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

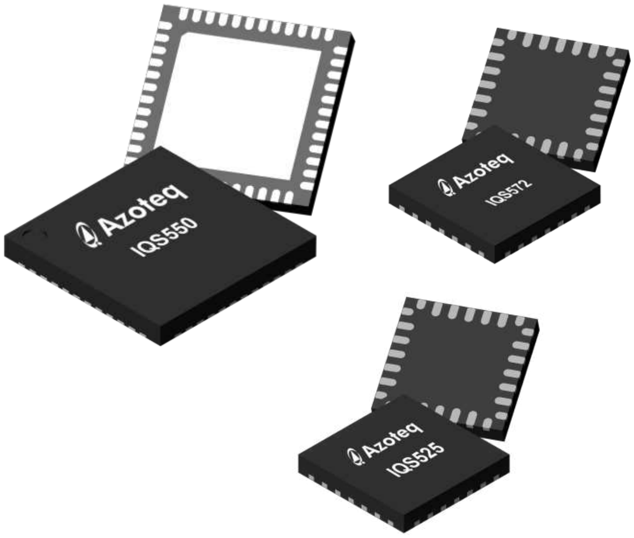
Tel: +86-755-8981 8866 Fax: +86-755-8427 6832
Email & Skype: info@chipsmall.com Web: www.chipsmall.com
Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China

IQS550/572/525-B000 – Capacitive Trackpad/Touchscreen Controller

Projected capacitive controller with proximity, touch, snap, trackpad outputs and gestures

The IQS5xx-B000 is a projected capacitive touch and proximity trackpad/touchscreen controller implementation on the IQS550, IQS572 and IQS525 platforms. The IQS5xx-B000 features best in class sensitivity, signal-to-noise ratio and automatic tuning of electrodes. Low power proximity detection allows extreme low power operation.

Main Features



- Proximity, touch and snap* on each channel
- Multi-touch support up to 5 fingers
- Single and multi-finger gestures
- 3584 x 2304 max resolution (IQS550)
- Scale, orientation and electrode layout selection
- I^2C communication interface
- ATI: automatic tuning for optimum sensitivity
- Supply Voltage 1.65V to 3.6V
- Proximity low power operation (<10uA)
- 3 Active and 2 low power modes
- Event and streaming modes
- Internal voltage regulator and reference capacitor
- On-chip noise detection and suppression

	IQS550	IQS572	IQS525
Maximum channels	150	72	25
Typical report rate (with single touch / all channels active)	100Hz	135Hz	190Hz
Maximum resolution (for shown Tx Rx configurations)	3584 x 2304 (15 x 10)	2048 x 1792 (9 x 8)	1280 x 768 (6 x 4)

Applications

- Compact Capacitive Keyboards
- Remote Control Trackpads
- Appliances
- Navigation devices
- Kiosks and POS terminals
- E-readers

T_A	QFN(7x7)-48	QFN(4x4)-28	QFN(4x4)-28
-40°C to 85°C	IQS550	IQS572	IQS525

*patented

Contents

List of Abbreviations

ALP	Alternate Low Power
ATI	Automatic Tuning Implementation
EMI	Electromagnetic Interference
ESD	Electrostatic Discharge
GND	Ground
GUI	Graphical User Interface
IC	Integrated Circuit
ICI	Internal Capacitor Implementation
IIR	Infinite Impulse Response
LP	Low Power
LTA	Long Term Average
MAV	Moving Average
ND	Noise Detect
THR	Threshold
TP	Trackpad
WDT	Watchdog Timer

1 Overview

The IQS550 / IQS572 / IQS525 are capacitive sensing controllers designed for multi-touch applications using projected capacitance touch panels. The device offers high sensitivity proximity wake-up and contact detection (touch) through a selectable number of sensor lines (Rxs and Txs).

The device has an internal voltage regulator and Internal Capacitor Implementation (ICI) to reduce external components. Advanced on-chip signal processing capabilities provide stable high performance with high sensitivity.

A trackpad consists of an array of sensors that are scanned at regular intervals. The controller uses the principle of projected capacitance charge transfer on the trackpad. When a conductive object such as a human finger approaches the sense plate it will decrease the detected capacitance. Thresholds are applied to the sensor data to identify areas that exhibit proximity and touch deviation. The contours of the touch areas are then translated to Cartesian position coordinates that are continuously monitored to identify gestures. A user has access to all of the data layers – the raw sensor data, the sensor proximity/touch status data, the XY coordinates as well as the gesture outputs.

Multiple filters are implemented to detect and suppress noise, track slow varying environmental conditions and avoid effects of possible drift. The Auto Tuning (ATI) allows for the adaptation to a wide range of touch screens without using external components.

An innovative addition, known as a snap*, is also available on each channel. This adds another channel output, additional to the proximity and touch.

The trackpad application firmware on the IQS5xx is very flexible in design, and can incorporate standard touch sensors, trackpad / touchscreen areas (giving XY output data) and conventional snap-dome type buttons, all providing numerous outputs such as proximity, touch, snap, touch strength, area and actual finger position all in one solution.

The IQS550, IQS572 and IQS525 devices ship with the bootloader only, since the designer must program custom IQS5xx-B000 firmware during production testing. The custom firmware is the IQS5xx-B000 trackpad firmware together with customer specific hardware settings exported by the GUI program.

*This datasheet applies to the following **IQS550** version:*
Product Number 40 / Project Number 15 / Version Number 2

*This datasheet applies to the following **IQS572** version:*
Product Number 58 / Project Number 15 / Version Number 2

*This datasheet applies to the following **IQS525** version:*
Product Number 52 / Project Number 15 / Version Number 2

*patented

2 Packaging and Pin-out

2.1 IQS550 - QFN48

The IQS550 is available in a QFN(7x7)-48 package.

