Atmel QTouch Surface Library

SAM D20 Peripheral Touch Controller

USER GUIDE

Table of Contents

1. Features

- Implements low-power, high-sensitivity, and robust capacitive touch surfaces
- Supports two touches with up to 2mm edge-to-edge touch separation
- Wake-up on surface touch from a standby current down to 4µA
- Scan rates up to 100Hz
- Supports up to 100 nodes with varying surface sizes
- Parasitic capacitance compensation for mutual capacitance mode
- Adjustable gain for superior sensitivity
- No need of temperature or VDD compensation
- Hardware noise filtering and noise signal de-synchronization for high conducted immunity
- Atmel provided QTouch Library firmware and QTouch Composer tool

Note:
PTC can be operated at 8MHz only for operating voltage range 2.5V -3.3V.

2. Product Support

For assistance related to QTouch capacitive touch sensing software libraries and related issues, contact your local Atmel sales representative or log on to myAtmel Design Support portal to submit a support request or access a comprehensive knowledge base.

If you do not have a myAtmel account, please visit http://www.atmel.com/design-support/ to create a new account by clicking on "Create Account" in the myAtmel menu at the top of the page.

Once logged in, you will be able to access the knowledge base, submit new support cases from the myAtmel page or review status of your ongoing cases.

3. Device Variants Supported

QTouch Surface Library for SAM D20 is available for the following device variants:

Series	Variant
SAM D20 J Series	ATSAM D20J18, ATSAM D20J17, ATSAM D20J16, ATSAM D20J15
SAM D20 G Series	ATSAM D20G18, ATSAM D20G17, ATSAM D20G16, ATSAM D20G15
SAM D20 E Series	ATSAM D20E18, ATSAM D20E17, ATSAM D20E16, ATSAM D20E15

4. Development Tools

The following development tools are required for developing QTouch Surface Library development using PTC SAM devices:

Development Environment for GCC Compiler:

- Atmel Studio 6.2.1548
- Atmel Software Framework 3.22.0.1370 or later versions

Development Environment for IAR Compiler:

- IAR Embedded Workbench for ARM 7.30.4.8187

Other Pre-requisites

- Atmel QTouch Composer 5.6.176 or higher
- Atmel QTouch Library 5.6.218 or higher

5. Surface Library

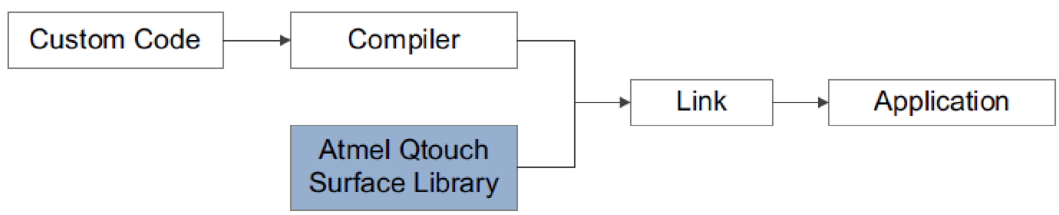
Atmel QTouch Surface Library makes it simple for developers to embed capacitive-touch surface functionality into general-purpose Atmel SAM D20 microcontroller applications. The royalty-free QTouch Surface Library provides library files for various device and supports a number of touch surface combinations, enabling both flexibility and efficiency in touch applications.

QTouch Surface Library can be used to develop single-chip solutions for various control applications, or to reduce the chip.

5.1. API Overview

QTouch Surface Library can be used to develop single-chip solutions for many control applications, or to reduce chip count in more complex applications.

Figure 5-1 QTouch Overview



For normal operation, it is sufficient to use the set of Regular APIs set for each method. The Helper APIs provide addedadditional flexibility to the user application.

Regular API	Helper API
`surf_init`	`surf_library_get_version_info`
`touch_mutlcap_sensor_config`	
`surf_calibrate_all`	
`surf_measure`	
`surf_low_power_start`	
`surf_low_power_stop`	

5.2. Two Dimensional Sensing

QTouch Surface uses mutual-capacitance technology to detect touches in two dimension. Mutual-capacitance technology inherently allows multi-touch operation and additional algorithms in QTouch Surface providing the capability to separate touches at 2mm edge-edge distance.

This section elaborates the important characteristics of two-dimensional sensing. All characteristics are partially dependent on the following design considerations: sensor design, touch diameter, movement speed, touch separation distance, noise & moisture condition, and filtering techniques.

5.2.1. Resolution

Resolution refers to the smallest change in touch movement that can be measured by the QTouch Surface library. The resolution depends on individual sensor's ability to reflect the movement as change in the signal level with respect to the neighbouring sensor's signal. When the change is higher in the signal level with respect to every unit of movement, a higher resolution can be supported.

QTouch Surface's resolution is expressed in Dots Per Inch (DPI).

For example, a 200 DPI system along X-axis can resolve and report 200 unique positions within an inch on the X-axis.

5.2.2. Accuracy

Accuracy of a reported touch position describes the difference between the actual touch location (X or Y) and the measured touch location in a particular measurement cycle. The measured deviation can be expressed in position or mm. The accuracy of a measured touch location depends on the quality of measured signals and the filtering techniques. Since the stability of the measured signals can be different during specific user actions (entering touch, exiting touch, stable stationary touch), the accuracy of a touch would differ accordingly. When a touch is entering or exiting, it could lead to signal changes and correspondingly lower accuracy. A stable stationary touch on the other hand would let the filtering techniques to stabilize the signal and hence a higher accuracy could be achieved.

5.2.3. Jitter

Jitter in touch position is described for stationary touches. Under stationary conditions, Jitter refers to the deviation in the reported touch location. The measured deviation can be expressed as position or mm. Jitter and Accuracy are closely related characteristics of a surface they are not the same. For example, under certain conditions, a surface could have low accuracy and high jitter or vice-versa.

5.2.4. Linearity

Linearity refers to the property of measured XY position accurately reflecting a moving touch. In a QTouch Surface, linearity can be improved using better sensor design and filtering techniques. Linearity error, under touch movement cases, can be measured as the deviation between a line representing the actual touch movement (X or Y) and another line representing the measured touch location. The measured deviation can be expressed as peak and average units either in position or mm. Here, "measured touch location" refers to the X-Y position reported by the QTouch Surface library.

Actual touch movement can be constructed using one of the following methods:

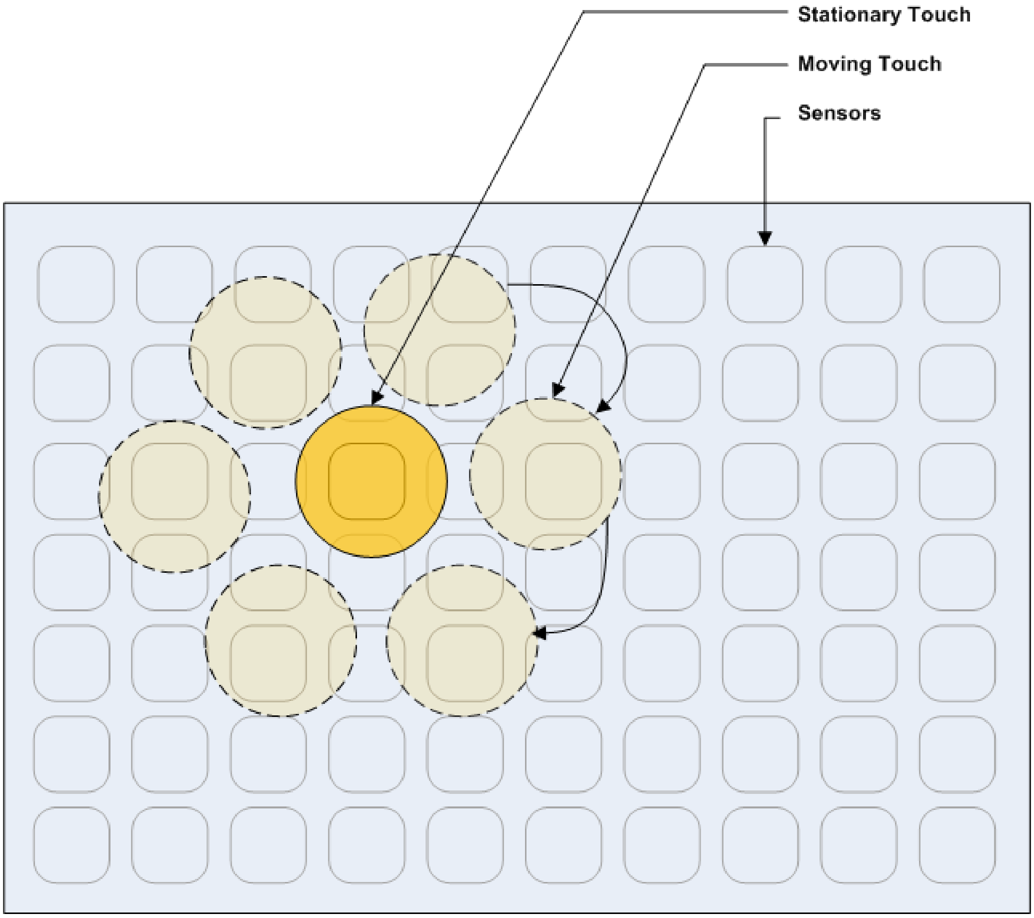
- In case of robotic testing, the test set-up parameters such as angle, speed and other parameters can be used to predict the touch location at every measurement instant
- In any test condition, a line of best fit can be constructed for the "measured touch location" to represent the "actual touch movement"

5.2.5. Touch Separation

Touch surface supports multi-touch measurement on the 2 Dimensional surface with 2mm edge to edge gap between touches. This means, more than one touch can be identified and reported successfully with all the touch parameters namely ID, XY position, area and strength.

Detecting touches that are close to each other is a challenging task. As long as the two touches are separated by at least one off-state sensor in-between, they can be identified easily by taking account of the low signal value observed on the separating node. The touch parameters can be calculated directly in this case. But, when touches get closer to each other, say 2mm edge-edge separation for a sensor size of 5mm, the two touches are merged and this makes the detection complicated. The touch surface uses the touch separation algorithm to separate these closest touches, calculates the touch parameters and reports them to the application. Crosstalk and shadowing are the two common problems that are encountered when touches become closer.

Figure 5-2 Cross-talk

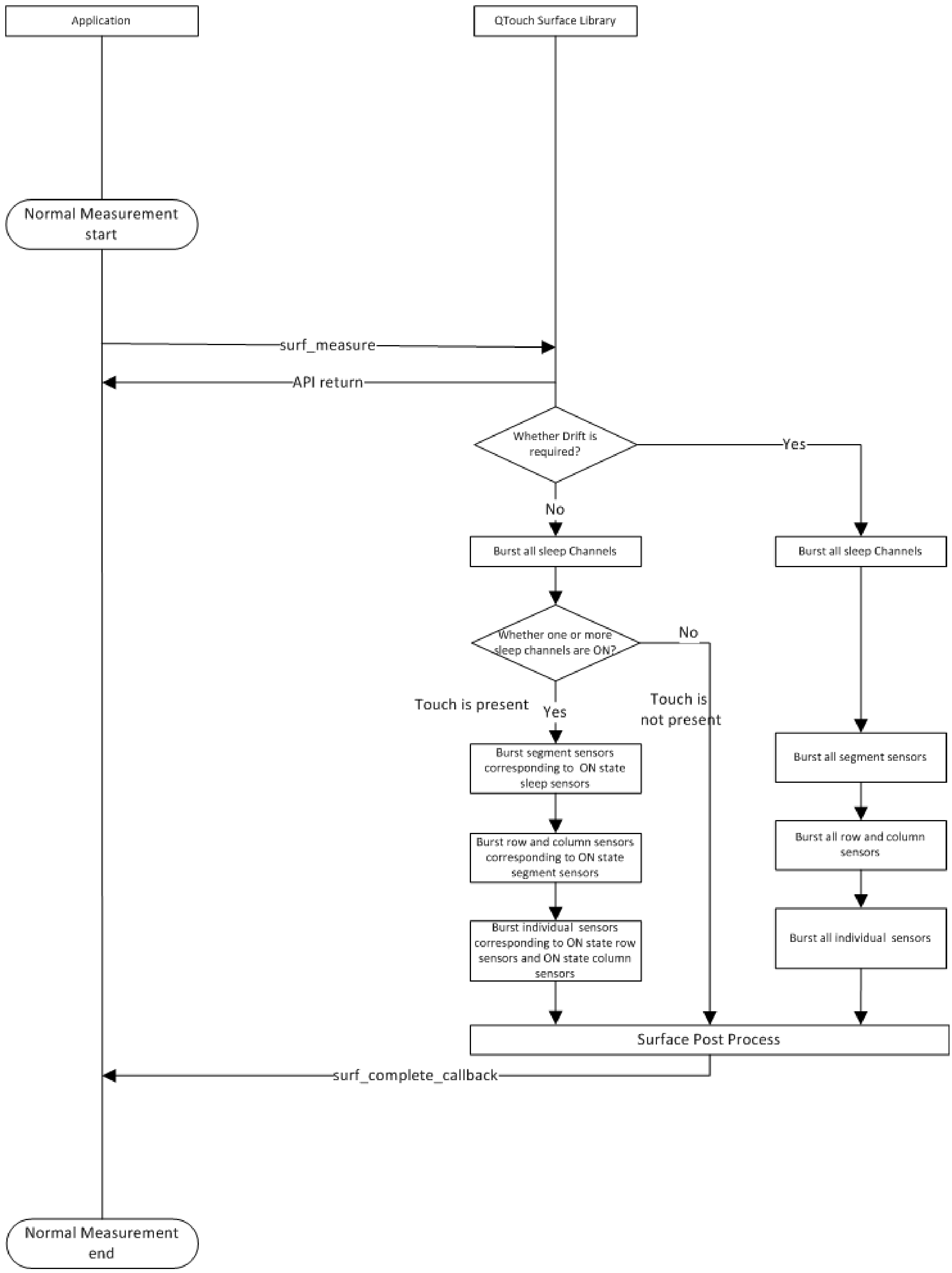


Crosstalk refers to the sudden rise/decrease in the position jitter of the stationary touch when the other touch comes closer to the stationary touch. Cross-talk can be measured/expressed in the same method as Jitter. Shadowing problem refers to the disappearance of a stationary touch when the other touch scrolls nearer to the stationary touch. Both the factors can be tested and monitored by rotating one touch 360 degrees against the stationary touch.

