreviations and Definitions

4.1 Definitions

- **AVR:** refers to a device(s) in the tinyAVR®, megaAVR®, XMEGA™ and UC3 microcontroller family.
- **ARM: refers to a device in the ATSAM ARM® basedmicrocontroller family.**
- **ATMEL QTouch Library**: The combination of libraries for both touch sensing acquisition methods (**QTouch and QMatrix).**
- **QTouch Technology**: A type of capacitive touch sensing technology using self capacitance - each channel has only one electrode.
- **QMatrix Technology**: A type of capacitive touch sensing technology using mutual capacitance – each channel has an drive electrode (X) and an receive electrode (Y).
- **Sensor**: A channel or group of channels used to form a touch sensor. Sensors are of 3 types (keys, rotors or sliders).
- **KEY**: a single channel forms a single KEY type sensor, also known as a BUTTON
- **ROTOR**, also known as a WHEEL, a group of channels forms a ROTOR sensor to detect angular position of touch.
    - A Rotor is composed of 3 channels for a QTouch acquisition method.
    - A Rotor can be composed of 3 to 8 channels for QMatrix acquisition method.
- **SLIDER**, a group of channels forms a SLIDER sensor to detect the linear position of touch.
    - A Slider is composed of 3 channels for a QTouch acquisition method.
    - A Slider can be composed of 3 to 8 channels for QMatrix acquisition method.
- **AKS**: Adjacent Key Suppression. See Section 5.4.5
- **SNS PIN**: Sense line for capacitive measurement using the QTouch Technology - connected to Cs.
- **SNSK PIN**: Sense Key line for capacitive measurement using the QTouch Technology - connected to channel electrode through Rs.
- **X Line**: The drive electrode (or drive line) used for QMatrix Technology.
- **Y Line**: The receive electrode (or receive line) used for QMatrix Technology.
- **Port Pair**: A combination of SNS port and SNSK port to which sensors are connected for QTouch technology. The SNS and SNSK ports used in a port pair can be located in the same AVR Port (8 pins for 4 sensors), or they may be in different 2 different AVR Ports (8+8 pins for 8 sensors).
- **Charge Cycle Period**: It is the width of the charging pulse applied to the channel capacitor.
- **Dwell Cycle**: In a QMatrix acquisition method, the duration in which charge coupled from X to Y is captured.
- **Acquisition**: A single capacitive measurement process.
- **Electrode**: Electrodes are typically areas of copper on a printed circuit board but can also be areas of clear conductive indium tin oxide (ITO) on a glass or plastic touch screen.

- **Intra-port:** A configuration for QTouch acquisition method libraries, when the sensor SNS and SNSK pins are available on the same port.
- **Inter-port:** A configuration for QTouch acquisition method libraries, when the sensor SNS and SNSK pins are available on distinct ports.

5 Generic QTouch Libraries

5.1 Introduction

ATMEL QTouch provides a simple to use solution to realize touch sensing solutions on a range of supported ATMEL AVR Microcontrollers. The QTouch libraries provide support for both QTouch and QMatrix acquisition methods.

Touch sensing using QMatrix or QTouch acquisition methods can be added to an application by linking the appropriate ATMEL QTouch Library for the AVR Microcontroller and using a simple set of API to define the touch channels and sensors and then calling the touch sensing API's periodically (or based on application needs) to retrieve the channel information and determine touch sensor states.

Figure 5-1 shows a typical configuration of channels when using an AVR and using the ATMEL QTouch Library. The ATMEL QTouch Library has been added to a host application running on an AVR microcontroller. The sample configuration illustrates using the library that supports eight touch channels numbered 0 to 7. The sensors are configured in the following order,

- Sensor 0 on channels 0 to 2 have been configured as a rotor sensor.
- Sensor 1 on channels 3 to 5 have been configured as a slider sensor.
- Sensor 2 on channel 6 is configured as key sensor.
- Sensor 3 on channel 7 is configured as key sensor.

The host application uses the QTouch Library API's to configure these channels and sensors, and to initiate detection of a touch using capacitive measurements.

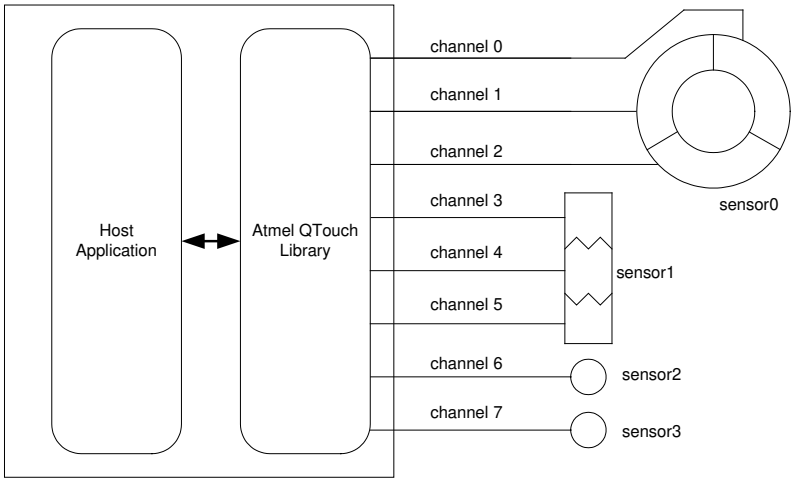


Figure 5-1 : Typical interface of the ATMEL QTouch library with the host application.