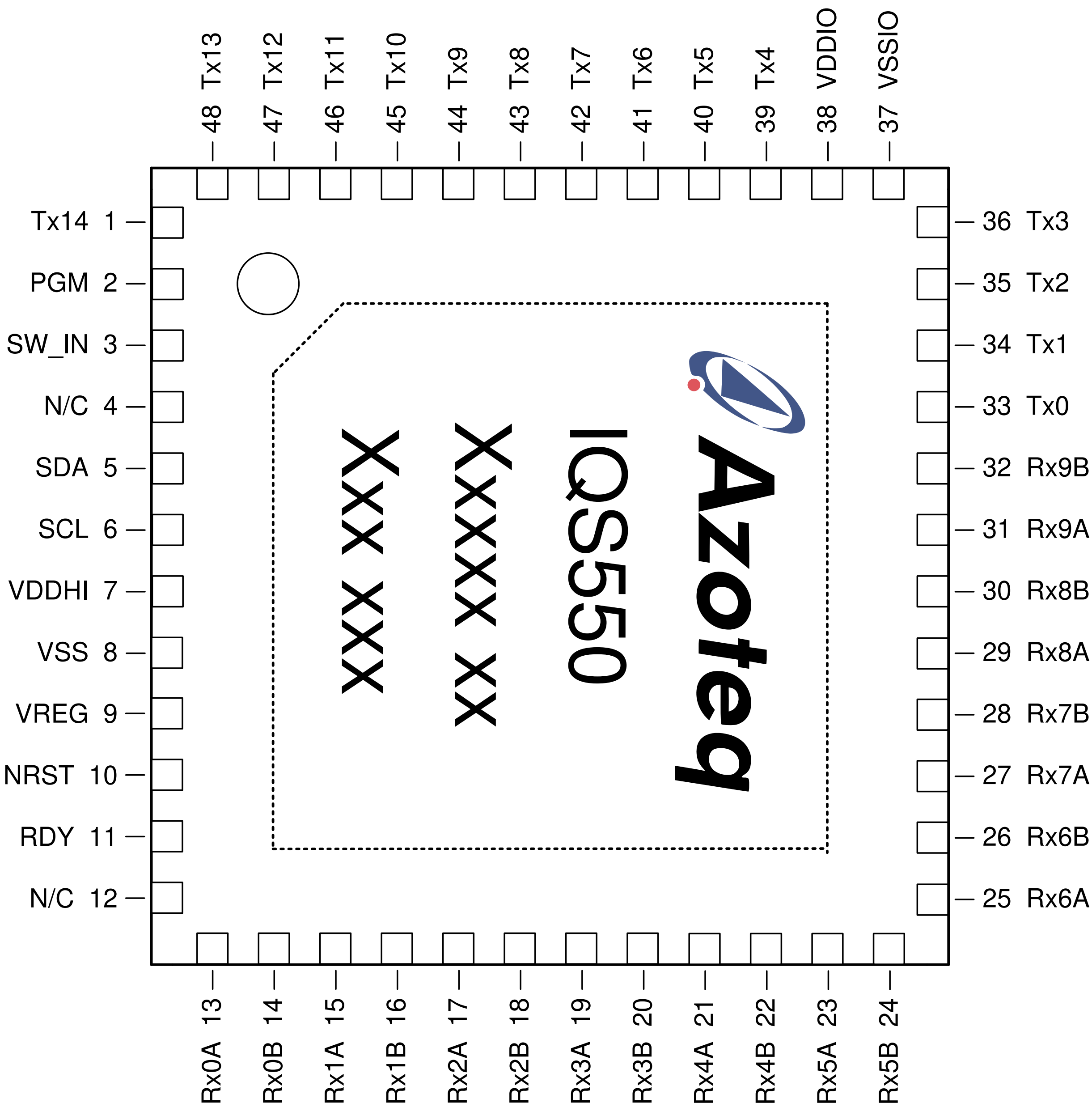


Figure 2.1 QFN Top View

Table 2.1 QFN48 Pin-out

Pin	Name	Description
1	Tx14	Transmitter electrode
2	PGM	Programming Pin
3	SW_IN	Wake-up from suspend and switch input
4	n/c	~
5	SDA	I^2C Data
6	SCL	I^2C Clock
7	VDDHI	Supply Voltage
8	VSS	Ground Reference
9	VREG	Internal Regulator Voltage
10	NRST	Reset (active LOW)
11	RDY	I^2C RDY
12	n/c	~
13	Rx0A	Receiver electrode
14	Rx0B	Note1
15	Rx1A	Receiver electrode
16	Rx1B	Note1
17	Rx2A	Receiver electrode
18	Rx2B	Note1
19	Rx3A	Receiver electrode
20	Rx3B	Note1
21	Rx4A	Receiver electrode
22	Rx4B	Note1
23	Rx5A	Receiver electrode
24	Rx5B	Note1
25	Rx6A	Receiver electrode
26	Rx6B	Note1
27	Rx7A	Receiver electrode
28	Rx7B	Note1
29	Rx8A	Receiver electrode
30	Rx8B	Note1
31	Rx9A	Receiver electrode
32	Rx9B	Note1
33	Tx0	Transmitter electrode
34	Tx1	Transmitter electrode
35	Tx2	Transmitter electrode
36	Tx3	Transmitter electrode
37	VSSIO	I/O Ground Reference
38	VDDIO	I/O Supply Voltage
39	Tx4	Transmitter electrode
40	Tx5	Transmitter electrode
41	Tx6	Transmitter electrode
42	Tx7	Transmitter electrode
43	Tx8	Transmitter electrode
44	Tx9	Transmitter electrode
45	Tx10	Transmitter electrode
46	Tx11	Transmitter electrode
47	Tx12	Transmitter electrode
48	Tx13	Transmitter electrode

Note1: Any of these can be configured through I^2C as the ProxSense® electrode.

2.2 IQS572 - QFN28

The IQS572 is available in a QFN(4x4)-28 package. The production version is shown below.

