IQS9150/IQS9151 DATASHEET

Multi-Function Trackpad IC: Proximity, Touch, Snap, Trackpad, and Gesture Functionality

1 Device Overview

The IQS9150/IQS9151 ProxSense® IC is a generic and configurable trackpad product aimed to be suitable for numerous design variations and requirements. This device has multitouch high-performance (linearity, accuracy, low-noise) trackpad outputs, integrated snap button options, and an on-chip gesture recognition engine. The IQS9150/IQS9151 features best in class sensitivity, signal-to-noise ratio and automatic tuning of electrodes. Furthermore there are user configurable virtual buttons, sliders, and wheels that can be superimposed onto the trackpad area, with easy-to-integrate virtual sensor outputs. Low power proximity detection allows extreme low power operation.

1.1 Main Features

- Highly flexible ProxSense® device
- Self-/Mutual-capacitive sensors configuration for device wake-up
- Ultra Low Power (ULP) wake-up on touch
- RF immunity
- Sensor flexibility:
  - Automatic sensor tuning for optimal sensitivity
  - Internal voltage regulator
  - On-chip noise filtering
  - Detection debounce and hysteresis
  - Wide range of capacitance detection
- I^{2}C communication interface with IRQ/RDY, up to Fast-Mode Plus (1 MHz)
- QFN52 (6×6×0.75 mm) – 0.40 mm pitch
- Wide input voltage supply range: 2.2 V to 3.5 V
- Wide operating temperature range: −40 °C to +85 °C
- Trackpad
  - Up to 7 fingers tracking
  - High resolution coordinate outputs
  - Fast response
  - Individual touch sensor
  - Snap dome detection
  - Integrated touch size output (area and strength) for touch integrity
  - Multi-finger gesture recognition engine
  - Electrode mapping for optimal PCB layout
  - Configurable coordinate resolution and orientation
  - Compatible with wide range of overlay materials and thicknesses
  - Compatible with multiple 1-and 2-layer sensor patterns
  - Adjustable sensing frequency offset for limiting potential interference
  - No calibration required - systems automatically compensated for mechanical & temperature changes
  - Virtual sensors:
    - Configurable virtual button, slider and wheel sensors
    - Change sensor locations and sizes without electrode changes required
    - Up to 16 virtual buttons
    - Up to 8 virtual sliders
    - Up to 4 virtual wheels

- Design and manufacturing support
  - Touch pattern layout drawing
  - Full FPC layout package (example & customised)
  - Test guide for touch pattern
  - RFI immunity design support
- Design simplicity
  - PC GUI software for debugging and obtaining optimal performance
  - Easily obtain setup defaults from GUI header file export
  - No production line calibration required
  - EEPROM compatibility for default settings storage for auto-startup

1.2 Applications

- Gaming controllers
- Headphones
- Notebooks
- Mobile Devices
- Tablet and notebook accessories
- Point-of-Sale (POS)
- Industrial and Specialised (Control panels, medical devices, aircraft cockpits)

1.3 System Overview

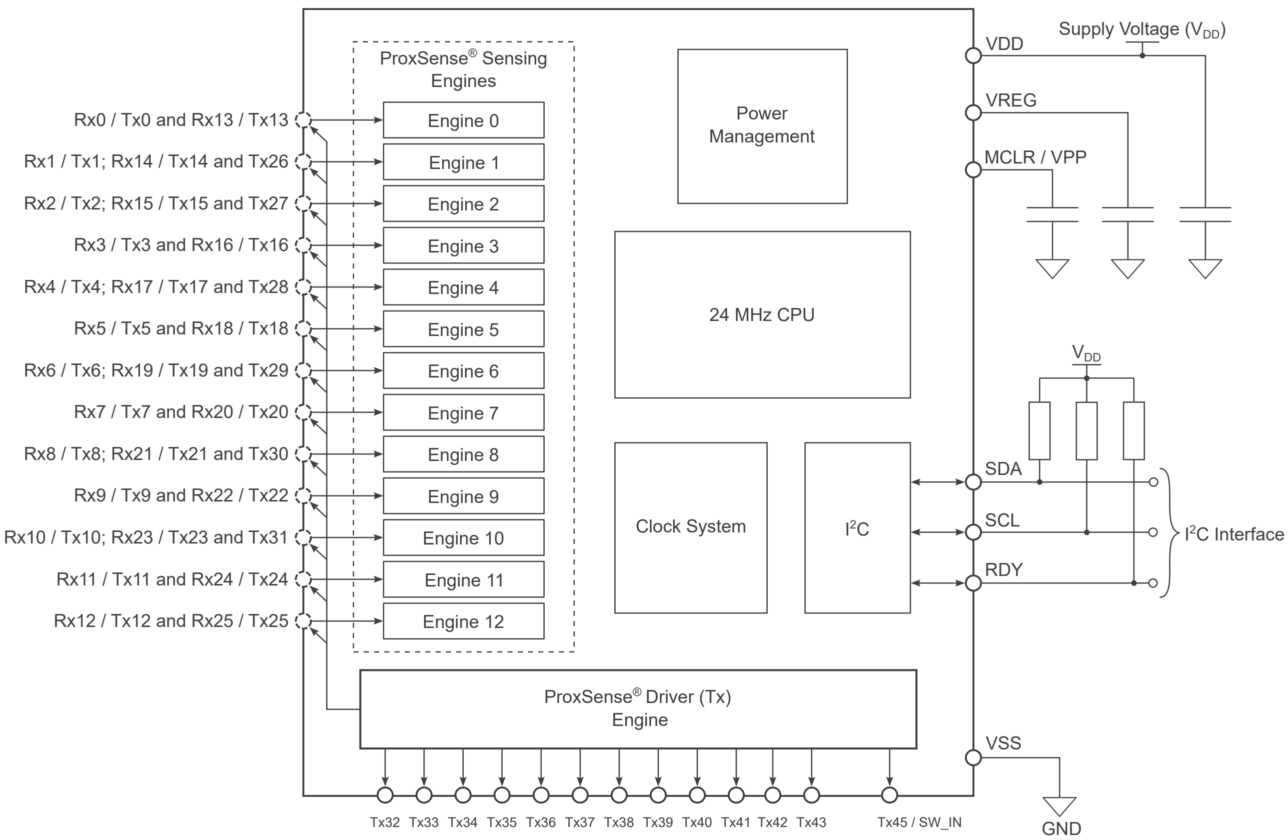


Figure 1.1: IQS9150/IQS9151 Block Diagram

1.4 Trackpad Size Summary

1.4.1 IQS9150

- Max Rxs: 26
- Max Txs: 22
- Max Channels: 506
- Max Trackpad Electrodes: 45
- Example Configurations:
  - Max Rectangular: 26 Rx x 19 Tx (494 channels, 45 electrodes)
  - Max Square: 23 Rx x 22 Tx (506 channels, 45 electrodes)

1.4.2 IQS9151

- Max Rxs: 13
- Max Txs: 22
- Max Channels: 156
- Max Trackpad Electrodes: 25
- Example Configurations:
  - Max Rectangular: 10 Rx x 15 Tx (150 channels, 25 electrodes)
  - Max Square: 13 Rx x 12 Tx (156 channels, 25 electrodes)

Contents

2 QFN52 Pinout

2.1 IQS9150

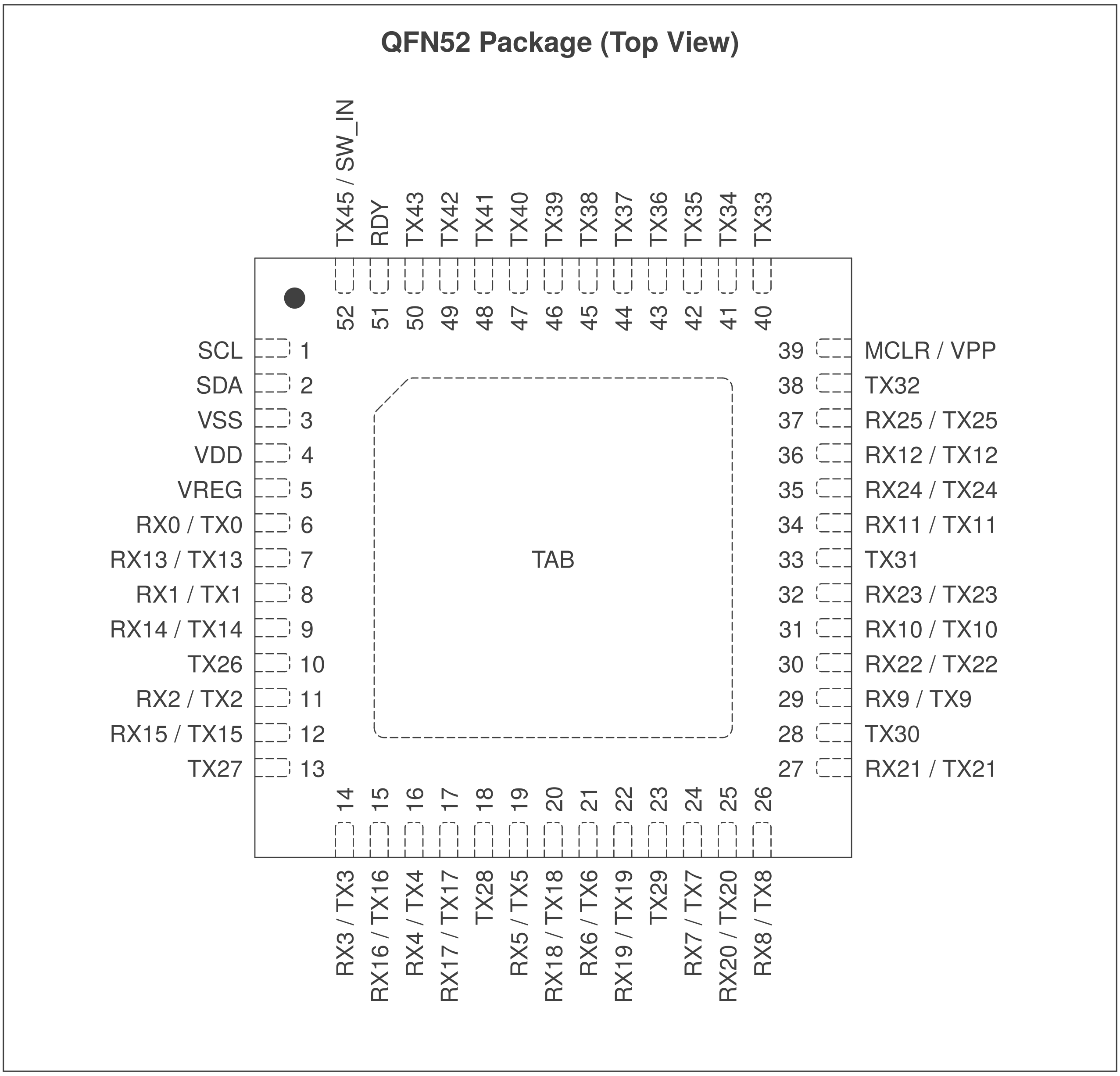


Figure 2.1: IQS9150 QFN52 Pinout

2.2 IQS9151

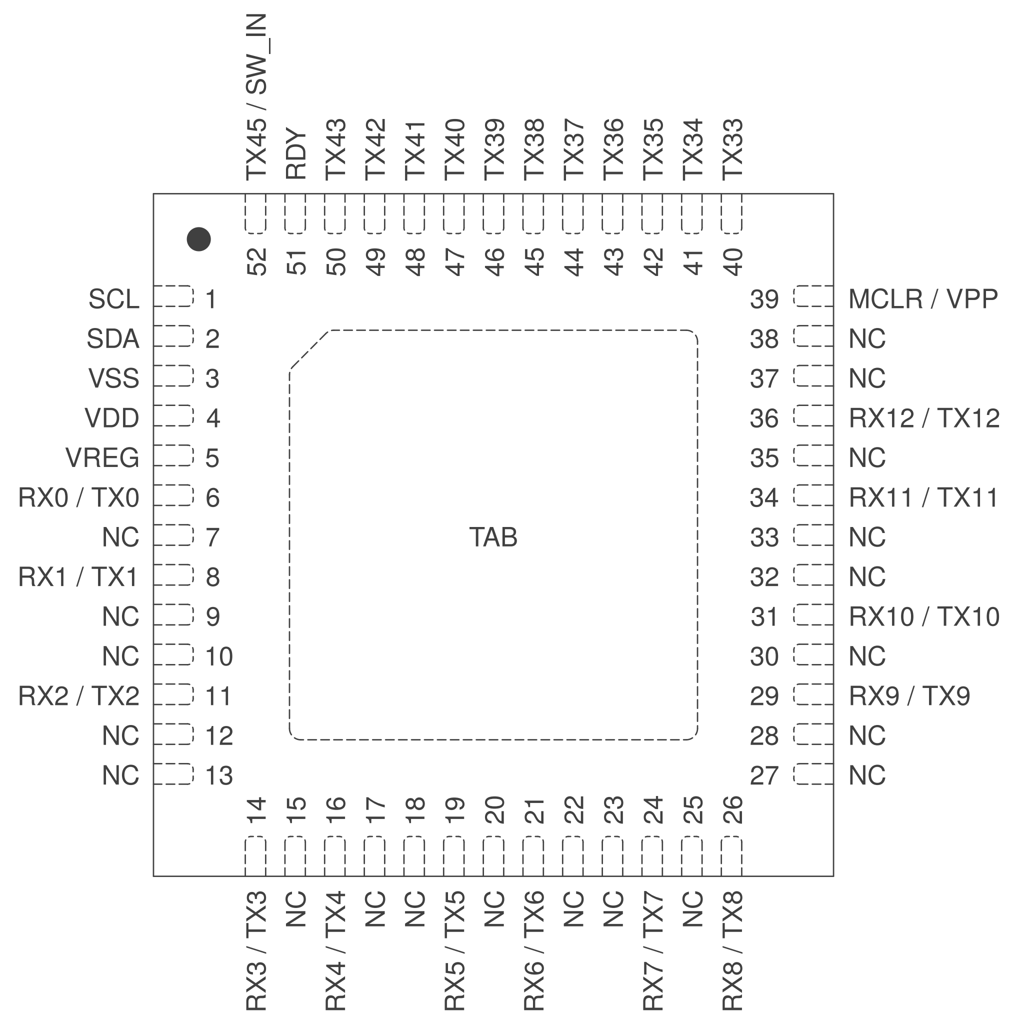


Figure 2.2: IQS9151 QFN52 Pinout

2.3 IQS9150/IQS9151 Pinout Descriptions

Table 2.1: QFN52 Pin Descriptions

Pin	IQS9150	IQS9151	Type[i]	Function	Description
1	SCL	SCL	I/O	I²C	I²C data
2	SDA	SDA	I/O	I²C	I²C clock
3	VSS	VSS	P	Power	Analog/digital ground
4	VDD	VDD	P	Power	Power supply input voltage
5	VREG	VREG	P	Power	Internally-regulated supply voltage
6	RX0 / TX0	RX0 / TX0	I/O	ProxSense®	Receiver or transmitter electrode

Continued on next page...

Table 2.1: QFN52 Pin Descriptions (Continued)

Pin	IQS9150	IQS9151	Type[i]	Function	Description
7	RX13 / TX13	NC	I/O	ProxSense®	Receiver or transmitter electrode
8	RX1 / TX1	RX1 / TX1	I/O	ProxSense®	Receiver or transmitter electrode
9	RX14 / TX14	NC	I/O	ProxSense®	Receiver or transmitter electrode
10	TX26	NC	O	ProxSense®	Transmitter electrode
11	RX2 / TX2	RX2 / TX2	I/O	ProxSense®	Receiver or transmitter electrode
12	RX15 / TX15	NC	I/O	ProxSense®	Receiver or transmitter electrode
13	TX27	NC	O	ProxSense®	Transmitter electrode
14	RX3 / TX3	RX3 / TX3	I/O	ProxSense®	Receiver or transmitter electrode
15	RX16 / TX16	NC	I/O	ProxSense®	Receiver or transmitter electrode
16	RX4 / TX4	RX4 / TX4	I/O	ProxSense®	Receiver or transmitter electrode
17	RX17 / TX17	NC	I/O	ProxSense®	Receiver or transmitter electrode
18	TX28	NC	O	ProxSense®	Transmitter electrode
19	RX5 / TX5	RX5 / TX5	I/O	ProxSense®	Receiver or transmitter electrode
20	RX18 / TX18	NC	I/O	ProxSense®	Receiver or transmitter electrode
21	RX6 / TX6	RX6 / TX6	I/O	ProxSense®	Receiver or transmitter electrode
22	RX19 / TX19	NC	I/O	ProxSense®	Receiver or transmitter electrode
23	TX29	NC	O	ProxSense®	Transmitter electrode
24	RX7 / TX7	RX7 / TX7	I/O	ProxSense®	Receiver or transmitter electrode
25	RX20 / TX20	NC	I/O	ProxSense®	Receiver or transmitter electrode
26	RX8 / TX8	RX8 / TX8	I/O	ProxSense®	Receiver or transmitter electrode
27	RX21 / TX21	NC	I/O	ProxSense®	Receiver or transmitter electrode
28	TX30	NC	O	ProxSense®	Transmitter electrode
29	RX9 / TX9	RX9 / TX9	I/O	ProxSense®	Receiver or transmitter electrode
30	RX22 / TX22	NC	I/O	ProxSense®	Receiver or transmitter electrode
31	RX10 / TX10	RX10 / TX10	I/O	ProxSense®	Receiver or transmitter electrode
32	RX23 / TX23	NC	I/O	ProxSense®	Receiver or transmitter electrode
33	TX31	NC	O	ProxSense®	Transmitter electrode
34	RX11 / TX11	RX11 / TX11	I/O	ProxSense®	Receiver or transmitter electrode
35	RX24 / TX24	NC	I/O	ProxSense®	Receiver or transmitter electrode
36	RX12 / TX12	RX12 / TX12	I/O	ProxSense®	Receiver or transmitter electrode
37	RX25 / TX25	NC	I/O	ProxSense®	Receiver or transmitter electrode
38	TX32	NC	O	ProxSense®	Transmitter electrode
39	MCLR / VPP	MCLR / VPP	I	GPIO	Master clear pin used for HW reset (active low), and VPP input for OTP
40	TX33	TX33	O	ProxSense®	Transmitter electrode
41	TX34	TX34	O	ProxSense®	Transmitter electrode
42	TX35	TX35	O	ProxSense®	Transmitter electrode
43	TX36	TX36	O	ProxSense®	Transmitter electrode
44	TX37	TX37	O	ProxSense®	Transmitter electrode
45	TX38	TX38	O	ProxSense®	Transmitter electrode
46	TX39	TX39	O	ProxSense®	Transmitter electrode
47	TX40	TX40	O	ProxSense®	Transmitter electrode
48	TX41	TX41	O	ProxSense®	Transmitter electrode
49	TX42	TX42	O	ProxSense®	Transmitter electrode

Continued on next page...

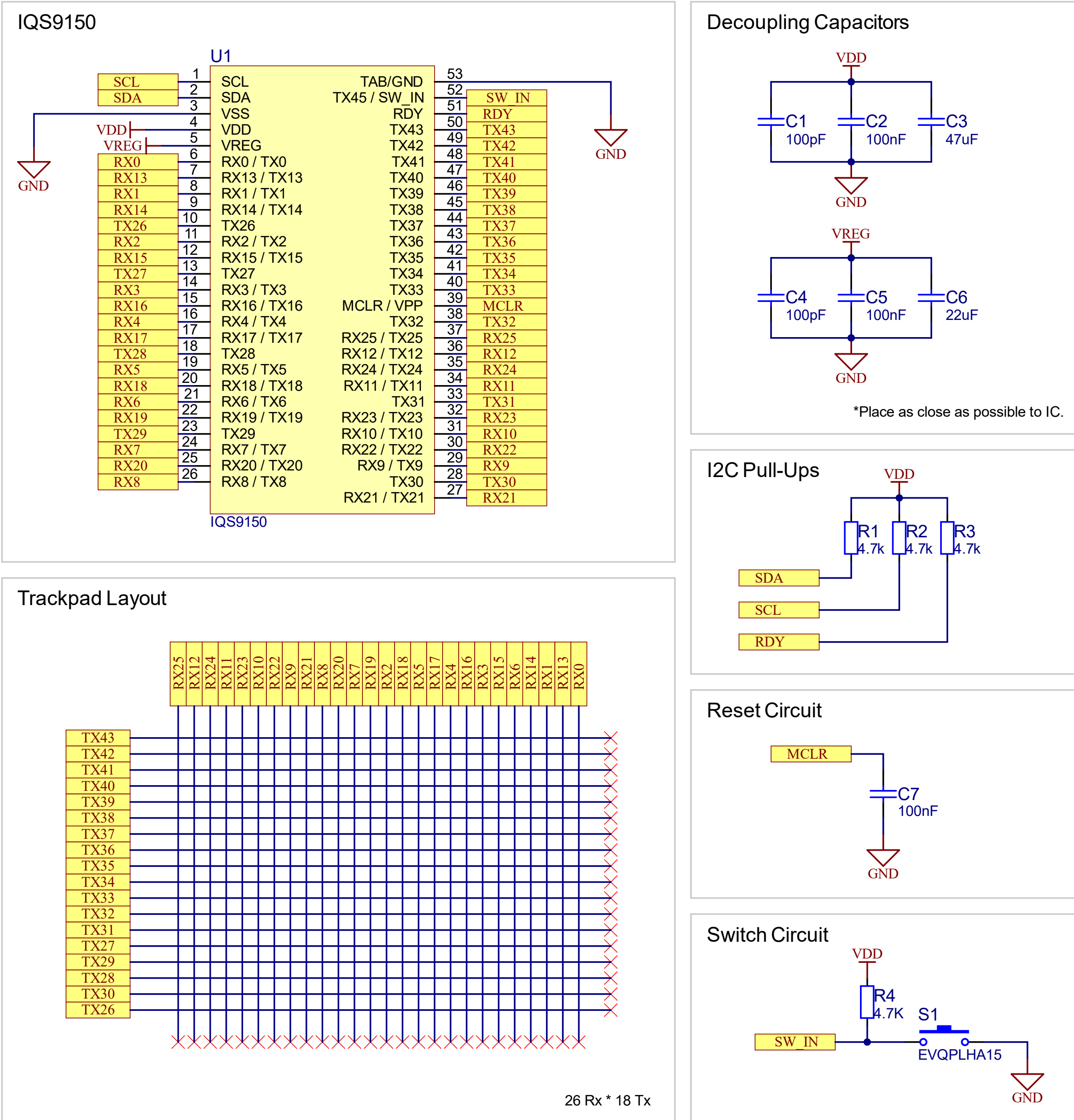
Table 2.1: QFN52 Pin Descriptions (Continued)

Pin	IQS9150	IQS9151	Type[i]	Function	Description
50	TX43	TX43	O	ProxSense®	Transmitter electrode
51	RDY	RDY	O	GPIO	Ready pin indicates communication window (active low)
52	TX45 / SW_IN	TX45 / SW_IN	I/O	GPIO	Transmit electrode or switch input