The QTouch libraries use minimal resources of the microcontroller. The sampling of the sensors is controlled by the QTouch library, while the sampling period is controlled by the application (possibly using timers, sleep periods, varying the CPU clock, external events like interrupts or communications, etc).

5.2 Acquisition Methods

There are two methods available for touch acquisition namely

1. QTouch acquisition method.
2. QMatrix acquisition method.

Libraries for AVR microcontrollers include both acquisition methods. Libraries for ATSAM microcontrollers include only QTouch acquisition method.

5.2.1 QTouch acquisition method

The QTouch acquisition method charges an electrode of unknown capacitance to a known potential. The resulting charge is transferred into a measurement capacitor (Cs). The cycle is repeated until the voltage across Cs reaches a voltage Vih. The signal level is the number of charge transfer cycles it took to reach that voltage. Placing a finger on the touch surface introduces external capacitance that increases the amount of charge transferred each cycle, reducing the total number of cycles required for Cs to reach the voltage. When the signal level (number of cycles) goes below the present threshold, then the sensor is reported to be in detected.

QTouch acquisition method sensors can drive single or multiple keys. Where multiple keys are used, each key can be set for an individual sensitivity level. Keys of different sizes and shapes can be used to meet both functional and aesthetic requirements.

NOTE: It is recommended to keep the size of the keys larger than 6mmx6mm to ensure reliable and robust measurements, although actual key design requirements also depend on panel thickness and material. Refer to the ATMEL Capacitive touch sensor design guide for details.

QTouch acquisition method can be used in two ways

- normal touch contact (i.e. when pressing buttons on a panel), and
- high sensitivity proximity mode (i.e. when a panel lights up before you actually contact it).

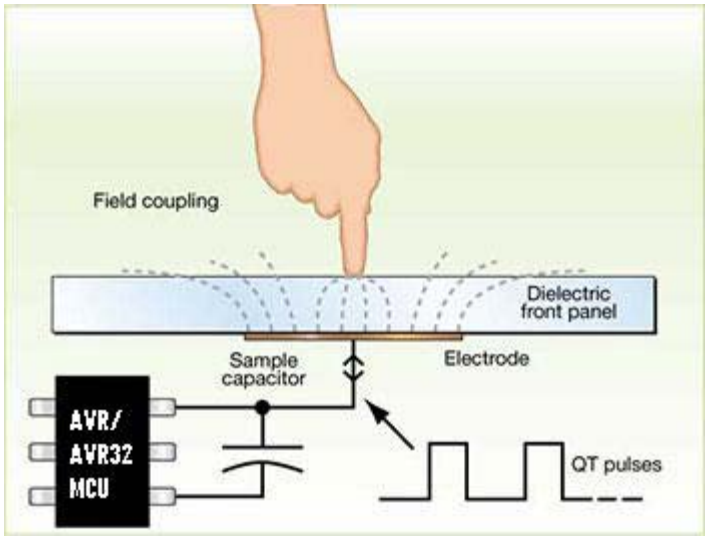
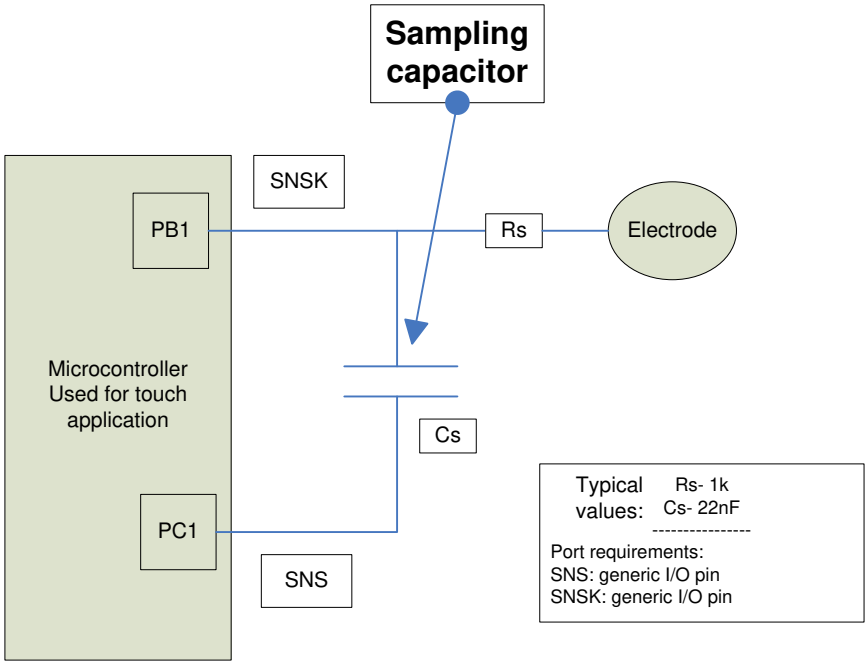