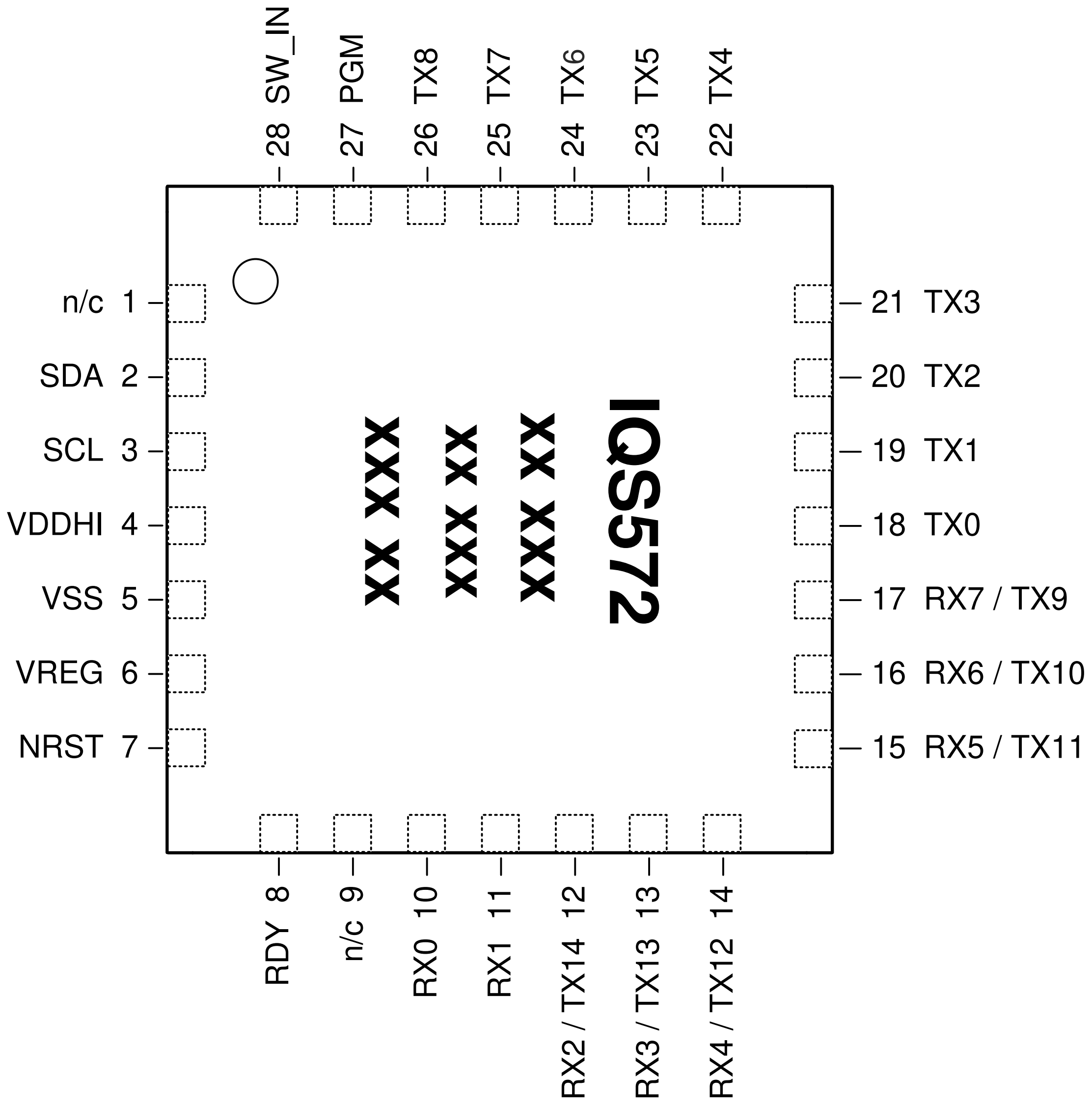


Figure 2.2 IQS572 QFN Top View

Table 2.2 IQS572 QFN28 Pin-out

Pin	Name	Description
1	n/c	~
2	SDA	I^2C Data
3	SCL	I^2C Clock
4	VDDHI	Supply Voltage
5	VSS	Ground Reference
6	VREG	Internal Regulator Voltage
7	NRST	Reset (active LOW)
8	RDY	I^2C RDY
9	n/c	~
10	Rx0	Receiver electrode
11	Rx1	Receiver electrode
12	Rx2	Receiver electrode
13	Rx3	Receiver electrode
14	Rx4	Receiver electrode
15	Rx5	Receiver electrode
16	Rx6	Receiver electrode
17	Rx7	Receiver electrode
18	Tx0	Transmitter electrode
19	Tx1	Transmitter electrode
20	Tx2	Transmitter electrode
21	Tx3	Transmitter electrode
22	Tx4	Transmitter electrode
23	Tx5	Transmitter electrode
24	Tx6	Transmitter electrode
25	Tx7	Transmitter electrode
26	Tx8	Transmitter electrode
27	PGM	Programming Pin
28	SW_IN	Wake-up from suspend and switch input

2.3 IQS525 - QFN28

The IQS525 is available in a QFN(4x4)-28 package. The production version is shown below.

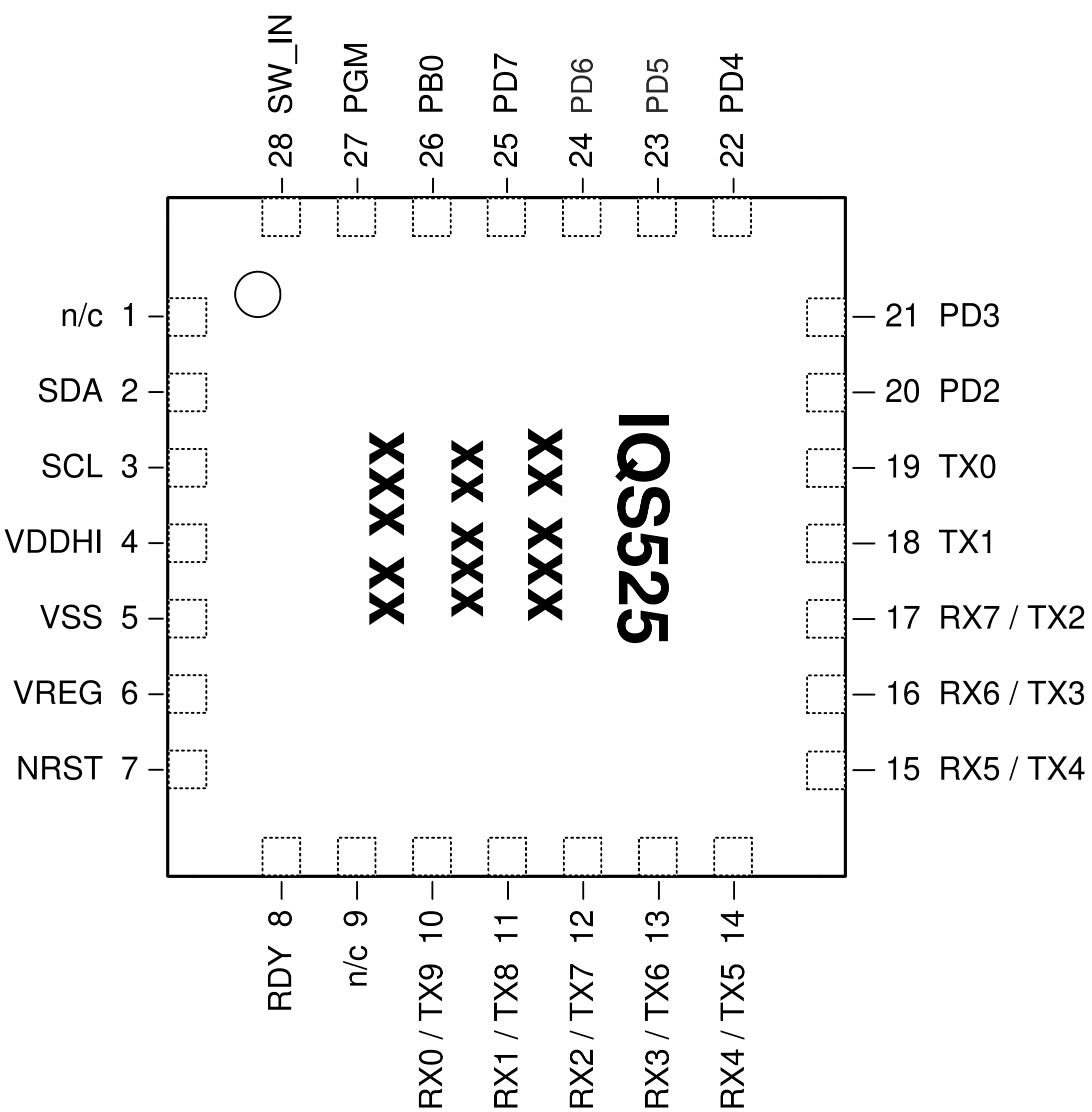


Figure 2.3 IQS525 QFN Top View

Table 2.3 QFN28 Pin-out

Pin	Name	Description
1	n/c	~
2	SDA	I²C Data
3	SCL	I²C Clock
4	VDDHI	Supply Voltage
5	VSS	Ground Reference
6	VREG	Internal Regulator Voltage
7	NRST	Reset (active LOW)
8	RDY	I²C RDY
9	n/c	~
10	Rx0	Receiver electrode
11	Rx1	Receiver electrode
12	Rx2	Receiver electrode
13	Rx3	Receiver electrode
14	Rx4	Receiver electrode
15	Rx5 / TX4	Receiver / Transmitter electrode
16	Rx6 / TX3	Receiver / Transmitter electrode
17	Rx7 / TX2	Receiver / Transmitter electrode
18	Tx1	Transmitter electrode
19	Tx0	Transmitter electrode
20	PD2	General purpose I/O
21	PD3	General purpose I/O
22	PD4	General purpose I/O
23	PD5	General purpose I/O
24	PD6	General purpose I/O
25	PD7	General purpose I/O
26	PB0	General purpose I/O
27	PGM	Programming Pin
28	SW_IN	Wake-up from suspend and switch input

3 ProxSense® Module

The IQS5xx contains a ProxSense® module that uses patented technology to measure and process the capacitive sensor data. The trackpad sensors are scanned one Tx transmitter at a time, until all have completed, with all enabled Rxs charging in each Tx time slot. The channel outputs (proximity, touch and snap) are the primary outputs from the sensors. These are processed further to provide secondary trackpad outputs that include finger position, finger size as well as on-chip gesture recognition.