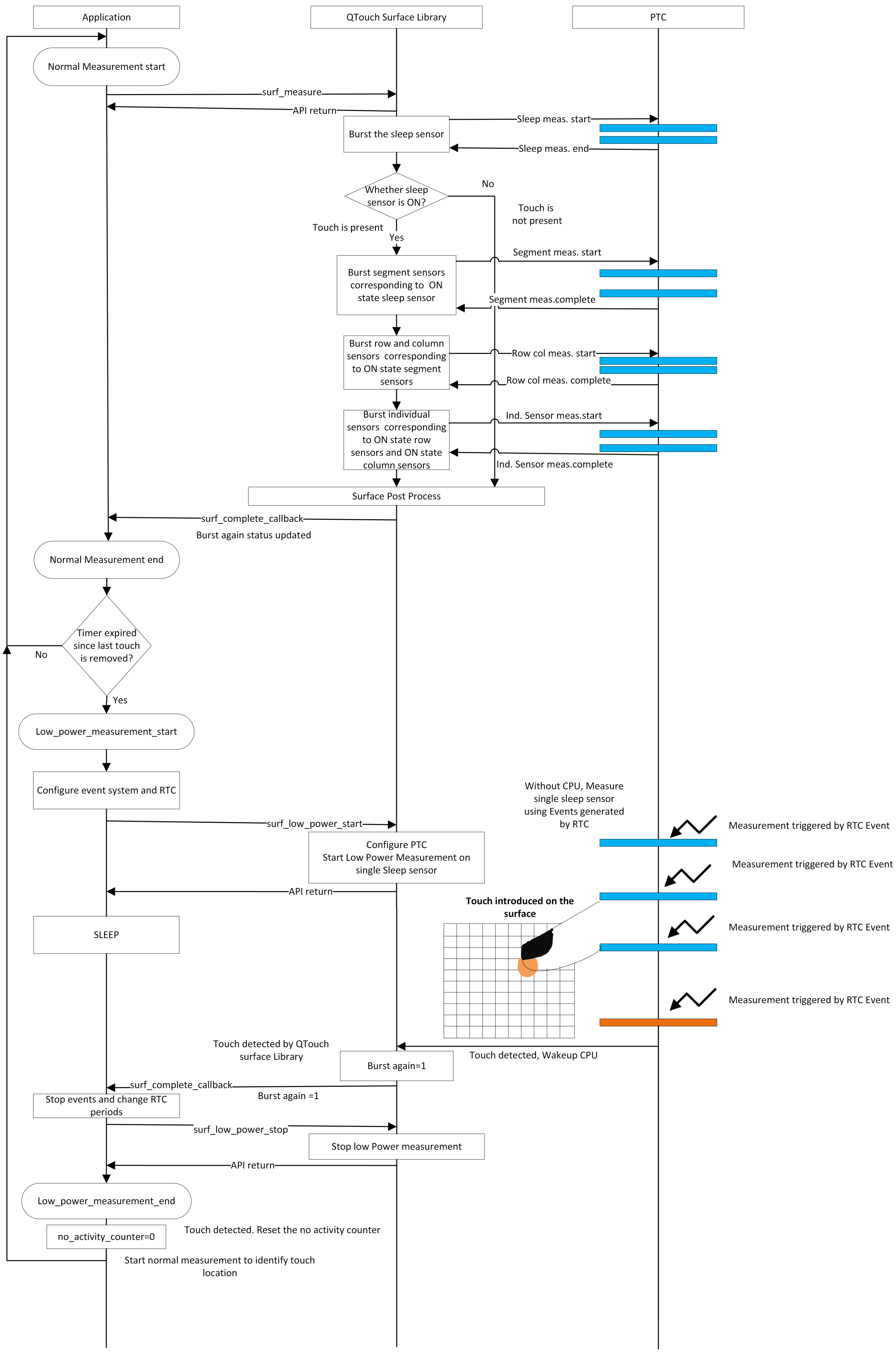
5.3. Sequence of Operation

The QTouch Surface Library can be configured either in normal measurement mode or in low power mode. In low power mode utilizes only one sleep channel and normal measurement mode utilizes more than 1 channel. This library provides these two different APIs for different purposes.

5.3.1. Normal Measurement Mode



5.3.2. Low Power Measurement Mode



5.4. Optimized Bursting

The QTouch Surface library is designed to save power, reduce CPU load and improve response time by using an optimized touch acquisition algorithm.To discern a touch on the touch surface, the number of sensors which are burst are optimized based on algorithms. This is referred as optimized bursting in this document. QTouch PTC allows multiple sensors to be grouped together for bursting. A single group of sensor takes almost the same amount of time and power as an individual sensor for bursting task. Such a group of sensors is known as a ‘Lumped sensor’.

The optimized way of touch acquisition is performed through four stages of bursting sequence. During this course of action, the individual nodes that constitutes the touch are identified, burst and processed.

The first three stages involves bursting over lumped sensor configurations namely Sleep, Segment and Row & Column while the last stage is over individual sensors. All the four stage of operation are performed in sequential manner wherein the result of previous stage is used to burst the sensors in the next stage. At the end of the fourth stage, the complete node information that belongs to the touch are captured and processed by the QTouch Surface library.

The table given below shows the bursting sequence under various scenarios. The Column Bursting signifies the sensors which will be burst in different scenarios as mentioned in the other columns.

S.No	Bursting	Sleep(No-touch) state regular measurement	Sleep(No-touch) state drifting measurement	Active state measurement
1	Sleep	All	All	All
2	Segment	None	All	Based on touch location
3	Row and Column	None	All	Based on touch location
4	Individual	None	All	Based on touch location

The successive sections describe the four type of sensors assuming a 10X * 10Y surface.

The sensors within a stage are described as:

Sa-b, where

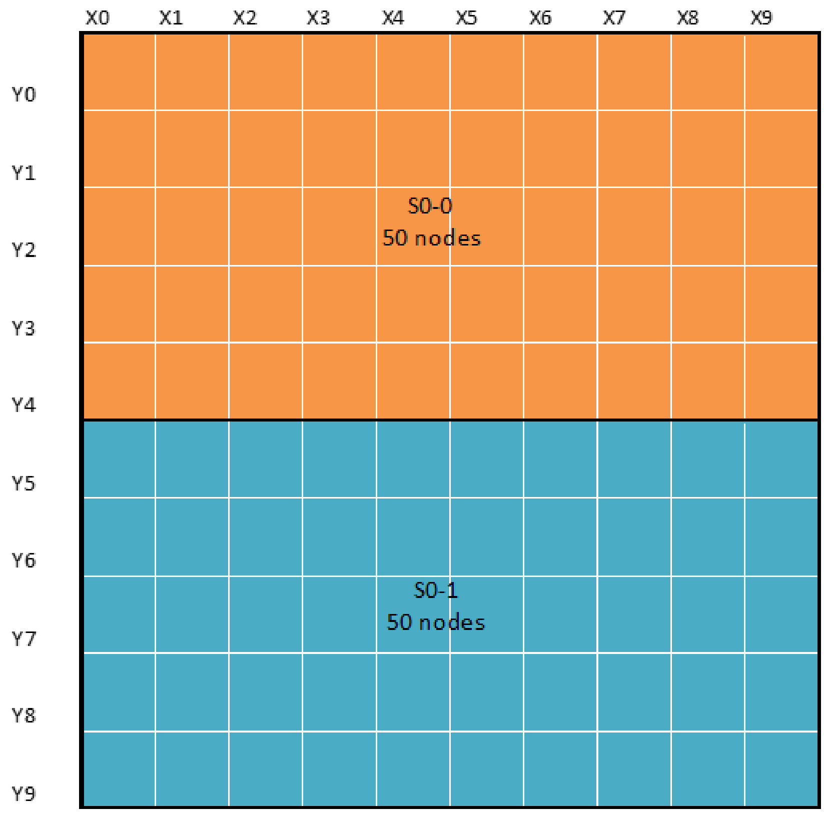
- ‘a’ stands for the stage (0-Sleep, 1-Segment, 2-Row and Column, 3-Individual Node)
- 'b’ identifies the sensor within the stage

5.4.1. Sleep Sensors

Sleep sensors can be formed by grouping sensors along rows or columns. In this example, the Lumped Sensors (S0-0 and S0-1) are formed by grouping rows. When QTouch composer is used to build a surface project, the composer chooses the ideal configuration of sleep sensors based on the following parameters:

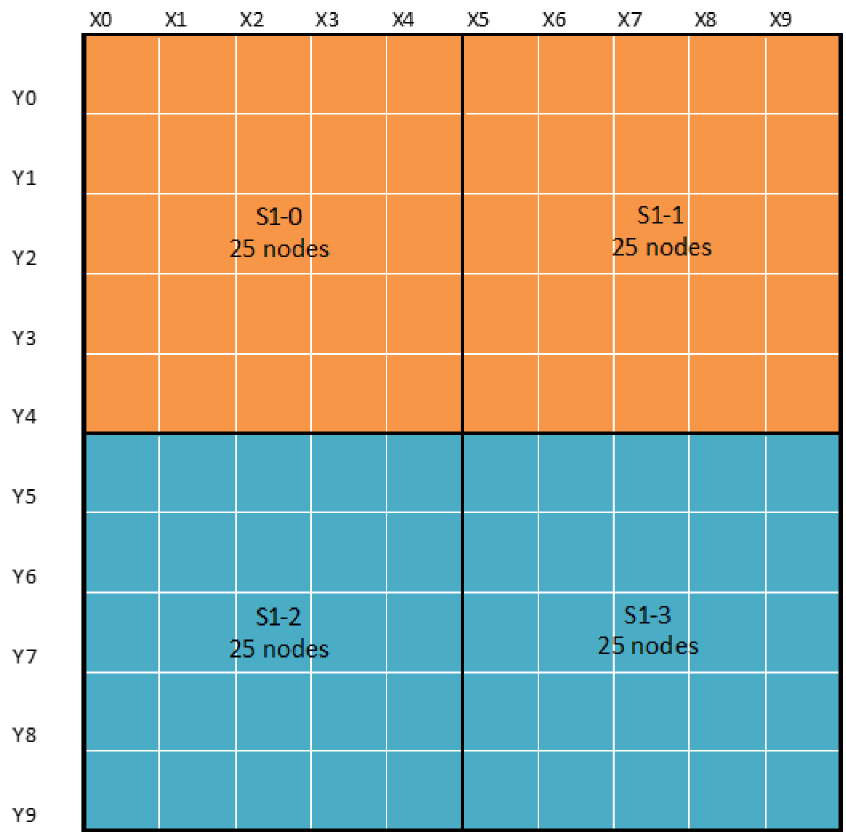
- Number of rows and columns in the surface
- Maximum number of individual nodes allowed per lump

Figure 5-3 Sleep Sensors with Lump Node Size =50



5.4.2. Segment Sensors

Figure 5-4 Segment Sensors

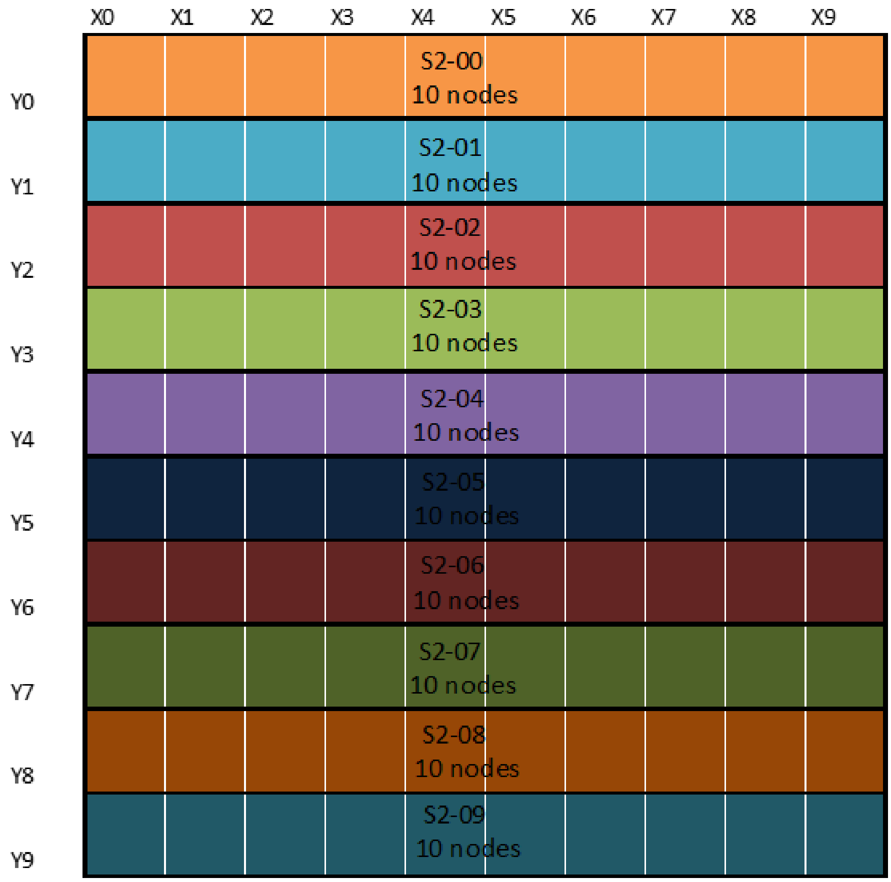


The Lumped Sensors arrangement for segment sensors is formed by splitting the Sleep Sensors along the other grouping axis. For example: If Sleep sensor grouping is along the Y axis, then the Lumped Sensors will be split along the X axis to form Segment Sensors.

The total number of Segment sensors would always be twice the number of sleep sensors

5.4.3. Row-Column Sensors

Figure 5-5 Row-Column Sensors



The Lumped Sensors arrangement for row and column sensors are formed by grouping the individual sensors on each row as one set of lumped sensors and similarly for the nodes on each column are grouped as another set of lumped sensors. Optimized bursting algorithm identifies rows/columns that needs to be burst from the previous stage and are burst accordingly.

5.4.4. Individual Nodes

This stage utilizes the output of the previous stage to identify the individual nodes that belongs to the touch/touches and they are burst sequentially in this stage. The illustrations in this section depict the operation regarding optimized bursting under no touch, touch and drifting scenarios. For more information, See Sequence of Operation.

Note:

- The number of sensors burst to locate a touch depends on various surface configuration parameters.
- The number of sensors burst in case of no-touch conditions depends on drift configuration, calibration status, etc.
- All the lumped sensors configuration described in the earlier sections can be configured by using QTouch Composer.
- The number of sensors burst in low power mode, changes with location of the touch, drifting period and event generation for low power measurement.

- The typical number of sensors burst in case of normal measurement (without low power mode) depends on the location of touch, sensors configuration, drifting period etc.