Figure 5-2 : QTouch Acquisition

QTouch circuits offers high signal-to-noise ratio, very good low power performance, and the easiest sensor layout.

5.2.1.1 Sensor schematics for a QTouch acquisition method design



Rs- Series resistor, Cs – Sample capacitor, PB1- PortB bit1, and PC1- PortC bit1

Figure 5-3 : Schematics for a QTouch acquisition method design

5.2.2 QMatrix acquisition method

QMatrix devices detect touch using a scanned passive matrix of electrode sets. A single QMatrix device can drive a large number of keys, enabling a very low cost-per-key to be achieved.

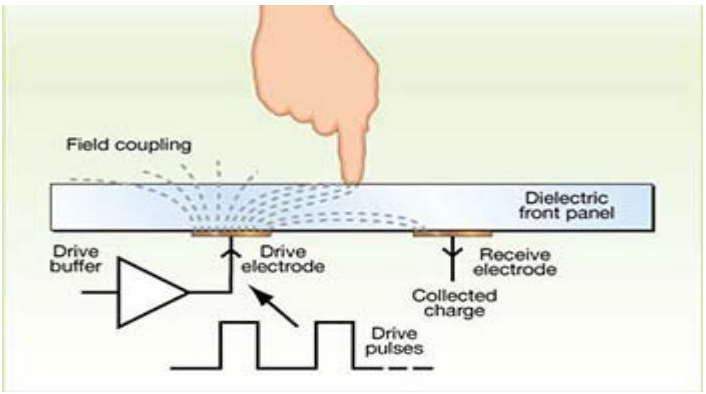


Figure 5-4 : QMatrix Acquisition method

QMatrix uses a pair of sensing electrodes for each channel. One is an emitting electrode into which a charge consisting of logic pulses is driven in burst mode. The other is a receive electrode that couples to the emitter via the overlying panel dielectric. When a finger touches the panel the field coupling is changed, and touch is detected. The drive electrode (or drive line) used for QMatrix charge transfer is labeled as the X line. The receiver electrode (or receive line) used for QMatrix charge transfer is labeled as the Y line.

QMatrix circuits offer good immunity to moisture films, extreme levels of temperature stability, superb low power characteristics, and small IC package sizes for a given key count.