The additional snap state is a unique sensor output that utilises capacitive technology to sense the depression of a metal dome snap button onto the customized sensor area. This gives an additional output above the traditional proximity and touch channel outputs.

For more information on capacitive sensing and charge transfers, please refer to the Azoteq Application Note AZD004.

For more information regarding design guidelines refer to the Application Note AZD068.

3.1 Channel Definition

A channel for a projected capacitive sensor consists of a Tx electrode that is in close proximity to an Rx electrode.

On a trackpad sensor (typically a diamond shape pattern), each intersection of an Rx and Tx row/column forms a capacitive sensing element which is referred to as a *channel*. Each channel has an associated count value, reference value, proximity, touch and snap (if enabled) status. The maximum number of Tx and Rx electrodes on the IQS550 device is 15x10, thus giving 150 channels in total.

3.2 Alternate Low-Power Channel (ALP)

If lower power consumption is required (ALP), LP1 and LP2 can be configured to utilise a single custom channel sensor, instead of sensing the trackpad channels. This channel has a lot of setup flexibility:

- Sensing method (CHARGE_TYPE): projected capacitive or self capacitive.
- Sensors: which Rxs (RX_GROUP / ALP Rx select) / Txs (*ALP Tx select*) are active during conversions.
- Reverse sensing: If enabled, negative deviations can also trigger proximity detection (PROX_REVERSE).
- Count value filtering: gives reliable proximity detection in noisy environments.
- Single channel: since the alternate channel is processed as only a single channel, much less processing is done, allowing for lower overall power consumption.

Since all Rxs return a count measurement, it means that the ALP channel can be a combination of numerous measurements. To reduce processing time (and this decrease current consumption) the measurements are added together, and processed as a single 'channel'.

3.3 Count Value

The capacitive sensing measurement returns a *count value* for each channel. Count values are inversely proportional to capacitance, and all outputs are derived from this them.

3.3.1 Trackpad Count Values

The individual trackpad channel count values (Count values) are unfiltered.

3.3.2 ALP Count Values

The combined count value (ALP count value) used for this channel is a summation of the individual count values (ALP individual count values) from each active Rx.

A count value filter is implemented on this channel to give stable proximity output for system wake-up from a low-power mode. It is recommended to leave this count filter enabled (ALP_COUNT_FILTER).

The amount of filtering can be modified (ALP count beta) if required. This beta is used as follows to determine the damping factor of the filter:

Count damping factor = Beta / 256

If the beta is small, the filtering is stronger, and if the beta is larger, the filtering is weaker.

3.3.3 Max Count

Each channel is limited to having a count value smaller than the configurable limit (Max count limit). If the ATI setting or hardware causes measured count values higher than this, the conversion will be stopped, and a value of '0' will be read for that relevant count value. Note that a '0' is also returned for a disabled channel.

3.3.4 Delta Value

The delta values (Delta values) are simply:

Delta = Count - Reference

3.4 Reference Value

User interaction is detected by comparing count values to reference values. The count value of a sensor represents the instantaneous capacitance of the sensor. The reference value of a sensor is the count value of the sensor that is slowly updated to track changes in the environment, and is not updated during user interaction.

The reference value is a two-cycle averaged of the count value, stored during a time of no user activity, and thus is a non-affected reference. The trackpad reference values are only updated from LP1 and LP2 mode when modes are managed automatically. Thus, if the system is controlled manually, the reference must also be managed and updated manually by the host.

3.4.1 Reference Update Time

The reference value is updated or refreshed according to a configurable interval (Reference update time), in seconds.

To ensure that the reference value is not updated during user interaction, it only executes from the LP1 and LP2 states, where no user interaction is assumed.

Setting the Reference update time to '0' will disable the updating of the reference values.

3.4.2 ALP Long-Term Average

The ALP channel does not have a snapshot reference value as used on the trackpad, but utilises a filtered long-term average value (ALP LTA value). The LTA tracks the environment closely for accurate comparisons to the measured count value, to allow for small proximity deviations to be sensed. The speed of LTA tracking can be adjusted with the ALP LTA beta. There is an ALP1 and ALP2, which are implemented in LP1 and LP2 respectively. This is to allow different settings for different report rates, so that the LTA tracking rate can remain the same.

3.4.3 Reseed

Since the *Reference* (or *LTA* for ALP channel) is critical for the device to operate correctly, there could be known events or situations which would call for a manual reseed. A reseed takes the latest measured counts, and seeds the *reference/LTA* with this value, therefore updating the value to the latest environment. A reseed command can be given by setting the corresponding bit (RESEED or ALP_RESEED).

3.5 Channel Outputs

For the trackpad channels, user interaction typically causes the count values to increase. The amount of deviation relative to the reference can be used to determine the output state of the channel, dependent on the sensitivities configured.

For a snap actuation, the count values decrease, and a negative deviation cause a snap output.

If the measured count value exceeds the selected threshold value for consecutive cycles, equal in number to the selectable debounce parameter, the output becomes set.

3.5.1 Proximity

This output (Prox_status) is set when a channels' count value deviates from the reference value by more than the selected threshold (Prox threshold).

The proximity threshold is the smallest difference between the count value and the reference value that would result in a proximity output. Small threshold values are thus more sensitive than large threshold values.

Note: For the trackpad channels (projected capacitive) the samples will increase with user interaction, thus the actual threshold is the reference value PLUS the threshold parameter. However, if an ALP channel is implemented in self capacitive mode, the samples will decrease during user interaction, thus the actual threshold is the reference value MINUS the threshold parameter.

3.5.2 Touch

This output (*Touch status*) is set when a channels' count value increases by more than the selected threshold.

The touch threshold for a specific channel is calculated as follows:

Threshold = Reference x (1 + Multiplier / 128)

A smaller fraction will thus be a more sensitive threshold.

A trackpad will have optimal XY data if all of the channels in the trackpad exhibit similar deltas under similar user inputs. In such a case all of the channels will have identical thresholds. In practise, sensor design and hardware restrictions could cause deltas which are not constant over the entire trackpad. It could then be required to select individual multiplier values. These (*Individual touch multiplier adjustment*) are signed 8-bit values and indicate how much the unsigned 8-bit global value (*Global touch multiplier*) must be adjusted. The threshold used for a specific channel (set and clear) is as follows:

Multiplier = Global + Individual adjust

A hysteresis can also be implemented because there are different touch multiplier parameters for setting a touch and clearing a touch. This hysteresis allows the channels to not flicker in and out of touch with noise.

3.5.3 Snap

When adding a metal snap-dome overlay to the trackpad pattern, an additional *snap* output (*Snap status*) is available. The device is able to distinguish between a normal 'touch' on the overlay and an actual button 'snap', which depresses the metal dome onto the Rx/Tx pattern. The design must be configured so that a snap on the metal dome will result in a channels' count value falling well below the reference for that channel.

If required, the function must be enabled (*Snap enabled channels*) for each channel on which snap is designed. *Only channels with snap must be marked as such, since channels are handled differently if they are snap channels, compared to non-snap channels.*

One global snap threshold (*Snap threshold*) is implemented as a delta value BELOW the reference. When a snap is performed, a sensor saturation effect causes the deviation to be negative.

Because it is only necessary to read the individual snap registers if a state change has occurred, a status bit (*SNAP_TOGGLE*) is added to indicate this. This is only set when there is a change of status of any snap channel.