Figure 5-6 Column Sensors (Columns are lumped)

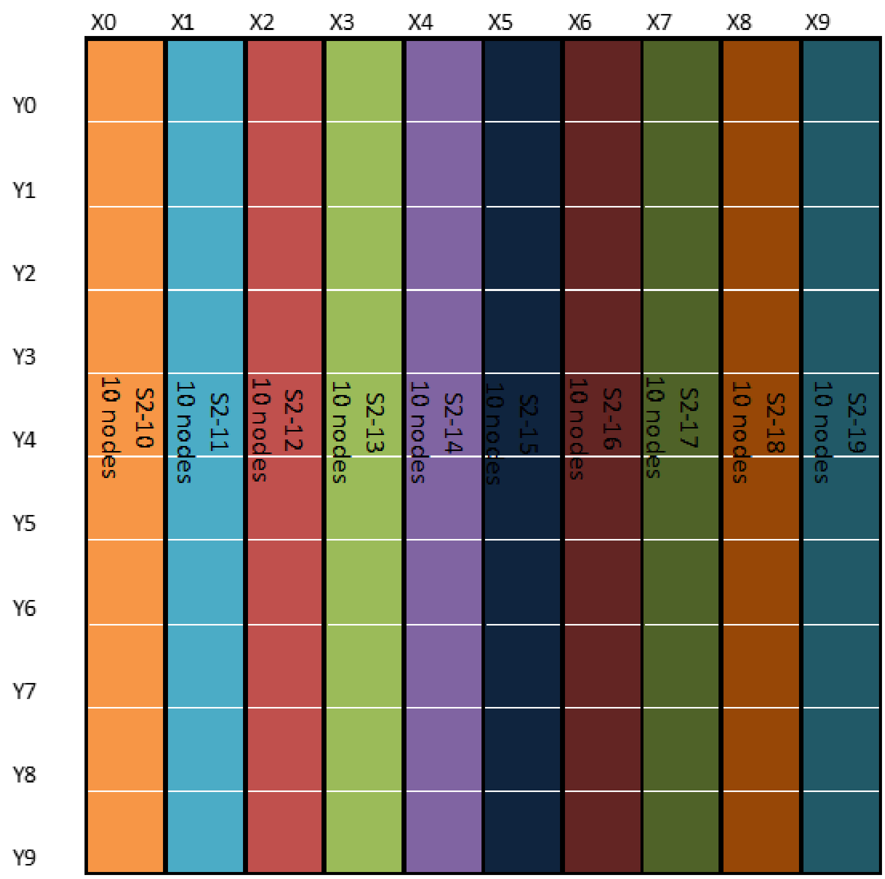


Figure 5-7 Optimized Bursting Illustration

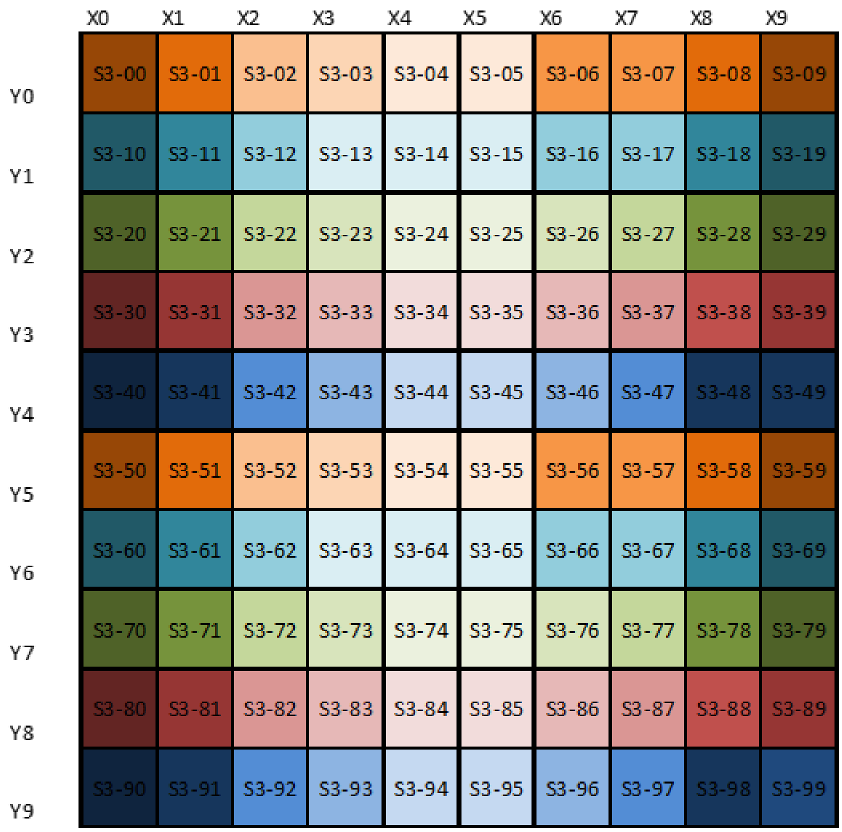


Figure 5-8 Optimized Bursting Illustration-Stage 0

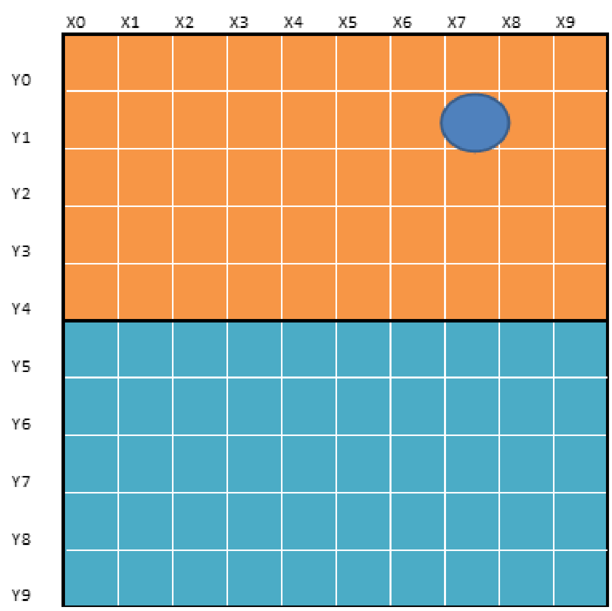


Figure 5-9 Optimized Bursting Illustration-Stage 1

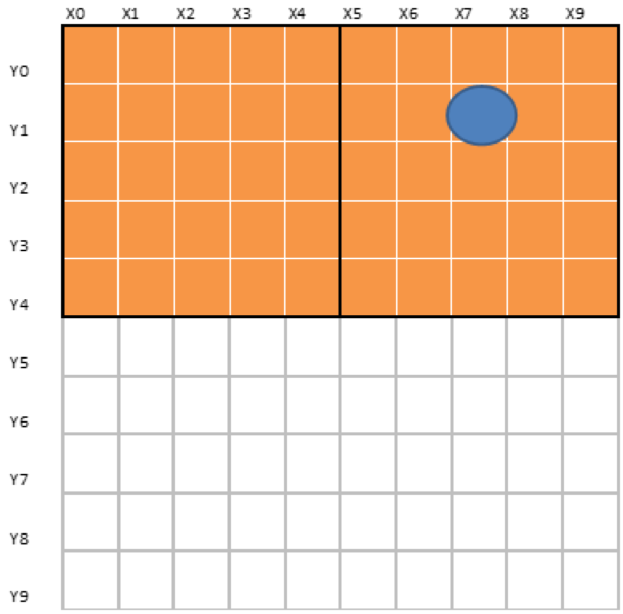


Figure 5-10 Optimized Bursting Illustration-Stage 2

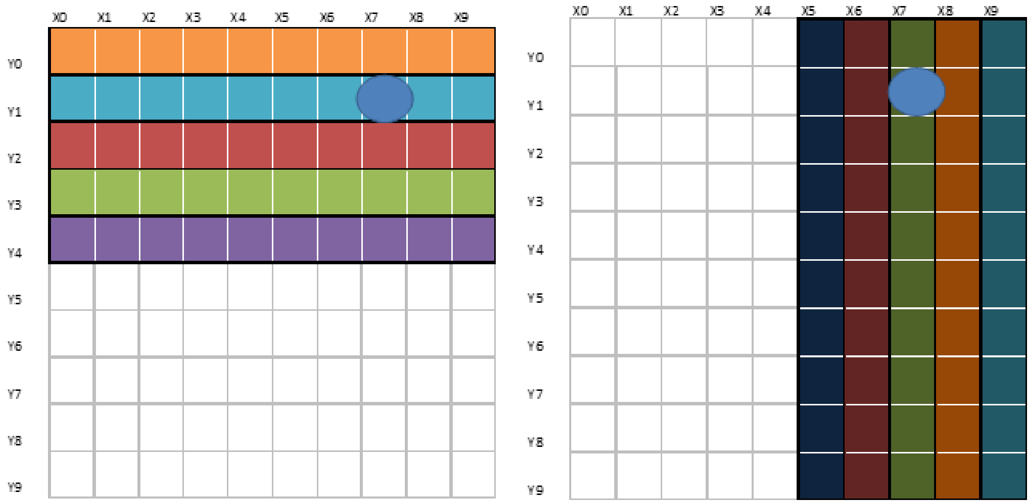
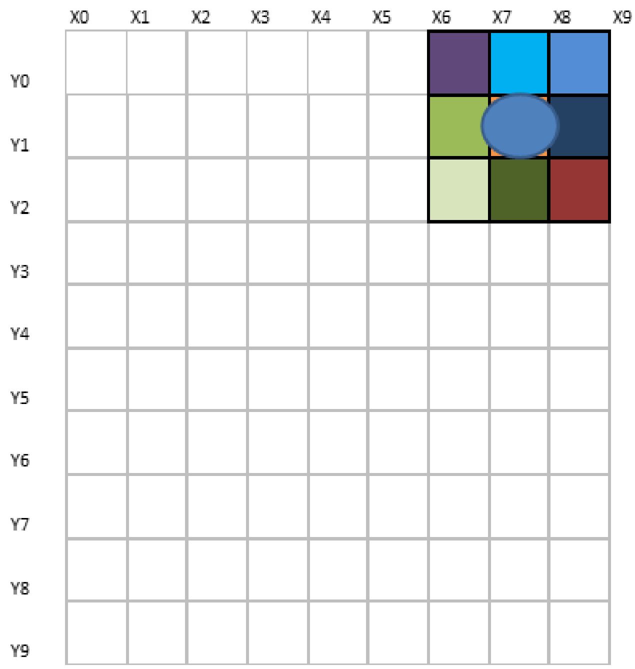
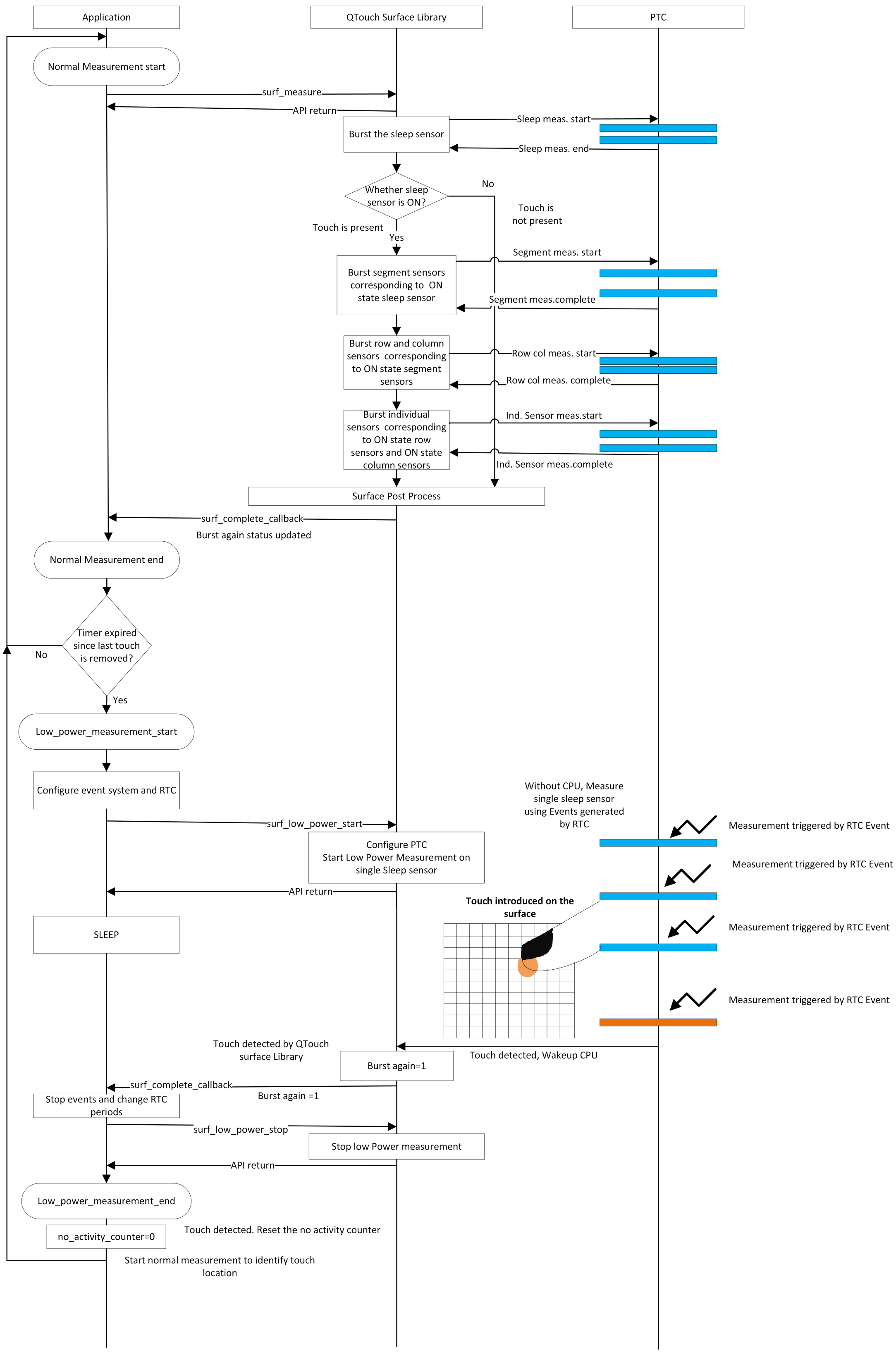


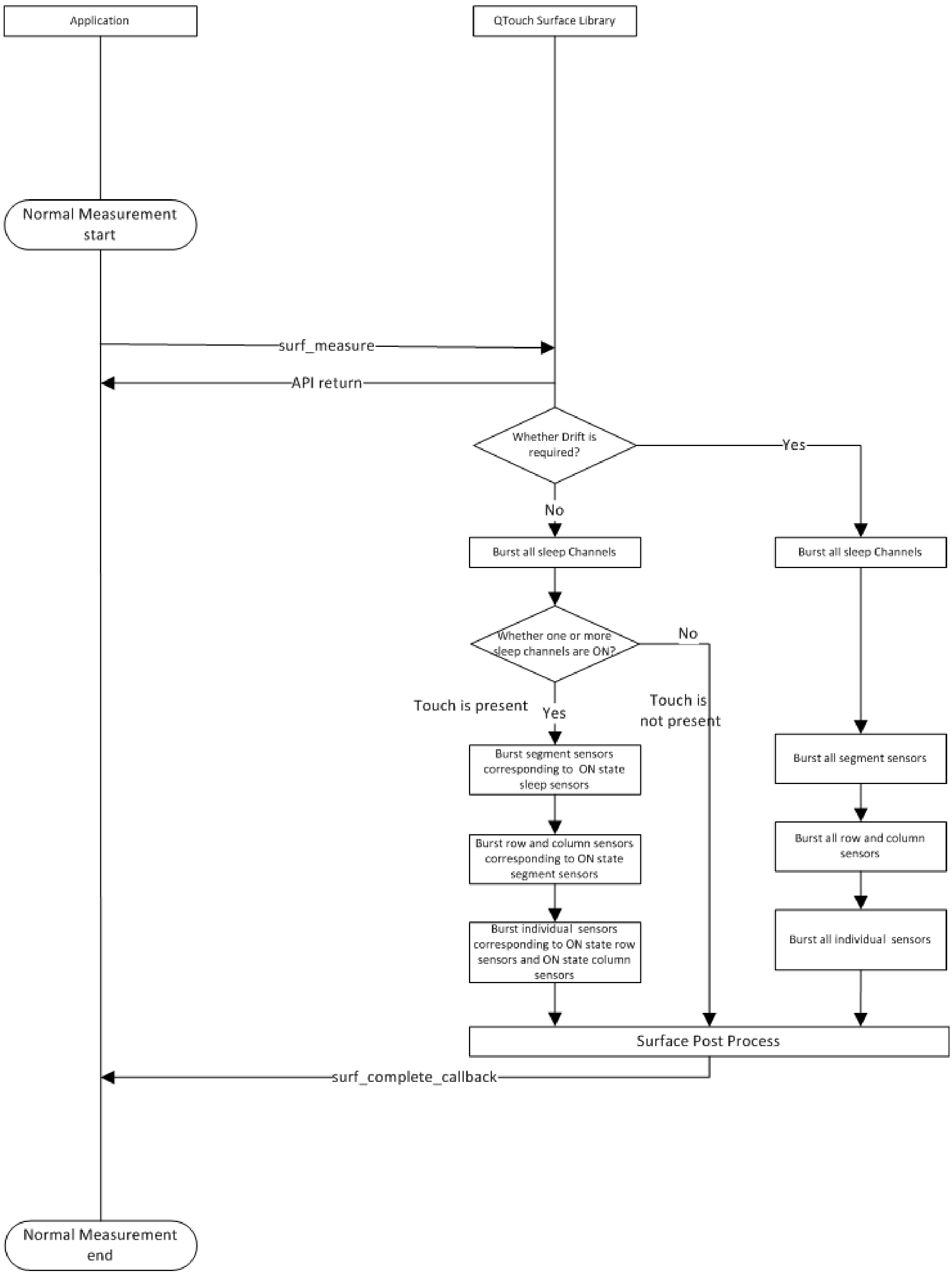
Figure 5-11 Optimized Bursting Illustration-Stage 3