5.2.3 Sensor schematics for a QMatrix acquisition method design

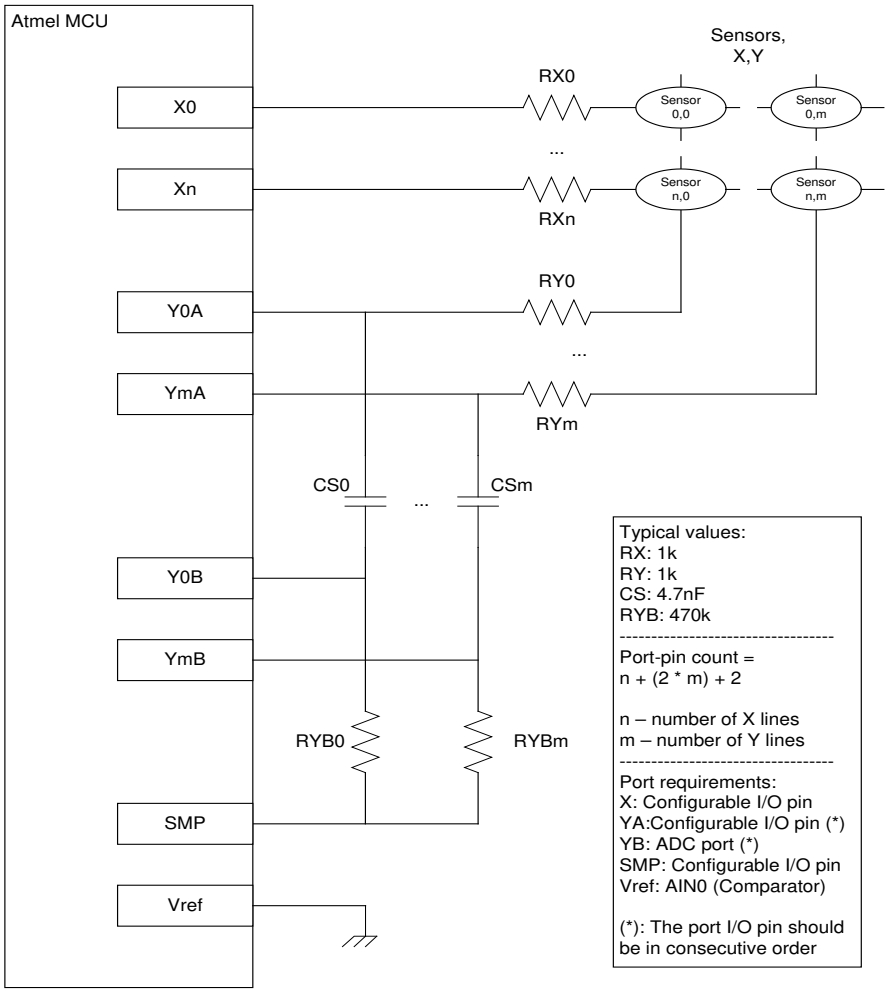


Figure 5-5 : Schematics for a QMatrix acquisition method design

5.3 Global settings common to all sensors of a specific acquisition method

The touch sensing using QTouch library could be fine tuned by using a number of configurable settings. This section explains the settings that are common to all sensors of a specific acquisition method like QMatrix or QTouch.

For example, if recalibration threshold (one of the global settings) of QMatrix acquisition method is set as 1, all QMatrix sensors will have recalibration threshold of 1.

5.3.1 Recalibration Threshold

Recalibration threshold is the level above which automatic recalibration occurs. Recalibration threshold is expressed as a percentage of the detection threshold setting. This setting is an enumerated value and its settings are as follows:

- Setting of 0 = 100% of detect threshold (RECAL_100)
- Setting of 1 = 50% of detect threshold (RECAL_50)
- Setting of 2 = 25% of detect threshold (RECAL_25)
- Setting of 3 = 12.5% of detect threshold (RECAL_12_5)
- Setting of 4 = 6.25% of detect threshold (RECAL_6_25)

However, an absolute value of 4 is the hard limit for this setting. For example, if the detection threshold is say, 40 and the Recalibration threshold value is set to 4. This implies an absolute value of 2 (40 * 6.25% = 2.5), but this is hard limited to 4.

Setting	Variable name	Data Type	Unit	Min	Max	Typical
Recalibration threshold	qt_recal_threshold	uint8_t	Enum	4	Detect threshold	1

5.3.2 Detect Integration

The QTouch Library features a detect integration mechanism, which acts to confirm detection in a robust fashion. The detect integrator (DI) acts as a simple signal filter to suppress false detections caused by spurious events like electrical noise.

A counter is incremented each time the sensor delta has exceeded its threshold and stayed there for a specific number of acquisitions, without going below the threshold levels. When this counter reaches a preset limit (the DI value) the sensor is finally declared to be touched. If on any acquisition the delta is not seen to exceed the threshold level, the counter is cleared and the process has to start from the beginning. The DI process is applicable to a ‘release’ (going out of detect) event as well.

For example, if the DI value is 10, then the device has to exceed its threshold and stay there for 10 acquisitions in succession without going below the threshold level, before the sensor is declared to be touched.

Setting	Variable name	Data Type	Unit	Min	Max	Typical
DI	qt_di	uint8_t	Cycles	0	255	4

5.3.3 Drift Hold Time

Drift Hold Time (DHT) is used to restrict drift on all sensors while one or more sensors are activated. It defines the length of time the drift is halted after a key detection.

This feature is useful in cases of high density keypads where touching a key or floating a finger over the keypad would cause untouched keys to drift, and therefore create a sensitivity shift, and ultimately inhibit any touch detection.

Setting	Variable name	Data Type	Unit	Min	Max	Typical
Drift hold time	qt_drift_hold_time	uint8_t	200 ms	1	255	20 (4s)

5.3.4 Maximum ON Duration

If an object unintentionally contacts a sensor resulting in a touch detection for a prolonged interval it is usually desirable to recalibrate the sensor in order to restore its function, perhaps after a time delay of some seconds.

The Maximum on Duration timer monitors such detections; if detection exceeds the timer's settings, the sensor is automatically recalibrated. After a recalibration has taken place, the affected sensor once again functions normally even if it still in contact with the foreign object.

Max on duration can be disabled by setting it to zero (infinite timeout) in which case the channel never recalibrates during a continuous detection (but the host could still command it).

Setting	Variable name	Data Type	Unit	Min	Max	Typical
Maximum ON Duration	qt_max_on_duration	uint8_t	200 ms	0	255	30 (6s)

5.3.5 Positive / Negative Drift

Drift in a general sense means adjusting reference level (of a sensor) to allow compensation for temperature (or other factor) effect on physical sensor characteristics. Decreasing reference level for such compensation is called Negative drift & increasing reference level is called Positive drift. Specifically, the drift compensation should be set to compensate faster for increasing signals than for decreasing signals.