[i] Pin Types: I = Input, O = Output, I/O = Input or Output, P = Power

3 Reference Schematics

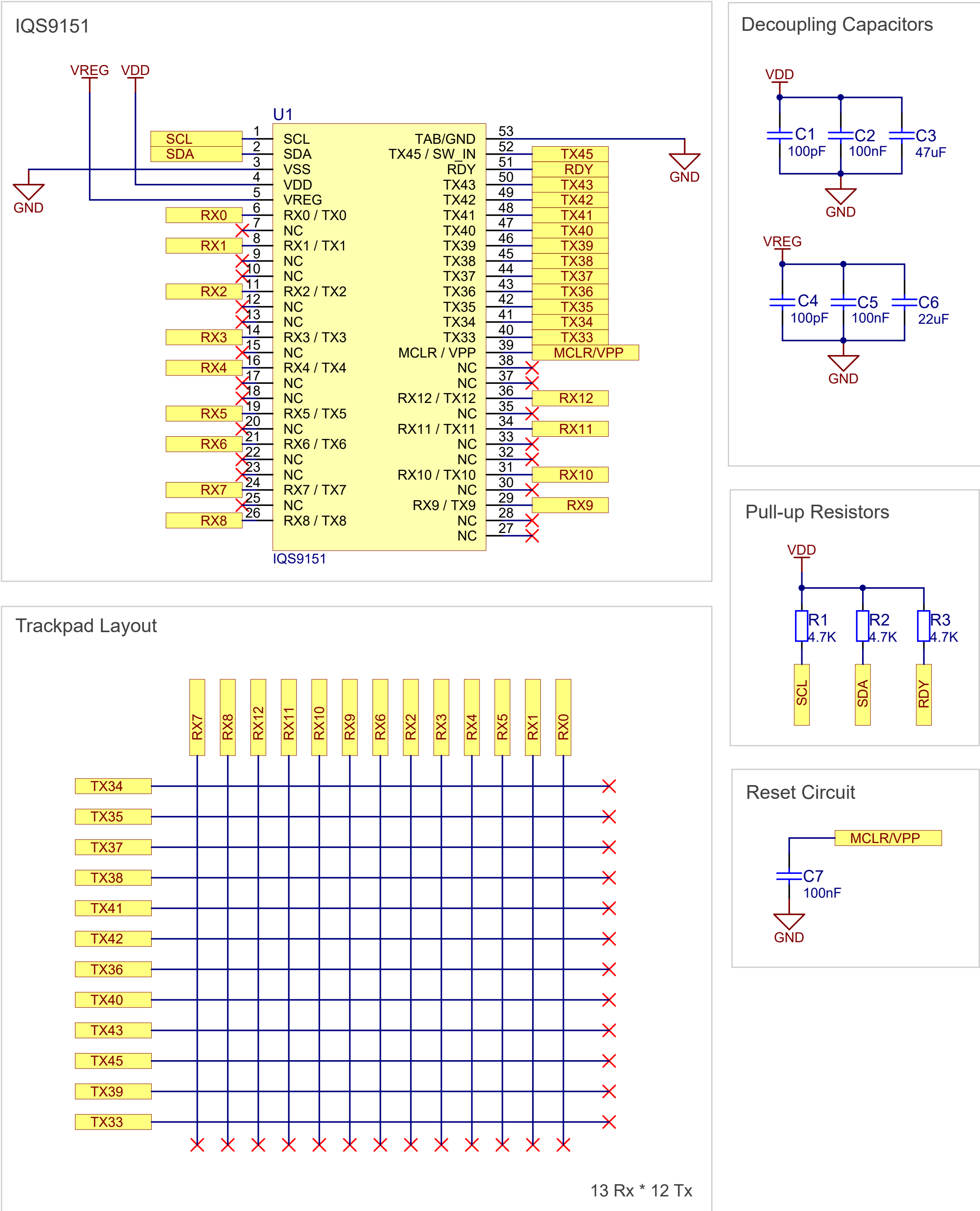
3.1 IQS9150



* Schematic subject to change without notice

Figure 3.1: IQS9150 Reference Schematic for 26 by 18 Trackpad Layout (468 Channels)

3.2 IQS9151



* Schematic subject to change without notice

Figure 3.2: IQS9151 Reference Schematic for 13 by 12 Trackpad Layout (156 Channels)

4 Electrical Characteristics

4.1 Absolute Maximum Ratings

Table 4.1: Absolute Maximum Ratings

Symbol	Rating	Min	Max	Unit
V_{DD}	Voltage applied at VDD pin (referenced to VSS)	−0.3	3.6	V
V_{IN}	Voltage applied to any ProxFusion® pin (referenced to VSS)	−0.3	V_{REG}	V
	Voltage applied to any other pin (referenced to VSS)	−0.3	V_{DD} + 0.3 (3.6 V max)	V
T_{stg}	Storage temperature	−40	85	°C

4.2 General Operating Conditions

Table 4.2: General Operating Conditions

Symbol	Parameter	Condition	Typ	Unit
F_{CLK}	Master clock frequency	F_{CLK} = 14 MHz	14	MHz
		F_{CLK} = 20 MHz	20	
		F_{CLK} = 24 MHz	24	
F_{PROX}	ProxFusion® engine clock frequency		16	MHz
V_{REG}	Internally-regulated supply output	F_{CLK} = 14 MHz	1.53	V
		$F_{CLK} \geq$ 20 MHz	1.80	

4.3 Recommended Operating Conditions

Table 4.3: Recommended Operating Conditions

Symbol	Parameter	Condition	Min	Rec[a]	Max	Unit
V_{DD}	Standard operating voltage, applied at VDD pin	F_{CLK} = 14 MHz	1.71		3.6	V
		$F_{CLK} \geq$ 20 MHz	2.2			
T_A	Operating free-air temperature		−20		85	°C
C_{VDD}	Recommended capacitor at VDD		C_{VREG}	2×C_{VREG}		µF
C_{VREG}	Recommended external buffer capacitor at VREG (ESR $\leq$ 200 mΩ)		10[b]	22	88	µF

[a] Recommended value
[b] Absolute minimum allowed capacitance value is 4.7 µF, after derating for voltage, temperature, and worst-case tolerance.

4.4 ProxSense® Electrical Characteristics

Table 4.4: Recommended Operating Conditions for ProxFusion® Pins

Symbol	Parameter	Min	Typ	Max	Unit
$Cx_{SELF-VSS}$	Capacitance between ground and external electrodes, in self-capacitance mode	1		400[a]	pF
$Cm_{CTx-CRx}$	Capacitance between transmitting and receiving electrodes, in mutual-capacitance mode	0.2		10[a]	pF
$Cp_{CRx-VSS}$	Capacitance between ground and external electrodes, in mutual-capacitance mode				pF
	F_{xfer} = 1 MHz			100[a]	
	F_{xfer} = 4 MHz			20[a]	
$\frac{Cp_{CRx-VSS}}{Cm_{CTx-CRx}}$	Capacitance ratio for optimal SNR in mutual-capacitance mode	1		50	
R_{CRx}, R_{CTx}	Series in-line resistance of Tx and Rx pins in mutual-capacitance mode	0		0.5[b,c]	kΩ
$R_{Cx(SELF)}$	Series in-line resistance of self-capacitance electrodes	0		1[c]	kΩ

[a] $R_{Cx} = 0\,\Omega$
[b] Series resistance of up to 500 Ω is recommended to prevent received and emitted EMI effects. Typical resistance also adds additional ESD protection.
[c] Series resistance limit is a function of F_{xfer} and the circuit time constant, RC. $R_{max} \times C_{max} = 1/(10 \times F_{xfer})$, where C is the pin capacitance to VSS.

4.5 ESD Rating

Table 4.5: ESD Rating

			Value	Unit
$V_{(ESD)}$	Electrostatic discharge voltage	Human-body model (HBM), per ANSI/ESDA/JEDEC JS-001[a]	±2000	V

[a] JEDEC document JEP155 states that 500 V HBM allows safe manufacturing with a standard ESD control process. Pins listed as ±2000 V may actually have higher performance.

4.6 Reset Levels

Table 4.6: Reset Levels

Parameter		Min	Typ	Max	Unit
VDD	Power-up (Reset trigger) – slope > 100 V/s			1.65	V
	Power-down (Reset trigger) – slope < −100 V/s	0.9			

4.7 MCLR Pin Levels and Characteristics

Table 4.7: MCLR Pin Characteristics

Parameter		Min	Typ	Max	Unit
V_{IL}	MCLR input low level voltage	$V_{SS} - 0.3$		$0.25 \times V_{DD}$	V
V_{IH}	MCLR input high level voltage	$0.75 \times V_{DD}$		$V_{DD} + 0.3$	V
R_{PU}	MCLR pull-up equivalent resistor		210		kΩ
t_{Trig}	MCLR input pulse width – ensure trigger	250			ns

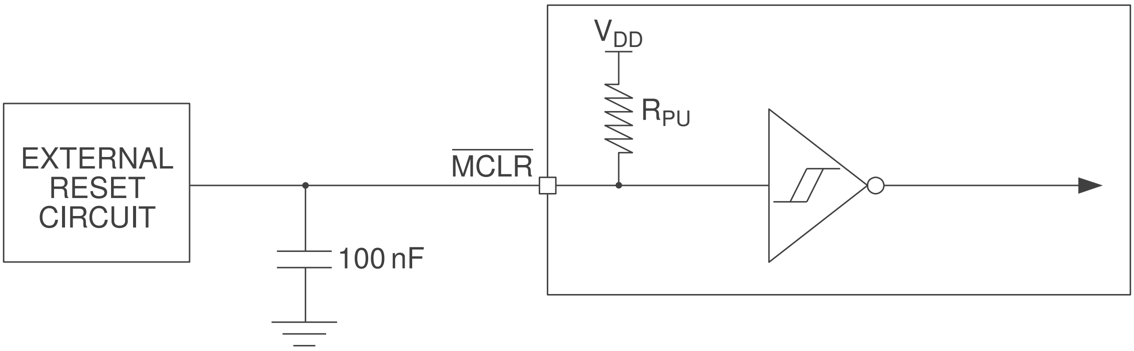


Figure 4.1: MCLR Pin Diagram

4.8 Digital I/O Characteristics

Table 4.8: Digital I/O Characteristics

Parameter		Test Conditions	Min	Max	Unit
V_{OL}	SDA & SCL output low voltage	$I_{sink} = 20\,mA$		0.3	V
	GPIO output low voltage	$I_{sink} = 10\,mA$		0.15	V
V_{OH}	Output high voltage	$I_{source} = 20\,mA$	$V_{DD} - 0.2$		V
V_{IL}	Input low voltage		$V_{SS} - 0.3$	$0.3 \times V_{DD}$	V
V_{IH}	Input high voltage		$0.7 \times V_{DD}$	$V_{DD} + 0.3$	V
I_{GPIO}	Output current sunk by any GPIO pin			10	mA
	Output current sourced by any GPIO pin			20	
C_b	SDA & SCL bus capacitance			550	pF

4.9 I²C Characteristics

Table 4.9: I²C Characteristics

Parameter		Min	Max	Unit
f_{SCL}	SCL clock frequency		1000	kHz
$t_{HD,STA}$	Hold time (repeated) START condition	0.26		µs
t_{LOW}	LOW period of the SCL clock	0.5		µs
t_{HIGH}	HIGH period of the SCL clock	0.26		µs
$t_{SU,STA}$	Set-up time for a repeated START condition	0.26		µs
$t_{HD,DAT}$	Data hold time	0		ns
$t_{SU,DAT}$	Data set-up time	50		ns
$t_{SU,STO}$	Set-up time for STOP condition	0.26		µs
t_{BUF}	Bus free time between a STOP and START condition	0.5		µs
t_{SP}	Pulse duration of spikes suppressed by input filter	0	50	ns

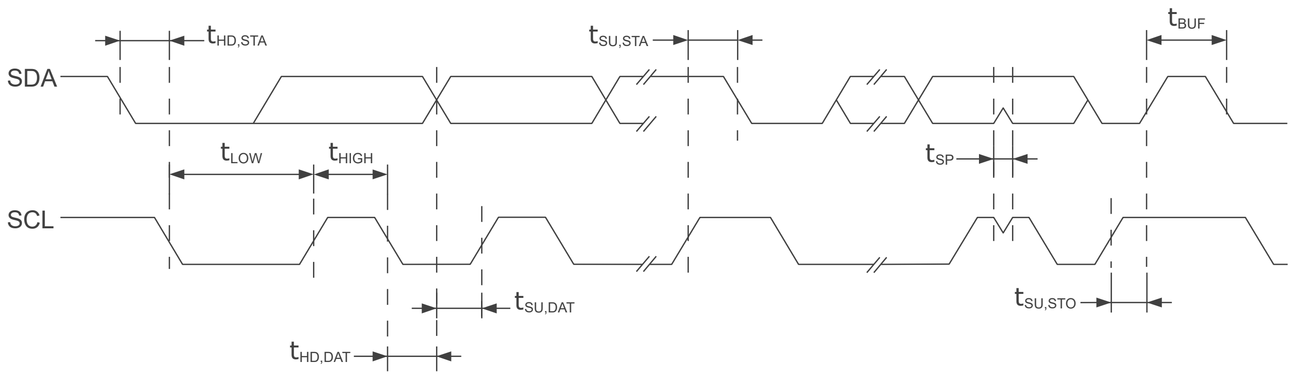


Figure 4.2: I²C Timing Diagram

4.10 Current Consumption

The current consumption of the IQS9150/IQS9151 is highly dependent on the specific parameters configured during initialisation. Therefore, Table 4.10 provided below serves as an illustration of the expected power consumption for similar configurations[i]. All measurements are taken without I²C communication, since the amount of data read via I²C will impact the current consumption and sampling period of the device. The device configurations outlined in the table represent practical setups commonly encountered in various applications.

Note: The 8 x 8 and the 10 x 15 column are applicable to both the IQS9150 and IQS9151.

- Trackpad Configuration:
  - Trackpad ATI Target = 250
  - Main Oscillator Selection = 24 MHz
  - Trackpad Conversion Frequency = 2.50 MHz
  - SH Bias = 5 µA

Table 4.10: Typical Current Consumption for a Range of Trackpad Sizes

Mode	Sampling Period [ms]	Current Consumption [mA]				
		8 x 8	10 x 15	20 x 14	23 x 22	26 x 19
Active[a]	10	2.2	3.7	5.5	9.1[c]	9.5[c]
Active[a]	15	1.5	2.6	3.7	6.3	6.2
Active[a]	20	1.1	1.9	2.8	4.8	4.7
Idle[b]	30	0.7	1.2	1.8	3.1	3.1
Idle[b]	50	0.4	0.7	1.1	1.9	1.9
Idle[b]	100	0.2	0.4	0.6	0.9	0.9

[a] Continuous movement in touch with a single 8 mm stylus.
[b] No touches in Idle Mode.
[c] Chosen sampling period was not achieved due to specific configuration.

Note: The LP1/2 mode uses the ALP channel, and the trackpad remains inactive during the LP modes. No touches are made in LP1/2 while measurements are taken.

- ALP Configuration 1:
  - ALP ATI Target = 300
  - Main Oscillator Selection = 24 MHz
  - Trackpad Conversion Frequency = 1.50 MHz
  - ALP Sensing Method = Self-capacitive
  - Active Tx Shield Enabled
  - All Rx and Tx electrodes enabled
  - LP1 Auto-Prox Disabled
  - LP2 Auto-Prox Enabled (Auto-Prox Cycles = 32)

[i] These measurements are based on bench testing and have not been characterised over large volumes.

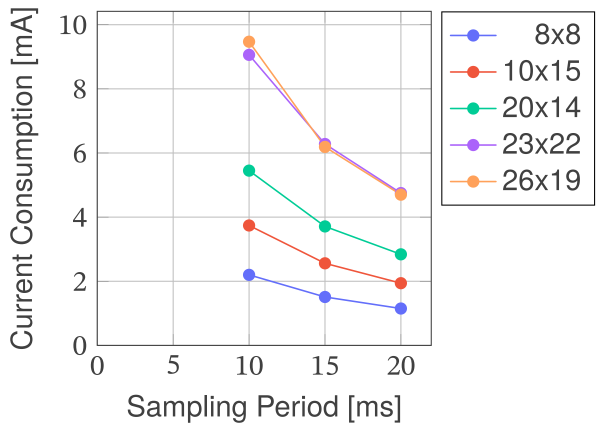
Table 4.11: Typical Current Consumption for Trackpads in ALP Mode with Configuration 1

Mode	Sampling Period [ms]	Current Consumption [µA]				
		8 x 8	10 x 15	20 x 14	23 x 22	26 x 19
ALP LP1	50	115	116	122	123	125
ALP LP1	100	59	60	63	63	64
ALP LP1	150	40	40	43	43	44
ALP LP1	200	31	31	33	33	33
ALP LP2	50	22	24	31	32	34
ALP LP2	100	12	13	16	17	18
ALP LP2	150	9	9	12	12	13
ALP LP2	200	7	8	9	10	10
ALP LP2	300	5	6	7	7	7
ALP LP2	400	5	5	6	6	6
ALP LP2	500	4	4	5	5	5

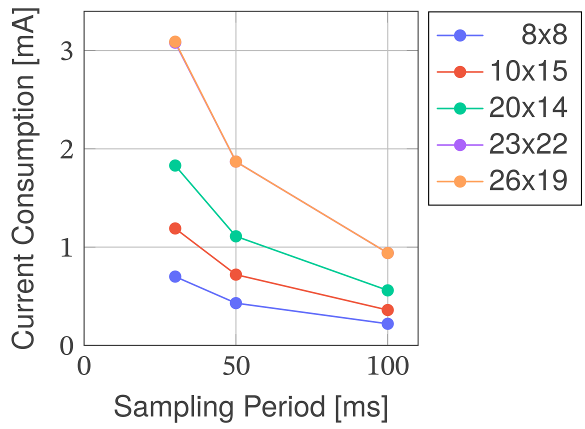
- ALP Configuration 2:
  - ALP ATI Target = 300
  - Main Oscillator Selection = 24 MHz
  - Trackpad Conversion Frequency = 1.50 MHz
  - ALP Sensing Method = Mutual-capacitive
  - SH Bias = 5 µA
  - Every alternate Rx and Tx electrode enabled
  - LP1 Auto-Prox Disabled
  - LP2 Auto-Prox Enabled (Auto-Prox Cycles = 32)

Table 4.12: Typical Current Consumption for Trackpads in ALP Mode with Configuration 2

Mode	Sampling Period [ms]	Current Consumption [µA]				
		8 x 8	10 x 15	20 x 14	23 x 22	26 x 19
ALP LP1	50	122	128	143	152	143
ALP LP1	100	62	65	73	78	73
ALP LP1	150	42	44	49	53	50
ALP LP1	200	32	34	38	40	38
ALP LP2	50	33	43	58	73	62
ALP LP2	100	18	22	30	38	32
ALP LP2	150	12	16	21	26	22
ALP LP2	200	10	12	16	19	17
ALP LP2	300	7	9	12	15	12
ALP LP2	400	5	7	9	10	9
ALP LP2	500	5	6	7	9	8

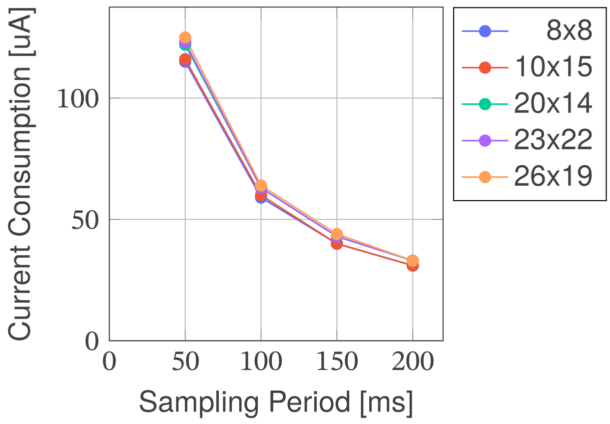


(a) Active Mode

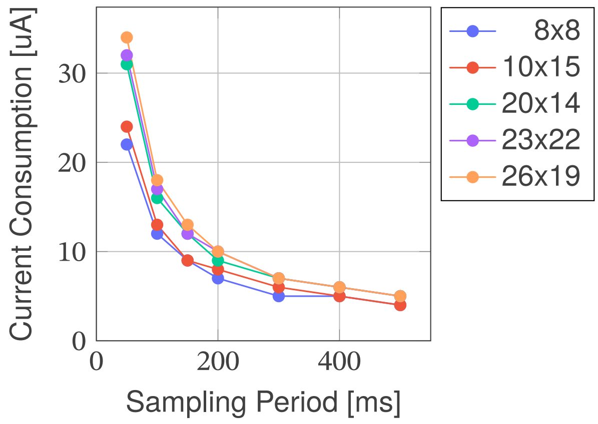


(b) Idle Mode

Figure 4.3: Typical Trackpad Current Consumption for a Range of Trackpad Sizes

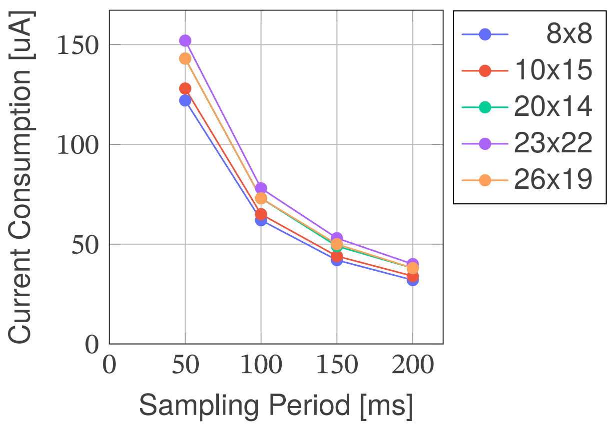


(a) ALP LP1 Mode - Config 1

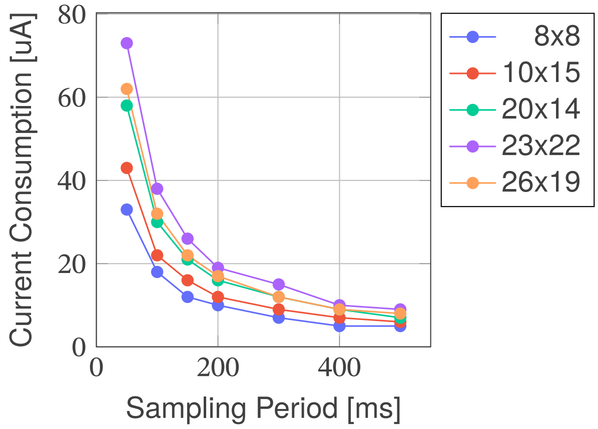


(b) ALP LP2 Mode - Config 1

Figure 4.4: Typical ALP Current Consumption for a Range of Trackpad Sizes



(a) ALP LP1 Mode - Config 2



(b) ALP LP2 Mode - Config 2

Figure 4.5: Typical ALP Current Consumption for a Range of Trackpad Sizes

5 ProxSense® Module

The IQS9150/IQS9151 contains a ProxSense® module that uses patented technology to measure and process the capacitive sensor data. The channel touch and snap are the primary sensor outputs. These are processed further to provide secondary trackpad outputs, including finger position, finger size, and on-chip gesture recognition.

5.1 ProxSense® Engine Consideration

The IQS9150 has 13 ProxSense® engines. Rx0 – Rx12 sensors are sensed simultaneously for trackpad sensing, and Rx13 – Rx25 simultaneously after that. Thus, if all 26 Rxs are enabled/used in a trackpad design, each Tx will consist of two cycles of sensing conversions, firstly the Rx0 – Rx12 sensors, and then the Rx13 – Rx25 sensors. The IQS9151 only has 13 Rxs, and therefore only one sensing cycle per Tx is needed.

There is thus no need for allocating channels into sensing cycles.

It is, however, advised that if 13 or fewer Rxs are used, that they are all allocated to the Rx0 - Rx12 group; this will allow for faster sampling periods since there will only be one conversion cycle needed per Tx. If more than 13 are used, it is advised to balance them between the two groups.

5.2 Trackpad Channels

On a trackpad sensor (typically a diamond-shaped pattern), each intersection of an Rx and Tx row/-column forms a mutual-capacitive sensing element, which is referred to as a *channel*. Each channel has an associated count value, reference value, touch status, and snap status (if enabled).

5.2.1 Channel Numbers

Trackpad channels are numbered from 0 to $((\textit{Total Rxs} \times \textit{Total Txs}) - 1)$. They are assigned from the top-left corner, first along the Rxs before stepping to the next Tx. The channel number must be known for some settings, such as configuring snap channels. Table 5.1 below is an example of an 8x12 trackpad's channel numbers:

Table 5.1: Channel Number Assignment

	Rx0 (Col 0)	Rx1 (Col 1)	Rx2 (Col 2)	Rx3 (Col 3)	Rx4 (Col 4)	Rx5 (Col 5)	Rx6 (Col 6)	Rx7 (Col 7)
Tx21 (Row 0)	0	1	2	3	4	5	6	7
Tx30 (Row 1)	8	9	10	11	12	13	14	15
Tx9 (Row 2)	16	17	18	19	20	21	22	23
Tx22 (Row 3)	24	25	26	27	28	29	30	31
Tx10 (Row 4)	32	33	34	35	36	37	38	39
Tx23 (Row 5)	40	41	42	43	44	45	46	47
Tx31 (Row 6)	48	49	50	51	52	53	54	55
Tx11 (Row 7)	56	57	58	59	60	61	62	63
Tx24 (Row 8)	64	65	66	67	68	69	70	71
Tx12 (Row 9)	72	73	74	75	76	77	78	79
Tx25 (Row 10)	80	81	82	83	84	85	86	87
Tx32 (Row 11)	88	89	90	91	92	93	94	95

5.3 Alternate Low-Power Channel (ALP)

To provide lower power consumption in LP1 and LP2, activity on the trackpad can be monitored by configuring an ALP channel (single combination sensor) instead of sensing the individual channels as done in Active/Idle modes. To utilise this ALP channel, it needs to be enabled in *ALP Setup*. If however it is not enabled, then the normal trackpad sensing will remain in LP1 and LP2. Since the alternate channel is processed as only a single channel, much less processing is done, allowing for lower overall power consumption. This channel has a lot of setup flexibility:

- *ALP Setup:*
  - *Count value filtering:* Gives reliable proximity detection in noisy environments.
  - *ALP sensing method:* Mutual-capacitive or self-capacitive.
  - *Rx electrode selection:* Which Rxs are active during ALP conversions.
- *ALP Tx Enable:*
  - *Tx electrode selection:* Which Txs are active during ALP conversions.
- *Other Settings:*
  - *Auto-prox:* Autonomous sensing cycles while the core is asleep, giving further power savings but similar wake-up capability.