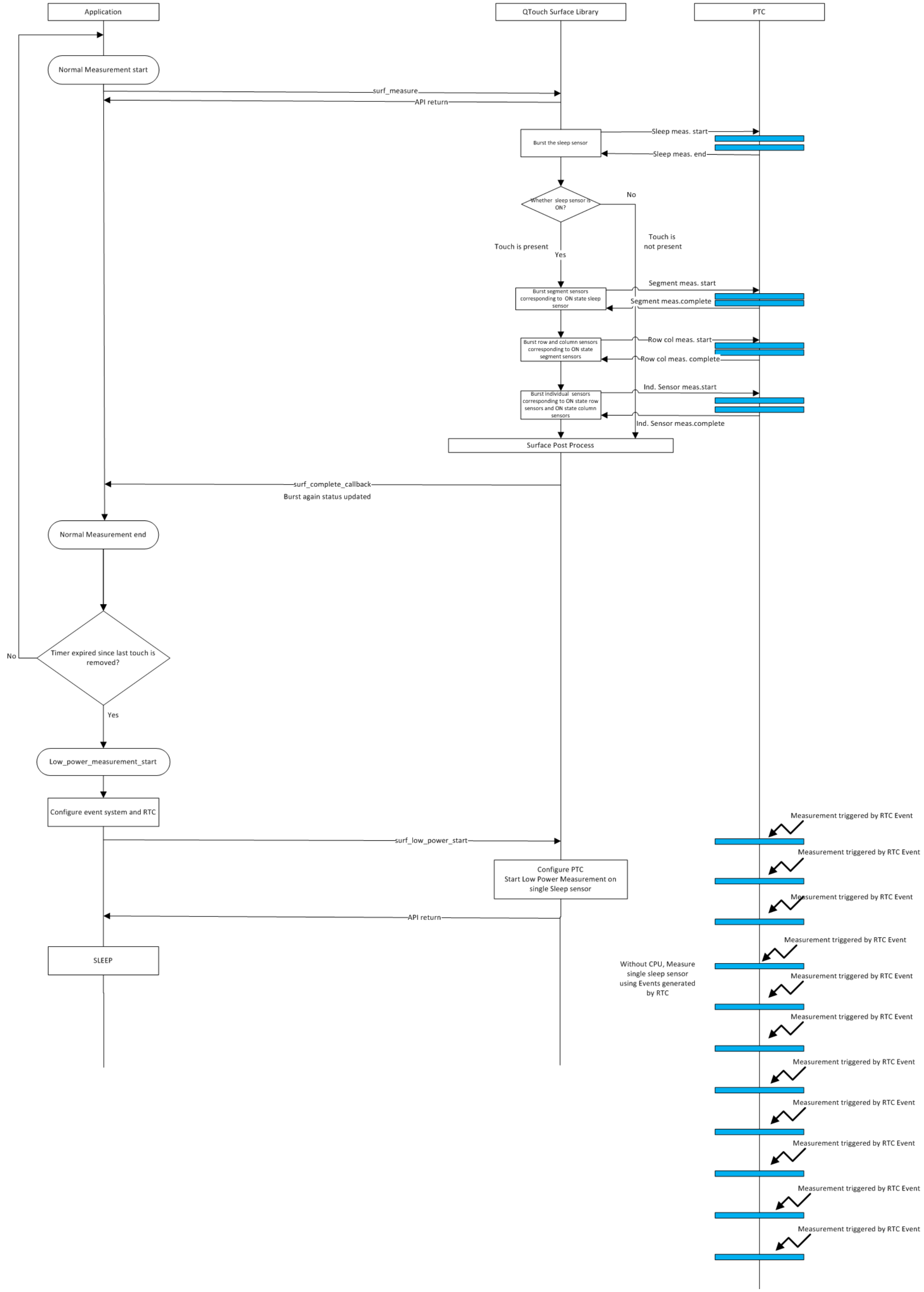
5.4.5. Low Power Measurement Mode



5.4.6. Normal Measurement Mode



5.4.7. Low Power Measurement Mode (No Touch)



5.4.8. Low Power Measurement Mode (Drifting)

Figure 5-12 Low Power Measurement Mode (Drifting)

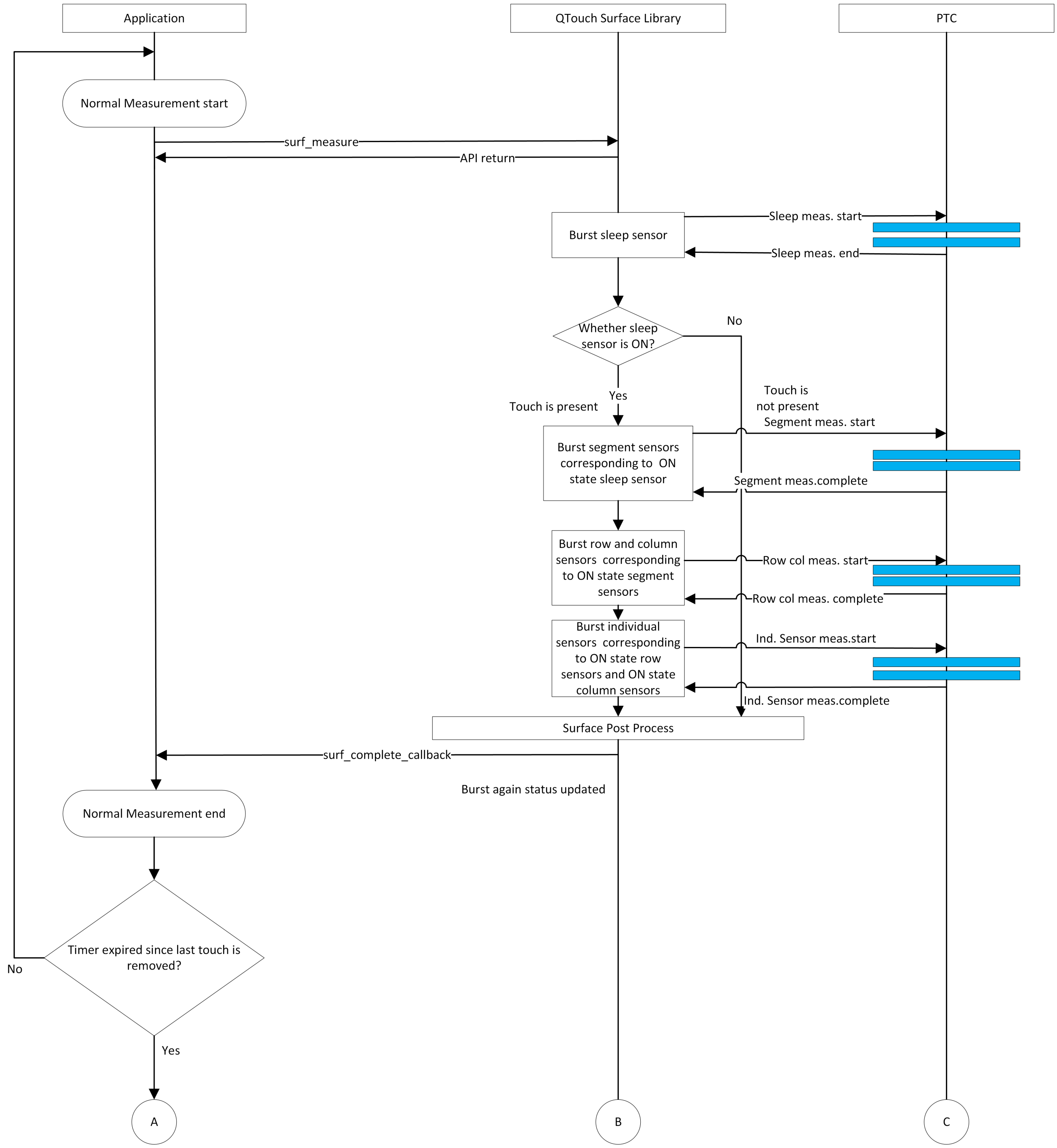
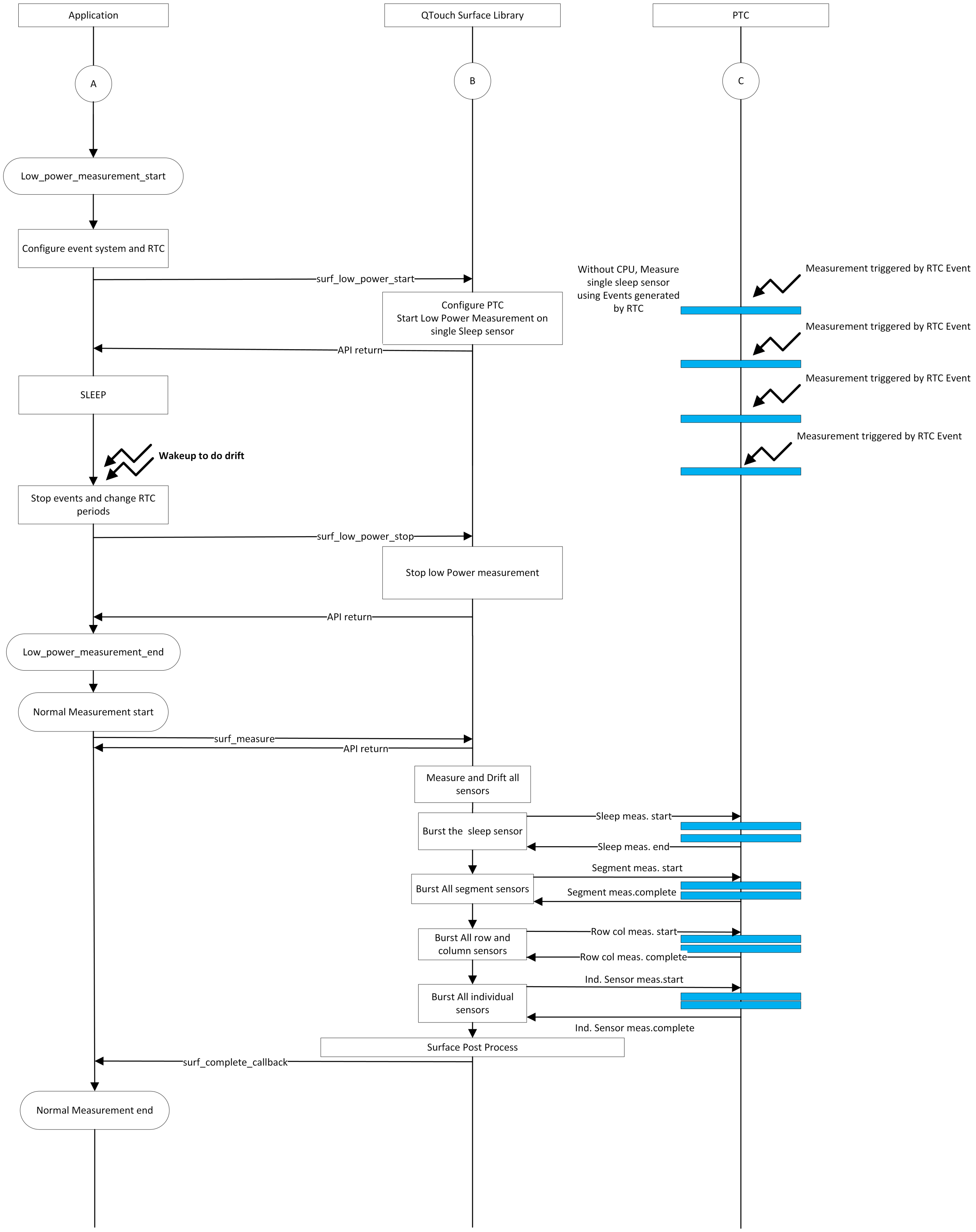


Figure 5-13 Low Power Measurement Mode (Drifting)... Continued



5.5. Surface Related Parameter Configuration

Feature	Parameter	Range	Typical	Recommendation
Maximum Touches	DEF_SURF_MAX_TCH	1 – 2	1	Setting the maximum touches to 2 will increase the response time and power consumption
				Set to 1 if the use-case for multiple touches can be ignored
Lump Node Size	DEF_MAX_LUMP_SIZE	9 – 100	30	Increasing Lump Node Size will allow fewer Sleep Channels and decrease power consumption
				If the parameter is too large, sensor CC value will be saturated
				Optimal Lump Node Size needs to be configured manually
Sleep Channels	DEF_SURF_NUM_SLEEP_CHANNELS	1 – 32	4	Fewer sleep channels will decrease power consumption
				This parameter is not configurable and will be set automatically based on the Lump Node Size
Sleep Channels Detect Threshold	SURF_SLEEP_CHANNELS_DT	1-255	80	Detect Threshold for the sleep channels
Segment Channels Detect Threshold	SURF_SEGMENTS_DT	1-255	50	Detect Threshold for the Segment channels
Row Line Channels Detect Threshold	SURF_ROW_LINE_DT	1-255	40	Detect Threshold for the Row Line channels
Column Line Channels Detect Threshold	SURF_COL_LINE_DT	1-255	40	Detect Threshold for the Column Line channels
Individual sensor Detect Threshold	SURF_IND_SEN_DT	1-255	20	Detect Threshold for Individual sensors

Feature	Parameter	Range	Typical	Recommendation
Detect Integrity	DEF_SURF_DI	0 – 255	0	Higher values will provide better immunity against noise spikes Set this to the lowest value that provides sufficient spike filtering
Max On Duration	DEF_SURF_MAX_ON_DURATION	0 – 255	0	This should be kept at the typical value This can be changed if special circumstances need to be addressed
Towards Touch Drift Period	DEF_SURF_TCH_DRIFT_PERIOD	1 – 127	20	Reference value will be adjusted by one count for every drift period mentioned here 1 unit mean 200 ms drift rate.
Away From Touch Drift Period	DEF_SURF_ATCH_DRIFT_PERIOD	1 – 127	5	Reference value will be adjusted by one count for every drift period mentioned here. 1 unit mean 200 ms drift rate.
Measurement time (in ms)	SURF_ACTIVE_TCH_SCAN_RATE_MS	0-1000	20	Scan Rate when touch is present
Off measurement period (in ms)	SURF_NO_TCH_SCAN_RATE_MS	0-10000	100	Scan Rate when there is no touch
Low Power trigger wait time (in ms)	NO_ACTIVITY_TRIGGER_TIME	0-50000	5000	Delay before low power mode is triggered when touch is present on the surface. This is Applicable only when Low Power Mode is enabled
DPI X Axis	DEF_SURF_DPI_X	50-160	160	Dots per inch on X axis Number of position per inch displayed along X Axis

Feature	Parameter	Range	Typical	Recommendation
DPI Y Axis	DEF_SURF_SIZE_IN_Y	50-160	160	Dots per inch on Y axis
				Number of position per inch displayed along Y Axis
Surface Size in X Axis(in mm)	DEF_SURF_SIZE_IN_X	20-120	50	Total Surface Size along X Axis
Surface Size in Y Axis(in mm)	DEF_SURF_SIZE_IN_Y	20-120	50	Total Surface Size along Y Axis
Surface Node Size in X Axis(in mm)	DEF_SURF_SENSOR_SIZE_IN_X	04-Oct	5	Surface Node Size along X Axis. This will be calculated internally.
Surface Node Size in Y Axis(in mm)	DEF_SURF_SENSOR_SIZE_IN_Y	04-Oct	5	Surface Node Size along Y Axis. This will be calculated internally by the project Builder.This is based on total surface size along X axis.
Total Position Resolution along the X Axis	DEF_SURF_TOT_RES_X	0-319	50	Number of Positions displayed along X axis for the whole surface. This will be calculated internally by the project Builder. This is based on total surface size along X axis.
Total Position Resolution along the Y Axis	DEF_SURF_TOT_RES_Y	0-319	50	Number of Positions displayed along Y axis for the whole surface. This will be calculated internally.