A reseed is executed if a snap is sensed for longer than the *Snap timeout* time (in seconds). A setting of 0 will never reseed. The timeout is reset if any snap is set or cleared.

3.5.4 Output Debounce

All the channel outputs (proximity, touch and snap) are debounced according to the selectable debounce values (*Prox debounce* / *Touch snap debounce*). Note that a debounce value of 1 means that two samples satisfying the condition must be met consecutively before the output is activated. The default *touch debounce* is set to 0 / no debouncing. This is due to the fact that with a 15x10 sensor, debouncing adds too much delay, and fast movements on the touch panel cannot be debounced fast enough to provide reliable XY output data.

3.5.5 Maximum Touch

An additional output is provided (*Max Touch*), and indicates the column and row of the channel with the largest touch deviation. This is usually only utilised when implementing discrete buttons, to reject any adjacent keys if they are located in close proximity to each other. If the Rxs and Txs are switched (*SWITCH_XY_AXIS*), the columns are the Txs, and the rows are the Rxs. If no touches are seen, then this will output 0xFF.

3.6 Auto Tuning (ATI)

The ATI is a sophisticated technology implemented in the new ProxSense® devices to allow optimal performance of the devices for a wide range of sensing electrode capacitances, without modification to external components. The ATI settings allow tuning of two parameters, ATI C Multiplier and ATI Compensation, to adjust the sample value for an attached sensing electrode.

For detailed information regarding the on-chip ATI technology, please refer to AZD027 and AZD061.

The main advantage of the ATI is to balance out small variations between trackpad hardware and IQS5xx variation, to give similar performance across devices.

3.6.1 ATI C Multiplier

All trackpad channels can be adjusted globally by modifying the global parameter (*Global ATI C*).

Although it is recommended to keep the same ATI C value for all trackpad channels, if different values are required (possibly for different trackpads), individual adjustments can be made. The ATI C value for each channel can be adjusted using 8-bit signed values (*ATI C individual adjust*) as follows:

ATI C = Global + Individual Adjust

The ALP channel has its own global ATI C parameter (*ALP ATI C*).

3.6.2 ATI Compensation & Auto ATI

The ATI Compensation value for each channel (*ATI compensation*) is set by means of an automated ATI procedure. The algorithm is executed after the *AUTO_ATI* bit is set. The ATI Compensation values are chosen so that each count value is close to the selected target value (*ATI target* / *ALP ATI target*).

The *AUTO_ATI* bit clears automatically on chip when the algorithm has completed.

The ATI routine will run for the channels of the current mode, for example, if the system is currently sensing the *alternate low-power* channel, the auto ATI will apply to it, similarly the algorithm will configure the trackpad channels if they are currently active.

The ALP channel has individual compensation values (*ALP ATI compensation*) for each enabled Rx.

The ALP ATI target value applies to each of the individual count values configured for the ALP channel.

Note: This routine will only execute after the communication window is terminated, and the I^2C communication will only resume again once the ATI routine has completed.

3.7 Automatic Re-ATI

3.7.1 Description

When enabled (*REATI* or *ALP_REATI*) the ATI algorithm will be repeated if certain conditions are met. One of the most important features of the Re-ATI is that it allows easy and fast recovery from an incorrect ATI, such as when performing ATI during user interaction with the sensor. This could cause the wrong ATI Compensation to be configured, since the user affects the capacitance of the sensor. A Re-ATI would correct this.

When a Re-ATI is performed on the IQS5xx, a status bit will set momentarily to indicate that this has occurred (*REATI_OCCURRED* / *ALP_REATI_OCCURRED*).

3.7.2 Conditions for Re-ATI to activate

1. **Reference drift**

A Re-ATI is performed when the reference of a channel drifts outside of the acceptable range around the ATI Target.

The boundaries where Re-ATI occurs for the trackpad channels and for the ALP channels are independently set via the drift threshold value (*Reference drift limit* / *ALP LTA drift limit*). The Re-ATI boundaries are calculated from the delta value as follows:

Re-ATI Boundary = ATI target ± Drift limit

For example, assume that the ATI target is configured to 800 and that the reference drift value is set to 50. If Re-ATI is enabled, the ATI algorithm will be repeated under the following conditions:

Reference > 850 or

Reference < 750

The ATI algorithm executes in a short time, so goes unnoticed by the user.

2. **Very large count values**

The configurable *Max count limit* is used to sense for unexpectedly large count values. A Re-ATI is triggered if the max count limit is exceeded for 15 consecutive cycles.

This limit is configured to be a value higher than the maximum count possible through user interaction, plus worst case noise on the count value, plus headroom. The monitoring of this assists in correcting for a Re-ATI which occurred during a snap press. If this does occur, after removing the snap, the counts are typically very high. If this was not monitored a stuck touch could occur.

3. **Decreased count value**

A considerable decrease in the count value of a non-snap channel is abnormal, since user interaction increases the count value. Therefore if a decrease larger than the configurable threshold (*Minimum count Re-ATI delta*) is seen on such a channel, it is closely monitored. If this is continuously seen for 15 cycles, it will trigger a Re-ATI. If the channel is a snap channel, this decrease is allowed since snap does cause count values to decrease.

3.7.3 ATI Error

After the ATI algorithm is performed, a check is done to see if there was any error with the algorithm. An ATI error is reported if one of the following is true for any channel after the ATI has completed:

- ATI Compensation <= *ReATI lower compensation limit*
- ATI Compensation >= *ReATI upper compensation limit*
- Count is already outside the Re-ATI range upon completion of the ATI algorithm.

If any of these conditions are met, the corresponding error flag will be set (*ATI ERROR* / *ALP ATI ERROR*). The flag status is only updated again when a new ATI algorithm is performed.

Re-ATI will not be repeated immediately if an ATI Error occurs. A configurable time (*Re-ATI retry time*) will pass where the Re-ATI is momentarily suppressed. This is to prevent the Re-ATI repeating indefinitely. An ATI error should however not occur under normal circumstances.

3.7.4 Design requirements

The Re-ATI can be very useful when ATI parameters are selected for which successful Re-ATI operation can be expected. With the conditions for Re-ATI mentioned above, it is clear that when the designer sets the ATI parameters, it is beneficial to select the ATI C and ATI Target so that the resulting ATI Compensation values are near the centre of the range. This ensures that with changing sensitivity, the ATI Compensation has the ability to increase/decrease in value without it easily becoming 0 or 255. In general, ATI Compensation values between 100 and 150 are desirable as they provide ample room for adjustment. Note that the range is dependent on the sensitivity requirements, and on the capacitance of the sensor.

3.8 Sensing Hardware Settings

Settings specific to the ProxSense® Module charge transfer characteristics can be changed.

The charge transfer frequency (f_{cc}) can be calculated as:

$$f_{cc} = \frac{16.10^6}{(2^{(7-CK\_FREQ)} \times (2 + UP + PASS + INC\_PHASE)} \text{ [Hz]}$$

where

$UP = 2^{(UPLEN-2)}$ (if UPLEN > 4)

$UP = UPLEN$ (if UPLEN ≤ 4)

$PASS = 2^{(PASSLEN-2)}$ (if PASSLEN > 4)

$PASS = PASSLEN$ (if PASSLEN ≤ 4)

Note: CK_FREQ, UPLEN and PASSLEN are the numerical values of the settings.