5.4 Count Value

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

5.4.1 Trackpad Count Values

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

5.4.2 ALP Count Values

A count value will be obtained from all enabled Rxs in the ALP sensor. The combined count values from all engines will form the ALP channel counts. To reduce processing time (and thus decrease current consumption), the measurements are added together (*ALP Channel Count*) and processed as a single channel. A count value filter is implemented on this channel to give stable proximity output for system wake-up from low-power mode. It is recommended to leave this count filter enabled in the *ALP Setup* register. The amount of filtering can be modified in *ALP Count Filter Beta* if required. This beta is used as follows to determine the damping factor of the filter:

$$\text{Count damping factor} = \frac{(8 \times \text{Beta} - 7)}{2048}$$

If the beta is small, the filtering is stronger (filtered count follows raw count slower), and if the beta is larger, the filtering is weaker (filtered count follows raw count faster).

5.4.3 Trackpad Delta Value

The signed delta values (*Trackpad Delta Values*) are simply:

$$\text{Delta} = \text{Count - Reference}$$

5.5 Reference Value/Long-Term Average (LTA)

User interaction is detected by comparing the measured count values to some reference value. The reference value/LTA of a sensor is slowly updated to track changes in the environment and is not updated during user interaction.

5.5.1 Trackpad References

The *Trackpad Reference Values* are a snapshot (identical to a reseed) of the count value, stored during a time of no user activity, and thus are a non-affected reference. The trackpad reference values are only updated from LP1 and LP2 mode when modes are managed automatically, where no user interaction is assumed. Thus, if the system is controlled manually, the reference must also be managed and updated manually by the host (not recommended).

The reference value is updated or refreshed according to a configurable interval (*Reference Update Time*), in seconds. The reference update time has a maximum setting of *60 seconds*.

5.5.2 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 Channel LTA*. The LTA tracks the environment closely for accurate comparisons to the measured count value and to allow for small proximity deviations to be sensed. The speed of LTA tracking can be adjusted in the *ALP LTA Filter Beta* registers. There is a beta for LP1 and LP2. This is to allow different settings for different sampling periods so that the speed of LTA tracking can remain the same. These beta settings are used in the same way as for the counts; see Section 5.4.2.

5.5.3 Reseed

Since the *Reference* (or *LTA* for ALP channel) is critical for the device to operate correctly, there could be known events or situations that 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 *TP Reseed* or *ALP Reseed* in the *System Control* register.

5.6 Channel Outputs

5.6.1 Trackpad Touch Output

The trackpad touch output *Touch Status* is set when a channel's count value increases by more than the selected threshold.

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

$$\text{Threshold} = \text{Reference} \times \left(1 + \frac{\text{Touch Set/Clear Threshold Multiplier}}{128}\right)$$

where *Multiplier* is an 8-bit unsigned value for both the *Touch Set Threshold Multiplier* and *Touch Clear Threshold Multiplier*, allowing a hysteresis to provide improved touch detection. A smaller fraction will thus be a more sensitive threshold.

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

$$\text{Adjusted Multiplier} = \text{Set/Clear Threshold Multiplier} + \text{Individual Threshold Adjustment}$$

5.6.2 Trackpad Snap Output

When adding a metal snap-dome overlay to the trackpad pattern, an additional snap output is available in the *Snap Status* register. 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 in the *Trackpad Snap Channel Enable* register 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.

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 the *Info Flags* register 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* (in seconds). A setting of '0' will never reseed. The timeout is reset if any snap is set or cleared.

The trackpad snap output *Snap Status* is set when a channel's snap count value decreases by more than the selected threshold.

The threshold for a snap channel is determined as follows:

$$\text{Threshold} = \text{Reference - Snap Threshold}$$

This output is set when a channel's count value decreases below the selected threshold – thus a delta setting. *Snap Set Threshold* is an 8-bit unsigned value for the 'set' threshold. *Snap Clear Threshold* is an 8-bit unsigned value for the 'clear' threshold, allowing a hysteresis to provide improved snap detection.

5.6.3 ALP Output

The *ALP Prox Status* flag in *Info Flags* is set when a channes count value deviates (positive or negative) from the LTA value by more than the selected threshold – thus a delta setting *ALP Output Threshold*. This can be used to implement proximity or touch detection, depending on the threshold used. In auto-prox mode, a deviation on any of the individual count values will wake the system from the auto-prox process. Since this is an individual unfiltered reading (compared to the filtered ALP Count value), it has a separate configurable *ALP Auto-Prox Threshold*, which is also a delta value for positive or negative deviations of the individual count values.

5.6.4 Output Debounce

There is no debounce on the trackpad touch or snap detection (or release). This is because debouncing adds too much delay, and fast movements on the touch panel cannot be debounced fast enough to provide reliable XY output data.

Debounce on the ALP output is, however, done to allow for stable proximity detection if needed. Two 8-bit unsigned values are used for the set and clear debounce parameters, *ALP Set Debounce* and *ALP Clear Debounce*.

5.7 Automatic Tuning Implementation (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 various parameters.

The main advantage of the ATI is to balance out small variations between trackpad hardware and the IQS9150/IQS9151 variation to give similar performance across devices and temperatures.

5.7.1 Trackpad ATI

The *Trackpad ATI Multiplier/Dividers* can be used to configure the base value for the trackpad channels. There is one global setting parameter for all the active trackpad channels for the coarse divider and one for the coarse multiplier. The coarse divider is a 5-bit setting (0-31) and the coarse multiplier a 4-bit setting (0-15). The coarse divider/multiplier are configured in the Azoteq GUI software in predefined sets of divider and multiplier combinations. This helps to simplify the configuration of this ATI parameter, and to help make sure optimal combinations are used.

The fine divider/multiplier is also used to configure the trackpad base value. There is one global setting parameter for all the active trackpad channels for the fine divider. The fine divider is a 5-bit setting (0-31) and the fine multiplier a 2-bit setting (0-2). It is recommended to set the fine multiplier to 1.

The *ATI Compensation Values* for each channel are set by the ATI procedure and are chosen so that each count value is close to the selected *ATI Target*.

The sensitivity of the trackpad channels increase in direct proportion to the ratio of the trackpad target counts to the trackpad base counts.

$$\text{Sensitivity} \propto \frac{\text{Target Counts}}{\text{Base Counts}}$$

The algorithm is queued by setting the *TP Re-ATI* bit in the *System Control* register. The *TP Re-ATI* bit clears automatically on chip when the algorithm has completed.

The queued re-ATI routine will execute as soon as the corresponding channels are sensed. For example, the trackpad re-ATI when the system is in Active, Idle-Touch, or Idle mode.

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

ATI Compensation are 10-bit values, thus 0 to 1023. The *ATI Compensation* can be scaled by means of the *Compensation Divider*. The 5-bit *Compensation Divider* values are also automatically configured together with the *ATI Compensation* during the ATI procedure.

5.7.2 ALP ATI

The *ALP ATI Mode* is configured in the *Config Settings* register. Users can choose between two options: *Full ATI* and *Compensation Only* ATI. In contrast to the manual user configuration for trackpad channels' ATI parameters, when *Full ATI* mode is selected, users set both an *ALP Base Target* and an *ALP ATI Target* for the automatic ATI parameter configuration of the ALP channel. The ALP channel uses both *ALP Coarse and Fine Dividers/Multipliers* in its configuration.

The *ALP Base Target* acts as a reference point for the ATI algorithm. The algorithm uses the *Coarse and Fine Dividers/Multipliers* to reach the *Base Target*, from which the *ALP Compensation* is incorporated to reach the *ALP ATI Target*. The *ALP ATI Target* value applies to each of the *ALP Individual Count* values configured for the ALP channel, resulting in the combined channel possessing an *ALP Count* value larger than the *ALP ATI Target*, as it is a sum of the individual Rx engine count values.

If the user selects *Compensation Only* for the *ALP ATI Mode*, the ATI parameters are configured in the same manner as those for the trackpad channels.

The ALP channel has individual *ALP Compensation* values and *ALP ATI Compensation Dividers* for each of the 13 ProxSense® engines.

The algorithm is queued by setting the *ALP Re-ATI* bit in the *System Control* register. The *ALP Re-ATI* bit clears automatically on chip when the algorithm has completed. The ALP channel will execute the re-ATI command when the system is in LP1 or LP2.

5.8 Automatic Re-ATI

5.8.1 Description

When *TP Re-ATI EN* or *ALP Re-ATI EN* are enabled in *Config Settings*, a re-ATI will be triggered 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. It is recommended to always have this enabled. When a re-ATI is performed on the IQS9150/IQS9151, a status bit (*TP/ALP Re-ATI Occurred*) will set momentarily in *Info Flags* to indicate that this has occurred.

5.8.2 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:

   $$\text{Re-ATI Boundary} = \text{ATI target} \pm \text{Drift limit}$$

   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:

   $$\text{Reference} > 850 \text{ or Reference} < 750$$

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

2. **Trackpad Negative Delta Re-ATI**

   A considerable decrease in the count value of a trackpad channel is abnormal since user interaction increases the count value. Therefore, if a decrease larger than the configurable threshold *Trackpad Negative Delta Re-ATI Value* is seen on such a trackpad channel, it is closely monitored. If this is continuously seen for 15 cycles, it will trigger a re-ATI.

3. **Trackpad Positive Delta Re-ATI**

   Enabling snap sensors presents an issue where, during an ATI, if a metal dome press occurs, an abnormally large positive delta is detected upon release – much larger than what would be expected from a regular user touch. To address this, if a positive delta exceeding the *Trackpad Positive Delta Re-ATI Value* is identified on a trackpad channel, it triggers a re-ATI after 15 consecutive cycles for recovery.

5.8.3 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 = 0 (min value)
- ATI Compensation = 1023 (max value)
- Count is already outside the re-ATI range upon completion of the ATI algorithm

If any of these conditions are met, the corresponding *ATI Error/ALP ATI Error* flag will be set in the *Info Flags* register. The flag status is only updated again when a new ATI algorithm is performed.

Note: 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. The Re-ATI retry time has a maximum setting of *60 seconds*.

6 Sensing Modes

The IQS9150/IQS9151 automatically switches between different charging modes dependent on user interaction and other aspects. This is to allow for fast response and low power consumption when applicable. The current *Charging Mode* can be read from the *Info Flags* register.

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

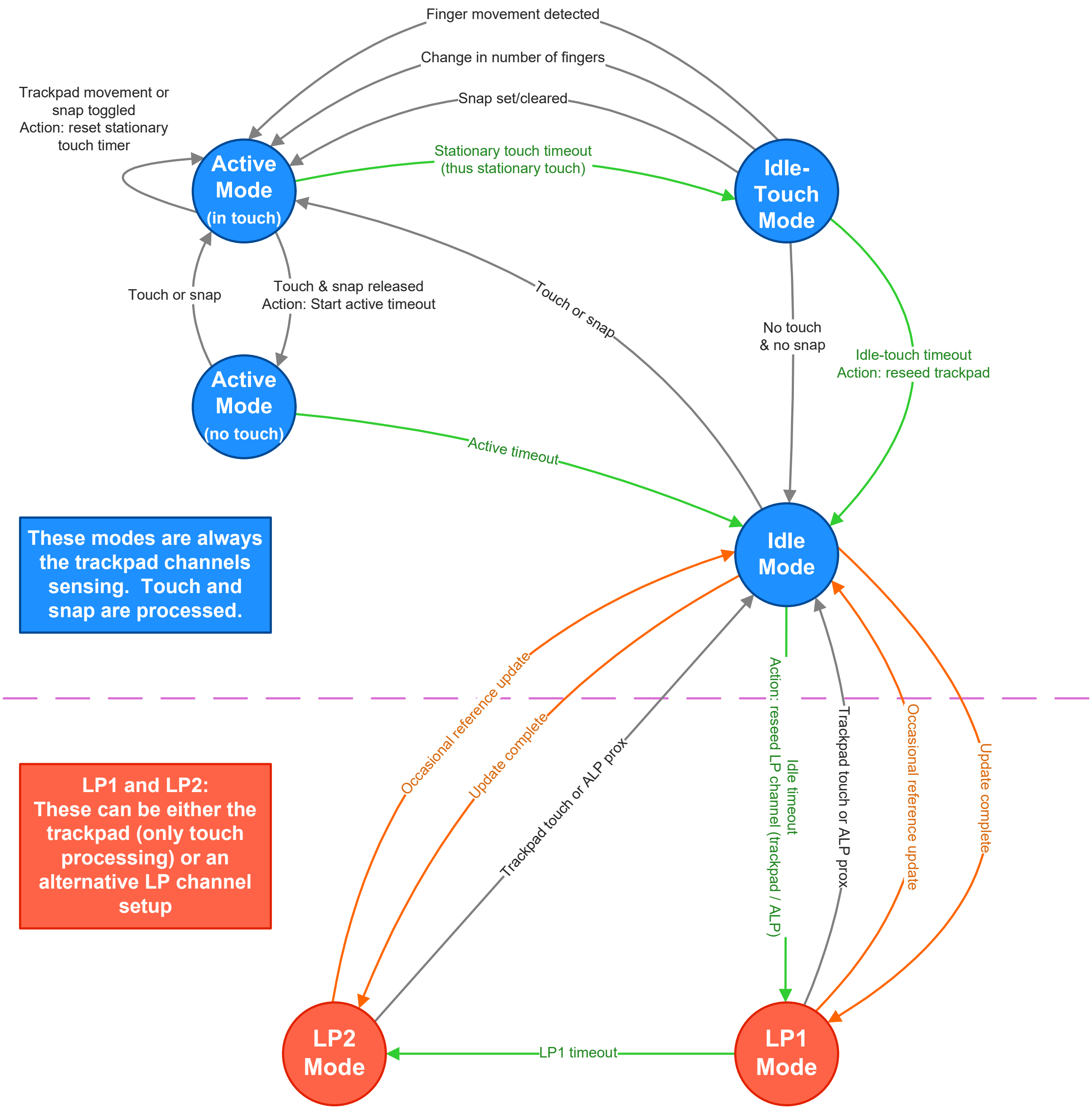


Figure 6.1: System Mode State Diagram

6.1 Sampling Period

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

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

- *Active Mode Sampling Period*
- *Idle-Touch Mode Sampling Period*
- *Idle Mode Sampling Period*
- *LP1 Mode Sampling Period*
- *LP2 Mode Sampling Period*

6.2 Mode Timeout

The timeout values are configurable, and once these durations have passed, the system will transition to the next state as depicted in Figure 6.1. You can adjust these durations by selecting your desired value (in seconds) for each specific timeout.

- *Stationary Touch Timeout*
- *Idle-Touch Mode Timeout*
- *Idle Mode Timeout*
- *LP1 Mode Timeout*
- *Active to Idle Mode Timeout (ms)*

Note: Active Mode includes two timeout settings:

- *Stationary Touch Timeout*, which triggers when the touch is stationary in Active Mode, transitioning the mode to Idle-Touch mode.
- *Active to Idle Mode Timeout*, which triggers upon touch/snap release.

A timeout value of '0' will result in a 'never' timeout condition.

6.3 Manual Control

The default method (manual control disabled) allows the IQS9150/IQS9151 to automatically switch between modes and update *Trackpad Reference Values* as shown in Figure 6.1. This requires no interaction from the master to manage the device and is the recommended option.

The master can manage various states and implement custom power modes when *Manual Control* is enabled in *Config Settings*. The master needs to control the mode (*Mode Select*) and also manage the reference values by reseeding (*TP Reseed*). Both settings are available in the *System Control* register.

7 Trackpad

7.1 Configuration

7.1.1 Size Selection

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

7.1.2 Trackpad Channel and Electrode Limitations

The IQS9150 product supports up to 506 channels, with a maximum of 45 electrodes. Up to 26 Rxs and up to 22 Txs can be configured, but the total number of Rxs and Txs combined cannot exceed 45 electrodes. The IQS9151 supports up to 156 channels, consisting of a maximum of 25 electrodes, with limitations of up to 13 Rxs and up to 22 Txs. Any trackpad size and configuration that fits into these limits are possible to implement.

7.1.3 Individual Channel Disabling

If the sensor is not a complete rectangle (this could be due to mechanical cut-outs or trackpad shape), there will be some channels that fall within the *Total Rxs*/*Total Txs* rectangle but do not exist. The channel numbers are still allocated for the complete rectangle (see Section 5.2.1). However, these channels can be disabled individually using the *Trackpad Channel Disable* registers.

7.1.4 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. The *RxTx Mapping* configures which actual Rx and Tx electrodes are used for the trackpad. The Rxs are specified first, up until the number of Rxs as defined by the *Total Rxs* register, then the Txs follow immediately.

Following the example in Table 5.1, the *RxTx Mapping* settings will be as follows:

RxTxMapping[0] = 0
RxTxMapping[1] = 1
RxTxMapping[2] = 2
RxTxMapping[3] = 3
RxTxMapping[4] = 4
RxTxMapping[5] = 5
RxTxMapping[6] = 6
RxTxMapping[7] = 7
RxTxMapping[8] = 21
RxTxMapping[9] = 30
RxTxMapping[10] = 9
RxTxMapping[11] = 22
RxTxMapping[12] = 10
RxTxMapping[13] = 23
RxTxMapping[14] = 31
RxTxMapping[15] = 11
RxTxMapping[16] = 24
RxTxMapping[17] = 12

RxTxMapping[18] = 25
RxTxMapping[19] = 32
RxTxMapping[20..44] = n/a

7.2 Trackpad Outputs

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

7.2.1 Number of Fingers

Number of Fingers in the *Trackpad Flags* register gives an indication of the number of active finger inputs on the trackpad.

7.2.2 Relative XY

If there is only one finger active, a *Relative X* and *Relative Y* value are 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.

7.2.3 Absolute XY

For all multi-touch inputs, the absolute finger positions are reported in the *Finger X/Y-Coordinate* registers, where the coordinate output is based on the selected resolution. This means that the coordinates will range between 0 and the selected *Resolution X/Y*.

7.2.4 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.

7.2.5 Area

The number of channels associated with a finger is provided in the *Finger Area* registers. This area is usually equal to or smaller than the number of touch channels under the finger.

7.2.6 Tracking Identification

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

7.3 Maximum Number of Multi-touches

The maximum number of allowed multi-touches is configurable *Max Multi-Touches* up to 7 points. If more than the selected value is sensed, the *Too Many Fingers* flag is set in the *Info Flags* register, and the XY data is cleared.

7.4 XY Resolution

The output resolution for the X and Y coordinates is 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.

7.5 Stationary Touch

A stationary touch is defined as a point that does not move outside a certain boundary within a specific time. This movement boundary or threshold can be configured in the *Stationary Touch Movement Threshold* register 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 for the *Stationary Touch Timeout (s)* period, where a lower duty cycle can be implemented to save power in applications where long touches are expected.

If movement is detected, the *Movement Detected* flag is set in *Trackpad Flags*.

7.6 Multi-touch Finger Split

The position algorithm looks at areas (polygons) of touches and calculates positional data from this. Two fingers near 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. The *Finger Split Factor* 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.

7.7 XY Output Flip & Switch

By default, X positions are calculated from the first column to the last column. Y positions are by default calculated from the first row to the last row. The X and/or Y output can be flipped by setting the relevant bits (*Flip X*/*Flip Y*) in *Trackpad Settings* to allow the [0, 0] coordinate 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.

Note: The channel numbers are still assigned the same way, first along the Rxs, then to the next Tx; it is not affected by this setting.

7.8 XY Position Filtering

Stable XY position data is available 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. It is recommended to keep both filters enabled for optimal XY data.

7.8.1 IIR Filter

The *IIR Filter*, if enabled in *Trackpad Settings*, can be configured to select between a dynamic and a static filter.

$$\text{Damping factor} = \frac{\text{Beta}}{256}$$

1. **Dynamic Filter**

   Relative to the speed of movement of a coordinate, 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 coordinate 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 Filter Bottom Speed*
- *XY Dynamic Filter Top Speed*
- *XY Dynamic Filter Bottom Beta*

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

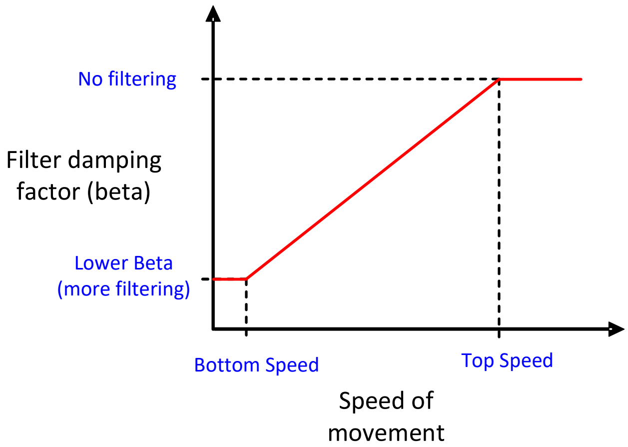


Figure 7.1: Dynamic Filter Parameters

2. **Static Filter**

   Coordinates filtered with a fixed but configurable damping factor (*XY Static Filter Beta*) are obtained using the static filter *IIR Static*. It is recommended that the dynamic filter is used due to the advantages of a dynamically changing damping value.

7.8.2 Jitter Filter

To prevent small finger coordinate movements for a stationary finger, a jitter filter is implemented. The *Jitter Filter* can be enabled in the *Trackpad Settings* register. The jitter filter will only allow initial movement once the finger has moved an initial configurable distance (*Jitter Filter Delta Threshold*) in either x or y.

7.9 X & Y Trim

Due to boundary conditions at the edges of the trackpad, it is unlikely that the X and Y extreme values will be achievable (0 and X/Y Resolution). To be able to achieve this, the edges can be trimmed with a configurable amount of *X Trim* or *Y Trim* on-chip. For example, say *X Trim* is set to 0, and a finger on the left of the trackpad gives a minimum X output of 48 and a maximum of 960 for a finger to the far right (for X resolution set to 1000). Then an X Trim = 50 could be used to trim away the 'dead' area, and the full 0 to 1000 range will be achievable.

7.10 Finger Confidence

For each finger on the trackpad, there is a *Finger Confidence* bit in the *Trackpad Flags* register to indicate whether there is confidence that this is a legitimate finger input. For normal finger inputs, the bit will be set (1), indicating high confidence that this is an acceptable trackpad input. If the finger area is larger than a configurable *Finger Confidence Threshold*, then the confidence bit related to that finger will clear (0), and it will remain cleared until that finger is removed.

7.11 Saturation

Sensor saturation is a non-ideal response from the touchpad to a specific user input. Saturation can be improved with design aspects. For more information, please see AZD068.

If any touch on the trackpad senses saturation occurring within the touch area, then the saturation bit will become set. Ideally, you would like your design to never have this set.

8 Gestures

The IQS9150/IQS9151 has an on-chip gesture recognition engine for single and two-finger gestures. The list of *Single Finger Gestures* and *Two-Finger Gestures* recognised by the device is as follows:

- Single finger gestures:
  - Single tap
  - Double tap
  - Triple tap
  - Press-and-Hold
  - Swipe X+ (with continuous swipe configurable)
  - Swipe X- (with continuous swipe configurable)
  - Swipe Y+ (with continuous swipe configurable)
  - Swipe Y- (with continuous swipe configurable)
  - Swipe and hold X+
  - Swipe and hold X-
  - Swipe and hold Y+
  - Swipe and hold Y-
  - Palm (Flat hand)
- Two-finger gestures:
  - Single tap
  - Double tap
  - Triple tap
  - Press-and-Hold
  - Zoom in
  - Zoom out
  - Vertical scroll
  - Horizontal scroll

Each gesture can individually be enabled or disabled by setting or clearing the corresponding bits in the relevant register, *Single Finger Gesture Enable* or *Two Finger Gesture Enable*.

Each gesture has parameters that define and configure its functionality.

8.1 Single, Double and Triple Tap Gesture

The tap gestures (*Single Tap, Double Tap, Triple Tap*) require that a touch be made and released in the same location and within a short period of time. Some small amount of movement from the initial coordinate is allowed to compensate for expected finger movement while tapping on the sensor. This bound is defined in register *Tap Distance*, which specifies the maximum deviation in pixels the touch is allowed to move before the tap gesture is no longer valid.

Similarly, the *Tap Time* register defines the maximum touch duration (in milliseconds) that will result in a valid gesture. The period is measured from the moment a touch is registered. The touch should be released before the *Tap Time* has elapsed for the tap to be reported.