Signals can drift because of changes in physical sensor characteristics over time and temperature. It is crucial that such drift be compensated for; otherwise false detections and sensitivity shifts can occur.

Drift compensation occurs only while there is no detection in effect. Once a finger is sensed, the drift compensation mechanism ceases since the signal is legitimately detecting an object. Drift compensation works only when the signal in question has not crossed the 'Detect threshold' level.

The drift compensation mechanism can be asymmetric; it can be made to occur in one direction faster than it does in the other simply by changing the appropriate setup parameters.

Signal values of a sensor tend to decrease when an object (touch) is approaching it or a characteristic change of sensor over time and temperature. Decreasing signals should not be compensated for quickly, as an approaching finger could be compensated for partially or entirely before even touching the channel (negative drift).

However, an object over the channel which does not cause detection, and for which the sensor has already made full allowance (over some period of time), could suddenly be removed leaving the sensor with an artificially suppressed reference level and thus become insensitive to touch. In the latter case, the sensor should compensate for the object's removal by raising the reference level relatively quickly (positive drift).

Setting	Variable name	Data Type	Unit	Min	Max	Typical
Negative Drift	qt_neg_drift_rate	uint8_t	200 ms	1	127	20 (4s)
Positive Drift	qt_pos_drift_rate	uint8_t	200 ms	1	127	5 (1s)

5.3.6 Positive Recalibration Delay

If any key is found to have a significant drop in signal delta, (on the negative side), it is deemed to be an error condition. If this condition persists for more than the positive recalibration delay, i.e., qt_pos_recal_delay period, then an automatic recalibration is carried out.

A counter is incremented each time the sensor delta is equal to the positive recalibration threshold and stayed there for a specific number of acquisitions. When this counter reaches a preset limit (the PRD value) the sensor is finally recalibrated. If on any acquisition the delta is seen to be greater than the positive recalibration threshold level, the counter is cleared and the positive drifting is performed.

For example, if the PRD value is 10, then the delta has to drop below the recalibration threshold and stay there for 10 acquisitions in succession without going below the threshold level, before the sensor is declared to be recalibrated.

Setting	Variable name	Data Type	Unit	Min	Max	Typical
Positive Recalibration Delay	qt_pos_recal_delay	uint8_t	cycles	1	255	3

5.4 Sensor specific settings

Apart from global settings as mentioned in the section above, touch sensing using QTouch library could also be fine tuned by more number of configurable settings.

This section explains the settings that are specific to each sensor. For example, sensor 0 can have a detect threshold (one of the sensor specific setting) that is different from sensor 1.

5.4.1 Detect threshold

A sensor's negative (detect) threshold defines how much its signal must drop below its reference level to qualify as a potential touch detect. The final detection confirmation must however satisfy the Detect Integrator (DI) limit. Larger threshold values desensitize sensors since the signal must change more (i.e. requires larger touch) in order to exceed the threshold level. Conversely, lower threshold levels make sensors more sensitive.

Threshold setting depends on the amount of signal swing that occurs when a sensor is touched. Thicker front panels or smaller electrodes usually have smaller signal swing on touch, thus require lower threshold levels.

Setting	Variable name	Data Type	Unit	Min	Max	Typical
Threshold	threshold	uint8_t	counts	3	255	10 – 20

5.4.2 Hysteresis

This setting is sensor detection hysteresis value. It is expressed as a percentage of the sensor detection threshold setting. Once a sensor goes into detect its threshold level is reduced (by the

hysteresis value) in order to avoid the sensor dither in and out of detect if the signal level is close to original threshold level.

- Setting of 0 = 50% of detect threshold value (HYST_50)
- Setting of 1 = 25% of detect threshold value (HYST_25)
- Setting of 2 = 12.5% of detect threshold value (HYST_12_5)
- Setting of 3 = 6.25% of detect threshold value (HYST_6_25)

Setting	Variable name	Data Type	Unit	Min	Max	Typical
Hysteresis	detect_hysteresis	uint8_t (2 bits)	Enum	HYST_6_25	HYST_50	HYST_6_25

5.4.3 Position Resolution

The rotor or slider needs the position resolution (angle resolution in case of rotor and linear resolution in case of slider) to be set. Resolution is the number of bits needed to report the position of rotor or slider. It can have values from 2bits to 8 bits.

Setting	Variable name	Data Type	Unit	Min	Reported position	Max	Reported position	Typical
Position Resolution	position_ resolution	uint8_t (3 bits)	-	2 bits	0 – 3	8 bits	0-255	8

5.4.4 Position Hysteresis

In case of QMatrix, the rotor or slider needs the position hysteresis (angle hysteresis in case of rotor and linear hysteresis in case of slider) to be set. It is the number of positions the user has to move back, before touch position is reported when the direction of scrolling is changed and during the first scrolling after the touch down.