For example, the default frequency is:

$$f_{cc} = \frac{16.10^6}{(2^{(7-7)} \times (2 + 4 + 3 + 0)} = 1.77MHz$$

The other hardware parameters are not discussed as they should only be adjusted under guidance of Azoteq support engineers.

4 Sensing Modes

The IQS5xx automatically switches between different charging modes dependent on user interaction and other aspects. This is to allow for fast response, and also low power consumption when applicable. The current mode can be read from the device (CHARGING_MODE).

The modes are best illustrated by means of the following state diagram.

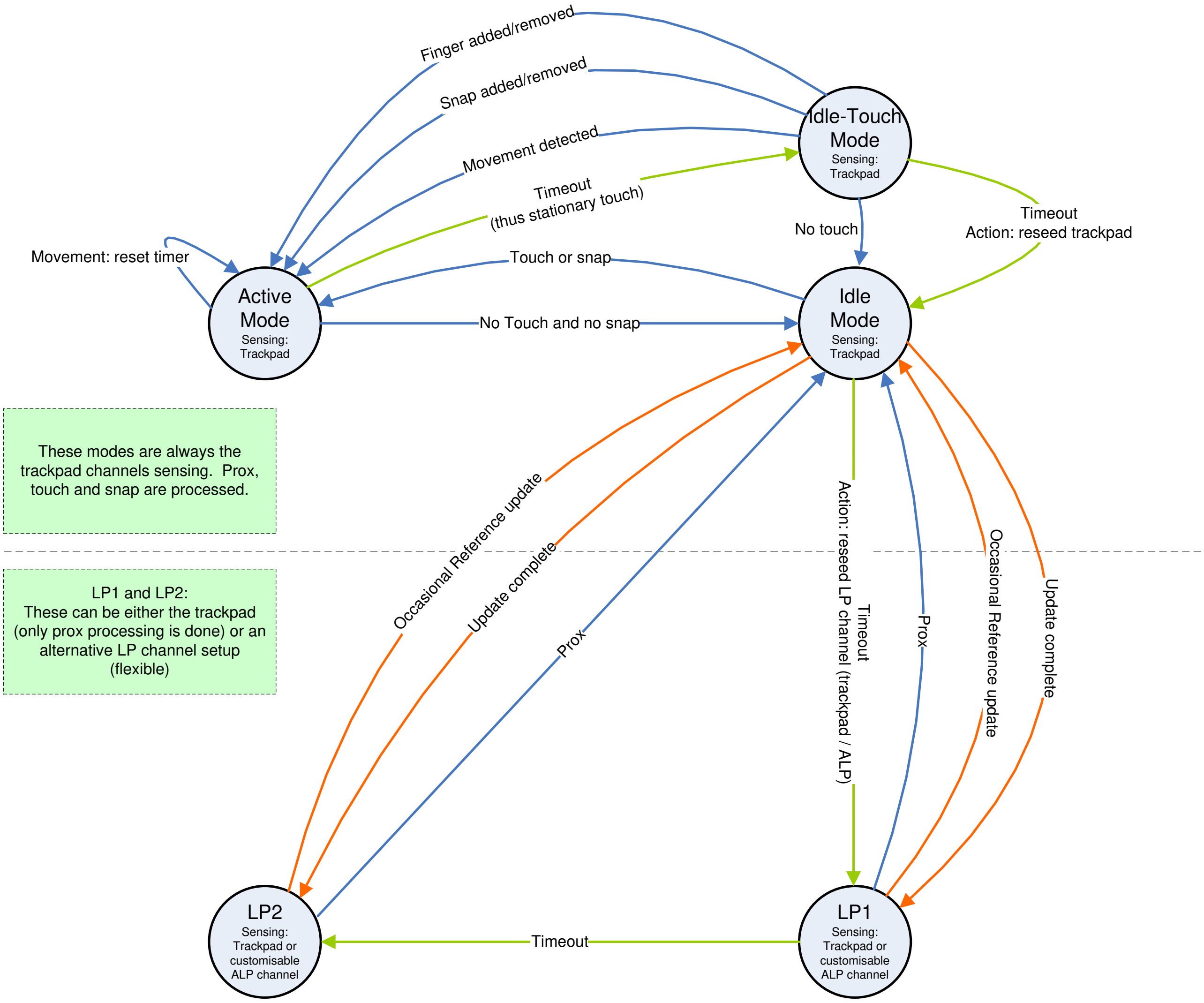


Figure 4.1 System Mode State Diagram

4.1 Report Rate

The report rate for each mode can be adjusted as required by the design. A faster report rate will have a higher current consumption, but will give faster response to user interaction. *Active mode* typically has the fastest report rate, and the other modes are configured according to the power budget of the design, and the expected response time.

The report rate is configured by selecting the cycle time (in milliseconds) for each mode:

- *Report rate Active mode*
- *Report rate Idle touch mode*
- *Report rate Idle mode*
- *Report rate LP1 mode*
- *Report rate LP2 mode*

4.1.1 Previous Cycle Time

The achieved report rate can be read (*Previous cycle time*) from the device each cycle; this is the previous cycles' length in milliseconds. If the desired rate is not achievable, that is, if processing and sensing takes longer than the specified time, a status flag (*RR_MISSED*) indicates that the rate could not be achieved.

4.2 Mode Timeout

The timeout values can be configured, and once these times have elapsed, the system will change to the next state according to the state diagram.

These times are adjusted by selecting a desired value (in seconds), for the specific timeout:

- *Timeout - Active mode*
- *Timeout - Idle touch mode*
- *Timeout - Idle mode*
- *Timeout - LP1 mode*

Note: the timeout for LP1 is set in multiples of 20s (thus a setting of '30' translates to 600s, or 10min).

A timeout value of 255 will result in a 'never' timeout condition.

4.3 Manual Control

The default method allows the IQS5xx to automatically switch between modes and update reference values as shown in Figure 4.1. This requires no interaction from the master to manage the IQS5xx.

The master can manage various states and implement custom power modes when *Manual Control* is enabled (*MANUAL_CONTROL*). The master needs to control the mode (*MODE_SELECT*), and also manage the reference values by reseeding (*RESEED*) or manually writing to the reference registers (*Reference values*).

5 Trackpad

5.1 Configuration

5.1.1 Size Selection

The total number of Rx and Tx channels used for trackpad purposes must be configured (*Total Rx* / *Total Tx*). This gives a rectangular area of channels, formed by rows and columns of Rx and Tx sensors.

5.1.2 Individual Channel Disabling

If the sensor is not a completed rectangle (this could be due to board cut-outs or trackpad shape), channels not implemented but falling within the *Total Rx* / *Total Tx* rectangle, must be individually disabled (*Active channels*).

5.1.3 Rx / Tx Mapping

The Rxs and Txs of the trackpad can be assigned to the trackpad in any order to simplify PCB layout and design. Rxs and Txs can however not be interchanged (for example you cannot use both Rxs and Txs for the columns of the trackpad).

For both the mapping registers (*Rx mapping* / *Tx mapping*) the first byte relates to the mapping of the first row/column, the next byte in the memory map is the next row/column, and so on.

Example: If a 5x5 trackpad was to be designed with Rx/Tx mapping to columns and rows as shown in Table 5.1, the Rx and Tx mapping registers would need to be set as follows:

Rx Mapping = {3, 0, 8, 1, 2}

Tx Mapping = {0, 1, 13, 12, 11}

Each value shown here is a byte in the memory map. The rest of the mapping bytes are 'don't care' since they are not used.