Note: If the number of sleep channels is 1, then low power mode is enabled by default in QT2 example project. Sensor board has to be touched in order to connect to the QTouch Analyser.

5.6. Surface Data Analysis on QTouch Analyzer

QTouch Surface Analyzer tool allows the users to view QTouch Surface data (i.e. signal, reference, delta and X, Y position co-ordinates of touch event).

QTouch Surface Analyzer is an effective tool for real-time visualization, analysis and debugging of Surface data, enabling easier optimization of Surface designs. To get appropriate sensitivity to a finger touch, QTouch Surface library parameters need to be tuned properly. Using QTouch Surface Analyzer, users must analyze sensor node signal and touch delta to appropriately tune sensor parameters such as Gain, Filter level, Detect threshold and global parameters such as drift rates.

QTouch Surface Analyzer for Surface supports the following views:

- Touch Reporter
- Graph View

- Surface View
- Tabular View

More details about these views are discussed in the following section.

5.6.1. Functionality Views

There are different views available in the QTouch Analyzer tool for in-depth analysis of various functionality provided by QTouch Surface library.

5.6.1.1. Touch Reporter View

The touch reporter tracks X, Y position co-ordinates of a moving finger on the touch Surface and shows patterns drawn over the touch Surface. It also represents the X, Y position co-ordinates and touch area in tabular format.

Note: The touch area is provided in terms of number of active sensor nodes.

The touch reporter view appears on left side of the QTouch Analyzer. The image below shows touch reporter view in QTouch Analyzer.

Figure 5-14 Touch Reporter View - Without Touch

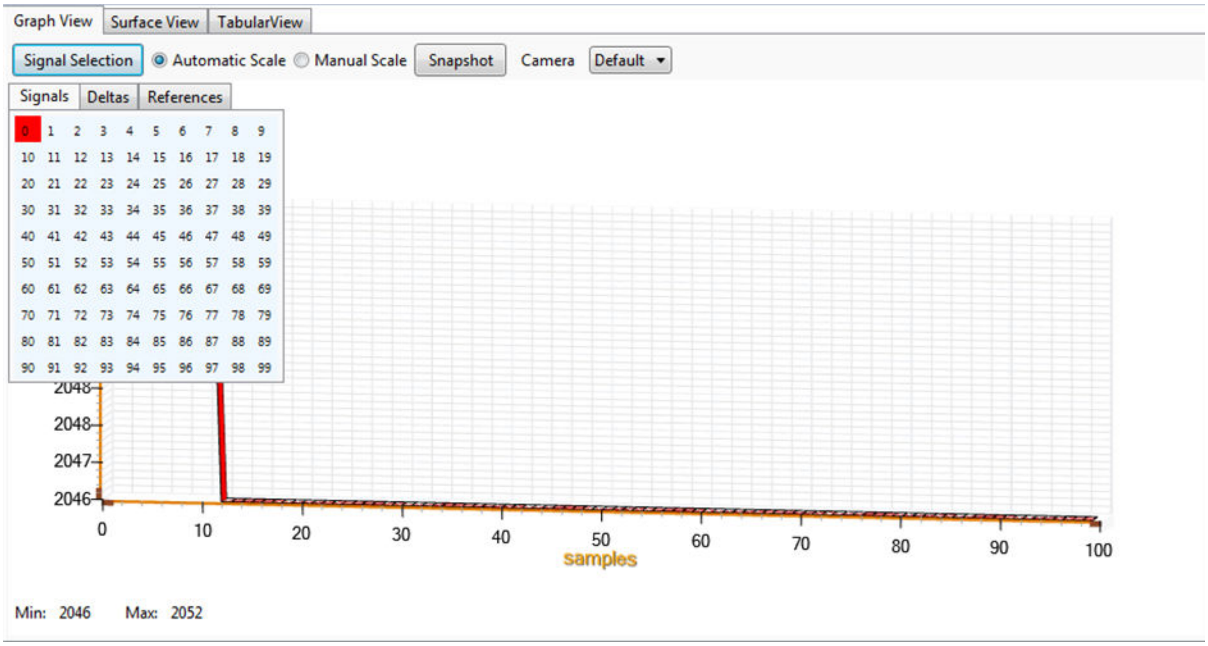


Figure 5-15 Touch Reporter View - With Single Touch

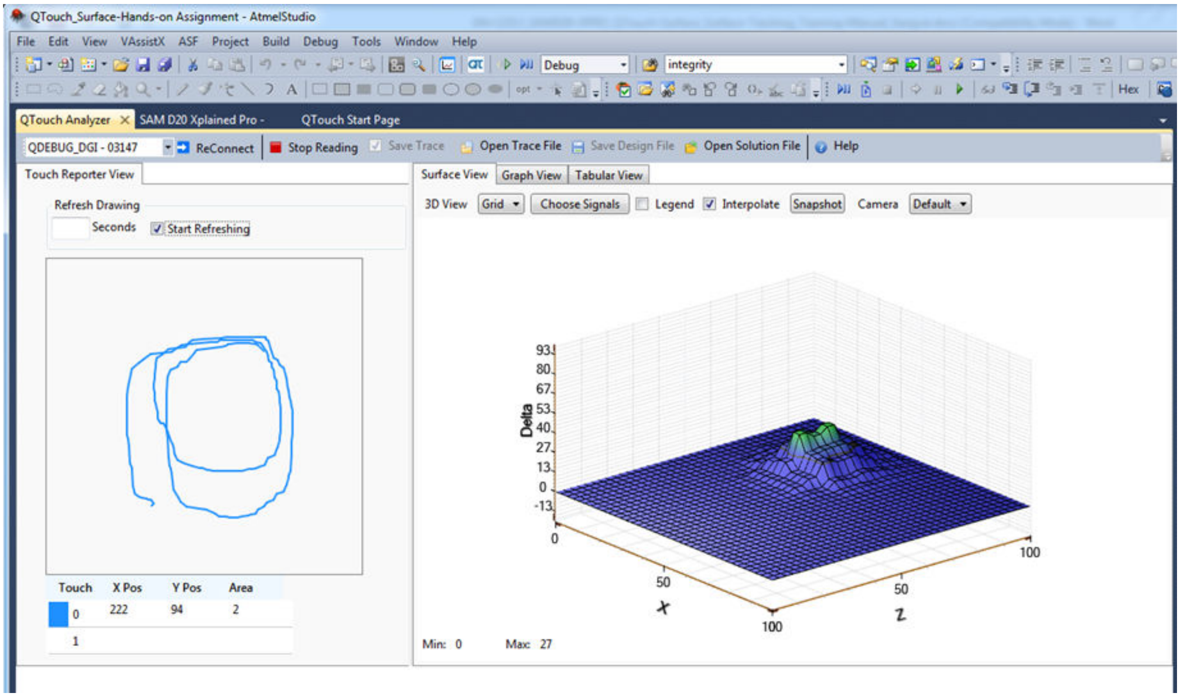
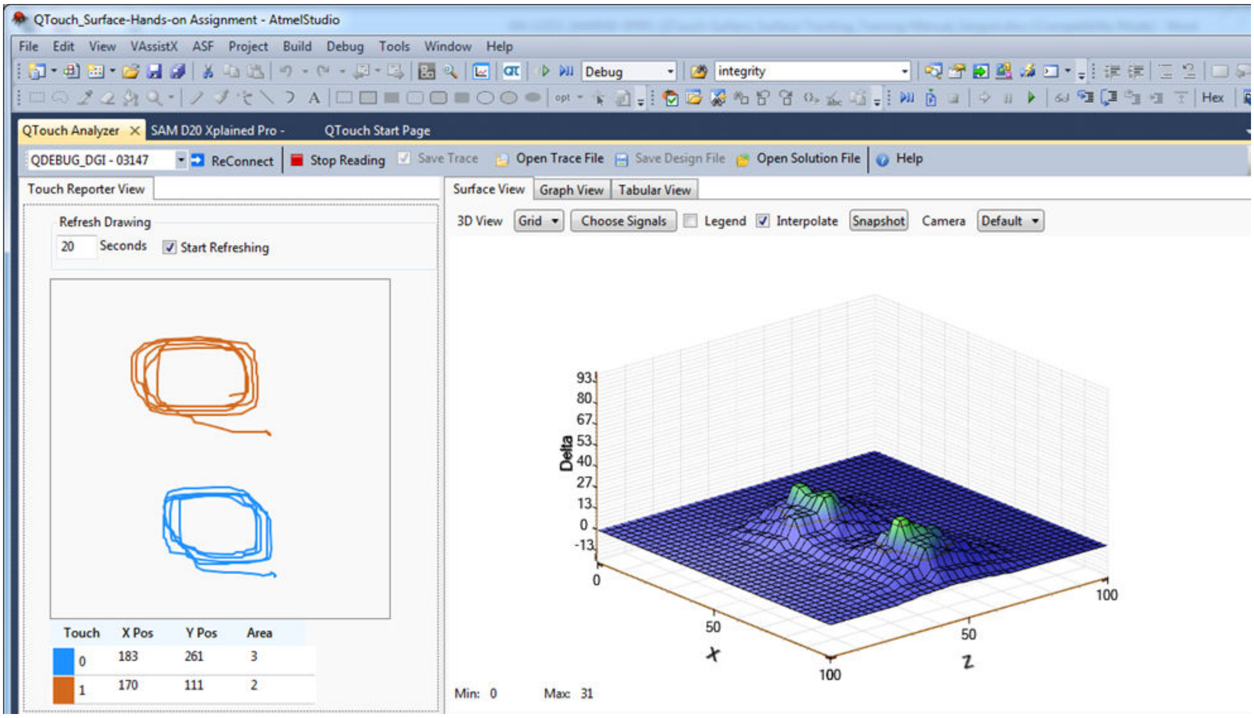


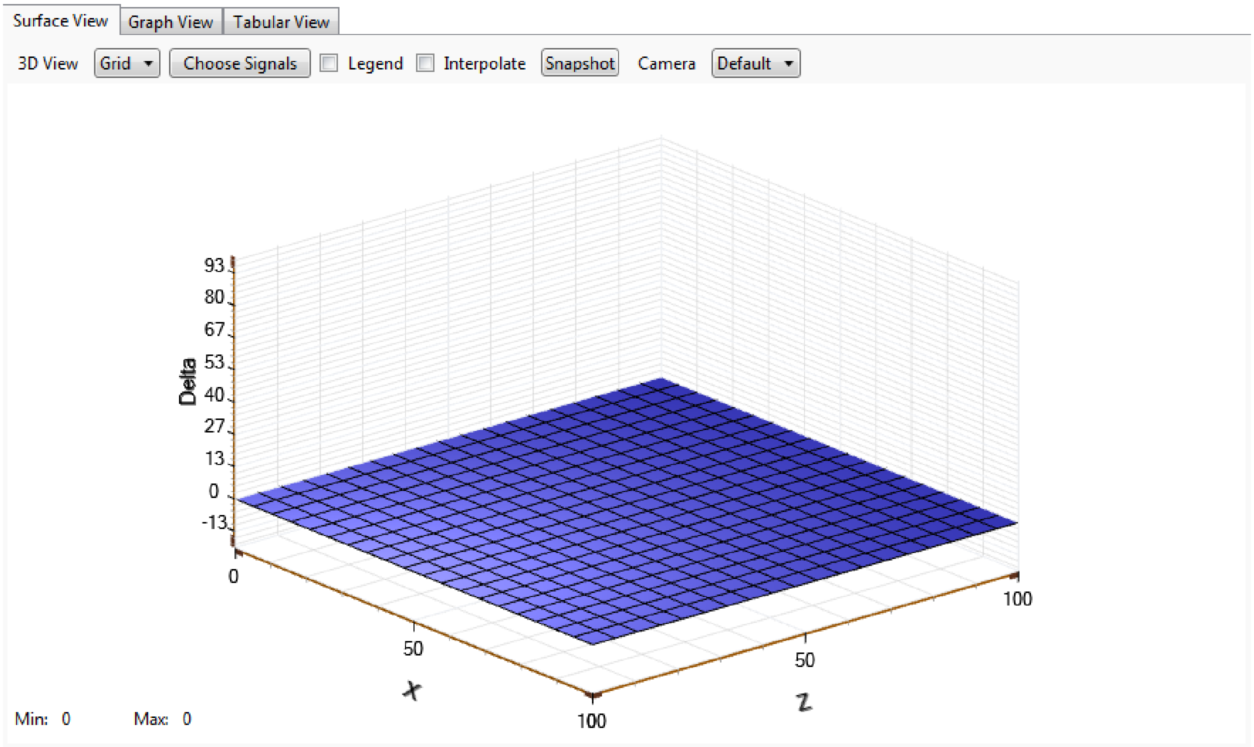
Figure 5-16 Touch Reporter View - With 2 Touches



5.6.1.2. Surface View

The surface view presents touch delta of surface sensor nodes in 3-Dimensional (3D) format. The image below shows surface view in the QTouch Analyzer.

Figure 5-17 Surface View



In the surface view, X-axis, Z-axis shows row sensor nodes and column sensor nodes of surface respectively. Y-axis represent delta value of the touched sensor nodes.