The *Air Time* parameter defines the maximum duration (in milliseconds) that is allowed between taps (while the finger is NOT touching the sensors) for double and triple taps to be detected. The next touch must be detected before the *Air Time* has expired, starting at the moment the previous touch is released, to continue the multiple tap sequence.

With double/triple taps enabled, the engine first needs to wait to confirm whether the current detected tap is part of a multi-tap gesture before the tap output can be provided. If subsequent taps are NOT enabled, the tap gesture will be immediately reported on the release of the tap touch. If subsequent taps ARE enabled, the current tap gesture will only be reported when the time specified by the *Air Time* parameter has elapsed and no further taps have begun. For example, double taps require an *Air Time* waiting period if, and only if, triple taps are enabled.

Since the gesture reports after the finger is removed and no XY data is available, the location of the tap gesture is placed in the *Gesture X* and *Gesture Y* registers.

The gesture engine will clear relative XY registers *Relative X* and *Relative Y* to prevent small cursor movement during tap detection.

Below are numerous scenarios illustrating the tap outputs.

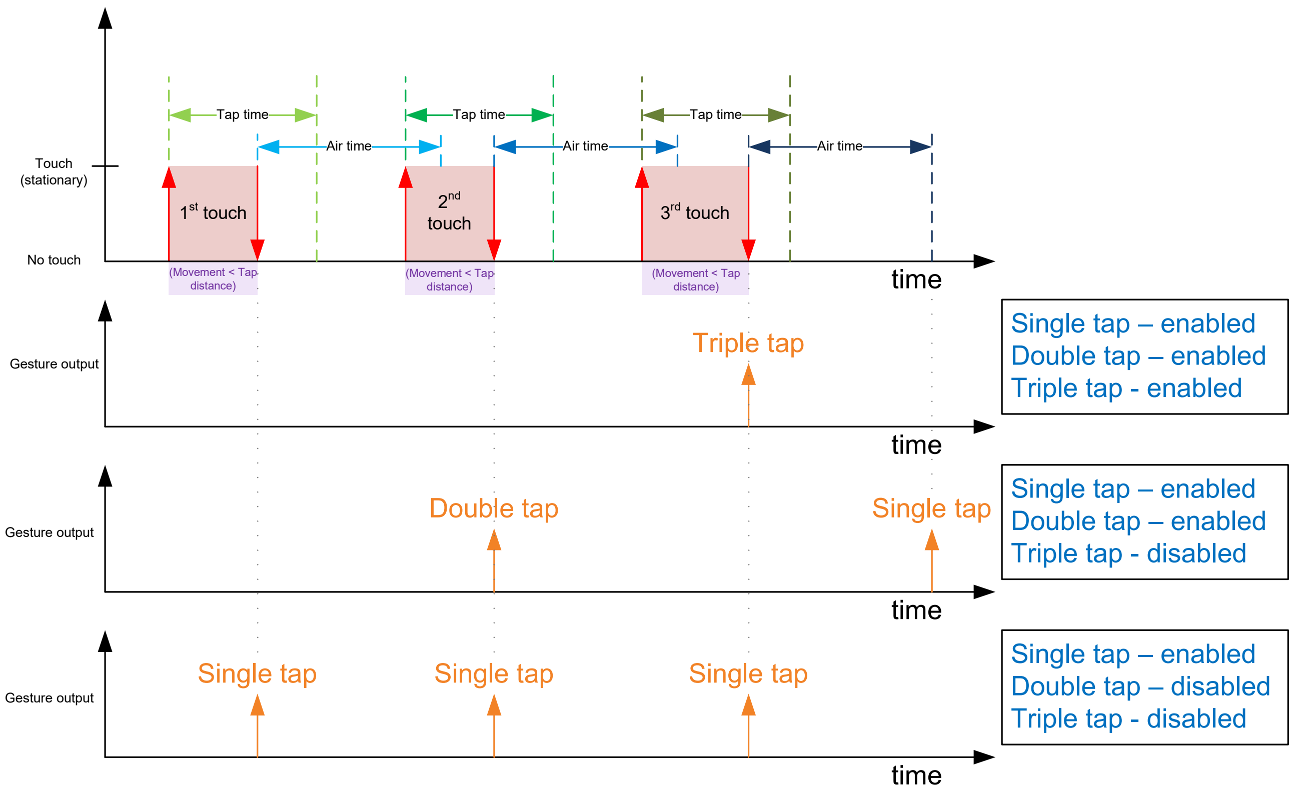


Figure 8.1: Three Taps - Output Scenarios

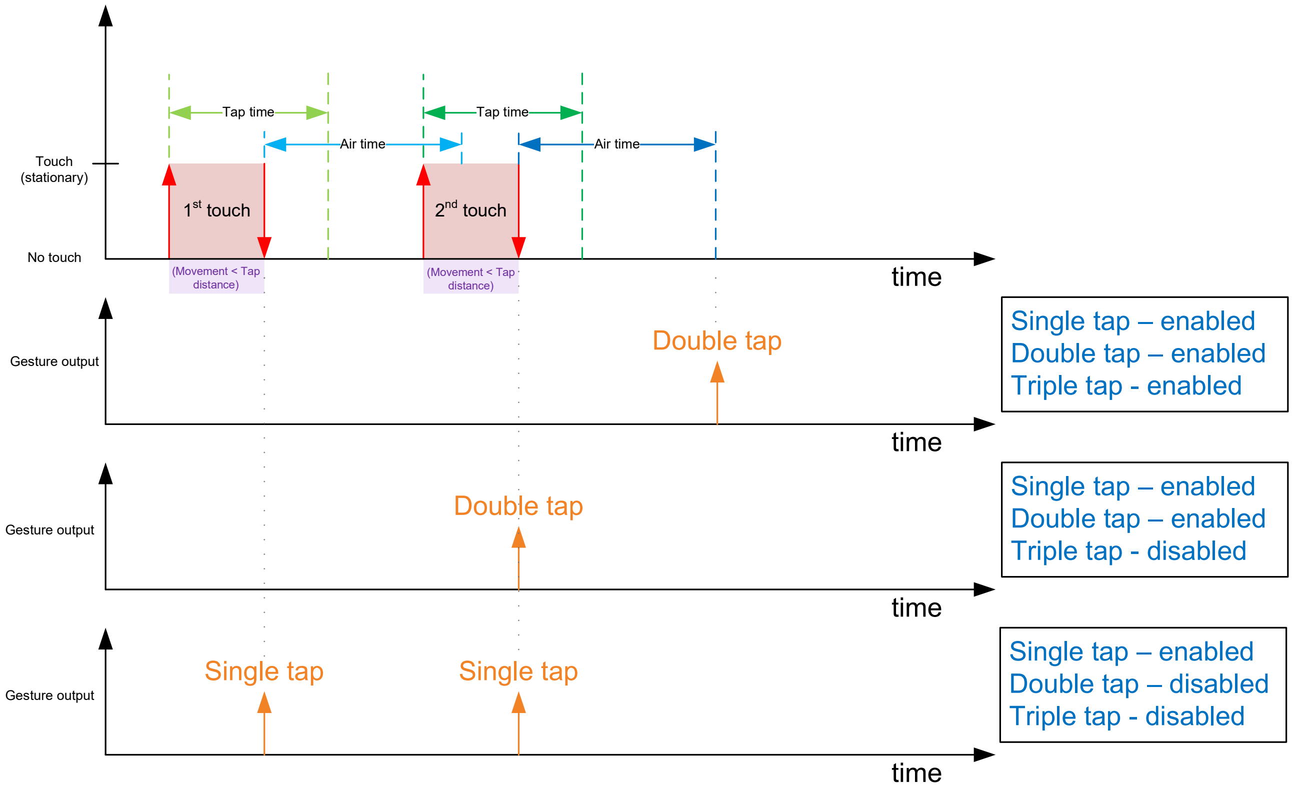


Figure 8.2: Two Taps - Output Scenarios

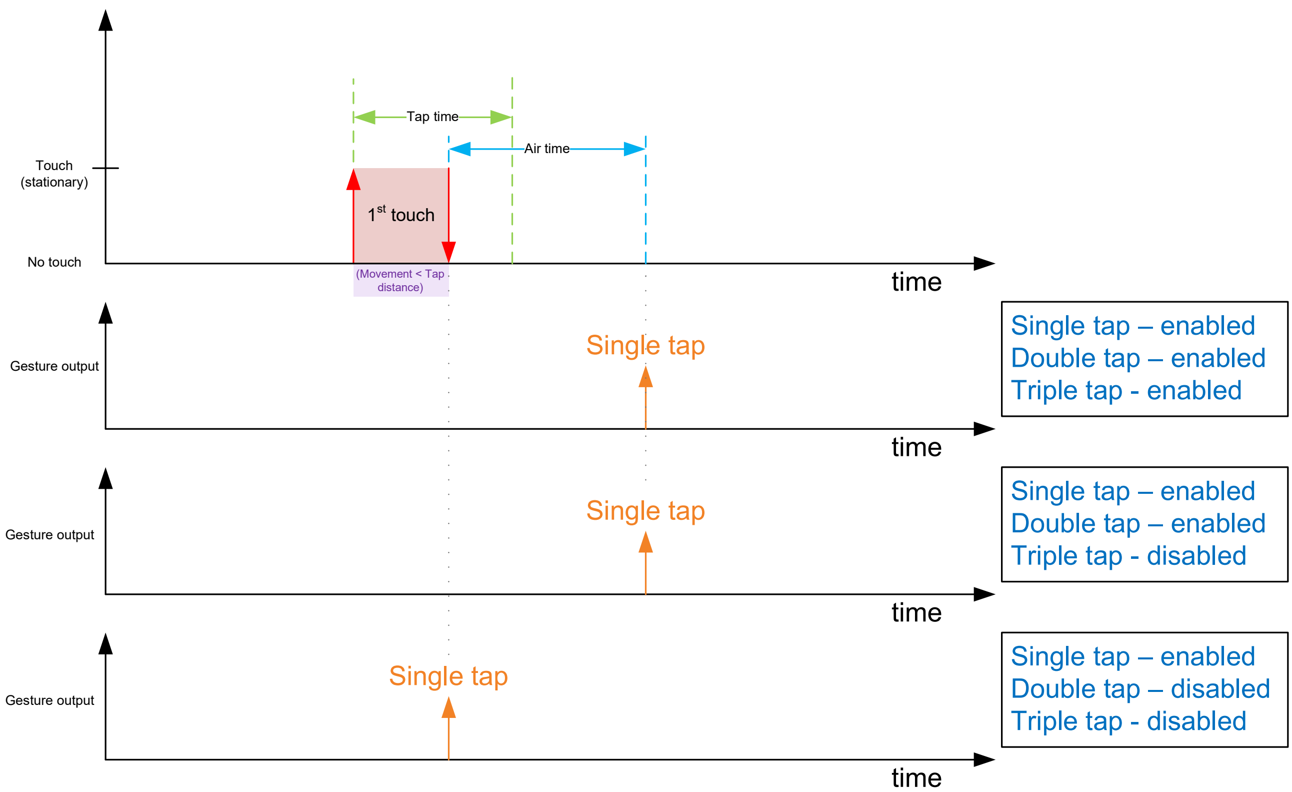


Figure 8.3: Single Tap - Output Scenarios

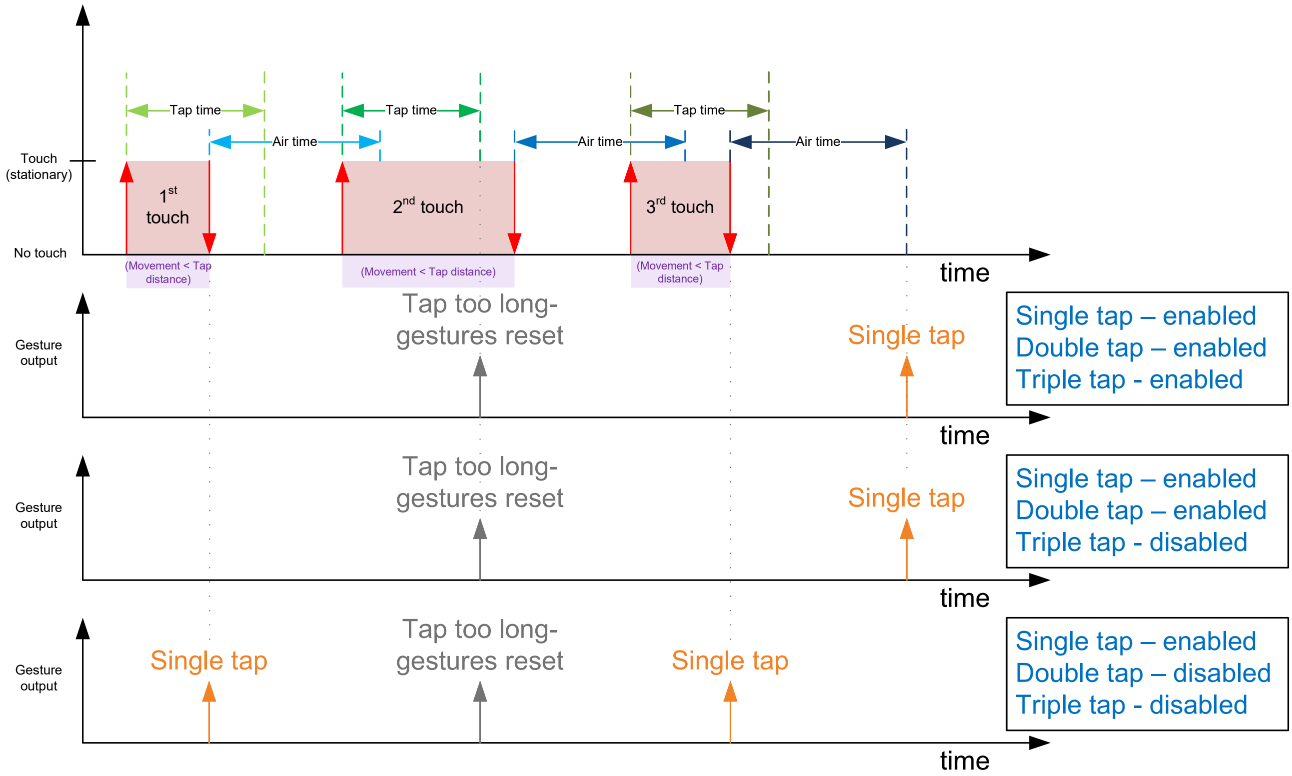


Figure 8.4: Tap Time Elapsed

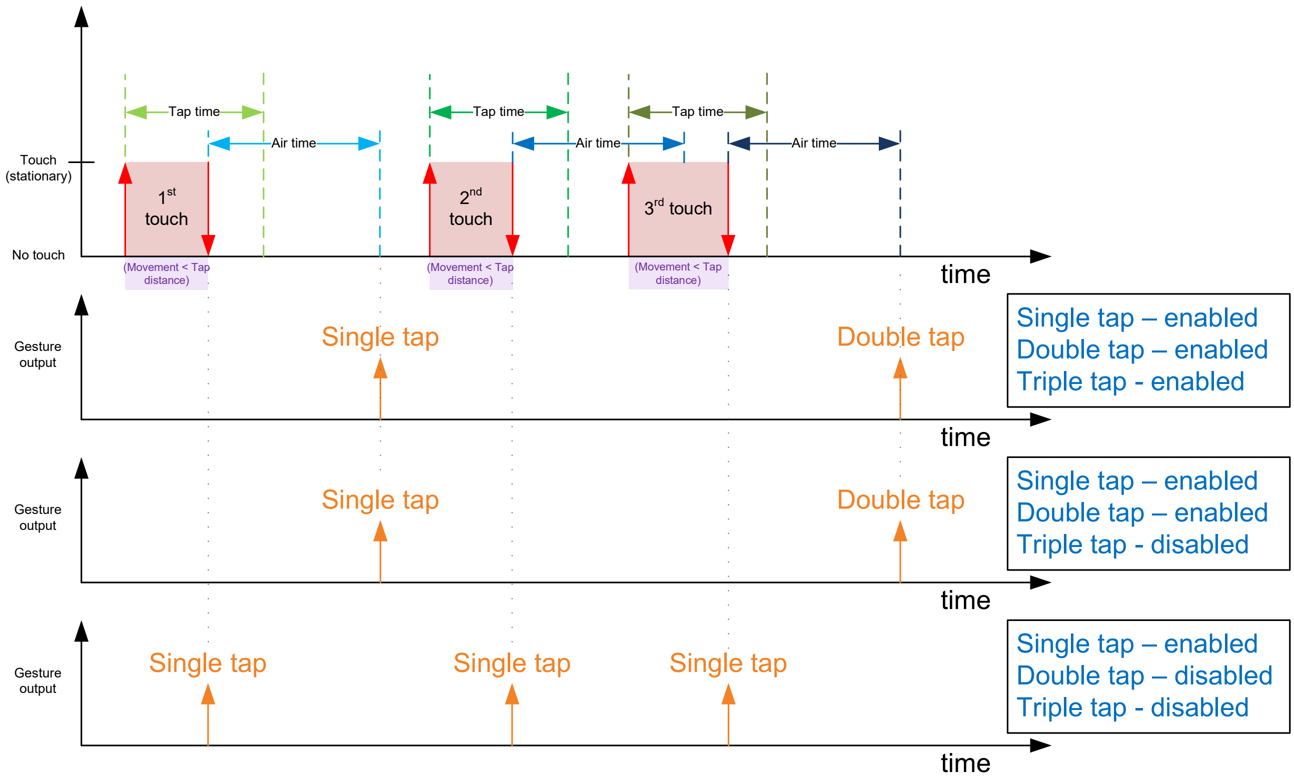


Figure 8.5: Air Time Elapsed

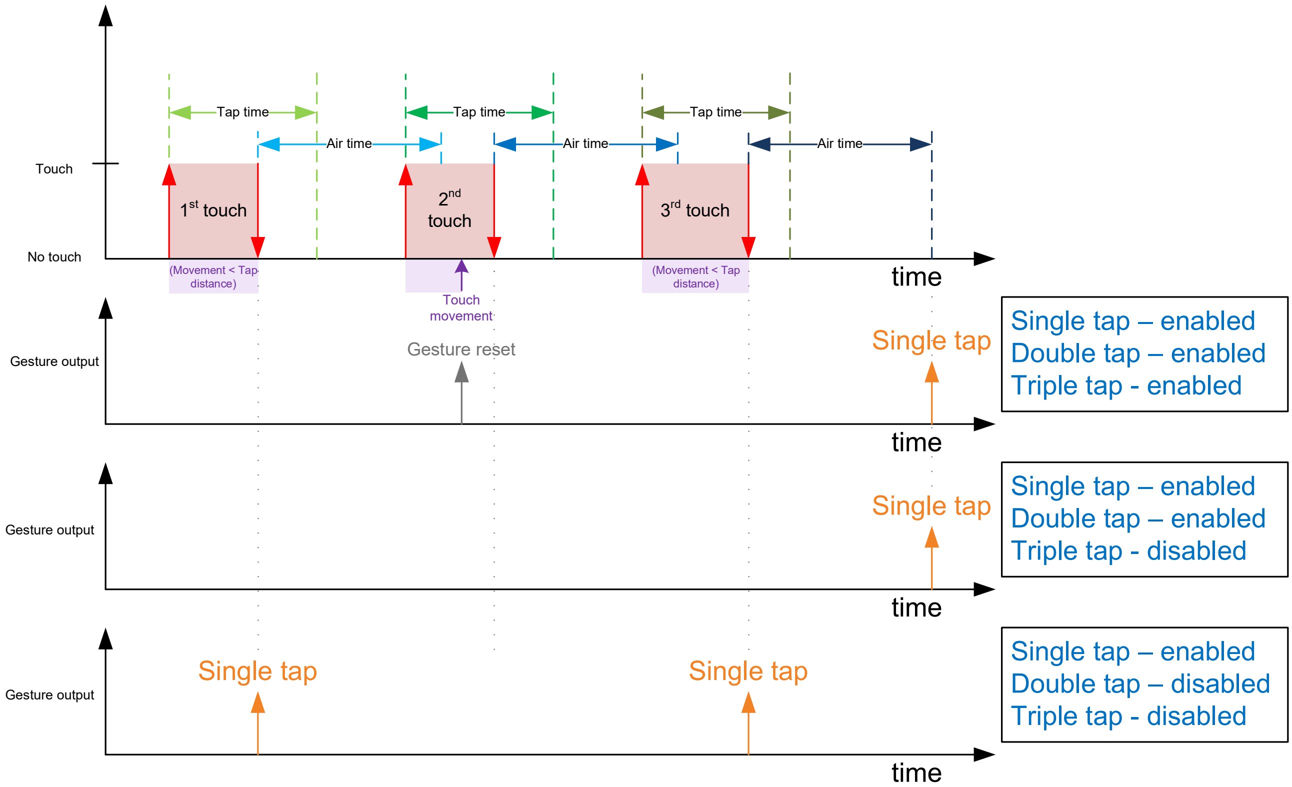


Figure 8.6: Finger Movement

8.2 Press-and-Hold Gesture

The same register that defines the bounds for the single tap gesture, *Tap Distance*, is used for the Press-and-Hold gesture.

If a touch remains within the given bound for longer than the *Hold Time* (in milliseconds), a Press-and-Hold gesture will be reported in the *Single Finger Gestures* register. The gesture will continue to be reported until the specific touch is released, even if finger movement resumes.

Similarly, there is also a two-finger Press-and-Hold, which requires two fingers in touch and follows the same conditions for activation. The two-finger Press-and-Hold gesture will be reported in the *Two-Finger Gestures* register.

Relative data will be reported in the *Gesture X/Y* registers once the gesture has been triggered. This allows for features such as drag-and-drop. For a one-finger Press-and-Hold gesture, the *Gesture X/Y* values will be exactly the same as the *Relative X/Y* register values. For a two-finger Press-and-Hold gesture, it will represent the relative movement of the average position of the fingers.

Once the gesture has triggered, the number of fingers must remain constant. For example, for a one-finger Press-and-Hold, the gesture will clear if there is ever not one finger in touch. Likewise, for a two-finger Press-and-Hold, there must always be two fingers in touch. If the gesture clears and there is still a touch, the *Gesture X/Y* registers will be zeroed, and the user must completely go out of touch before any gestures will be reported again.

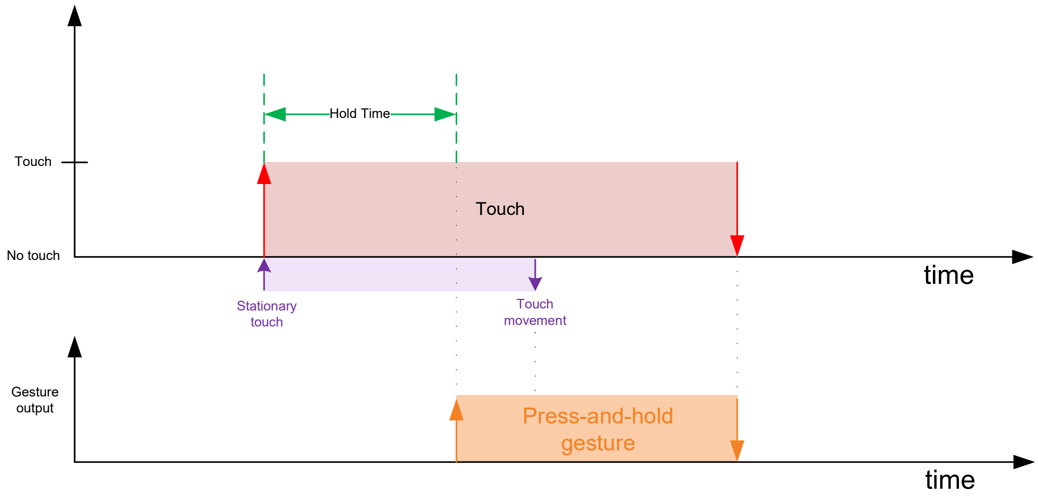


Figure 8.7: Press-and-Hold

8.3 Swipe Gesture

8.3.1 Single Swipe

All four swipe gestures (*Swipe X+, Swipe X-, Swipe Y+, Swipe Y-*) 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. If the touch is moving away from the origin, it is considered a positive swipe (+). 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 registers *Swipe Initial X-Distance* and *Swipe Initial Y-Distance*.
2. The distance in (1) is covered within the time specified in *Swipe Time* (in milliseconds).
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 respective swipe gesture will be reported for 1 cycle when all these conditions are met. The relative distance travelled each cycle will be reported in the *Relative X/Y* registers throughout.

The value in register *Swipe Angle* is calculated as 64 tanθ, where θ is the desired angle (in degrees).