Table 5.1 Mapping Example

	Column number (mapped Rx)				
Row number (mapped Tx)	**0 (Rx3)**	**1 (Rx0)**	**2 (Rx8)**	**3 (Rx1)**	**4 (Rx2)**
0 (Tx0)	5x5 Trackpad				
1 (Tx1)					
2 (Tx13)					
3 (Tx12)					
4 (Tx11)					

5.1.4 Rx / Tx Selections

On the IQS525 and IQS572, some Rxs can be configured to take on Tx functionality. The preferred option is to keep them as Rxs, but if more Txs are needed in the design, they can be configured as such in the *RxToTx* register. This allows for elongated trackpads or sliders to be implemented on the two devices. The corresponding Rx or Tx number is then used in the mapping registers to configure the order of the electrodes.

5.2 Trackpad Outputs

The channel count variation (deltas) and touch status outputs are used to calculate finger location data.

5.2.1 Number of Fingers

This gives an indication of the number of active finger inputs on the trackpad (*Number of fingers*).

5.2.2 Relative XY

If there is only one finger active, a *Relative X* and *Relative Y* value is available. This is a signed 2's complement 16-bit value. It is a delta of the change in X and Y, in the scale of the selected output resolution.

Note: Gestures also use these registers to indicate swipe, scroll and zoom parameters.

5.2.3 Absolute XY

For all the multi-touch inputs, the absolute finger position (*Absolute X/Y*), in the selected resolution (*Resolution X/Y*) of the trackpad, is available.

5.2.4 Touch Strength

This value (*Touch strength*) indicates the strength of the touch by giving a sum of all the deltas associated with the finger, and therefore varies according to the sensitivity setup of the sensors.

5.2.5 Area

The number of channels associated with a finger is provided here. This area is usually equal to or smaller than the number of touch channels under the finger.

5.2.6 Tracking / Identification

The fingers are tracked from one cycle to the next, and the same finger will be located in the same position in the memory map. The memory location thus identifies the finger.

5.3 Max Number of Multi-touches

The maximum number of allowed multi-touches is configurable (*Max multi-touches*) up to 5 points. If more than the selected value is sensed, a flag is set (*TOO_MANY_FINGERS*) and the XY data is cleared.

5.4 XY Resolution

The output resolution for the X and Y coordinates are configurable (*X/Y Resolution*). The on-chip algorithms use 256 points between each row and column. The resolution is defined as the total X and total Y output range across the complete trackpad.

5.5 Palm Rejection

A maximum finger size/area (*Palm reject threshold*) can be set up to allow for palm rejection or similar input suppression. This feature can be enabled or disabled (*PALM_REJECT*), and when a palm reject condition is sensed, a status flag will indicate this result (*PALM_DETECT*). All XY outputs are also suppressed during palm detection. Palm reject is latched on for the timeout period

(*Palm reject timeout*) to prevent erratic behaviour before and after the palm is seen. This timeout sets in increments of 32ms.

5.6 Stationary Touch

A stationary touch is defined as a point that does not move outside of a certain boundary within a specific time. This movement boundary or threshold can be configured (*Stationary touch movement threshold*), and is defined as a movement in either X or Y in the configured resolution.

The device will switch to *Idle-Touch* mode when a stationary point is detected, where a lower duty cycle can be implemented to save power in applications where long touches are expected.

If movement is detected, a status flag (*TP_MOVEMENT*) is set.

5.7 Multi-touch Finger Split

The position algorithm looks at areas (polygons) of touches, and calculates positional data from this. Two fingers in close proximity to each other could have areas touching, which would merge them incorrectly into a single point. A finger split algorithm is implemented to separate these merged polygons into multiple fingers. There is a *finger split aggression factor* which can be adjusted to determine how aggressive this finger splitting must be implemented. A value of '0' will not split polygons, and thus merge any fingers with touch channels adjacent (diagonally also) to each other.

5.8 XY Output Flip & Switch

By default, X positions are calculated from the first column (usually Rx0) to the last column. Y positions are by default calculated from the first row (usually Tx0) to the last row. The X and/or Y output can be flipped (*FLIP_X* / *FLIP_Y*), to allow the [0, 0] co-ordinate to be defined as desired. The X and Y axes can also be switched (*SWITCH_XY_AXIS*) allowing X to be the Txs, and Y to be along the Rxs.

5.9 XY Position Filtering

Stable XY position data is available from the IQS5xx due to two on-chip filters, namely the Moving Average (MAV) filter, and the Infinite Impulse Response (IIR) filter. The filters are applied to the raw positional data in the aforementioned order. It is recommended to keep both of the filters enabled for optimal XY data.

5.9.1 MAV Filter

If enabled (*MAV_FILTER*), raw XY points from the last two cycles are averaged to give the filter output.

5.9.2 IIR Filter

The IIR filter, if enabled (*IIR_FILTER*), can be configured to select between a dynamic and a static filter (*IIR_SELECT*).

The damping factor is calculated from the selected *Beta* as follows:

Damping factor = Beta / 256

5.9.2.1 Dynamic Filter

Relative to the speed of movement of a co-ordinate, the filter dynamically adjusts the amount of filtering (damping factor) performed. When fast movement is detected, and quick response is required, less filtering is done. Similarly when a co-ordinate is stationary or moving at a slower speed, more filtering can be applied.

The damping factor is adjusted depending on the speed of movement. Three of these parameters are adjustable to fine-tune the dynamic filter if required (*XY dynamic bottom beta* / *XY dynamic lower speed* / *XY dynamic upper speed*).

The speed is defined as the distance (in the selected resolution) travelled in one cycle (pixels/cycle).

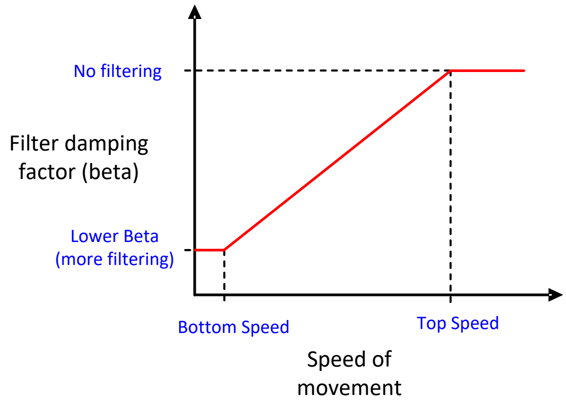


Figure 5.1 Dynamic Filter Parameters

5.9.2.2 Static Filter

Co-ordinates filtered with a fixed but configurable damping factor (*XY static beta*) are obtained when using the static filter. It is recommended that the dynamic filter is used due to the advantages of a dynamically changing damping value.

6 Gestures

The IQS5xx has an on-chip gesture recognition feature. The list of recognisable gestures includes:

- 1 finger gestures (*GESTURE_EVENTS_0*):
  - A single tap
  - A press and hold
  - Swipe X+
  - Swipe X-
  - Swipe Y+
  - Swipe Y-
- 2 finger gestures (*GESTURE_EVENTS_1*):
  - 2 simultaneous taps
  - Scroll
  - Zoom

Each single finger gesture can individually be enabled and disabled by setting or clearing the corresponding bits in the register *SINGLE_FINGER_GESTURES*. The multi finger gestures can be enabled and disabled via the register *MULTI_FINGER_GESTURES*.

All gestures are calculated relative to their starting coordinates, i.e., the first coordinate at which the touch was detected. Furthermore, if at any time during a gesture, more than the required number of touches is detected, the gesture will be invalidated.