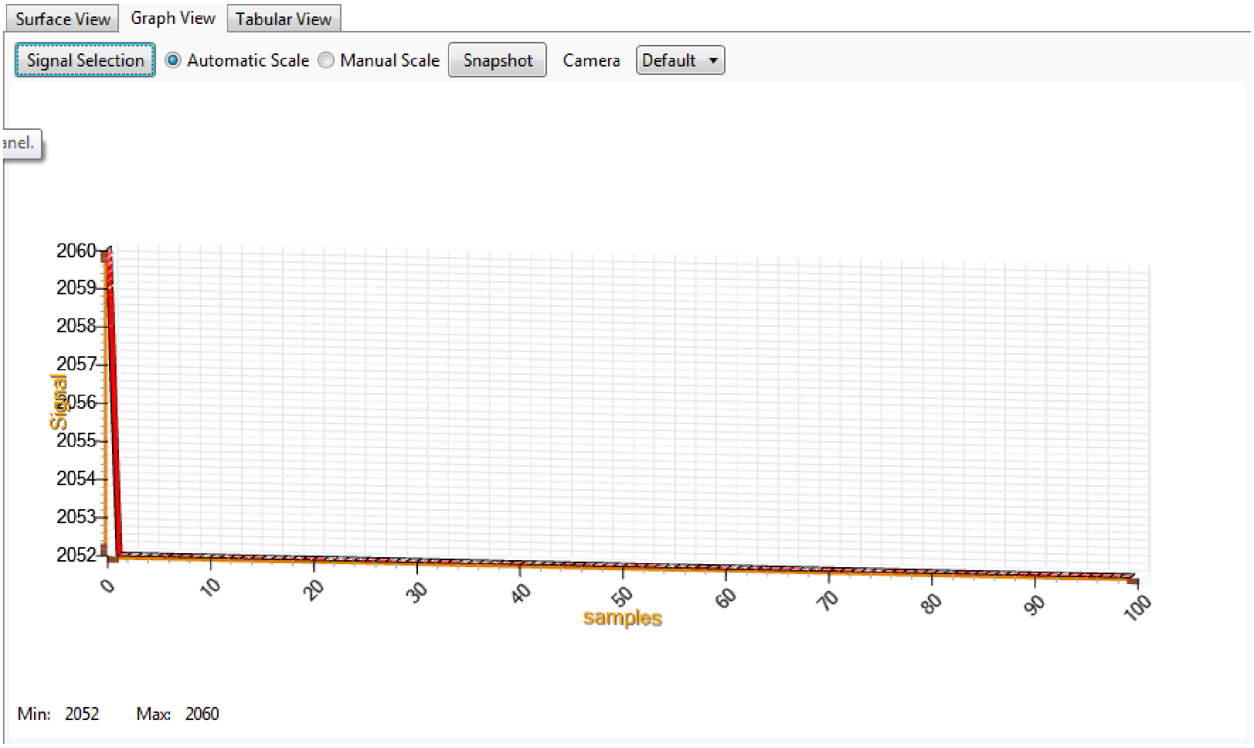
Surface view can plot the sensor delta values either in 3D grid view or in 3D mesh view. User can choose the preferable option from “3D View” drop down menu.

Surface view also allows the user to display a set of selected sensor node delta values alone.

5.6.1.3. Graph View

The graph view allows the user to view surface sensor node’s signal, reference and delta values in graphical format. The image below shows graph view in the QTouch Analyzer tool.

Figure 5-18 Graph View - Signal Selection



In the graph view, X-axis shows time stamp of the captured samples and Y-axis shows sensors data (i.e. signal, reference and delta value). The scale on the Y-axis can be set either automatically or manually by selecting the “Automatic Scale” and “Manual Scale” radio buttons respectively.

Note: Use the tabs to switch between signal, delta, and reference selection. Once the sensor nodes are selected for display, click on “Signal Selection”. The “Signal Selection” window disappears.

5.6.1.4. Tablular View

The tabular view represents surface sensor node delta values in tabular format. The image below shows tabular view in the QTouch Analyzer.The following figure provides the deltas for the sleep channels. Similarly, to view the deltas of the segments, Row-Column, the appropriate Tab has to be clicked.

Figure 5-19 Tabular View - Sleep Channel Deltas

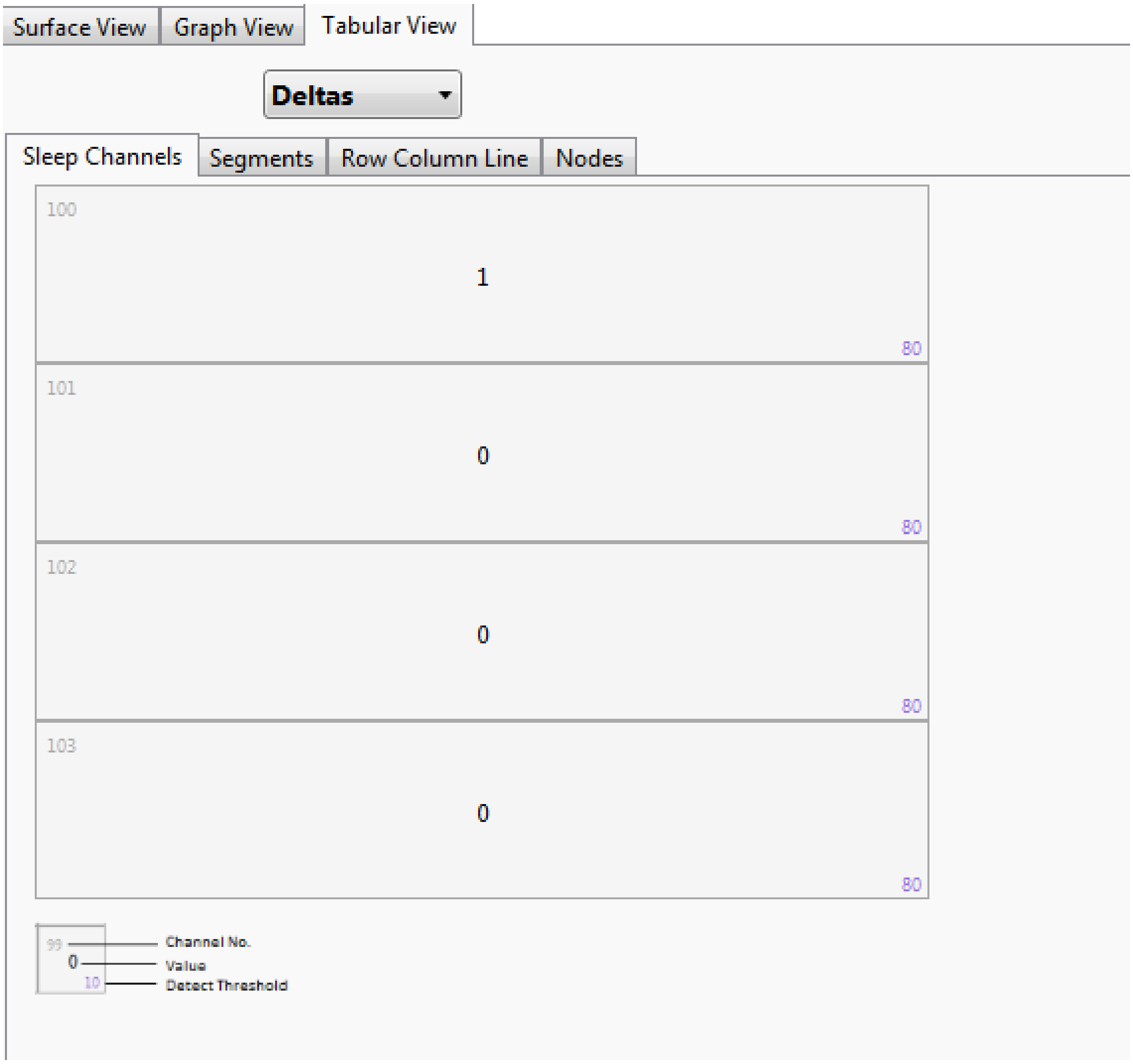
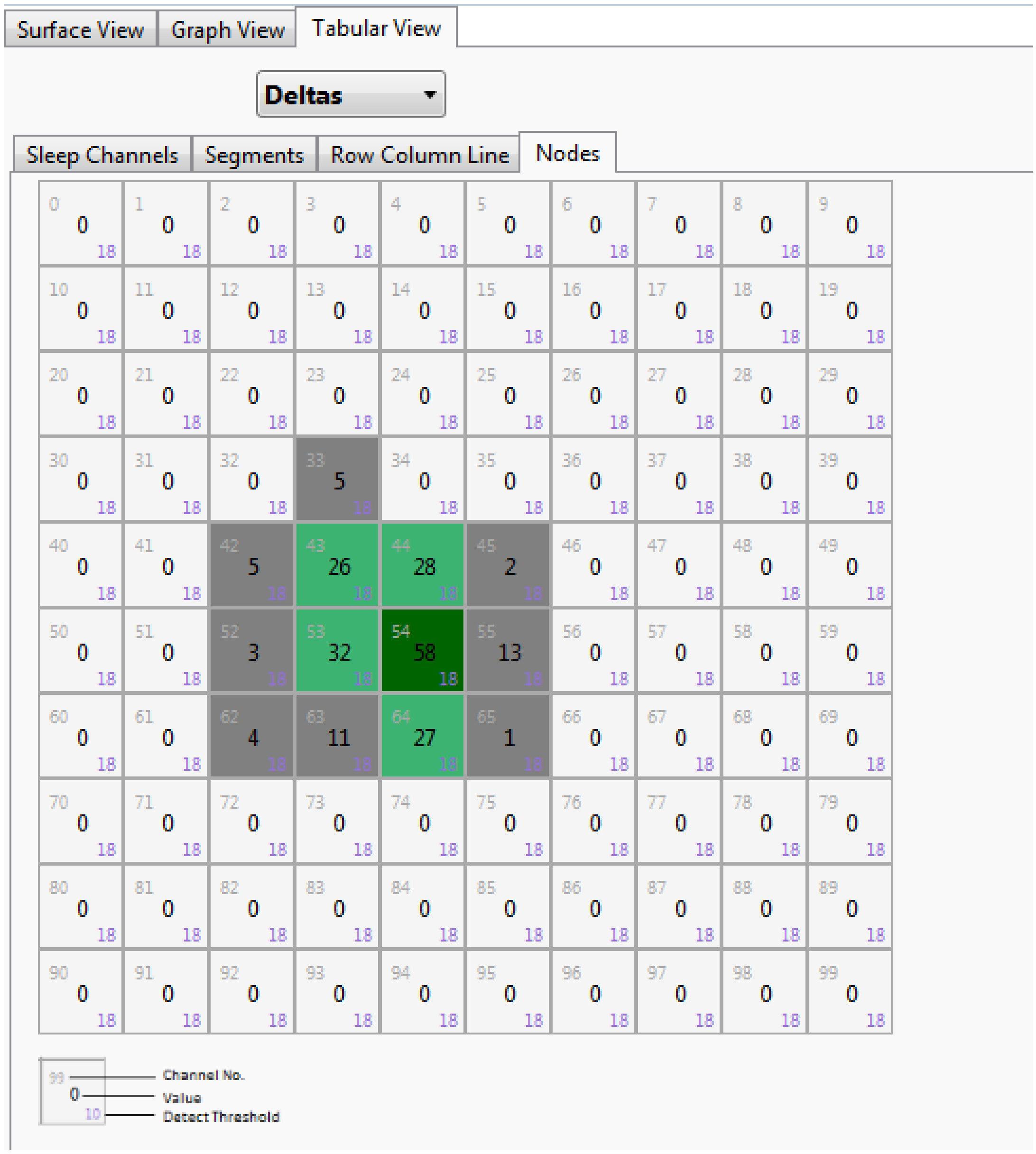


Figure 5-20 Tabular View - Individual Node deltas



The sensor node which has the highest peak delta will be shown in dark green colour. The sensor nodes whose delta is above the detect threshold are highlighted with light green color. And the nodes which has the delta below Detect threshold are highlighted with light grey colour.

6. Surface Library API

QTouch Surface Library API for PTC can be used for touch sensor pin configuration, acquisition parameter setting as well as periodic sensor data capture and status update operations. The QTouch Library in turn interfaces with the PTC module to perform the necessary action. The PTC module interfaces with the external capacitive touch sensors and is capable of performing mutual capacitance method measurements.

6.1. Macro

6.1.1. Touch State Configuration

`SURF_GET_TOUCH_STATE (TOUCH_ID)`

To get the state of the particular touch id(whether detect or not) that corresponds to the touch id specified using the TOUCH_ID.

The macro returns either 0 or 1.

If the bit value is 0, the touch is not in detect If the bit value is 1, the touch is in detect

```
#define SURF_GET_TOUCH_STATE(TOUCH_ID) surf_status.surf_tch_state & (1 <<
(TOUCH_ID % 8))
```

6.1.2. Position Related Configurations

`SURF_GET_X_POSITION (TOUCH_ID)`

To get the X position for the particular touch id.

```
#define SURF_GET_X_POSITION(TOUCH_ID)
surf_status.ptr_surf_tch_status[TOUCH_ID].x_position SURF_GET_Y_POSITION
(TOUCH_ID)
```

To get the Y position for the particular touch id.

```
#define SURF_GET_Y_POSITION(TOUCH_ID)
surf_status.ptr_surf_tch_status[TOUCH_ID].y_position
```

6.1.3. Area Related Configurations

`SURF_GET_TOUCH_AREA (TOUCH_ID)`

To get the area for the particular touch id.

```
#define SURF_GET_TOUCH_AREA(TOUCH_ID)
surf_status.ptr_surf_tch_status[TOUCH_ID].area
```

6.1.4. Surface Size Configurations

`DEF_SURF_SIZE_IN_X`

This macro will be filled by composer.

The user has to provide the total surface size along the X Axis in mm.Based on this, composer will calculate the Total Resolution on X axis and the Sensor (Node) size in X-axis.

```
#define DEF_SURF_SIZE_IN_X 50
```

`DEF_SURF_SIZE_IN_Y`

This macro will be filled by composer.

The user has to provide the total surface size along the Y Axis in mm. Based on this, composer will calculate the Total Resolution on Y axis and the Sensor (Node) size in Y-axis.

```
#define DEF_SURF_SIZE_IN_Y 50
```

6.2. Typedef

Field	Unit	Description
`threshold_t`	`uint8_t`	An unsigned 8-bit number setting a sensor detection threshold
`sensor_id_t`	`uint8_t`	Sensor number type

6.3. Enumeration

6.3.1. PTC Series Resistor Setting (tag_rsel_val_t)

For mutual capacitance mode, this series resistor is switched internally on the Y-pin.

Example:

- `RSEL_VAL_0` sets internal series resistor to 0 Ohms
- `RSEL_VAL_20` sets internal series resistor to 20 Kohms
- `RSEL_VAL_50` sets internal series resistor to 50 Kohms
- `RSEL_VAL_100` sets internal series resistor to 100 Kohms

Data Fields

- `PRSC_DIV_SEL_1`
- `PRSC_DIV_SEL_2`
- `PRSC_DIV_SEL_4`
- `PRSC_DIV_SEL_8`

6.3.2. PTC Clock Pre-scalar Setting

Refer `touch_configure_ptc_clock()` API in `surface.c`

Example: If generic clock input to PTC = 4 MHz,

- `PRSC_DIV_SEL_1` sets PTC Clock to 4 MHz
- `PRSC_DIV_SEL_2` sets PTC Clock to 2 MHz
- `PRSC_DIV_SEL_4` sets PTC Clock to 1 MHz
- `PRSC_DIV_SEL_8` sets PTC Clock to 500 KHz

Data Fields

- `PRSC_DIV_SEL_1`
- `PRSC_DIV_SEL_2`
- `PRSC_DIV_SEL_4`
- `PRSC_DIV_SEL_8`

6.3.3. Sensor Hysteresis Setting (tag_hysteresis_t)

A sensor detection hysteresis value.