Hysteresis can range from 0 (1 position) to 7 (8 positions). The hysteresis is carried out at 8 bits resolution internally and scaled to desired resolution; therefore at resolutions lower than 8 bits there might be a difference of 1 reported position from the hysteresis setting, depending on where the touch is detected.

At lower resolutions, where skipping of the reported positions is observed, hysteresis can be set to 0 (1 position). At Higher resolutions (6 ..8bits) , it would be recommended to have a hysteresis of at least 2 positions or more.

NOTE:
It is not valid to have a hysteresis value more than the available bit positions in the resolution.
Ex: do not have a hysteresis value of 5 positions with a resolution of 2 bits (4 positions).

Setting	Variable name	Data Type	Unit	Min	Max	Typical
Position Hysteresis	position_hysteresis	uint8_t (3 bits)	-	0	7	3

NOTE:
Position hysteresis is not valid (unused) in case of QTouch acquisition method libraries.

5.4.5 Adjacent Key Suppression (AKS)

In designs where the sensors are close together or set for high sensitivity, multiple sensors might report detect simultaneously if touch is near them. To allow applications to determine the intended single touch, the touch library provides the user the ability to configure a certain number of sensors in an AKS group.

When a group of sensors are in the same AKS group, then only the first strongest sensor will report detection. The sensor reporting detection will continue to report detection even if another sensor's delta becomes stronger. The sensor stays in detect until its delta falls below its detection threshold, and then if any more sensors in the AKS group are still in detect then the strongest will report detection. So at any given time only one sensor from each AKS group will be reported to be in detect.

The library provides the ability to configure any sensor to be included in any one of the Adjacent Key Suppression Groups (AKS Group).

Setting	Variable name	Data Type	Unit	Min	Max	Typical
AKS Group	aks_group	uint8_t (3 bits)	Enum	0 (off)	7	0 (off)

5.5 Using the Sensors

5.5.1 Avoiding Cross-talk

In ATMEL QTouch library variants that use QTouch acquisition technology, adjacent sensors are not measured at the same time. This prevents interference due to cross-talk between adjacent channels, but at the same time some sensor configurations take longer to measure than others. For example, if an 8-channel device is configured to support 8 keys, then the library will measure the keys on channels 0, 2, 4, and 6 parallely, followed by keys on channels 1, 3, 5, and 7. If the same device is configured, say, to support 4 keys, putting them either on all the odd channels or on all the even channels means that they can all be measured simultaneously.

This means the library calls are faster, and the device can use less power. So, it is recommended that the appropriate channel numbers are used when using less than the maximum number of channels available for the device to ensure optimum performance. In a similar sense for faster execution and reduced power consumption, it is also advisable to use intra-port sensor configuration instead of inter-port sensor configuration while using 4 channels on the same port.

5.5.2 Multiple measurements

The library will not automatically perform multiple measurements on a sensor (Ex: To resolve for instance Detect Integration or recalibration.). The user is given the option to perform the measurement multiple times if certain conditions are met. This will enable the user to implement the time critical code thereby making the qt_measure_sensors() a non-blocking API .The host application has to perform multiple measurements, based on the need. The global flag QTLIB_BURST_AGAIN indicating that multiple measurements are needed is passed to the user. This is BIT8 of the return value from the qt_measure_sensors( ) API. The main_<devicename>.c has the example usage to perform multiple measurements.

If QTLIB_BURST_AGAIN = 1, multiple measurements are needed to

- To compensate for drift
- Resolve re-calibration
- Resolve calibration.
- Resolve detect integration.

If QTLIB_BURST_AGAIN = 0, multiple measurements are not needed and the user can execute the host application code. Apart from QTLIB_BURST_AGAIN, various flags are provided to the user to perform the multiple measurements based on the need of the host application to act to specific situation. Description of the these flags can be found in the section5.6.5.6

Note: To maintain robustness and timing of the touch sensing measurement, it is recommended that the user calls the qt_measure_sensors() immediately if the flag QT_BURST_AGAIN=1. However, the user is allowed to run time- critical section (not more than few instructions) of the host application comprising on the touch sensing timing.