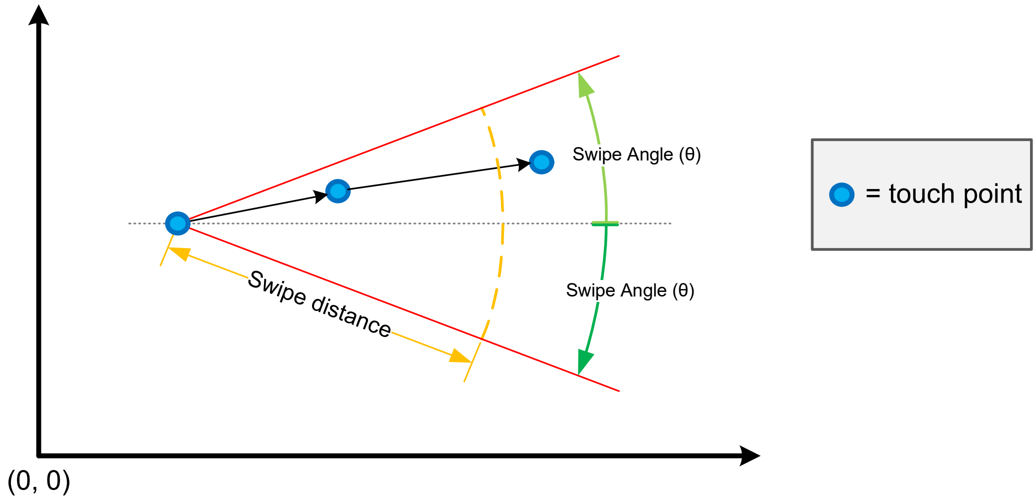


Figure 8.8: Illustration of the Swipe Angle Requirement

The relative X and Y movement used to determine the swipe is available in the *Gesture X/Y* registers. The swipe angle and distance can be calculated from the data reported in these registers. This allows customers with orientation-sensing capability to normalise the swipe to the orientation of the product. The *Swipe Angle* parameter should be set to obtain 45 degrees (thus always allowing a swipe), and the master can accept or reject swipes depending on the adjusted swipe angle.

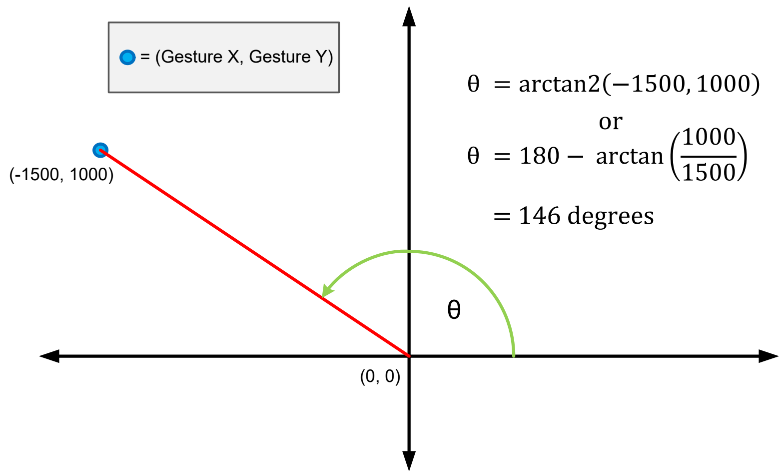


Figure 8.9: Swipe Angle Calculation from Gesture X/Y

Once the initial swipe has been detected, additional swipe outputs can be triggered in one of two ways during the same touch interaction.

8.3.2 Swipe-and-Hold

With Swipe-and-hold (*Swipe Hold X+, Swipe Hold X-, Swipe Hold Y+, Swipe Hold Y-*) enabled in the *Single Finger Gesture Enable* register, the additional swipe gestures will be triggered by a stationary touch. For a Swipe-and-Hold gesture to be reported, a single swipe must be detected (Section 8.3.1), and then the finger that performed the swipe must become stationary. To be stationary, the finger's movement must be less than the *Tap Distance* for the duration of the *Hold Time*. This is similar to the Press-and-Hold gesture. At this point the relevant output (*Swipe Hold X+, Swipe Hold X-, Swipe Hold Y+, Swipe Hold Y-*) will be reported in the *Single Finger Gestures* register and will then only clear upon release of the finger. While one of the swipe-and-hold flags is set, relative finger movement will be reported in the *Gesture X/Y* registers.

The same termination logic as for the Press-and-Hold gesture is applied once a swipe-and-hold gesture is detected. In other words, if another finger enters touch, the gesture is cleared, and the *Gesture X/Y* values are reset to zero.

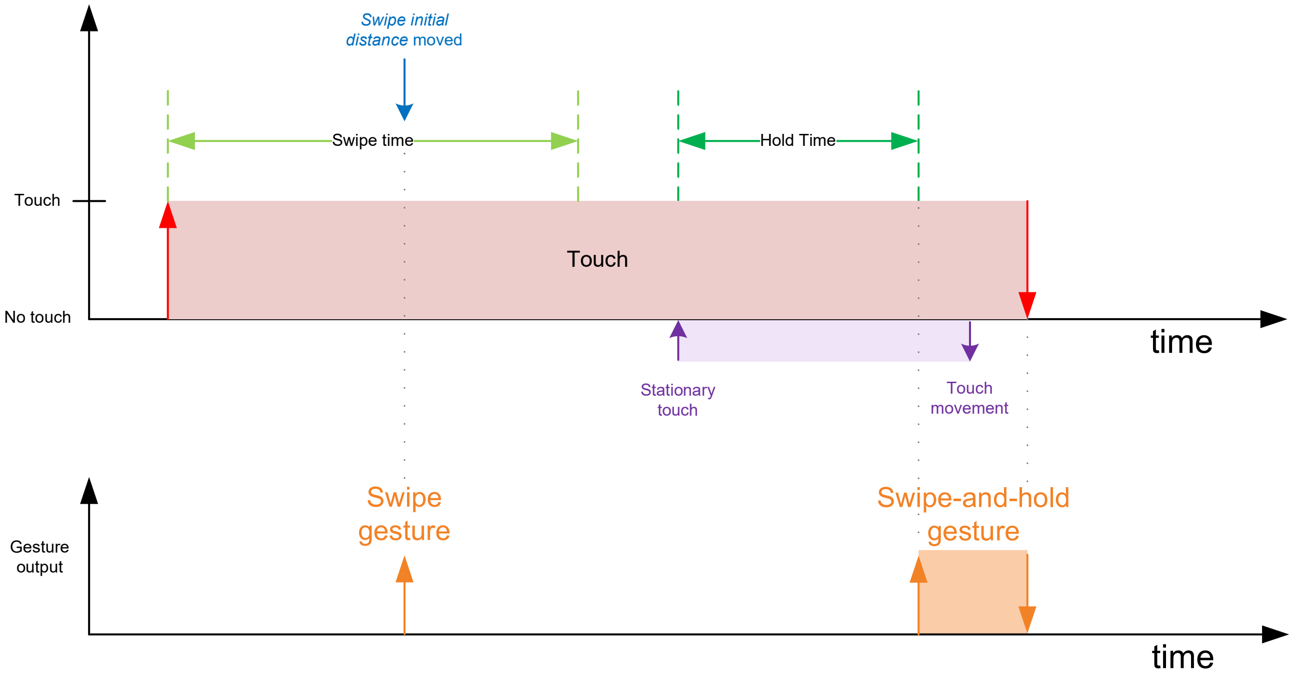


Figure 8.10: Swipe-and-Hold Gesture

8.3.3 Consecutive Swipe

With Swipe-and-hold disabled, it is possible to generate consecutive swipe gesture events during the same swipe gesture by defining the *Swipe Consecutive X-Distance* and *Swipe Consecutive Y-Distance* (pixels). Once the initial swipe gesture has been reported, additional swipe outputs will be generated when the movement exceeds the consecutive threshold and the angle satisfies the *Swipe Angle* condition, and will continue in this manner until the finger is released. The reference point for the consecutive swipe distance is the location where the previous swipe was detected.

Note: For consecutive swipes, the time limit *Swipe Time* is no longer applied.

The *Swipe Consecutive Distance* is used to evaluate consecutive swipes along the same axis. To switch swipe axes, the *Swipe Initial Distance* must be met along the axis being switched to. The consecutive threshold is normally a shorter distance than the initial distance, meaning switching the axis is slightly more difficult to achieve, preventing unwanted direction changes.

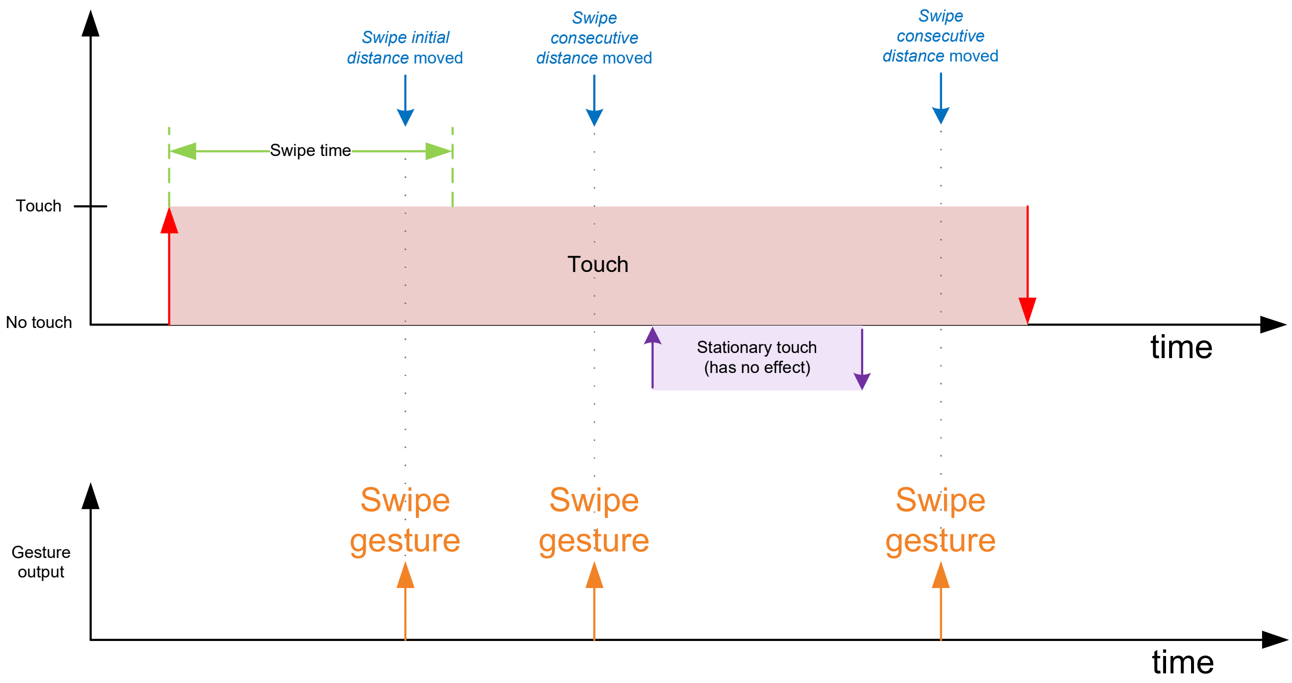


Figure 8.11: Consecutive Swipe with Pause

8.4 Palm Gesture (Flat Hand Gesture)

The palm gesture is used to detect the presence of a flat hand on the trackpad. Since a hand is not a perfectly flat surface, it is not expected that all channels on the trackpad will detect a touch. For this reason, the palm gesture requires a configurable *Palm Gesture Threshold* number of channels to detect touch simultaneously for the *Palm Gesture* to be reported in the *Single Finger Gestures* register. Normally a high percentage of the total channels, larger than the largest allowed touch, are selected as the *Palm Gesture Threshold*. Once the palm gesture has been detected, it will require a full release (no touches) before the gesture is cleared.

Relative movement in the *Relative X/Y* registers will still be reported if it occurs. The user must determine whether the master will ignore the data or not.

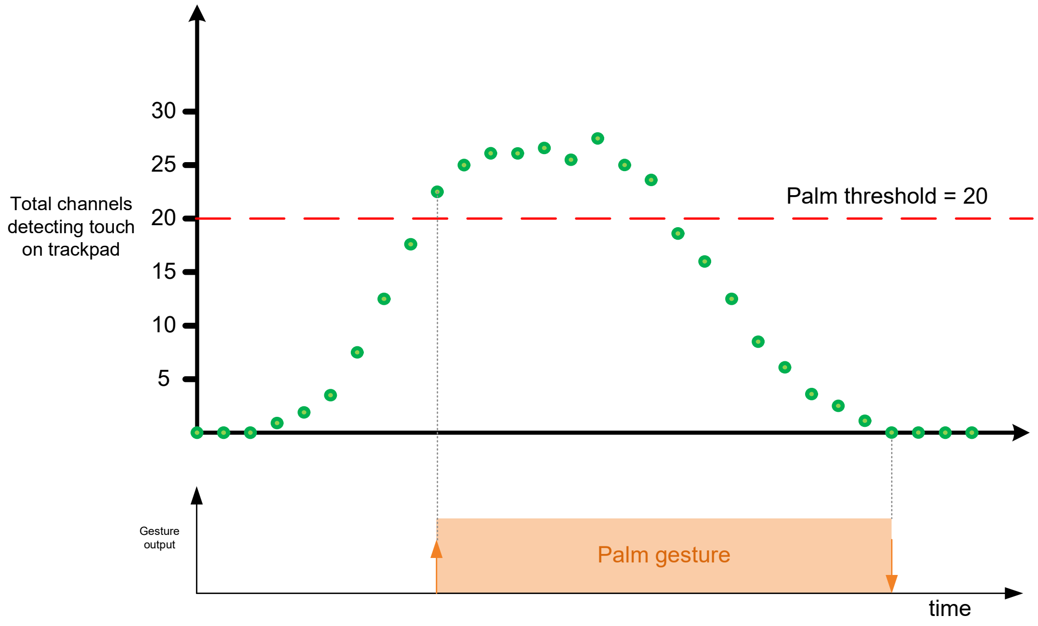


Figure 8.12: Palm Gesture

8.5 Two-Finger Tap

The simultaneous tap gestures require two single-finger tap gestures to occur simultaneously. For this reason, the two-finger tap gestures use the same parameters (*Tap Time*, *Air Time*, and *Tap Distance*) as those of the single-finger tap gestures.

8.6 Scroll

A scroll gesture is identified by two simultaneous and parallel moving touches. A scroll gesture will be reported in the *Two-Finger Gestures* register once the average distance travelled by the two touches in pixels exceeds the value stored in register *Scroll Initial Distance*. Once the initial scroll has been detected, a scroll gesture will be reported when the average distance travelled by the two touches in pixels exceeds the value stored in *Scroll Consecutive Distance*, measured from the point at which the initial scroll was detected.

Similar to the swipe gestures, the scroll gestures are bound by a given angle to the axis (*Scroll Angle*). The value in this register is calculated as 64 tanθ, where θ is the desired angle (in degrees).

The direction of the scroll gesture is defined by the reported *Gesture X* (horizontal scroll) and *Gesture Y* (vertical scroll) data. For instance, a positive *Gesture X* value will correspond with the direction of a swipe X+ gesture. 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 *Gesture X* value and a zero *Gesture Y* value. This value relates to the movement/size of the scroll gesture.

During a scroll gesture, *Relative X/Y* data will be reported in accordance with the standard non-gesture implementation, based on the finger assignments.

8.7 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 in the *Two-Finger Gestures* register. Thereafter, the register *Zoom Consecutive Distance* (pixels) 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.

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

This gesture will terminate if the two touches ever merge into one.

9 Virtual Sensors

The IQS9150/IQS9151 possesses the capability to create easy-to-use virtual sensors within the trackpad sensor area. Adjustable touch buttons, sliders, and wheels with configurable sizes and shapes can be superimposed onto the trackpad sensors. This allows for the creation of easily customisable touch sensors without the need for hardware electrode layout modification or added complexity. The key benefit lies in the ability to reuse the same trackpad PCB (thus no hardware changes) for various designs with different touch sensor requirements, by simply modifying the virtual sensors and their required configuration in firmware.

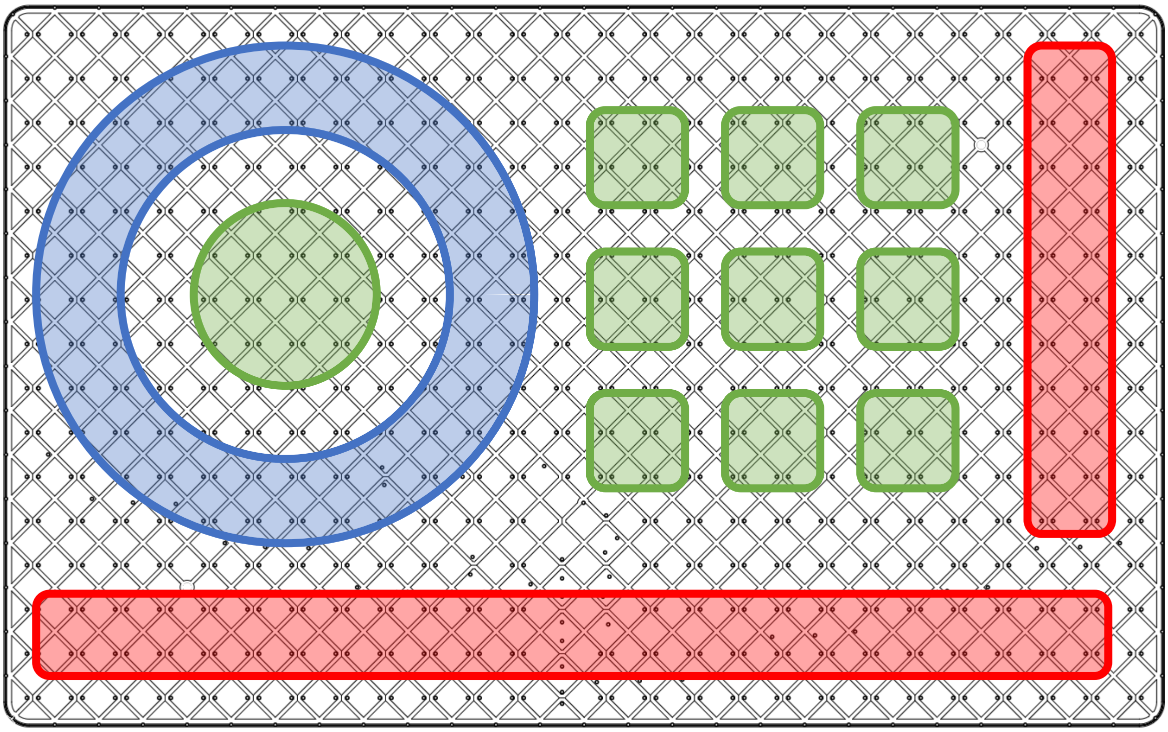


Figure 9.1: Virtual Sensors

For these virtual sensors, it is suggested that the designer should add finger guides to the overlay material/structure, LED sensor indicators, or something similar to help identify the sensor location.

9.1 Maximum Virtual Sensors

The *Number of Virtual Sensors Enabled* register specifies the count of activated virtual sensors. Each type of virtual sensor has its own configurable number of enabled sensors (*Number of Buttons/Sliders/Wheels*). The maximum number of distinct sensors allowed is specified in the table below.

Table 9.1: Maximum Virtual Sensors

Virtual Sensor Type	Maximum Allowed
Button	16
Slider	8
Wheel	4

9.2 Maximum Fingers Per Sensor

For virtual buttons, only one touch (a finger) is detected. Thus, if more than one trackpad finger is within the button area, it will simply indicate a touch, with no indication there is more than one. For the sliders and wheels, however, two-finger inputs can be detected per sensor, and their corresponding slider or wheel locations reported. This allows for multi-touch control of these virtual two sensor types. In a similar manner to the trackpad XY data, the location of the output remains fixed for a specific finger and thus identifies the corresponding finger by report location.

9.3 Buttons

A maximum of sixteen virtual buttons can be implemented on the trackpad. Additionally, any Rx/Tx trackpad channel can function as a standalone touch 'button' sensor, with the output simply obtained from the touch bit in the *Touch Status* register. However, utilising virtual buttons offers the advantage of flexibility in firmware configuration. This includes the ability to easily relocate, resize, or alter the shape of buttons solely through firmware modifications. Also, employing virtual buttons ensures uniform touch sensitivity across the entire button area, eliminating the need for intricate electrode designs.

9.3.1 Button Output

A virtual button has a touch output bit, indicating whether the button is pressed or not. The touch output can be seen in the *Button Output* register, where bit 0 corresponds to Button 0, bit 1 to Button 1, and so forth.

9.3.2 Button Setup

The location of the virtual button is configured by defining its top-left trackpad X,Y coordinate and also its bottom-right coordinate, *Button Top-Left X/Y* and *Button Bottom-Right X/Y*. Any trackpad touch within this bounding box will activate the corresponding button output. There is no limitation on the size or shape of the button, simply that it must fall within the trackpad coordinate space.

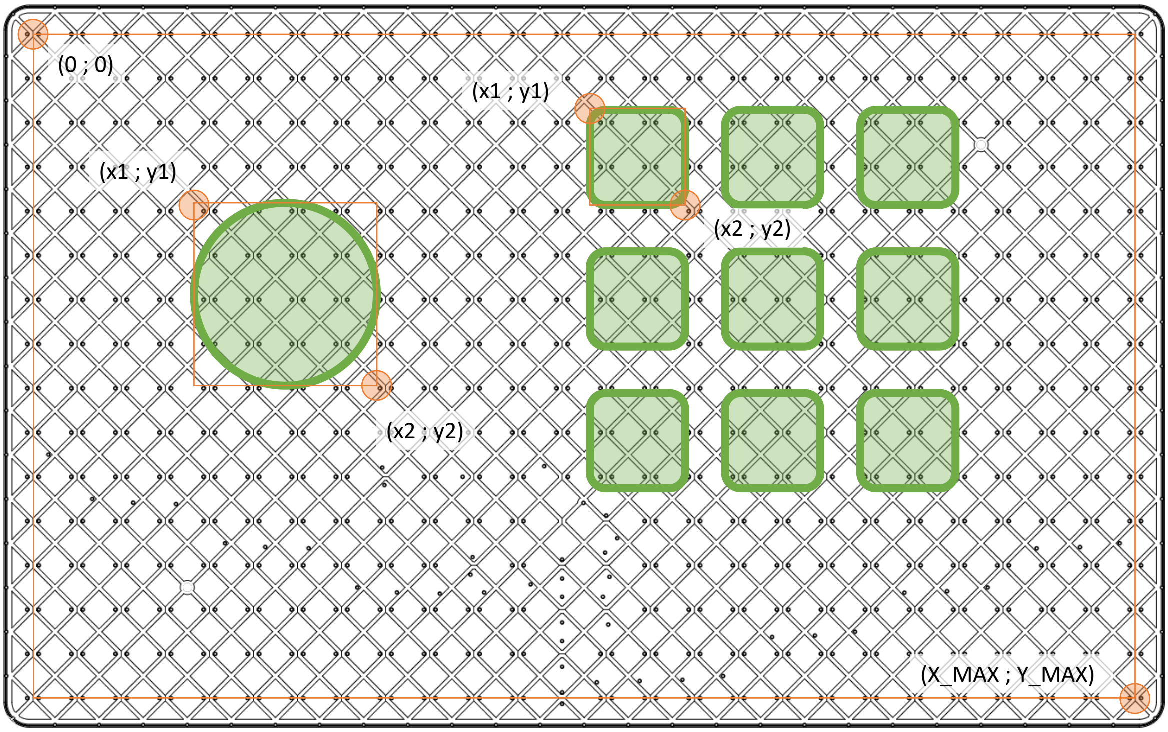


Figure 9.2: Virtual Button Setup

9.4 Sliders

The IQS9150/IQS9151 can implement up to eight virtual slider sensors. Unlike the virtual buttons, the sliders provide a position output showing the user location on the virtual slider. Each slider allows for up to 2 touch inputs simultaneously, allowing for innovative user interfaces on the multi-touch slider sensors.

9.4.1 Slider Output

The *Slider Output* is a positional output that ranges from 0 to the configured *Slider Resolution* value. This output value can be configured according to the implemented slider requirements. To allow the extremes of the slider to be easily activated by the user near the slider ends, a *Slider Deadzone* is configurable, which is an area at the extremes of the slider where an output (either 0 or the slider resolution value) is detected and output. The global parameter (applicable to all the virtual sliders) is also configured in terms of trackpad pixels and defines the trackpad coordinate distance that will provide an unchanged slider output before the slider effectively begins adjusting its output.