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

`HYST_x` = hysteresis value is x% of detection threshold value (rounded down).

Note: A minimum value of 2 is used.

6.3.4. Sensor Recalibration Threshold (tag_recal_threshold_t)

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

`RECAL_x` = recalibration threshold is x% of detection threshold value (rounded down).
Note: A minimum value of 4 is used.

Example: If detection threshold = 40,

- `RECAL_100` = 40 (100% of 40)
- `RECAL_50` = 20 (50% of 40)
- `RECAL_25` = 10 (25% of 40)
- `RECAL_12_5` = 5 (12.5% of 40)
- `RECAL_6_25` = 4 (6.25% of 40 = 2, but value is hard limited to 4)

Data Fields

- `RECAL_100`
- `RECAL_50`
- `RECAL_25`
- `RECAL_12_5`
- `RECAL_6_25`
- `MAX_RECAL` Max value of enum type for testing

6.3.5. PTC Acquisition Frequency Mode Setting (tag_freq_mode_sel_t)

The frequency mode setting option enables the PTC acquisition to be configured for the following modes.

- Frequency hopping and spread spectrum disabled.
- Frequency hopping enabled with median filter.
- Frequency spread spectrum enabled without median filter.

Range:

- `FREQ_MODE_NONE` (no frequency hopping & spread spectrum) to `FREQ_MODE_SPREAD` (spread spectrum without median filter)

Data Fields

- `FREQ_MODE_NONE` 0u. The PTC runs at constant speed. This mode is suitable for best power consumption.
- `FREQ_MODE_HOP` 1u. The PTC runs with three frequencies for subsequent measurement cycles and applies the median filter among the resultant values.
- `FREQ_MODE_SPREAD` 2u. The PTC runs with spread spectrum enabled for jittered delay based acquisition.

6.3.6. Gain Setting

Gain per touch channel.

Gain is applied on a per-channel basis to allow a scaling-up of the touch sensitivity on contact. Note: delta on touch contact, not the resting signal which is measured on each sensor.

Range: `GAIN_1` (no scaling) to `GAIN_32` (scale-up by 32)

Data Fields

- `GAIN_1`
- `GAIN_2`
- `GAIN_4`
- `GAIN_8`
- `GAIN_16`
- `GAIN_32`

6.3.7. Library Error Code (tag_touch_ret_t)

Touch Library error codes.

Data Fields

- `TOUCH_SUCCESS` Successful completion of touch operation.
- `TOUCH_ACQ_INCOMPLETE` Library is busy with pending previous touch measurement.
- `TOUCH_INVALID_INPUT_PARAM` Invalid input parameter.
- `TOUCH_INVALID_LIB_STATE` Operation not allowed in the current touch library state.
- `TOUCH_INVALID_MUTLCAP_CONFIG_PARAM` Invalid mutual capacitance configuration input parameter.
- `TOUCH_INVALID_RECAL_THRESHOLD` Invalid recalibration threshold input value.
- `TOUCH_INVALID_CHANNEL_NUM` Channel number parameter exceeded total number of channels configured.
- `TOUCH_INVALID_SENSOR_TYPE` Invalid sensor type. Sensor type must NOT be `SENSOR_TYPE_UNASSIGNED`.
- `TOUCH_INVALID_SENSOR_ID` Invalid sensor number parameter.

6.3.8. Surface Return (surf_ret_t)

This enumeration describes the types of return from the surface library APIs.

Data Fields

- `SURF_SUCCESS` Successful execution of surface library API
- `SURF_INVALID_INPUT_PARAM` Parameter passed is Invalid
- `SURF_INVALID_LIB_STATE` Surface Library state is Invalid
- `SURF_INVALID_CHANNEL_NUM` Channel Number passed is Invalid
- `SURF_INVALID_SENSOR_ID` Sensor Id passed is Invalid

6.3.9. Filter Level Setting

Touch library `FILTER LEVEL` setting. The filter level setting controls the number of samples acquired to resolve each acquisition. A higher filter level setting provides improved signal to noise ratio under noisy conditions, while increasing the total time for measurement which results in increased power consumption. Refer `filter_level_t in touch_api_samd.h`

Range: `FILTER_LEVEL_1` (one sample) to `FILTER_LEVEL_64` (64 samples).

Data Fields

- FILTER_LEVEL_1
- FILTER_LEVEL_2
- FILTER_LEVEL_3
- FILTER_LEVEL_8
- FILTER_LEVEL_16
- FILTER_LEVEL_32
- FILTER_LEVEL_64

6.3.10. Touch Channel (tag_channel_t)

Sensor start and end channel type of a Sensor. Channel number starts with value 0.

Data Fields

CHANNEL_0 to CHANNEL_255

6.3.11. PTC Clock Pre-scalar Setting

Refer touch_configure_ptc_clock() API in surface.c

Example: If generic clock input to PTC = 4 MHz,

- PRSC_DIV_SEL_1 sets PTC Clock to 4 MHz
- PRSC_DIV_SEL_2 sets PTC Clock to 2 MHz
- PRSC_DIV_SEL_4 sets PTC Clock to 1 MHz
- PRSC_DIV_SEL_8 sets PTC Clock to 500 KHz

Data Fields

- PRSC_DIV_SEL_1
- PRSC_DIV_SEL_2
- PRSC_DIV_SEL_4
- PRSC_DIV_SEL_8

6.4. Datastructures

6.4.1. Mutual Capacitance Sensor Configuration (touch_mutlcap_config_t)

Touch Library mutual capacitance configuration input type.

Field	Unit	Description
num_channels	uint16_t	Number of channels
num_sensors	uint16_t	Number of sensors
num_rotors_and_sliders	uint8_t	Not Applicable
global_param	touch_global_param_t	Global parameter structure
touch_mutlcap_acq_param	touch_mutlcap_acq_param_t	Sensor acquisition parameter info
*p_data_blk	uint8_t	Pointer to data block buffer

Field	Unit	Description
*buffer_size	uint16_t	Size of data block buffer
*p_mutlcap_xy_nodes	const uint16_t	Pointer to xy nodes
mutl_quick_reburst_enable	uint8_t	Quick re-burst enable
touch_filter_data_t *p_filter_data	void(* filter_callback)	Mutual capacitance filter callback
enable_freq_auto_tune	uint8_t	Frequency auto tune enable
enable_noise_measurement	uint8_t	Noise measurement enable
nm_buffer_cnt	uint8_t	Memory allocation buffer

6.4.2. Touch Library Input Configuration (touch_config_t)

Touch Library Input Configuration Structure.

Field	Unit	Description
p_mutlcap_config	touch_mutlcap_config_t	Pointer to mutual capacitance configuration structure.
ptc_isr_lvl	uint8_t	PTC ISR priority level

6.4.3. Global Sensor Configuration Info (tag_touch_global_param_t)

The surface input configuration data are passed through this structure.

Field	Unit	Description
di	uint8_t	Not Applicable
atch_drift_rate	uint8_t	Not Applicable
tch_drift_rate	uint8_t	Not Applicable
max_on_duration	uint8_t	Not Applicable
drift_hold_time	uint8_t	Sensor drift hold time
atch_recal_delay	uint8_t	Sensor away from touch recalibration delay
cal_seq_1_count	uint8_t	Sensor calibration dummy burst count
cal_seq_2_count	uint8_t	Sensor calibration settling burst count
recal_threshold	recal_threshold_t	Sensor away from touch recalibration threshold
auto_tune_sig_stability_limit	uint16_t	Stability limit for frequency auto tune feature
auto_freq_tune_in_cnt	uint8_t	Frequency auto tune In counter

Field	Unit	Description
nm_sig_stability_limit	uint16_t	Stability limit for noise measurement
nm_noise_limit	uint8_t	Noise limit
nm_enable_sensor_lock_out	nm_sensor_lockout_t	Sensor lockout feature variable
nm_lockout_countdown	uint8_t	Lockout countdown for noise measurement

6.4.4. Sensor Acquisition Parameter (tag_touch_mutlcap_acq_param_t)

Pointer to gain per node.

Field	Unit	Description
*p_mutlcap_gain_per_node	const gain_t	Sensor acquisition parameter
touch_mutlcap_freq_mode	Freq_mode_sel_t	Setup acquisition frequency mode
mutlcap_ptc_prsc	prsc_div_sel_t	PTC clock prescaler value
mutlcap_resistor_value	rsel_val_t	PTC series resistor value
p_mutlcap_hop_freqs	*freq_hop_sel_t	Pointer to acquisition frequency settings
mutlcap_filter_level	filter_level_t	Filter level
mutlcap_auto_os	auto_os_t	Auto oversampling
mutlcap_ptc_prsc_cc_cal	prsc_div_sel_t	PTC clock prescale value during CC cal
mutlcap_resistor_value_cc_cal	rsel_val_t	PTC sense resistor value during CC cal

6.4.5. Sensor Info (tag_sensor_t)

This is the start of your concept.

Keyword	Type	Description
state	uint8_t	Sensor state (calibrate, on, off, filter-in, filter-out, disable, pos-recal)
General_counter	uint8_t	General purpose counter used for calibrating,drifting etc
Ndil_counter	uint8_t	Counter used for detect integration
Type_aks_pos_hyst	uint8_t	• Bits 7..6: sensor type: {00: key,01: rotor,10: slider,11: reserved} • Bits 5..3: AKS group (0..7): 0 = no AKS group • Bit 2 : positive recal flag bits 1..0: hysteresis
From_channel	uint8_t	Sensor from channel for keys: from channel = to channel. • Rotors: Top channel. • Sliders : Left most channel. **Note:** Only need to_channel for keys in rotors/sliders build

6.4.6. Measure Data Type (tag_touch_measure_data_t)

Touch library measure data type.

Data Fields

Field	Unit	Description
measurement_done_touch	volatile uint8_t	Flag set by touch_mutlcap_measure_complete_callback() function when a latest Touch status is available.
acq_status	touch_acq_status_t	Status of touch measurement.
num_channel_signals	uint16_t	Length of the measured signal values list.
*p_channel_signals	uint16_t	Pointer to measured signal values for each channel.
num_channel_references	uint16_t	Length of the measured reference values list.
* p_channel_references	uint16_t	Touch status of each sensor.
num_sensor_states	uint8_t	Number of sensor state bytes.
*p_sensor_states	uint8_t	Pointer to touch status of each sensor.
num_rotor_slider_values	uint8_t	Length of the rotor and slider position values list.
*p_rotor_slider_values	uint8_t	Pointer to rotor and slider position values.
num_sensors	uint16_t	Length of the sensors data list.
* p_cc_calibration_vals	uint16_t	Pointer to calibrated compensation values for a given sensor channel.
*p_sensors	sensor_t	Pointer to sensor data.
*p_sensor_noise_status	uint8_t	Pointer to noise status of the sensors.
*p_nm_ch_noise_val	uint16_t	Pointer to noise level value of each channel.
cc_calib_status_flag	uint8_t	Flag is set when cc_calibration gets over.

6.4.7. Surface Input Configuration(surf_config_t)

Touch library global parameter.

Field	Unit	Description
surf_hw_config	surf_hw_config_t	Hardware configuration parameters required for surface library are passed through this structure.
surf_pos_prop	surf_pos_prop_t	Configuration related to surface position is configured using this structure.
surf_global_param	surf_global_param_t	Configuration related to global parameters for the surface library is configured.
surf_tch_config	surf_tch_config_t	Properties related to touch for the surface library is configured through this structure.
ptr_surf_seg_size	surf_seg_size_t *	Pointer to the surface segments

Field	Unit	Description
ptr_surf_data_block	uint8_t *	Pointer to the surface data block allocated by the application
surf_data_block_size	uint16_t	Size of the surface data block

6.4.8. Surface Hardware Configuration (surf_hw_config_t)

The hardware configuration parameters required for surface library are passed through this structure.

Field	Unit	Description
num_total_channels	uint16_t	Total number of channels required for the surface library.
		Maximum value: 132
		Minimum value: 18
num_x_lines	uint8_t	Number of X lines.
		Maximum value: 10
		Minimum value: 3
num_y_lines	uint8_t	Number of Y lines.
		Maximum value: 10
		Minimum value: 3
num_sleep_channels	uint8_t	Total Number of Sleep channels.

6.4.9. Surface Position Property Configuration (surf_pos_prop_t)

The configuration related to surface position is configured using this structure.

Field	Unit	Description
dpi_x	uint16_t	Dots Per Inch along the X Axis.
dpi_y	uint16_t	Dots per Inch along the Y Axis.
total_resolution_x	uint16_t	Total resolution along the X axis. This field will be calculated by the composer by using the dpi and number of inches along that axis.
total_resolution_y	uint16_t	Total resolution along the Y axis. This field will be calculated by the composer by multiplying the dpi and number of inches along that axis.
size_of_sensor_in_mm_on_x	uint8_t	Size of sensor along x-axis in mm.
size_of_sensor_in_mm_on_y	uint8_t	Size of sensor along y-axis in mm.
pos_hysteresis	uint8_t	Position hysteresis for the surface library.

Both resolution on x and y axis will use the size of sensor configured and calculate the total number of inches present in the hardware board on both axis.

6.4.10. Surface Global Configuration (surf_global_param_t)

The configuration related to global parameters for the surface library is configured using this structure.

Field	Unit	Description
tch_drift_period	uint8_t	Towards Touch Drift Rate. Range : 1 to 127
atch_drift_period	uint8_t	Away from Touch Drift Rate. Range : 1 to 127
(*surf_complete_callback)	uint8_t (*) (uint16_t)	The complete callback function pointer which gets called from the surface library after the surface measurement is completed.

Note: Towards touch drift rate and away from touch drift rate should be in multiples of each other.

6.4.11. Surface Touch Property (surf_tch_prop_t)

The properties related to touch for the surface library is configured through this structure.

Field	Unit	Description
num_touches	uint8_t	Number of touches required for surface
tch_di	uint8_t	Detect intergrator for the surface library
tch_MOD	uint8_t	Max On duration for the surface library

6.4.12. Surface Library Version Information (surf_libver_info_t)

Surface library version information is part of this structure.

Surface library version information.

Product id for Surface Library is 203.

Firmware version is formed of major, minor, and patch version as follows:

- `SLIB_MAJOR_VERSION` = 1
- `SLIB_MINOR_VERSION` = 0
- `SLIB_PATCH_VERSION` = 1

```
fw_version= (SLIB_MAJOR_VERSION<< 8) | (SLIB_MINOR_VERSION<< 4) |
(SLIB_PATCH_VERSION)
```

Data Fields

Field	Type	Description
chip_id	uint32	Chip identification number
product_id	uint16_t	Product identification number.
fw_version	uint16_t	Library version number.

6.4.13. Surface Status Information(surf_status_t)

The status information from the surface library is passed through this structure to the application.

Data Fields

Field	Type	Description
num_active_touches	uint8_t	Number of active touches present on the touch surface.
surf_tch_state	uint8_t	Each bit corresponds to ON and OFF state of each touch.Bit 0 , if set to 0 signifies touch state is OFF for touch id 0 and if set to 1 means touch is in ON State.
		Similarly, Bit 1 represents ON and OFF for touch id 1.
ptr_surf_tch_status	surf_tch_status_t*	Pointer to individual touch ID status.
ptr_surf_cyc_burst_mask	uint32*	This variable is internal to the Surface library and should not be Read/Write by the customer application.