6.1 Single Tap

The single tap gesture requires that a touch is made and released in the same location and within a short period of time. Some small amount of movement from the initial coordinate must be allowed to compensate for shift in the finger coordinate during the release. This bound is defined in register *Tap distance*, which specifies the maximum deviation in pixels the touch is allowed to move before a single tap gesture is no longer valid.

Similarly, the *Tap time* register defines the maximum duration in ms that will result in a valid gesture. That is, the touch should be released before the time period in *Tap time* is reached.

A valid single tap gesture will be reported (*SINGLE_TAP*) in the same processing cycle as the touch release was detected, and will be cleared on the next cycle. No movement will be reported in the relative XY registers (*Relative_X* and *Relative_Y*) during this gesture.

Since the gesture reports after the finger is removed, the location of the tap gesture is placed in the *Absolute X/Y* registers of finger 1 at this time. With *Number of fingers* set to 0, this will not look like an active finger, and is just a repetition of the location of the tap that has occurred for the main controller to utilise.

6.2 Press and Hold

The same register that defines the bounds for the single tap gesture (*Tap distance*) is used for the press and hold gesture. If the touch deviates more than the specified distance, the gesture is no longer valid.

However, if the touch remains within the given bound for longer that the period in ms, defined as the sum of the register values in *Tap time* and *Hold time*, a press and hold gesture will be reported (*PRESS AND HOLD*). The gesture will continue to be reported until the touch is released or if a second touch is registered.

No data will be reported in *Relative X* and *Relative Y* before the defined maximum hold period is reached, however, the relative data will be reported thereafter. This allows for features such as drag-n-drop.

6.3 Swipe (X-, X+, Y-, Y+)

All four swipe gestures work in the same manner, and are only differentiated in their direction. The direction is defined with respect to the origin (0, 0) of the trackpad, typically at Rx0, Tx0 (Channel 0). If the touch is moving away from the origin, it is considered a positive swipe (+) and if it is moving towards the origin, it is a negative swipe (-). Whether the swipe is of the type X or Y is defined by which axis the touch is moving approximately parallel to.

A swipe gesture event is only reported when a moving touch meets all three of the following conditions:

1. A minimum distance is travelled from its initial coordinates, as defined in pixels by the value in register *Swipe initial distance*.
2. The distance in (1) is covered within the time specified in *Swipe initial time* (in ms).
3. The angle of the swipe gesture, as determined by its starting coordinate and the coordinate at which conditions (1) and (2) were first met, does not exceed the threshold in *Swipe angle* with regards to at least 1 of the axes. The value in register *Swipe angle* is calculated as $64 \tan\theta$, where θ is the desired angle (in degrees).

The respective swipe gesture will be reported for 1 cycle (*SWIPE X-, X+, Y- Y+*) when all of these conditions are met. The relative distance travelled will be reported in registers *Relative X* and *Relative Y* throughout.

It is also possible to generate consecutive swipe gesture events during the same swipe gesture by defining the swipe gesture settings in registers *Swipe consecutive distance* [pixels] and *Swipe consecutive time* [ms]. Once the initial swipe gesture conditions are met as defined above, the parameters of *Swipe initial distance* [pixels] and *Swipe initial time* [ms] will be replaced with these. Also, the gesture engine will reset its properties, thus evaluating the current touch's movement as if its initial coordinate was at the point at which the previous swipe gesture was recognised and as if it first occurred at that point in time.

The consecutive events allow for the continuous stream of swipe events for a single action by the user. However, once the initial conditions are satisfied, the direction of the swipe gesture is fixed. For example, if a swipe X+ gesture is recognised by the engine, the consecutive swipe gestures will also be of type X+. And the 3rd condition will only be evaluated against the X axis.

In the case that only a single event is desired, the settings in *Swipe consecutive distance* can be set to its maximum value and *Swipe consecutive time* set to zero. This would make it impossible to meet these conditions on a standard trackpad.

6.4 2 Finger Tap

The simultaneous tap gesture simply requires two tap gestures to occur simultaneously. For this reason the gesture uses the same parameters (*Tap distance* and *Tap time*) as that of the tap gesture. It is also confined to the same conditions for the output to be reported (*2 FINGER TAP*).

6.5 Scroll

A scroll gesture is identified by two simultaneous and parallel moving touches. A scroll gesture will be reported (*SCROLL*) once the average distance travelled by the two touches in pixels exceeds the value stored in register *Scroll initial distance*. Thereafter, a scroll gesture will continuously be reported until one of the touches is released or if a zoom gesture is validated.

Similar to the swipe gestures, the scroll gestures are also bounded by a given angle to the axis (*Scroll angle*). The value in this register is calculated as $64\tan\theta$, where θ is the desired angle (in degrees). This condition is only enforced during the initial validation stage of the scroll gesture.

The direction of the scroll gesture is defined by the reported relative X (horizontal scroll) and Y (vertical scroll) data. For instance, a positive relative X value will correspond with the direction of a swipe X+ gesture. Unlike the swipe gestures, a scroll gesture may alternate between a positive and negative direction without requiring the validation of the initial conditions. However, switching between the axes will require the validation.

At any given stage during a scroll gesture, only the axis applicable to the gesture will have a non-zero value in its relative data register. For example, a scroll parallel to the X-axis will have a non-zero *Relative X* value and a zero *Relative Y* value. This value relates to the movement of the scroll gesture.

6.6 Zoom

Zoom gestures require two touches moving toward (zoom out) or away (zoom in) from each other. Similar to the scroll and swipe gestures, the zoom requires that an initial distance threshold in the register *Zoom initial distance* [pixels] is exceeded before a zoom gesture is reported (*ZOOM*). Thereafter, the register Zoom Consecutive Distance defines the distance threshold for each zoom event that follows the initial event. The direction/axis along which the two touches move is not relevant.

Switching from a zoom in to a zoom out gesture, or vice versa, requires that the initial conditions be met in the opposite direction before the switch can occur. Alternating between a zoom and a scroll gesture requires the same.

The size of each zoom event will be reported in *Relative X*, where the negative sign indicates a zoom out gesture and a positive sign a zoom in gesture.

6.7 Switching Between Gestures

For all single finger gestures it is necessary to release all touches before any new gesture can be made and validated. However, for the scroll and zoom gestures, it is possible to alternate between the gestures and their directions without releasing any touches.

A switch between multi-touch gestures includes

- Alternating between scroll axes
- Alternating between zoom in and out
- Going from a scroll to a zoom gesture
- Going from a zoom to a scroll gesture
- Releasing any one of the two touches
- Having more than 2 touches on the trackpad at any given moment.

A release of 1 of the touches will require a new touch be generated before any multi-touch gesture can be validated. The multi-touch gestures require 2, and only 2, touches at all time during the gesture.

7 Additional Features

7.1 Non-volatile Defaults

The designer can use the supplied GUI to easily configure the optimal settings for different setups. The design specific firmware is then exported by the GUI, and programmed onto the IQS5xx. These parameters are used as the default values after start-up, without requiring any setup from the master.

Two registers (*Export file version number*) are available so that the designer can label and identify the exported HEX file with the corresponding settings. This allows the master to verify if the device firmware has the intended configuration as required.

7.2 Automated Start-up

The IQS5xx is programmed with the trackpad application firmware, bundled with settings specifically configured for the current hardware as described in Section 7.1. After power-up the IQS5xx will automatically use the settings and configure the device accordingly.