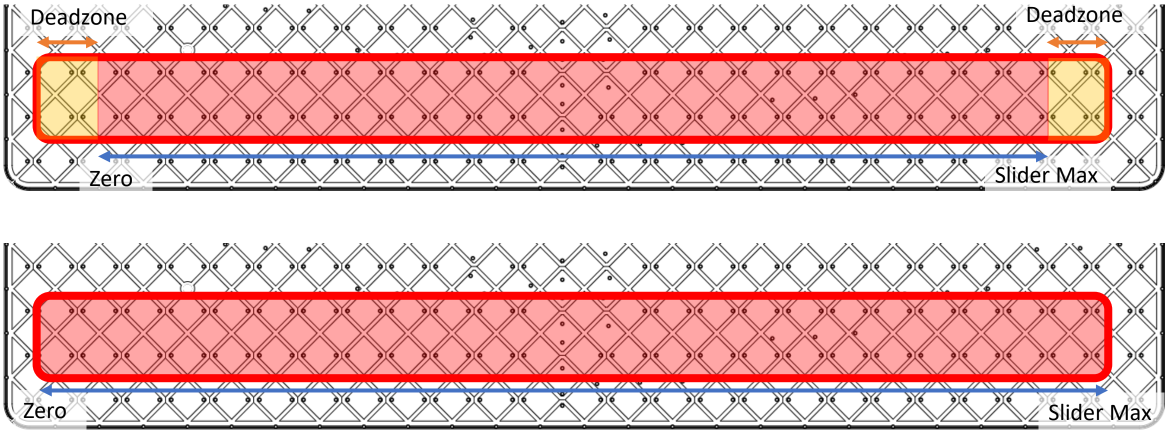


Figure 9.3: Virtual Slider Output and Deadzones

If a slider is not active (no touch on the slider), then in a similar manner to the trackpad X, Y output, it will output a slider position of 65535 (0xFFFF).

9.4.2 Slider Setup

The location, shape, and size of each slider are configured in the same manner as the virtual buttons by defining a top-left and bottom-right trackpad (X, Y) coordinate, *Slider Top-Left X/Y* and *Slider Bottom-Right X/Y*.

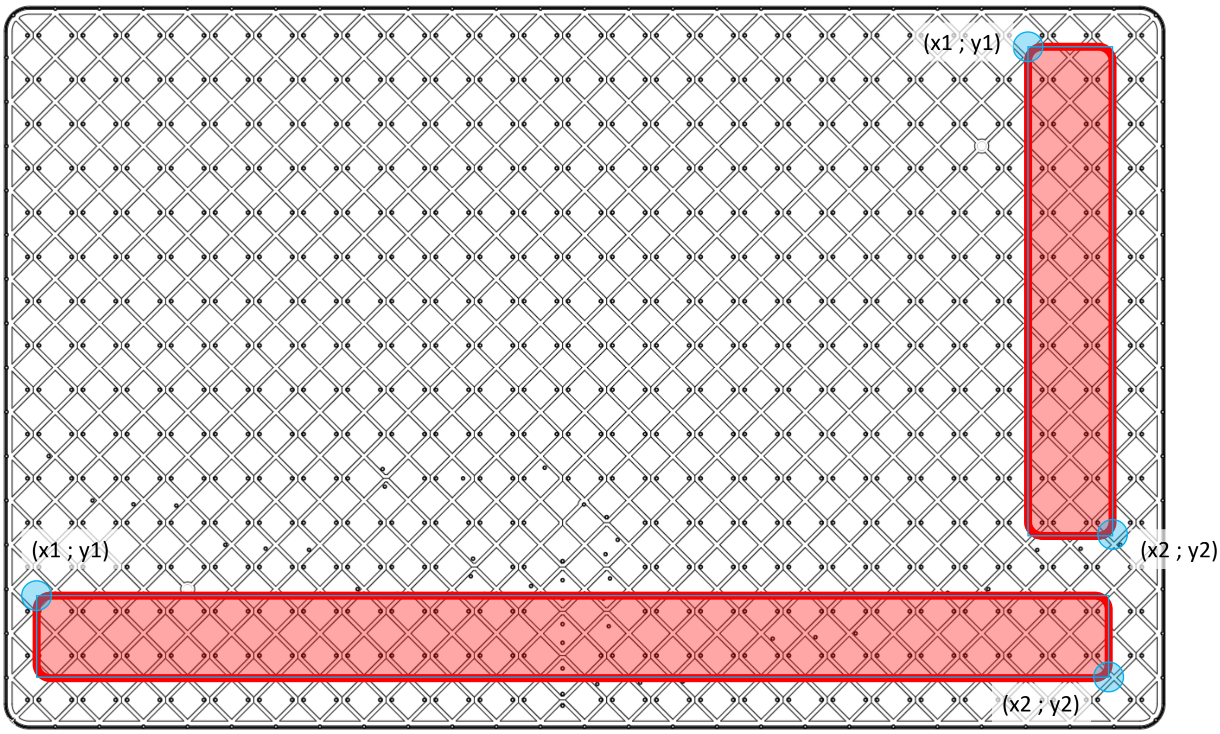


Figure 9.4: Virtual Slider Setup

The orientation of each slider (whether it is horizontal or vertical) is determined by the shape of the slider. If the size in x of the slider is larger than in y, then it is a horizontal slider calculated using trackpad x-coordinates; otherwise, it is a vertical slider calculated using the y-coordinates of the trackpad.

9.5 Wheels

Up to four-wheel sensors can be enabled on the IQS9150/IQS9151. A wheel sensor is a defined ring/doughnut shape that will output wheel coordinates around the wheel circumference for up to 2 fingers simultaneously.

9.5.1 Wheel Output

The *Wheel Output* register provides position output ranging from 0 to the configured *Wheel Resolution* value, similar to the sliders. The wheel output on an analogue watch starts from 0 at 3 o'clock and increases counter-clockwise. The maximum wheel output is also located at 3 o'clock, where it then wraps back to 0, as depicted in Figure 9.5 below.

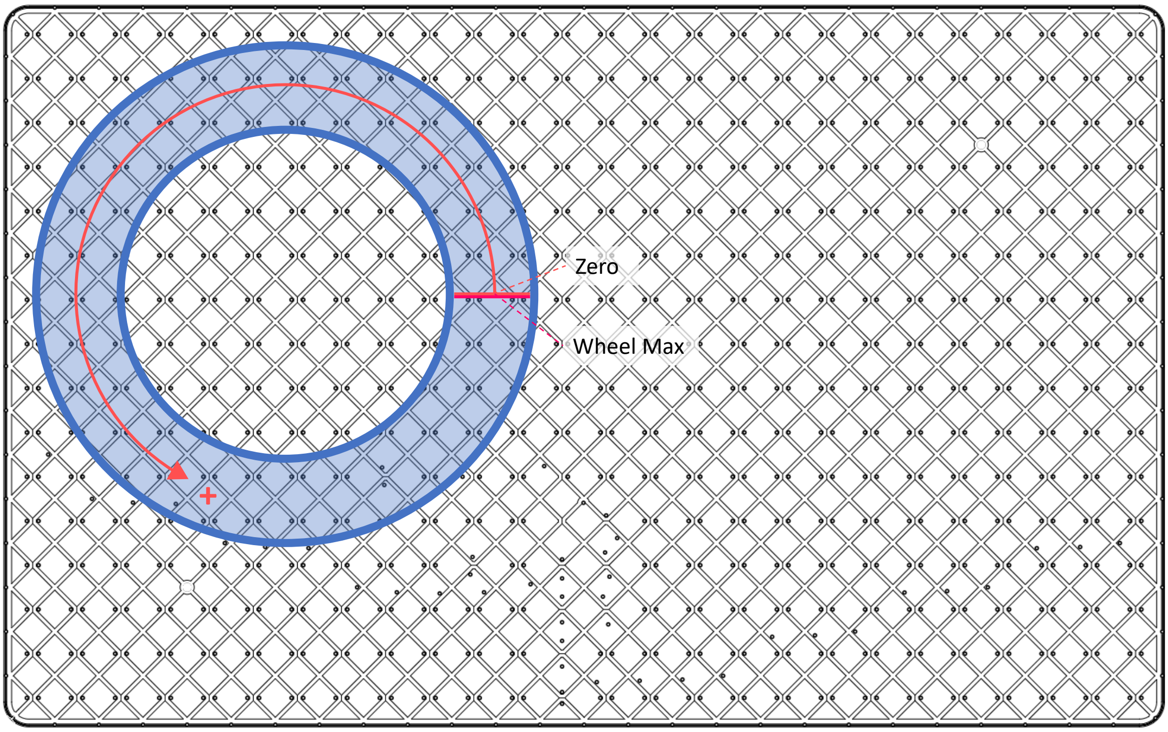


Figure 9.5: Virtual Wheel Output

Again, like the sliders, if no touch is sensed on the wheel, then it will output a wheel position of 65535 (0xFFFF).

9.5.2 Wheel Setup

The location and size of the wheel are configured by defining 3 parameters. Firstly, the *Wheel Centre X/Y* centre coordinate of the wheel location is configured. From this centre point, the *Wheel Inner Radius* and *Wheel Outer Radius* must be defined to indicate the wheel's inner and outer circumference boundaries.

Note: Since the trackpad X and Y coordinates are used to determine a virtual wheel, it is crucial to select the X and Y resolution such that they yield identical pixels per mm. This ensures that the calculation of the virtual wheel results in a round shape rather than an elongated oval shape.

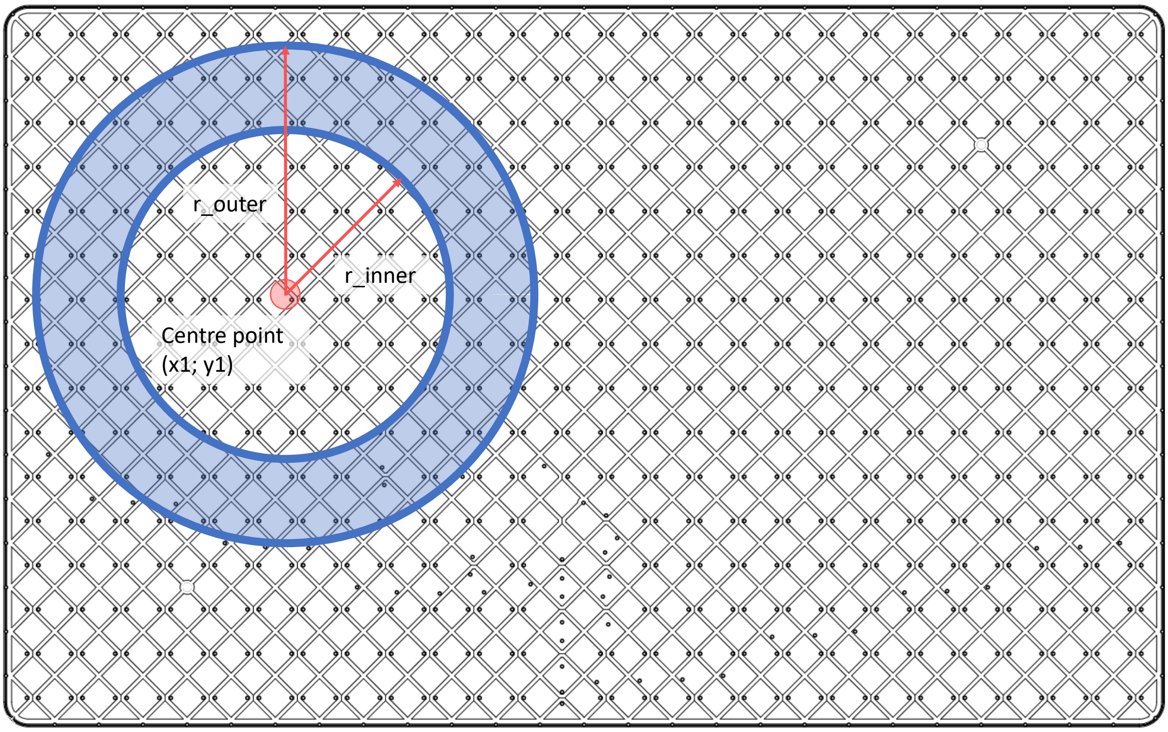


Figure 9.6: Virtual Wheel Setup

10 Hardware Settings

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

Note: Below some are described; the other hardware parameters are not discussed as they should only be adjusted under the guidance of Azoteq support engineers.

10.1 Main Oscillator

The main oscillator frequency can be configured to 14 MHz, 20 MHz or 24 MHz and is configured in the *Other Settings* register. When 20 MHz or 24 MHz is selected, the minimum VDD allowed increases; please see Section 4.3 for details.

10.2 Charge Transfer Frequency

The charge transfer frequency (f_{xfer}) can be configured using the IQS9150/IQS9151 PC GUI software. The charge transfer parameter section can be viewed in Appendix A.17. For high-resistance sensors (such as ITO), it might be needed to decrease f_{xfer}.

10.3 Reset

10.3.1 Reset Indication

After a reset, the *Show Reset* bit will be set in the *Info Flags* register by the system to indicate the reset event occurred. This bit will clear when the master sets the *Ack Reset* in the *System Control* register; if it becomes set again, the master will know a reset has occurred and can react appropriately.

Note: *Event Mode* will not work until the *Ack Reset* has been used to clear the *Show Reset* bit. This allows I^2C to always become active again if an unexpected reset has occurred, allowing the master to react accordingly to the *Show Reset* flag, such as writing the start-up settings if needed.

10.3.2 Software Reset

The IQS9150/IQS9151 can be reset through an I^2C command by setting the *SW Reset* bit in the *System Control* register. This reset will take effect shortly after the SW Reset bit has been set and the I^2C communication window terminates.

10.3.3 Hardware Reset

The MCLR pin (active low) can be used to reset the device when outside an I^2C communication window. For more details, see Section 4.7.

11 Additional Features

11.1 GUI for Parameter Setup

The Azoteq product GUI can configure the optimal settings required for the specific hardware. The device performance can be easily monitored and evaluated in the graphical environment until the optimal configuration is obtained.

Once the optimal configuration is obtained in the GUI, a header file can be exported containing the parameters to configure the IQS9150/IQS9151. To configure it correctly, these parameters need to be written to the device after every power-up.

Two bytes *Settings version number* are available so that the designer can label and identify the settings version. This allows the master to verify if the device firmware has the intended configuration as required.

11.1.1 Manual Start-up

The device will be programmed with defaults not necessarily applicable to the current application. It is recommended that the whole memory map is overwritten with all data from the header file to be sure all settings are as intended. Once this has been done, set the re-ATI bits for the trackpad and ALP channel so that the ATI can be executed on the intended settings.

11.2 Suspend

The IQS9150/IQS9151 can be placed into a suspended state, where no processing is performed, minimal power is consumed (<3 µA), and the device retains existing data. This state is entered after the communication session that sets the *Suspend* bit in the *System Control* register terminates.

The device can be woken from suspend by forcing I^{2}C communication (see Section 12.9.2) and clearing the suspend bit in that communication session. An automatic reseed of the trackpad is triggered after the device is woken from suspend since it cannot be guaranteed that the reference values are still relevant.

11.3 Watchdog Timer (WDT)

A watchdog timer is implemented to improve system reliability.

The working of this timer is as follows:

- A software timer t_{WDT} is linked to the LFTMR (Low Frequency Timer) running on the 'always on' Low Frequency Oscillator.
- This timer is reset at a strategic point in the main loop.
- Failing to reset this timer will cause the appropriate ISR (interrupt service routine) to run.
- This ISR performs a software-triggered POR (Power on Reset).
- The device will reset, performing a full cold boot.

11.4 RF Immunity

The IQS9150/IQS9151 has immunity to high-power RF noise. To improve the RF immunity, extra decoupling capacitors are suggested on V_{REG} and V_{DD}.

Place decoupling capacitors on V_{REG} and V_{DD} according to the reference schematic in Section 3. All decoupling capacitors should be placed as close as possible to the V_{DD} and V_{REG} pads.

If needed, series resistors can be added to Rx electrodes to reduce RF coupling into the sensing pads. Normally these are in the range of 100 Ω – 1 kΩ. PCB ground planes also improve noise immunity.

11.5 Switch Input

The switch input feature of the IQS9150/IQS9151 provides designers with the flexibility to implement switch functionality according to specific application requirements. The IQS9150/IQS9151 includes a dedicated Switch I/O pin. Developers can control the activation or deactivation of the switch functionality using the *Switch Enable* setting in the *Other Settings* register. The *Switch Polarity* setting allows users to configure the switch as active-high or active-low. For an active-low configuration, a pull-up resistor is recommended to ensure proper functionality. However, the behaviour of the switch ultimately depends on the external hardware setup to ensure that the input state (high or low) corresponds correctly to whether the switch is pressed or not pressed.

The *Switch Pressed* flag in the *Info Flags* register indicates the current status of the switch.

11.6 Additional Non-Trackpad Channels

Unused mutual capacitive channels can be used to design additional buttons or sliders.

Note: The channels will still provide XY data output, which can be ignored (or utilised) by the master.

Note: The additional sensors will have to use the same global ATI and sensitivity parameters, so careful sensor design is needed to ensure that these parameters are applicable.

It is suggested that the button sensor design is identical to the trackpad sensor, with the same overlay material and thickness. Please contact Azoteq if you consider this option.

11.7 Version Information

Version Information is subject to change before the product release. For up-to-date information, please get in touch with Azoteq.

12 I^2C Interface

12.1 I^2C Module Specification

The device features a standard two-wire I^2C interface, complemented by an RDY (ready interrupt) line, supporting a maximum bit rate of up to 1 Mbps. The memory structures accessible over the I^2C interface are byte-addressable with 16-bit address values. 16-bit or 32-bit values are packed with little-endian byte order and are stored in word-aligned addresses.

- Standard two-wire interface with additional RDY interrupt line
- *Fast-Mode Plus* I^2C with up to 1Mbps bit rate
- 7-bit device address
- 16-bit register address
- Little-endian

12.2 I^2C Address

The IQS9150/IQS9151 has a default I^2C address of 0x56.

Alternatively, the *I^2C Slave Address* parameter in the memory map can be updated to configure the I^2C slave addresses. An *I^2C Update Key* needs to be written to force the address update and prevent accidental overwriting of this. When the current I^2C communication window ends and the *I^2C Update Key* is set to 0xA3, the system will update the I^2C slave address. The *I^2C Update Key* register will automatically go back to 0x00 after the address has been updated.

12.2.1 Reserved I^2C Address

When communicating with the IQS9150/IQS9151, it will acknowledge (ACK) communication attempts made to an address derived from its slave address. This derived address is obtained by flipping the least significant bit of the slave address.

For example, if the slave address of the IQS9150/IQS9151 is 0x56 (1010110 in binary), the derived address for ACK would be 0x57 (1010111 in binary), obtained by changing the LSB from 0 to 1. However, it's important to note that this derived address is reserved for internal use and should not be used. Even though the device will acknowledge communication attempts to this address, it will not function as normal and therefore should be avoided.

12.3 Memory Map Addressing

All memory locations are 16-bit addressable in little-endian byte order.

12.4 Memory Map Data

Each 16-bit memory map address addresses a byte (8 bits), making the memory map byte-addressable. Since the data is packed in a little-endian sequence, a 16-bit value starting at, for example, address 0x1014 will have its least significant byte at address 0x1014 and its most significant byte at address 0x1015.

12.5 Read and Write Operations

12.5.1 I²C Read From Specific Address

The read operation is displayed in Figure 12.1. The master device watches for the IQS9150's/IQS9151's RDY line to drop low, signalling the availability of fresh data and a window for communication. It is always best to hold off on starting I²C transactions until the RDY line goes low. The master starts communication by sending a start condition, the device address, and a write command in response to the RDY interrupt. After receiving an acknowledgement from the IQS9150/IQS9151, the master device will send two bytes that specify the register address. The device address and a read command will be sent by the master after a repeated start condition. Then, while the master acknowledges each byte, the IQS9150/IQS9151 will keep transmitting data from the requested address. The read operation ends when the master generates a stop condition and does not acknowledge the last byte received.

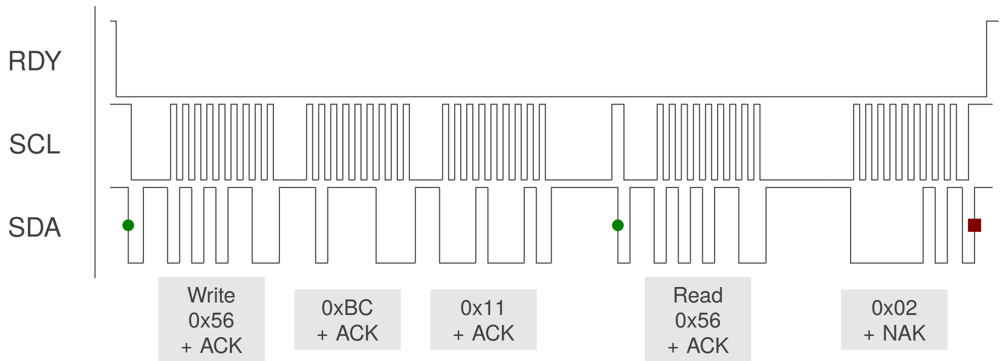


Figure 12.1: I²C Read Example - Read System Control Register 0x11BC Before Modifying

12.5.2 I²C Write To Specific Address

The write operation is displayed in Figure 12.2. Similar to the read transaction, when the RDY interrupt is triggered, the master initiates communication by sending a start condition followed by the device address along with a write command. The IQS9150/IQS9151 will respond with an acknowledgement, after which the master device will transmit two bytes defining the register address. The slave acknowledges the register address bytes. The master may then write a series of bytes to the register address and the addresses which follow, with each byte being acknowledged by the slave. The write operation is ended when the master produces a stop condition.

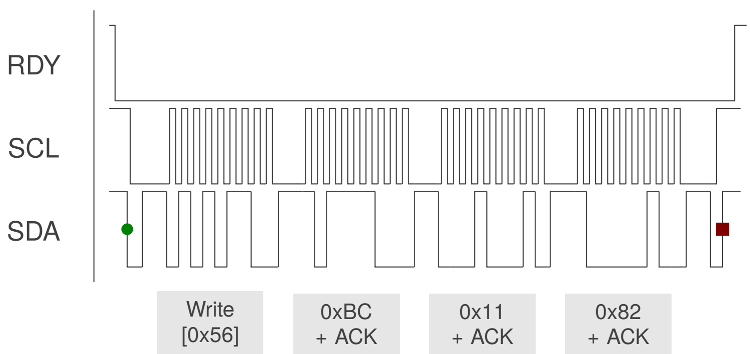


Figure 12.2: I²C Write Example - Write 0x82 (Ack Reset bit) to System Control Register 0x11BC

Note: When modifying registers, it's recommended to read the register first, make the necessary modifications, and then write the updated value back to the IQS9150/IQS9151 register to prevent unintentional bit settings.

- Read the System Control Register (0x11BC) as illustrated in Figure 12.1.
- Set the Ack Reset bit using the bitwise OR operator (Current register value OR 0x80).
- Example: 0x02 OR 0x80 = 0x82.
- Write the value 0x82 to Register 0x11BC as shown in Figure 12.2.

12.6 I²C Timeout

If the communication window is not serviced within the *I²C Timeout* period (in milliseconds), the session is ended (RDY goes HIGH), and processing continues as normal. This allows the system to continue and keep reference values up to date even if the master is not responsive; however, the corresponding data was missed/lost, and this should be avoided.

12.7 Terminate Communication

With the *Terminate Comms Window* setting cleared in the *Config Settings* register, a standard I²C STOP ends the current communication window. If multiple I²C transactions need to be done, then they should be strung together using repeated-start conditions instead of giving a STOP. This will allow the communication to occur in the same session. Allowing an I²C STOP to terminate the communication window is the recommended method and is illustrated in Figures 12.1 and 12.2.