6.4.14. Touch ID Status Information (surf_tch_status_t)

The status information for the individual touch ID status is provided through this structure to the application.

Data Fields

Field	Unit	Description
tch_id	uint8_t	The touch id corresponds to individual touches. If touch is Invalid, the touch id value is 255.Otherwise, touch id is 0 for first touch, touch id is 1 for second touch and so on.
int_tch_state	uint8_t	This variable is internal to the Surface library
area	uint8_t	Total Area of the Touch
x_position	uint16_t	X position of the touch
y_position	uint16_t	Y position of the touch

6.5. Global Variables

6.5.1. Mutual Capacitance Sensor Measured Data Pointer

`(touch_measure_data_t *p_mutlcap_measure_data)`

This holds the Mutual capacitance method measured data pointer.

6.5.2. Sensor Detect Threshold

`uint8_t sensor_detect_threshold[DEF_MUTLCAP_NUM_CHANNELS]`

This variable is used to hold the Sensor detect thresholds for each stage.

6.6. Functions

6.6.1. Surface Setup and Configuration

```
touch_ret_t touch_mutlcap_sensor_config (sensor_type_t sensor_type, channel_t
from_channel, channel_t to_channel, aks_group_t aks_group, threshold_t
detect_threshold, hysteresis_t detect_hysteresis, resolution_t
position_resolution, uint8_t position_hysteresis, sensor_id_t * p_sensor_id) .
```

This API can be used to configure a sensor of type key.

Data Fields

Field	Description
sensor_type	Can be of type key.
from_channel	The first channel of the individual node
to_channel	The last channel of the individual node
aks_group	The AKS group (if any) the sensor is in
detect_threshold	Not Applicable
detect_hysteresis	The sensor detection hysteresis value
position_resolution	The resolution of the reported position value
position_hysteresis	The hysteresis for position value (available only for mutual cap)
p_sensor_id	The sensor id value of the configured sensor is updated by the Touch Library

Note: The values of `from_channel` and `to_channel` are always identical.

Returns:

`touch_ret_t`: Touch Library Error status.

6.6.2. Surface Library Initialization

The following API is used to initialize the surface library with channel configuration and pin configuration provided by the user.

```
surf_ret_t surf_init(surf_config_t *ptr_surf_config , surf_status_t
*ptr_surf_status,touch_config_t *ptr_touch_config
```

Field	Unit	Description
ptr_surf_config	surf_config_t *	The surface library input configuration are passed through this structure
ptr_surf_status	surf_status_t *	The surface library will update the status information in this pointer
ptr_touch_config	touch_config_t *	The touch configuration are passed through this structure

Return:

`surf_ret_t.`

6.6.3. Surface Calibration All

The following API is used to calibrate the touch surface.

```
surf_ret_t surf_calibrate_all()
```

Field Descritpions

Field	Unit	Description
none		

6.6.4. Normal Measurement

The following API is used to initiate the measurement of QTouch surface library.

```
surf_ret_t surf_measure(surf_current_time_t current_time_ms)
```

Field	Unit	Description
`Current_time_ms`	`surf_current_time_t`	Current time in milliseconds passed from application to library.

6.6.5. Surface Library Version

The following API is used to get the surface library version information.

```
surf_ret_t surf_library_get_version_info( surf_libver_info_t
*ptr_surf_libver_info).
```

Field	Unit	Description
`ptr_surf_libver_info`	`surf_libver_info_t *`	Pointer to the surface libary version information.

Return:

`surf_ret_t.`

6.6.6. Low Power Operation

The following API will be used to configure the surface library for low power measurement.

```
surf_ret_t surf_low_power_start()
```

Field	Unit	Description
none		

Return:

`surf_ret_t.`

The following API will be used to stop the low power measurement.

```
surf_ret_t surf_low_power_stop()
```

Field	Unit	Description
none		

Return:

`surf_ret_t.`

7. Touch Surface Characterization

This section includes charatcterization details such as Power, CPU Load, Response Time, Memory and Touch performance.

7.1. Configuration Details

The configuration settings provided in this table are common to the other characterization data provided in this section.

Table 7-1 Common Configuration Settings

Parameter	Value
CPU Frequency	48 MHz
AUTO_OS	AUTO_OS_DISABLE
FREQ_MODE	FREQ_MODE_SPREAD
ToolChain	IAR
Vcc	3.3 V
Toolchain Optimization	Level:High Type:Size
DPI	162
State	Sleep: No touch is present Active: One or more touch is present

7.2. Power Consumption

The typical power consumption values for Touch Surface application are mentioned in this section. The settings to be applied for this characterization includes values from Configuration Details and the additional setting specified for each table.

Table 7-2 PTC_CLK_FREQUENCY = 8MHz and FILTER_LEVEL_16

Surface Dimension	State	Drifting Periodicity(seconds)	Number of touches	Active Scan Rate (ms)	Sleep Scan Rate (ms)	Current (uA)
10X * 10Y	Sleep	2.4	0	20	100	72
10X * 10Y	Sleep	24	0	20	100	38
10X * 10Y	Active	NA	1	20	100	1350
10X * 10Y	Active	NA	2	40	100	1226
6X * 5Y	Sleep	Disabled	0	20	100	2.5
6X * 5Y	Sleep	2.4	0	20	100	33
6X * 5Y	Sleep	24	0	20	100	5.6
6X * 5Y	Active	NA	1	20	100	1064

Surface Dimension	State	Drifting Periodicity(seconds)	Number of touches	Active Scan Rate (ms)	Sleep Scan Rate (ms)	Current (uA)
6X * 5Y	Active	NA	2	40	100	772
4X * 4Y	Sleep	Disabled	0	20	100	2.5
4X * 4Y	Sleep	2.4	0	20	100	28
4X * 4Y	Sleep	24	0	20	100	4.8
4X * 4Y	Active	NA	1	20	100	868

Table 7-3 PTC_CLK_FREQUENCY = 8MHz and FILTER_LEVEL_32

Surface Dimension	State	Drifting Periodicity(seconds)	Number of touches	Active Scan Rate (ms)	Sleep Scan Rate (ms)	Current (uA)
10X * 10Y	Sleep	2.4	0	20	100	79
10X * 10Y	Sleep	24	0	20	100	40
10X * 10Y	Active	NA	1	20	100	1490
10X * 10Y	Active	NA	2	40	100	1358
6X * 5Y	Sleep	Disabled	0	20	100	2.6
6X * 5Y	Sleep	2.4	0	20	100	36
6X * 5Y	Sleep	24	0	20	100	6.3
6X * 5Y	Active	NA	1	20	100	1240
6X * 5Y	Active	NA	2	40	100	842
4X * 4Y	Sleep	Disabled	0	20	100	2.6
4X * 4Y	Sleep	2.4	0	20	100	30
4X * 4Y	Sleep	24	0	20	100	5
4X * 4Y	Active	NA	1	20	100	937

Table 7-4 PTC_CLK_FREQUENCY = 4MHz and FILTER_LEVEL_16

Surface Dimension	State	Drifting Periodicity(seconds)	Number of touches	Active Scan Rate (ms)	Sleep Scan Rate (ms)	Current (uA)
10X * 10Y	Sleep	2.4	0	20	100	81
10X * 10Y	Sleep	24	0	40	100	42
10X * 10Y	Active	NA	1	20	100	1550
10X * 10Y	Active	NA	2	40	100	1414

Surface Dimension	State	Drifting Periodicity(seconds)	Number of touches	Active Scan Rate (ms)	Sleep Scan Rate (ms)	Current (uA)
6X * 5Y	Sleep	Disabled	0	20	100	2.8
6X * 5Y	Sleep	2.4	0	20	100	36
6X * 5Y	Sleep	24	0	20	100	6.7
6X * 5Y	Active	NA	1	20	100	1348
6X * 5Y	Active	NA	2	40	100	870
4X * 4Y	Sleep	Disabled	0	20	100	2.9
4X * 4Y	Sleep	2.4	0	20	100	31
4X * 4Y	Sleep	24	0	20	100	5.4
4X * 4Y	Active	NA	1	20	100	968

Table 7-5 PTC_CLK_FREQUENCY = 4MHz and FILTER_LEVEL_32

Surface Dimension	State	Drifting Periodicity(seconds)	Number of touches	Active Scan Rate (ms)	Sleep Scan Rate (ms)	Current (uA)
10X * 10Y	Sleep	2.4	0	40	100	93
10X * 10Y	Sleep	24	0	40	100	45
10X * 10Y	Active	NA	1	40	100	962
10X * 10Y	Active	NA	2	60	100	1104
6X * 5Y	Sleep	Disabled	0	20	100	3
6X * 5Y	Sleep	2.4	0	40	100	41
6X * 5Y	Sleep	24	0	40	100	6.9
6X * 5Y	Active	NA	1	40	100	810
6X * 5Y	Active	NA	2	60	100	738
4X * 4Y	Sleep	Disabled	0	40	100	3
4X * 4Y	Sleep	2.4	0	40	100	33
4X * 4Y	Sleep	24	0	40	100	5.9
4X * 4Y	Active	NA	1	40	100	584

7.3. CPU Load

The typical CPU loading values for Touch Surface is mentioned in this section. CPU loading by Surface library is expressed as a percentage of measurement periodicity.

Example: If CPU utilization by Surface library is 2ms and surface measurement cycle is 10ms, then the CPU load is (2ms * 100 /10ms) = 20% .

The settings to be applied for this section includes values from Configuration Details and the additional setting specified above each table. When the CPU load is described as zero % in sleep state, it is because of the low power feature.

Table 7-6 PTC_CLK_FREQUENCY = 8MHz, FILTER_LEVEL_16 and Drifting = Disabled

Hardware	State	Number of touches	Active Scan Rate (ms)	Sleep Scan Rate (ms)	CPU load (%)
10X * 10Y	Sleep	0	20	100	0.46
10X * 10Y	Active	1	20	100	25.35
10X * 10Y	Active	2	40	100	19.39
6X * 5Y	Sleep	0	40	100	0
6X * 5Y	Active	1	20	100	19.07
6X * 5Y	Active	2	40	100	11.84
4X * 4Y	Sleep	0	20	100	0
4X * 4Y	Active	1	20	100	13.36

Table 7-7 PTC_CLK_FREQUENCY = 8MHz, FILTER_LEVEL_32 and Drifting = Disabled

Hardware	State	Number of touches	Active Scan Rate (ms)	Sleep Scan Rate (ms)	CPU load (%)
10X * 10Y	Sleep	0	20	100	0.46
10X * 10Y	Active	1	20	100	22.53
10X * 10Y	Active	2	40	100	19.24
6X * 5Y	Sleep	0	40	100	0
6X * 5Y	Active	1	20	100	16.52
6X * 5Y	Active	2	40	100	11.79
4X * 4Y	Sleep	0	20	100	0
4X * 4Y	Active	1	20	100	12.71

Table 7-8 PTC_CLK_FREQUENCY = 4MHz, FILTER_LEVEL_16 and Drifting = Disabled

Hardware	State	Number of touches	Active Scan Rate (ms)	Sleep Scan Rate (ms)	CPU load (%)
10X * 10Y	Sleep	0	20	100	0.49
10X * 10Y	Active	1	20	100	27.48
10X * 10Y	Active	2	40	100	21.99
6X * 5Y	Sleep	0	40	100	0
6X * 5Y	Active	1	20	100	21.07

Hardware	State	Number of touches	Active Scan Rate (ms)	Sleep Scan Rate (ms)	CPU load (%)
6X * 5Y	Active	2	40	100	11.16
4X * 4Y	Sleep	0	20	100	0
4X * 4Y	Active	1	20	100	14.26

Table 7-9 PTC_CLK_FREQUENCY = 4MHz, FILTER_LEVEL_32 and Drifting = Disabled

Hardware	State	Number of touches	Active Scan Rate (ms)	Sleep Scan Rate (ms)	CPU load (%)
10X * 10Y	Sleep	0	40	100	0.49
10X * 10Y	Active	1	40	100	13.54
10X * 10Y	Active	2	60	100	14.63
6X * 5Y	Sleep	0	40	100	0
6X * 5Y	Active	1	40	100	10.62
6X * 5Y	Active	2	60	100	7.46
4X * 4Y	Sleep	0	40	100	0
4X * 4Y	Active	1	40	100	7.22

7.4. Response Time

The typical Response time of Touch Surface Application is mentioned in this section. Response time is described as the time difference between the start of a measurement cycle and the corresponding status output availability. The settings to be applied for this section includes values from Configuration Details and the additional setting specified on each table.

Table 7-10 PTC_CLK_FREQUENCY = 8MHz and FILTER_LEVEL_16

Hardware	State	Number of touches	Active Scan Rate (ms)	Sleep Scan Rate (ms)	Response time (ms)
10X * 10Y	Active	1	20	100	11.68
10X * 10Y	Active	2	40	100	21.16
6X * 5Y	Active	1	20	100	9.76
6X * 5Y	Active	2	40	100	13.47
4X * 4Y	Active	1	20	100	6.86

Table 7-11 PTC_CLK_FREQUENCY = 8MHz and FILTER_LEVEL_32

Hardware	State	Number of touches	Active Scan Rate (ms)	Sleep Scan Rate (ms)	Response time (ms)
10X * 10Y	Active	1	20	100	16.52
10X * 10Y	Active	2	40	100	32.14

Hardware	State	Number of touches	Active Scan Rate (ms)	Sleep Scan Rate (ms)	Response time (ms)
6X * 5Y	Active	1	20	100	14.13
6X * 5Y	Active	2	40	100	20.76
4X * 4Y	Active	1	20	100	10.15

Table 7-12 PTC_CLK_FREQUENCY = 4MHz and FILTER_LEVEL_16

Hardware	State	Number of touches	Active Scan Rate (ms)	Sleep Scan Rate (ms)	Response time (ms)
10X * 10Y	Active	1	20	100	17.66
10X * 10Y	Active	2	40	100	33.64
6X * 5Y	Active	1	20	100	15.23
6X * 5Y	Active	2	40	100	21.48
4X * 4Y	Active	1	20	100	10.57

Table 7-13 PTC_CLK_FREQUENCY = 4MHz and FILTER_LEVEL_32

Hardware	State	Number of touches	Active Scan Rate (ms)	Sleep Scan Rate (ms)	Response time (ms)
10X * 10Y	Active	1	40	100	28.43
10X * 10Y	Active	2	60	100	55.36
6X * 5Y	Active	1	40	100	24.98
6X * 5Y	Active	2	60	100	35.95
4X * 4Y	Active	1	40	100	17.47

7.5. Memory

The typical memory requirements for QTouch Surface application is specified in the following table.

Table 7-14 PTC_CLK_FREQUENCY = 8MHz and FILTER_LEVEL_16

Flash (KBytes)	RAM (KBytes)	
10X * 10Y	20.8	5.6
6X * 5Y	21.7	3.6
4X * 4Y	21.5	3.2

Note: In *6X * 5Y* and *4X * 4Y* configurations, Low Power sensor is used.Surface Size (X * Y).

8. Revision History

Doc. Rev.	Date	Comments
Rev.B	04/2015	Included information on two-dimensional sensing
Rev.A	02/2015	Initial document release

Atmel Corporation 1600 Technology Drive, San Jose, CA 95110 USA **T:** (+1)(408) 441.0311 **F:** (+1)(408) 436.4200 | **www.atmel.com**