5.5.3 Guard Channel

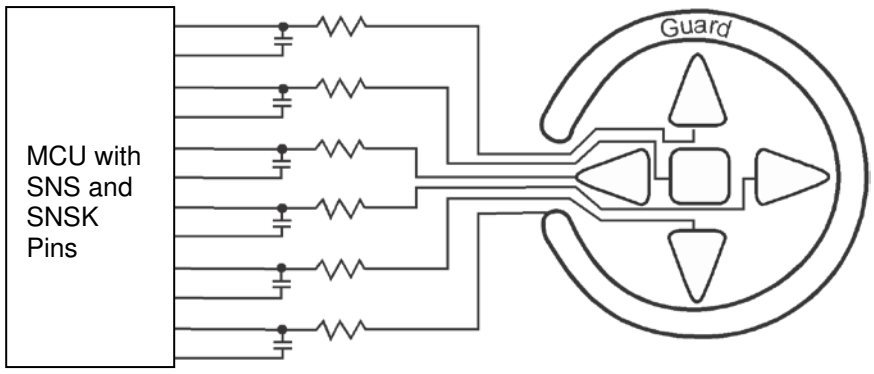
Guard channel in Qtouch Acquisition Method allows one key to be configured as a guard channel to help prevent false detection. Guard channel keys should be more sensitive than the other keys (physically bigger or larger Cs).To enable key as guard channel, the designated key is connected to a sensor pad which detects the presence of touch and overrides any output from the other keys using the AKS feature.

The key can be configured to have a guard channel function by adjusting a number of independent settings. The Guard channel is designed so that it is likely to be activated unless a key is accurately touched.

The guard channel sensor must be set up so that it is slightly more sensitive than the keys that it surrounds. The exact amount of increase depends on the application and is best determined by experimentation.
There are three methods of increasing the sensor sensitivity that can be used in combination:

1. Increasing the size of the sensor.
2. Increasing the value of the Sample Capacitor (Cs).
3. Adjust the detection threshold for the sensor.

The sensor size and capacitor values should be altered to establish the base sensitivity for the sensor. Once these values have been established, the detection threshold can be used to fine tune the sensor.



The Above figure illustrates how a Guard sensor/key is to be visualized.It has six keys and five keys are surrounded by a Guard Channel.

Please refer QTAN0031 for further information on Guard Channel.
http://www.atmel.com/dyn/resources/prod_documents/QTAN0031(2).pdf

5.6 QTouch API and Usage

The Atmel QTouch library provides support for many devices. This chapter explains the touch library for such devices without any hardware support.

5.6.1 QTouch Library API

This section describes the QTouch library Application Programming Interface (API) for touch sensing using QTouch and QMatrix acquisition methods.

Using the API, Touch sensors and the associated channels can be defined. Once touch sensing has been initiated by the user, the host application can use the API to make touch measurements and determine the status of the sensors.

5.6.2 touch_api.h - public header file

The *touch_api.h* header file is the public header file which needs to be included in users application and it has the type definitions and function prototypes of the API's listed in sections 5.6.3 , 5.6.4 and 5.6.5

The touch_api.h header file is located in the library distribution in the following directory.

- *..\Atmel_QTouch_Libraries_5.x\Generic_QTouch_Libraries\include*

5.6.3 Type Definitions and enumerations used in the library

5.6.3.1 Typedefs

This section lists the type definitions used in the library.

Typedef	Notes
uint8_t	unsigned 8-bit integer
int8_t	signed 8-bit integer
uint16_t	unsigned 16-bit integer
int16_t	signed 16-bit integer
uint32_t	unsigned 32-bit integer
threshold_t	unsigned 8-bit integer used for setting a sensor detection threshold

5.6.3.2 Enumerations

This section lists the enumerations used in the QTouch Library.

5.6.3.2.1 sensor_type_t

Enumeration sensor_type_t
Use Define the type of the sensor

Values	Comment
SENSOR_TYPE_UNASSIGNED	Channel is not assigned to any sensor
SENSOR_TYPE_KEY	Sensor is of type KEY
SENSOR_TYPE_ROTOR	Sensor is of type ROTOR

SENSOR_TYPE_SLIDER	Sensor is of type SLIDER

5.6.3.2.2 aks_group_t

Enumeration aks_group_t

Use Defines the Adjacent Key Suppression (AKS) groups each sensor may be associated with (see section 5.3.4 Maximum ON Duration)

AKS is selectable by the system designer
7 AKS groups are supported by the library

Values	Comment
NO_AKS_GROUP	NO AKS group selected for the sensor
AKS_GROUP_1	AKS Group number 1
AKS_GROUP_2	AKS Group number 2
AKS_GROUP_3	AKS Group number 3
AKS_GROUP_4	AKS Group number 4
AKS_GROUP_5	AKS Group number 5
AKS_GROUP_6	AKS Group number 6
AKS_GROUP_7	AKS Group number 7

5.6.3.2.3 channel_t

Enumeration channel_t

Use The channel numbers used in the library.

When using the QTouch acquisition method, the channel numbers have a one to one mapping to the pin numbers of the port being used.