The alternative option with the *Terminate Comms Window* setting set is that an I²C command is needed to terminate the communication window. For this configuration, an I²C STOP will NOT terminate the communication window. This can be done by writing 0xEEEE, followed by a STOP as follows:

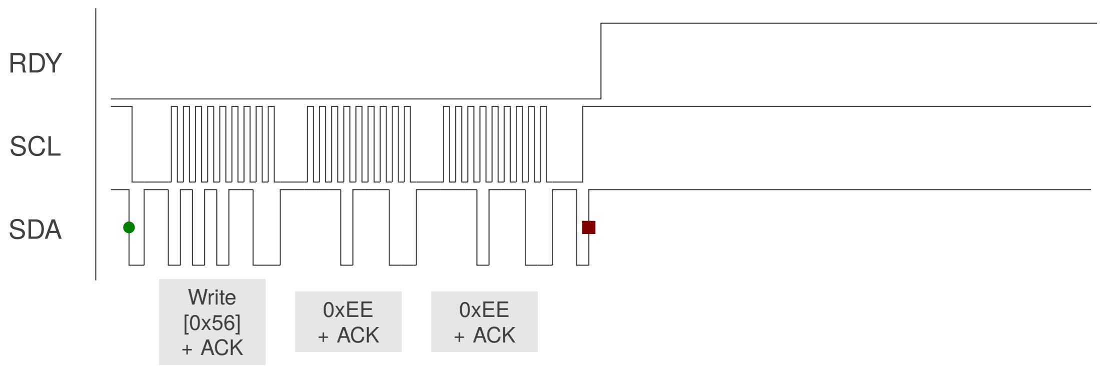


Figure 12.3: Terminate Comms Diagram

12.8 RDY / IRQ

The IQS9150/IQS9151 includes an open-drain active-low RDY signal, indicating when updated data and a communication window are ready. While the master can communicate with the device at any time according to the *Force Comms Method* setting, it is recommended to use the RDY signal for optimal power consumption. Integrating the RDY signal as an interrupt input allows the master MCU to efficiently read and write data.

The device provides both streaming and event modes. In streaming mode, the RDY line toggles continuously, whereas in event mode, the RDY toggles only when a specific event occurs. The types of events that trigger the RDY window are configurable in the *Config Settings* register.

12.9 Event Mode Communication

The device can be set up to bypass the communication window when no activity is sensed by setting the *Event Mode* bit in the *Config Settings* register. This is usually enabled since the master does not want to be interrupted unnecessarily during every cycle if no activity occurs. The communication will resume (RDY will indicate available data) if an enabled event occurs. It is recommended that the RDY be placed on an interrupt-on-pin-change input on the master.

Note that *Event Mode* will only implement if *Show Reset* has been cleared in *Info Flags*; see Section 10.3.1. An example of how to do this can be seen in Section 12.4.

12.9.1 Events

Numerous events can be individually enabled in the *Config Settings* register to trigger communication; they are:

- Gesture events (*Gesture Event*): Enabled gestures will trigger event.
- Trackpad events (*TP Event*): Event triggered if there is a change in X/Y value or if a finger is added or removed from the trackpad.
- Touch events (*TP Touch Event*): Event only triggers if a channel has a change in a touch state. This is mostly aimed at channels that are used for traditional buttons, where you want to know only when a status is changed.
- Re-ATI (*Re-ATI Event*): One communication cycle is given to indicate the re-ATI occurred.
- Proximity/Touch on ALP (*ALP Event*): Event given on state change.
- Switch event (*Switch Event*): With the switch input enabled, if the switch changes state, then this event will trigger.
- Snap events (*Snap Event*): Triggers if a snap channel has a change in state.

12.9.2 Force Communication / Polling

The master can initiate communication even while RDY is HIGH (inactive). The default method (*Force Comms Method* set to '0') is that the IQS9150/IQS9151 will clock stretch until an appropriate time to complete the I^2C transaction. The master firmware will not be affected (if clock stretching is correctly handled).

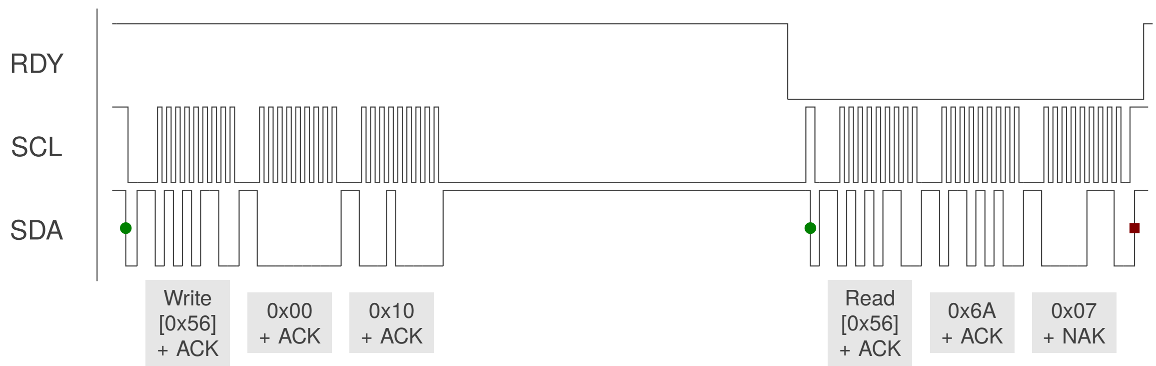


Figure 12.4: Clock Stretch Comms Diagram

If the associated clock stretching cannot be allowed, then an alternative *Force Comms Method* can be enabled in the *Config Settings* register. To achieve this, the master will communicate when RDY is not active (thus forcing comms), and it will write a comms request to the device. This comms request is as follows:

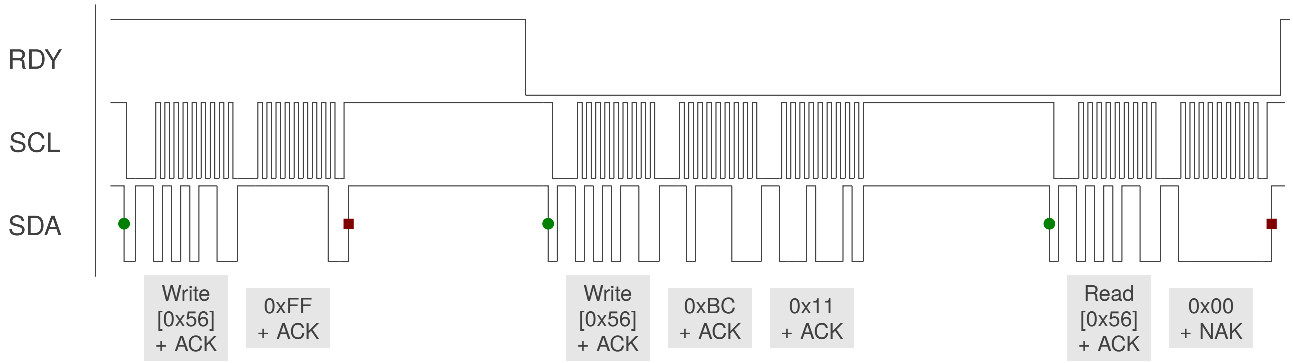


Figure 12.5: Force Comms Diagram

After this request for communication has been sent, the next available communication window will become available as normal (thus RDY going LOW).

For optimal program flow, it is suggested that RDY is used to sync new data. The forced/polling method is only recommended if the master must perform I^2C and Event Mode is active.

13 Ordering Information

13.1 Ordering Code

13.1.1 IQS9150

IQS9150 zzz ppb

Table 13.1: IQS9150 Order Code Description

IC NAME				IQS9150
CONFIGURATION	zzz	=	000	NRFND
			001	Default I^2C Address = 0x56
PACKAGE TYPE	pp	=	QF	QFN-52 Package
BULK PACKAGING	b	=	R	QFN-52 Reel (3000 pcs/reel)

Example : IQS9150-001QFR

13.1.2 IQS9151

IQS9151 zzz ppb

Table 13.2: IQS9151 Order Code Description

IC NAME				IQS9151
CONFIGURATION	zzz	=	000	Default I^2C Address = 0x56
PACKAGE TYPE	pp	=	QF	QFN-52 Package
BULK PACKAGING	b	=	R	QFN-52 Reel (3000 pcs/reel)

Example : IQS9151-000QFR

13.2 QFN52 Top Markings

13.2.1 IQS9150

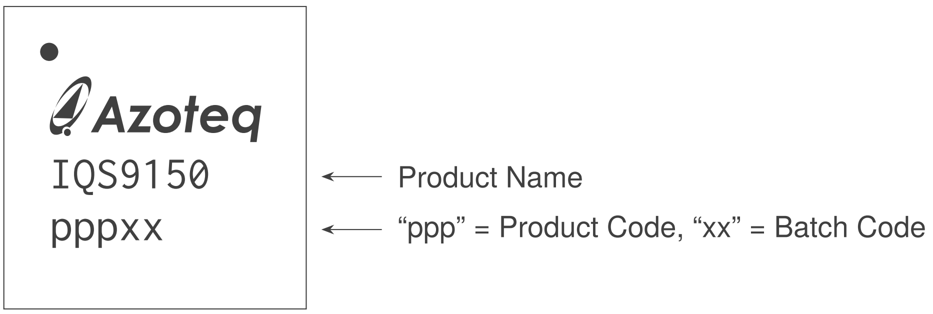


Figure 13.1: IQS9150-QFN52 Package Top Marking

13.2.2 IQS9151

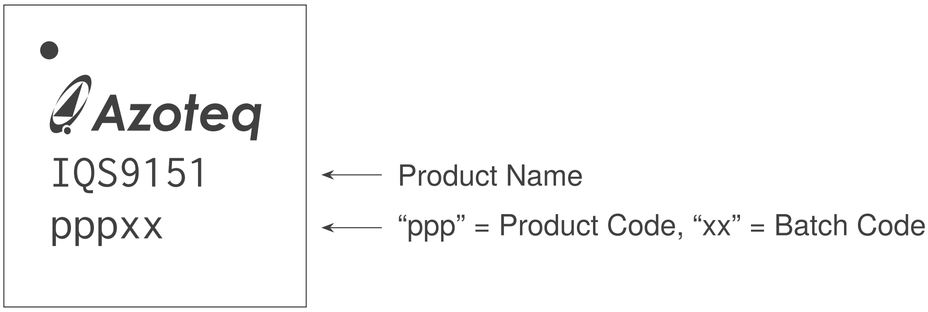


Figure 13.2: IQS9151-QFN52 Package Top Marking

13.2.3 Generic QFN52 Top Marking

When purchasing complete sub-assemblies from Azoteq, the IC on the assembly may have this generic top marking.

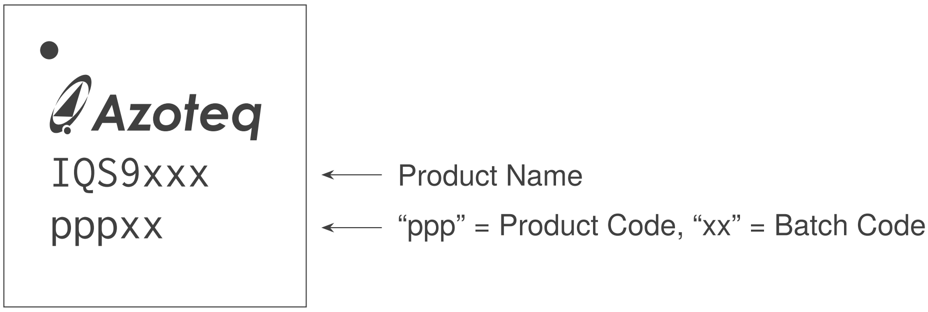


Figure 13.3: QFN52 Generic Package Top Marking

14 QFN52 Package Information

14.1 QFN52 Package Outline

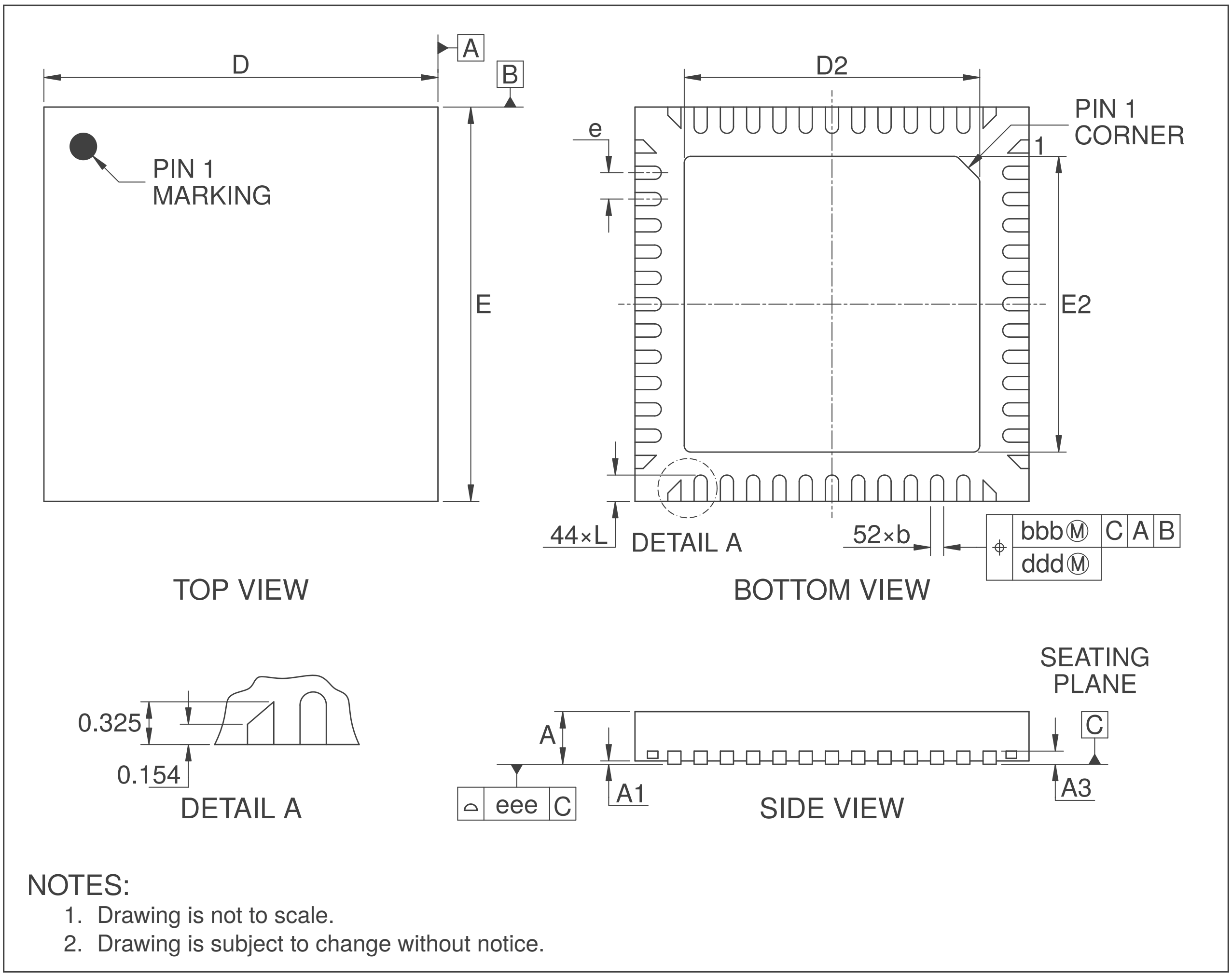


Figure 14.1: QFN52 Package Outline Visual Description

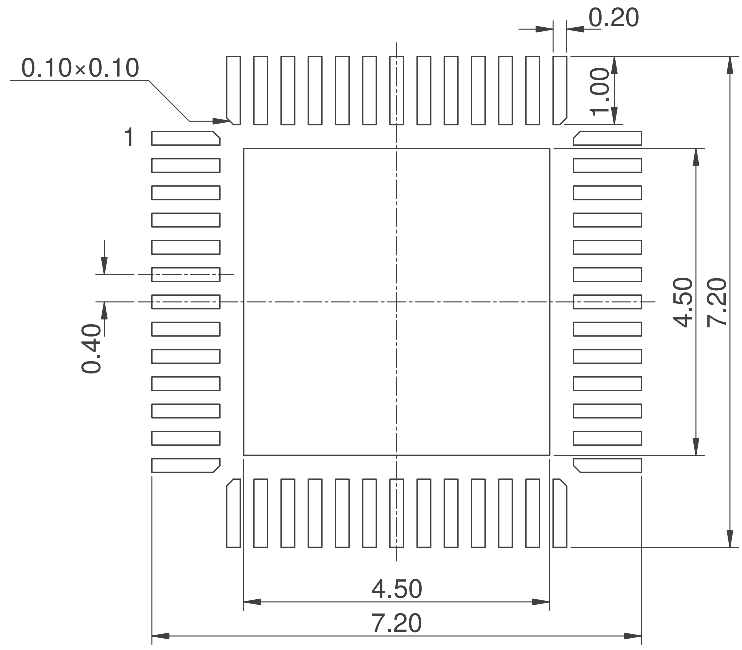
Table 14.1: QFN52 Package Dimensions [mm]

Dimension	Millimeters		
	Min	Typ	Max
A	0.70	0.75	0.80
A1	0.00	0.02	0.05
A3	0.20 REF		
D	5.95	6.00	6.05
E	5.95	6.00	6.05
D2	4.40	4.50	4.60
E2	4.40	4.50	4.60
b	0.15	0.20	0.25
e	0.40 BSC		
L	0.35	0.40	0.45

Table 14.2: QFN52 Package Tolerances [mm]

Tolerance	Millimeters
bbb	0.10
ddd	0.05
eee	0.08

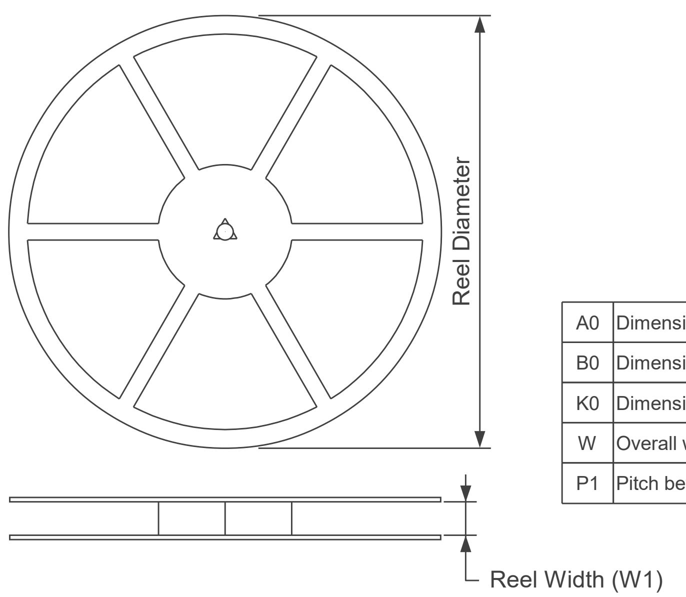
14.2 QFN52 Recommended Footprint

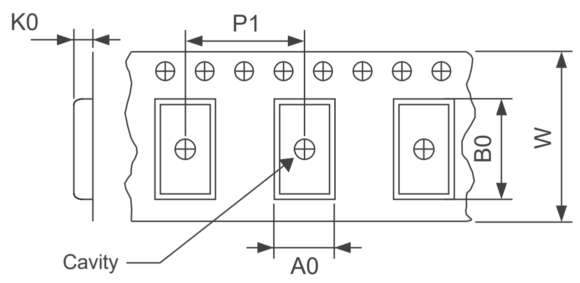


NOTES:
1. Dimensions are expressed in millimeters.
2. Drawing is not to scale.
3. Drawing is subject to change without notice.
4. Final dimensions may vary due to manufacturing tolerance considerations.

Figure 14.2: QFN52 Recommended Footprint

14.3 Tape and Reel Specifications





A0	Dimension designed to accommodate the component width
B0	Dimension designed to accommodate the component length
K0	Dimension designed to accommodate the component thickness
W	Overall width of the carrier tape
P1	Pitch between successive cavity centers

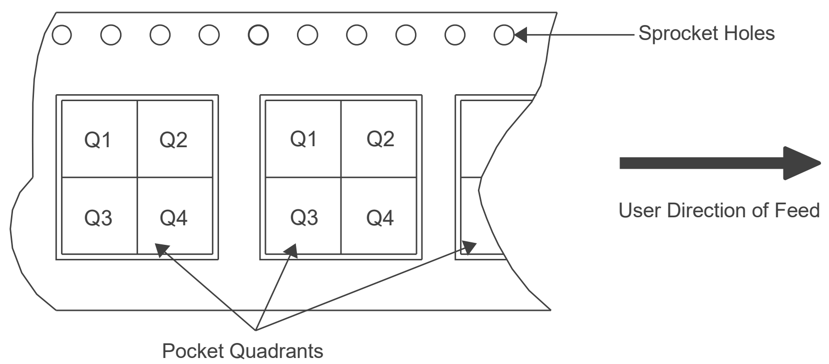


Figure 14.3: Tape and Reel Specification

Table 14.3: Tape and Reel Specifications

Package Type	Pins	Dimension [Millimeters]							Pin 1 Quadrant
		Reel Diameter	Reel Width	A0	B0	K0	P1	W	
QFN52	52	330.0	16.4	6.3	6.3	1.1	12.0	16.0	Q2

15 I²C Memory Map - Register Descriptions

For a more detailed description please see Appendix A.

Address	Length	Description	Notes
Read Only		**Version Information**	
0x1000 – 0x1013	20	Product Information	Appendix A.1
Read Only		**Device Data**	
0x1014	2	Relative X	Section 7.2.2
0x1016	2	Relative Y	
0x1018	2	Gesture X	Section 8.1 and 8.3
0x101A	2	Gesture Y	
0x101C	2	Single Finger Gestures	Appendix A.2
0x101E	2	Two-Finger Gestures	Appendix A.3
0x1020	2	Info Flags	Appendix A.4
0x1022	2	Trackpad Flags	Appendix A.5
0x1024	2	Finger 1 X-Coordinate	Section 7.2.3
0x1026	2	Finger 1 Y-Coordinate	
0x1028	2	Finger 1 Touch Strength	Section 7.2.4
0x102A	2	Finger 1 Area	Section 7.2.5
⋮	40	⋮	⋮
0x1054	2	Finger 7 X-Coordinate	Section 7.2.3
0x1056	2	Finger 7 Y-Coordinate	
0x1058	2	Finger 7 Touch Strength	Section 7.2.4
0x105A	2	Finger 7 Area	Section 7.2.5
Read Only		**Channel Data**	
0x105C	88	Touch Status	Appendix A.28
0x10B4	2	ALP Channel Count	Section 5.4.2
0x10B6	2	ALP Channel LTA	Section 5.5.2
0x10B8	26	ALP Individual Counts	Appendix A.6
0x10D2	88	Snap Status	Appendix A.28
0x112A	2	Button Output	Appendix A.7
0x112C	32	Slider Output	Appendix A.8
0x114C	16	Wheel Output	Appendix A.9
Read-Write		**Trackpad Configuration**	
0x115C	26	ALP ATI Compensation	Appendix A.10
0x1176	1	I²C Update Key	Section 12.2
0x1177	1	I²C Slave Address	
0x1178	1	Settings Minor Version	Section 11.1
0x1179	1	Settings Major Version	
0x117A	2	Trackpad ATI Multiplier/Dividers (Global)	Appendix A.11
0x117C	26	ALP ATI Multiplier/Dividers	
0x1196	2	Trackpad ATI Target	Section 5.7.1
0x1198	2	ALP ATI Target	Section 5.7.2
0x119A	2	ALP ATI Base Target	

Continued on next page

0x119C	2	Trackpad Negative Delta Re-ATI Value	Section 5.8
0x119E	2	Trackpad Positive Delta Re-ATI Value	
0x11A0	1	Trackpad Reference Drift Limit	
0x11A1	1	ALP LTA Drift Limit	
Read-Write		**Device Configuration**	
0x11A2	2	Active Mode Sampling Period (ms)	Section 6.1
0x11A4	2	Idle-Touch Mode Sampling Period (ms)	
0x11A6	2	Idle Mode Sampling Period (ms)	
0x11A8	2	LP1 Mode Sampling Period (ms)	
0x11AA	2	LP2 Mode Sampling Period (ms)	
0x11AC	2	Stationary Touch Timeout (s)	Section 6.2
0x11AE	2	Idle-Touch Mode Timeout (s)	
0x11B0	2	Idle Mode Timeout (s)	
0x11B2	2	LP1 Mode Timeout (s)	
0x11B4	2	Active to Idle Mode Timeout (ms)	
0x11B6	1	Re-ATI Retry Time (s)	Section 5.8.3
0x11B7	1	Reference Update Time (s)	Section 5.5.1
0x11B8	2	I^2C Timeout (ms)	Section 12.6
0x11BA	1	Snap Timeout	Section 5.6.2
0x11BB	1	Reserved (0x00)	-
0x11BC	2	System Control	Appendix A.12
0x11BE	2	Config Settings	Appendix A.13
0x11C0	2	Other Settings	Appendix A.14
0x11C2	4	ALP Setup	Appendix A.15
0x11C6	6	ALP Tx Enable	Appendix A.16
0x11CC	1	Touch Set Threshold Multiplier	Section 5.6.1
0x11CD	1	Touch Clear Threshold Multiplier	
0x11CE	1	ALP Output Threshold (Delta)	Section 5.6.3
0x11CF	1	ALP Auto-Prox Threshold (Delta)	
0x11D0	1	ALP Set Debounce	Section 5.6.4
0x11D1	1	ALP Clear Debounce	
0x11D2	1	Snap Set Threshold (Delta)	Section 5.6.2
0x11D3	1	Snap Clear Threshold (Delta)	
0x11D4	1	ALP Count Filter Beta - LP1 Mode	Section 5.4.2 and 5.5.2
0x11D5	1	ALP LTA Filter Beta - LP1 Mode	
0x11D6	1	ALP Count Filter Beta - LP2 Mode	
0x11D7	1	ALP LTA Filter Beta - LP2 Mode	
0x11D8	3	Trackpad Conversion Frequency	Appendix A.17
0x11DB	3	ALP Conversion Frequency	
0x11DE	2	Trackpad Hardware Settings	Appendix A.18
0x11E0	2	ALP Hardware Settings	
Read-Write		**Trackpad Configuration**	
0x11E2	1	Trackpad Settings	Appendix A.19
0x11E3	1	Total Rxs	Section 7.1.1
0x11E4	1	Total Txs	
0x11E5	1	Max Multi-Touches	Section 7.3