When using the QMatrix acquisition method, the channel numbers are ordered in a matrix sequence

Values	Comment
CHANNEL_0	Channel number : 0
CHANNEL_1	Channel number : 1
CHANNEL_2	Channel number : 2
CHANNEL_3	Channel number : 3
.....	Channel number: ..
Upto CHANNEL (N-1)	Channel number N-1 : for an N Channel library

The maximum number of channels supported is dependent on the library variant. Possible values of N are as listed below

Acquisition method	Device type	Possible values of N (Maximum number of channels)
QTouch acquisition	8-bit	4,8,16
	32-bit	8, 16, 32
QMatrix Acquisition	8-bit	8,16,32,64

5.6.3.2.4 hysteresis_t

Enumeration Hysteresis_t

Use Defines the sensor detection hysteresis value. This is expressed as a percentage of the sensor detection threshold.

This is configurable per sensor.

HYST_x = hysteresis value is x percent of detection threshold value (rounded

down).
Note that a minimum value of 2 is used as a hard limit. Example: if detection threshold = 20, then:
HYST_50 = 10 (50 percent of 20)
HYST_25 = 5 (25 percent of 20)
HYST_12_5 = 2 (12.5 percent of 20)
HYST_6_25 = 2 (6.25 percent of 20 = 1, but set to the hard limit of 2)

Values	Comment
HYST_50	50% Hysteresis
HYST_25	25% Hysteresis
HYST_12_5	12.5% Hysteresis
HYST_6_25	6.25% Hysteresis

5.6.3.2.5 resolution_t

Enumeration resolution_t

Use For rotors and sliders, the resolution of the reported angle or position.
RES_x_BIT = rotor/slider reports x-bit values.
Example: if slider resolution is RES_7_BIT, then reported positions are in the range 0…127.

Values	Comment
RES_1_BIT	1 bit resolution : reported positions range 0 – 1
RES_2_BIT	2 bit resolution : reported positions range 0 – 3
RES_3_BIT	3 bit resolution : reported positions range 0 – 7
RES_4_BIT	4 bit resolution : reported positions range 0 – 15
RES_5_BIT	5 bit resolution : reported positions range 0 – 31
RES_6_BIT	6 bit resolution : reported positions range 0 – 63
RES_7_BIT	7 bit resolution : reported positions range 0 – 127
RES_8_BIT	8 bit resolution : reported positions range 0 – 255

5.6.3.2.6 recal_threshold_t

Enumeration recal_threshold_t

Use A sensor recalibration threshold. This is expressed as a percentage of the sensor detection threshold.

This is for automatic recovery from false conditions, such as a calibration while sensors were touched, or a significant step change in power supply voltage.
If the false condition persists the library will recalibrate according to the settings of the recalibration threshold.

This setting is applicable to all the configured sensors.

Usage :
RECAL_x = recalibration threshold is x percent of detection threshold value (rounded down).
Note: a minimum value of 4 is used.
Example: if detection threshold = 40, then:
RECAL_100 = 40 (100 percent of 40)
RECAL_50 = 20 (50 percent of 40)
RECAL_25 = 10 (25 percent of 40)
RECAL_12_5 = 5 (12.5 percent of 40)
RECAL_6_25 = 4 (6.25 percent of 40 = 2, but value is limited to 4)

Values	Comment

RECAL_100	100% recalibration threshold
RECAL_50	50% recalibration threshold
RECAL_25	25% recalibration threshold
RECAL_12_5	12.5% recalibration threshold
RECAL_6_25	6.25% recalibration threshold

5.6.4 Data structures

This section lists the data structures that hold sensor status, settings, and diagnostics information

5.6.4.1 qt_touch_status_t

Structure qt_touch_status_t
Input / Output Output from the Library
Use Holds the status (On/ Off) of the sensors and the linear and angular positions of sliders and rotors respectively

Fields	Comment
sensor_states[]	For Sensor, the sensor_states. Bit “n” = state of nth sensor : Bit Value 0 - indicates the sensor is not in detect Bit Value 1 - indicates the sensor is in detect
rotor_slider_values[]	Rotors angles or slider positions if rotors and sliders are used. These values are valid when sensor states shows that the corresponding rotor or slider is in detect

The macro that can get the sensor state when the sensor number is provided can be something as below:

```
#define     GET_SENSOR_STATE(SENSOR_NUMBER)
      qt_measure_data.qt_touch_status.sensor_states[(SENSOR_NUMBER/8)] &
(1 << (SENSOR_NUMBER % 8))
```

The host application can use this macro to act accordingly, the following example shows how to toggle a IO pin (PD2) based on the sensor0 state.(Set PD2 if sensor0 is in detect, and clear PD2 if sensor0 is not in detect)

```
Ex:   /*Set pin PD2 direction as output*/
      DDRD |= (1u << PORTD2);
      if (GET_SENSOR_STATE(0) !=0)
      {     PORTD |= (1u << PORTD2); /* Set PORTD2 */
      }
      else {
            PORTD &= ~(1u << PORTD2); /* Clear PORTD2 */
      }
```

5.6.4.2 qt_touch_lib_config_data_t

Structure qt_touch_lib_config_data_t
Input / Output Input to the library