Continued on next page

0x11E6	2	X Resolution	Section 7.4
0x11E8	2	Y Resolution	
0x11EA	2	XY Dynamic Filter Bottom Speed	Section 7.8
0x11EC	2	XY Dynamic Filter Top Speed	
0x11EE	1	XY Dynamic Filter Bottom Beta	
0x11EF	1	XY Static Filter Beta	
0x11F0	1	Stationary Touch Movement Threshold	Section 7.5
0x11F1	1	Finger Split Factor	Section 7.6
0x11F2	1	X Trim Value	Section 7.9
0x11F3	1	Y Trim Value	
0x11F4	1	Jitter Filter Delta Threshold	Section 7.8.2
0x11F5	1	Finger Confidence Threshold	Section 7.10
Read-Write		**Gesture Configuration**	
0x11F6	2	Single Finger Gesture Enable	Appendix A.20 and A.21
0x11F8	2	Two-Finger Gesture Enable	
0x11FA	2	Tap Time (ms)	Section 8.1
0x11FC	2	Air Time (ms)	
0x11FE	2	Tap Distance (pixels)	
0x1200	2	Hold Time (ms)	Section 8.2
0x1202	2	Swipe Time (ms)	Section 8.3
0x1204	2	Swipe Initial X-Distance (pixels)	
0x1206	2	Swipe Initial Y-Distance (pixels)	
0x1208	2	Swipe Consecutive X-Distance (pixels)	
0x120A	2	Swipe Consecutive Y-Distance (pixels)	
0x120C	1	Swipe Angle (64tan(deg))	
0x120D	1	Scroll Angle (64tan(deg))	Section 8.6
0x120E	2	Zoom Initial Distance	Section 8.7
0x1210	2	Zoom Consecutive Distance	
0x1212	2	Scroll Initial Distance	Section 8.6
0x1214	2	Scroll Consecutive Distance	
0x1216	2	Palm Gesture Threshold	Section 8.4
Read-Write		**Trackpad Electrode & Channel Configuration**	
0x1218	46	RxTx Mapping	Appendix A.22
0x1246	88	Trackpad Channel Disable	Appendix A.28
0x129E	88	Trackpad Snap Channel Enable	
0x12F6	506	Individual Touch Threshold Adjustments	Appendix A.23
Read-Write		**Virtual Sensor Configuration**	
0x14F0	2	Number Of Virtual Sensors Enabled	Appendix A.24
Read-Write		**Virtual Button Configuration**	
0x14F2	2	Button 0 Top-Left X	Section 9.3
0x14F4	2	Button 0 Top-Left Y	
0x14F6	2	Button 0 Bottom-Right X	
0x14F8	2	Button 0 Bottom-Right Y	
⋮	112	⋮	⋮

Continued on next page

0x156A	2	Button 15 Top-Left X	Section 9.3
0x156C	2	Button 15 Top-Left Y	
0x156E	2	Button 15 Bottom-Right X	
0x1570	2	Button 15 Bottom-Right Y	
Read-Write		**Virtual Slider Configuration**	
0x1572	2	Slider Deadzone	Section 9.4.1
0x1574	2	Slider 0 Top-Left X	Section 9.4
0x1576	2	Slider 0 Top-Left Y	
0x1578	2	Slider 0 Bottom-Right X	
0x157A	2	Slider 0 Bottom-Right Y	
0x157C	2	Slider 0 Resolution	
⋮	60	⋮	⋮
0x15BA	2	Slider 7 Top-Left X	Section 9.4
0x15BC	2	Slider 7 Top-Left Y	
0x15BE	2	Slider 7 Bottom-Right X	
0x15C0	2	Slider 7 Bottom-Right Y	
0x15C2	2	Slider 7 Resolution	
Read-Write		**Virtual Wheel Configuration**	
0x15C4	2	Wheel 0 Centre X	Section 9.5
0x15C6	2	Wheel 0 Centre Y	
0x15C8	2	Wheel 0 Inner Radius	
0x15CA	2	Wheel 0 Outer Radius	
0x15CC	2	Wheel 0 Resolution	
⋮	20	⋮	⋮
0x15E2	2	Wheel 3 Centre X	Section 9.5
0x15E4	2	Wheel 3 Centre Y	
0x15E6	2	Wheel 3 Inner Radius	
0x15E8	2	Wheel 3 Outer Radius	
0x15EA	2	Wheel 3 Resolution	
Read-Write		**Engineering Configuration**	
0x2000	2	Engineering Configuration	Appendix A.25
0x2002	1	Main Oscillator Step Up	-
0x2003	1	Main Oscillator Step Down	
0x2004	2	Main Oscillator Step Threshold	
Read Only		**Trackpad Channel Information**	
0xA000	1012	Trackpad Count Values	Appendix A.27
0xB000	1012	Trackpad Reference Values	
0xC000	1012	Trackpad Delta Values	
0xD000	1012	Trackpad ATI Compensation Values	Appendix A.26
0xE000	6	Unique Identifier	-

A Memory Map Descriptions

A.1 Product Information

Table A.1: IQS9150 / IQS9151 Product Info

Address	Length	Description	IQS9150	IQS9151
0x1000	2	Product Number	0x076A	0x09BC
0x1002	2	Product Major Version	0x0001	0x0001
0x1004	2	Product Minor Version	0x0002	0x0000
0x1006	4	Product SHA	-	
0x100A	2	Library Number	0x037D	
0x100C	2	Library Major Version	0x0001	
0x100E	2	Library Minor Version	0x0000	
0x1010	4	Library SHA	-	

A.2 Single Finger Gestures (0x101C)

Bit	15	14	13	12	11	10	9	8
Description	Swipe and Hold Y-	Swipe and Hold Y+	Swipe and Hold X-	Swipe and Hold X+	Swipe Y-	Swipe Y+	Swipe X-	Swipe X+

Bit	7	6	5	4	3	2	1	0
Description	Reserved			Palm Gesture	Press-and-Hold	Triple Tap	Double Tap	Single Tap

- Bit 15: **Swipe and Hold Y-** - Swipe and hold in negative Y direction
  - 0: No gesture
  - 1: Swipe and hold in negative Y direction occurred
- Bit 14: **Swipe and Hold Y+** - Swipe and hold in positive Y direction
  - 0: No gesture
  - 1: Swipe and hold in positive Y direction occurred
- Bit 13: **Swipe and Hold X-** - Swipe and hold in negative X direction
  - 0: No gesture
  - 1: Swipe and hold in negative X direction occurred
- Bit 12: **Swipe and Hold X+** - Swipe and hold in positive X direction
  - 0: No gesture
  - 1: Swipe and hold in positive X direction occurred
- Bit 11: **Swipe Y-** - Swipe in negative Y direction
  - 0: No gesture
  - 1: Swipe in negative Y direction occurred
- Bit 10: **Swipe Y+** - Swipe in positive Y direction
  - 0: No gesture
  - 1: Swipe in positive Y direction occurred
- Bit 9: **Swipe X-** - Swipe in negative X direction
  - 0: No gesture
  - 1: Swipe in negative X direction occurred
- Bit 8: **Swipe X+** - Swipe in positive X direction
  - 0: No gesture
  - 1: Swipe in positive X direction occurred
- Bit 7-5: Unused
- Bit 4: **Palm Gesture** - Indicates a Palm gesture
  - 0: No gesture
  - 1: Palm gesture occurred

- Bit 3: **Press-and-Hold** - Indicates a Press-and-Hold gesture
  - 0: No gesture
  - 1: Press-and-Hold occurred
- Bit 2: **Triple Tap** - Indicates a triple tap gesture
  - 0: No gesture
  - 1: Triple tap occurred
- Bit 1: **Double Tap** - Indicates a double tap gesture
  - 0: No gesture
  - 1: Double tap occurred
- Bit 0: **Single Tap** - Indicates a single tap gesture
  - 0: No gesture
  - 1: Single tap occurred

A.3 Two Finger Gestures (0x101E)

Bit	15	14	13	12	11	10	9	8
Description	Reserved							

Bit	7	6	5	4	3	2	1	0
Description	Horizontal Scroll	Vertical Scroll	Zoom Out	Zoom In	Press-and-Hold	Triple Tap	Double Tap	Single Tap

- Bit 15-8: Unused
- Bit 7: **Horizontal Scroll** - Indicates a horizontal scroll gesture
  - 0: No gesture
  - 1: Horizontal scroll gesture occurred
- Bit 6: **Vertical Scroll** - Indicates a vertical scroll gesture
  - 0: No gesture
  - 1: Vertical scroll gesture occurred
- Bit 5: **Zoom Out** - Indicates a zoom out gesture
  - 0: No gesture
  - 1: Zoom out gesture occurred
- Bit 4: **Zoom In** - Indicates a zoom in gesture
  - 0: No gesture
  - 1: Zoom in gesture occurred
- Bit 3: **Press-and-Hold** - Indicates a Press-and-Hold gesture
  - 0: No gesture
  - 1: Press-and-Hold occurred
- Bit 2: **Triple Tap** - Indicates a triple tap gesture
  - 0: No gesture
  - 1: Triple tap occurred
- Bit 1: **Double Tap** - Indicates a double tap gesture
  - 0: No gesture
  - 1: Double tap occurred
- Bit 0: **Single Tap** - Indicates a single tap gesture
  - 0: No gesture
  - 1: Single tap occurred

A.4 Info Flags (0x1020)

Bit	15	14	13	12	11	10	9	8
Description	Snap Toggled	Switch Toggled	TP Touch Toggled	ALP Prox Toggled	Global Snap	Switch Pressed	Global TP Touch	ALP Prox Status

Bit	7	6	5	4	3	2	1	0
Description	Show Reset	ALP Re-ATI Occurred	ALP ATI Error	Re-ATI Occurred	ATI Error	Charging Mode		

- Bit 15: **Snap Toggled** - Snap detection status of a snap channel toggled
  - 0: Snap output did not toggle
  - 1: Snap output toggled
- Bit 14: **Switch Toggled** - Switch detection status toggled
  - 0: Switch input did not toggle
  - 1: Switch input toggled
- Bit 13: **TP Touch Toggled** - Touch detection status of a trackpad channel toggled
  - 0: Touch status did not toggle
  - 1: Touch status toggled
- Bit 12: **ALP Prox Toggled** - Prox detection status of ALP channel toggled
  - 0: ALP Prox status did not toggle
  - 1: ALP Prox status toggled
- Bit 11: **Global Snap** - Global snap detection status of any snap channel
  - 0: No output detected
  - 1: Output detected
- Bit 10: **Switch Pressed** - Switch pressed status
  - 0: No switch press detected
  - 1: Switch press detected
- Bit 9: **Global TP Touch** - Touch detection status of any TP channel
  - 0: No TP touch detected
  - 1: TP touch detected
- Bit 8: **ALP Prox Status** - Prox/Touch detection status of ALP channel
  - 0: No output detected
  - 1: Output detected
- Bit 7: **Show Reset** - Indicates a reset
  - 0: Reset indication has been cleared by host, writing to *Ack Reset*
  - 1: Reset has occurred and indication has not been cleared by host
- Bit 6: **ALP Re-ATI Occurred** - Alternate Low Power channel Re-ATI Status
  - 0: No re-ATI
  - 1: Re-ATI has just completed on alternate LP channel
- Bit 5: **ALP ATI Error** - Alternate Low Power ATI error status
  - 0: Most recent ATI process was successful
  - 1: Most recent ATI process was unsuccessful
- Bit 4: **Re-ATI Occurred** - Trackpad re-ATI status
  - 0: No re-ATI
  - 1: Re-ATI has just completed on the trackpad
- Bit 3: **ATI Error** - Error condition seen on latest trackpad ATI procedure
  - 0: Most recent ATI process was successful
  - 1: Most recent ATI process was unsuccessful
- Bit 2-0: **Charging Mode** - Indicates current mode
  - 000: Active mode
  - 001: Idle-touch mode
  - 010: Idle mode
  - 011: LP1 mode
  - 100: LP2 mode

A.5 Trackpad Flags (0x1022)

Bit	15	14	13	12	11	10	9	8
Description	Reserved	Finger 7 Confidence	Finger 6 Confidence	Finger 5 Confidence	Finger 4 Confidence	Finger 3 Confidence	Finger 2 Confidence	Finger 1 Confidence

Bit	7	6	5	4	3	2	1	0
Description	Saturation	Main Osc Stepped	Too Many Fingers	Movement Detected	Number of Fingers			

- Bit 15: Unused
- Bit 14-8: **Finger Confidence** - Confidence that the touch detected is a legitimate finger input
  - 0: Not confident that the touch is a finger input
  - 1: Confident that the touch is a finger input
- Bit 7: **Saturation** - Saturation detection status
  - 0: No saturation detected
  - 1: Saturation detected
- Bit 6: **Main Osc Stepped** - Frequency adjusted due to detected noise
  - 0: Oscillator not adjusted
  - 1: Oscillator adjusted
- Bit 5: **Too Many Fingers** - Indicates more than allowed fingers detected
  - 0: Number of fingers within maximum selected value
  - 1: Number of fingers exceeds maximum selected value
- Bit 4: **Movement Detected** - Trackpad finger movement detected
  - 0: No touches, or all touches stationary (see Section 7.5)
  - 1: Movement of finger(s) detected on trackpad
- Bit 3-0: **Number of Fingers** - Number of fingers detected on trackpad
  - 0000: No fingers on trackpad
  - 0001: 1 fingers active
  - 0010: 2 fingers active
  - 0011: 3 fingers active
  - 0100: 4 fingers active
  - 0101: 5 fingers active
  - 0110: 6 fingers active
  - 0111: 7 fingers active

A.6 ALP Individual Counts (0x10B8)

Address	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x10B8	ALP Counts Rx0 and/or Rx13															
0x10BA	ALP Counts Rx1 and/or Rx14															
0x10BC	ALP Counts Rx2 and/or Rx15															
0x10BE	ALP Counts Rx3 and/or Rx16															
0x10C0	ALP Counts Rx4 and/or Rx17															
0x10C2	ALP Counts Rx5 and/or Rx18															
0x10C4	ALP Counts Rx6 and/or Rx19															
0x10C6	ALP Counts Rx7 and/or Rx20															
0x10C8	ALP Counts Rx8 and/or Rx21															
0x10CA	ALP Counts Rx9 and/or Rx22															
0x10CC	ALP Counts Rx10 and/or Rx23															
0x10CE	ALP Counts Rx11 and/or Rx24															
0x10D0	ALP Counts Rx12 and/or Rx25															

A.7 Button Output (0x112A)

Bit	15	14	13	12	11	10	9	8
Description	Button 15	Button 14	Button 13	Button 12	Button 11	Button 10	Button 9	Button 8

Bit	7	6	5	4	3	2	1	0
Description	Button 7	Button 6	Button 5	Button 4	Button 3	Button 2	Button 1	Button 0

- Button state for Button 0 - Button 15
  - 0: Virtual touch button not pressed
  - 1: Virtual touch button pressed

A.8 Slider Output (0x112C)

Address	Length	Description
0x112C	2	Slider 0 output for Finger 1
0x112E	2	Slider 0 output for Finger 2
0x1130	2	Slider 1 output for Finger 1
0x1132	2	Slider 1 output for Finger 2
0x1134	2	Slider 2 output for Finger 1
0x1136	2	Slider 2 output for Finger 2
0x1138	2	Slider 3 output for Finger 1
0x113A	2	Slider 3 output for Finger 2
0x113C	2	Slider 4 output for Finger 1
0x113E	2	Slider 4 output for Finger 2
0x1140	2	Slider 5 output for Finger 1
0x1142	2	Slider 5 output for Finger 2
0x1144	2	Slider 6 output for Finger 1
0x1146	2	Slider 6 output for Finger 2
0x1148	2	Slider 7 output for Finger 1
0x114A	2	Slider 7 output for Finger 2

A.9 Wheel Output (0x114C)

Address	Length	Description
0x114C	2	Wheel 0 output for Finger 1
0x114E	2	Wheel 0 output for Finger 2
0x1150	2	Wheel 1 output for Finger 1
0x1152	2	Wheel 1 output for Finger 2
0x1154	2	Wheel 2 output for Finger 1
0x1156	2	Wheel 2 output for Finger 2
0x1158	2	Wheel 3 output for Finger 1
0x115A	2	Wheel 3 output for Finger 2

A.10 ALP ATI Compensation (0x115C)

Address	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	ALP Compensation Divider					ALP Compensation									
0x115C	Rx0 and/or Rx13															
0x115E	Rx1 and/or Rx14															
0x1160	Rx2 and/or Rx15															
0x1162	Rx3 and/or Rx16															
0x1164	Rx4 and/or Rx17															
0x1166	Rx5 and/or Rx18															
0x1168	Rx6 and/or Rx19															
0x116A	Rx7 and/or Rx20															
0x116C	Rx8 and/or Rx21															
0x116E	Rx9 and/or Rx22															
0x1170	Rx10 and/or Rx23															
0x1172	Rx11 and/or Rx24															
0x1174	Rx12 and/or Rx25															

- Bit 15: Unused
- Bit 14-10: **ALP Compensation Divider**
- Bit 9-0: **ALP Compensation**

A.11 Trackpad and ALP Multipliers / Divider (0x117A / 0x117C)

Address	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Fine Mult		Fine Divider					Coarse Multiplier				Coarse Divider				
0x117A	Trackpad															
0x117C	ALP Rx0 + Rx13															
0x117E	ALP Rx1 + Rx14															
0x1180	ALP Rx2 + Rx15															
0x1182	ALP Rx3 + Rx16															
0x1184	ALP Rx4 + Rx17															
0x1186	ALP Rx5 + Rx18															
0x1188	ALP Rx6 + Rx19															
0x118A	ALP Rx7 + Rx20															
0x118C	ALP Rx8 + Rx21															
0x118E	ALP Rx9 + Rx22															
0x1190	ALP Rx10 + Rx23															
0x1192	ALP Rx11 + Rx24															
0x1194	ALP Rx12 + Rx25															

- Bit 15-14: **Fine Multiplier**
  - 2-bit value between 1 and 2
  - Recommend to keep 1
- Bit 13-9: **Fine Divider**
  - 5-bit value between 1 and 21
  - Recommend to keep above 6

- Bit 8-5: **Coarse Multiplier**
  - 4-bit value between 1 and 15
  - Use Azoteq recommended sets as defined in GUI software
- Bit 4-0: **Coarse Divider**
  - 5-bit value between 1 and 31
  - Use Azoteq recommended sets as defined in GUI software

A.12 System Control (0x11BC)

Bit	15	14	13	12	11	10	9	8
Description	Tx Short Test	Reserved			Suspend	Reserved	SW Reset	Reserved

Bit	7	6	5	4	3	2	1	0
Description	Ack Reset	ALP Re-ATI	TP Re-ATI	ALP Reseed	TP Reseed	Mode Select		

- Bit 15: **Tx Short Test** - Tx short test
  - 0: Normal operation
  - 1: Enable Tx short test configuration
- Bit 14-12: Unused
- Bit 11: **Suspend** - Suspend IQS9150/IQS9151
  - 0: No action
  - 1: Place IQS9150/IQS9151 into suspend after the communication window terminates
- Bit 10: Unused
- Bit 9: **SW Reset** - Reset the device
  - 0: No action
  - 1: Reset device after communication window terminates
- Bit 8: Unused
- Bit 7: **Ack Reset** - Acknowledge a reset
  - 0: No action
  - 1: Acknowledge the reset by clearing *Show Reset* flag
- Bit 6: **ALP Re-ATI** - Queue a re-ATI on ALP channel
  - 0: No action
  - 1: Perform re-ATI when ALP channel is sensed again
- Bit 5: **TP Re-ATI** - Queue a re-ATI on trackpad channels
  - 0: No action
  - 1: Perform re-ATI when trackpad channels are sensed again
- Bit 4: **ALP Reseed** - Queue a reseed on ALP channel
  - 0: No action
  - 1: Reseed the LTA of the ALP channel when it is sensed again
- Bit 3: **TP Reseed** - Queue a reseed on trackpad channels
  - 0: No action
  - 1: Reseed reference values of the trackpad channels when it is sensed again
- Bit 2-0: **Mode Select** - Select mode (only applicable in *Manual Mode)*
  - 000: Active mode
  - 001: Idle-Touch mode
  - 010: Idle mode
  - 011: LP1 mode
  - 100: LP2 mode

A.13 Configuration Settings (0x11BE)

Bit	15	14	13	12	11	10	9	8
Description	Snap Event Enable	Switch Event Enable	TP Touch Event Enable	ALP Event Enable	Re-ATI Event Enable	TP Event Enable	Gesture Event Enable	Event Mode

Bit	7	6	5	4	3	2	1	0
Description	Manual Control	Terminate Comms Window	Sleep During Conversions	Force Comms Method	ALP Re-ATI Enable	TP Re-ATI Enable	ALP ATI Mode	Sleep Option

- Bit 15: **Snap Event Enable** - Enable snap triggering event
  - 0: Toggle of snap status does not trigger an event
  - 1: Toggle of snap status triggers an event
- Bit 14: **Switch Event Enable** - Enable switch triggering event
  - 0: Toggle of switch status does not trigger an event
  - 1: Toggle of switch status triggers an event
- Bit 13: **TP Touch Event Enable** - Enable trackpad touch triggering event
  - 0: Toggle of trackpad touch status does not trigger an event
  - 1: Toggle of trackpad touch status triggers an event
- Bit 12: **ALP Event Enable** - Enable alternate LP channel detection triggering event
  - 0: Toggle of alternate channel prox/touch status does not trigger an event
  - 1: Toggle of alternate channel prox/touch status triggers an event
- Bit 11: **Re-ATI Event Enable** - Enable Re-ATI generating an event
  - 0: Re-ATI occurring does not trigger an event
  - 1: Re-ATI occurring triggers an event
- Bit 10: **TP Event Enable** - Enable trackpad events
  - 0: Trackpad finger movement or finger up/down will not trigger event
  - 1: Trackpad finger movement or finger up/down will trigger event
- Bit 9: **Gesture Event Enable** - Enable gesture events
  - 0: Gestures will not trigger event
  - 1: Gestures will trigger event
- Bit 8: **Event Mode** - Enable event mode communication
  - 0: I²C is presented each cycle (except auto-prox cycles)
  - 1: I²C is only initiated when an enabled event occurs
- Bit 7: **Manual Control** - Override automatic mode switching
  - 0: Modes are automatically controlled by IQS9150/IQS9151 firmware (recommended)
  - 1: Manual control of modes are handled by host
- Bit 6: **Terminate Comms Window** - Alternative method to terminate comms (see Section 12.7)
  - 0: I²C stop ends comms
  - 1: *Terminate Comms Window* command, followed by an I²C STOP end comms
- Bit 5: **Sleep During Conversions** - Prevent processing during conversions
  - 0: Normal (recommended)
  - 1: Sleep (increases latency and current consumption)
- Bit 4: **Force Comms Method** - Force comms method selection (while RDY not LOW)
  - 0: Forcing comms will clock stretch until a comms window (Normal I²C outside RDY window)
  - 1: A comms window must be requested with a command (no stretching) outside comms window
- Bit 3: **ALP Re-ATI Enable** - Automatic Re-ATI on alternate LP channel
  - 0: Re-ATI is disabled for alternate LP channel
  - 1: Re-ATI is enabled for alternate LP channel (recommended)
- Bit 2: **TP Re-ATI Enable** - Automatic Re-ATI on trackpad
  - 0: Re-ATI is disabled for trackpad channels
  - 1: Re-ATI is enabled for trackpad channels (recommended)
- Bit 1: **ALP ATI Mode** - ALP ATI mode
  - 0: Compensation only
  - 1: Full ATI
- Bit 0: **Sleep Option** - Internal device sleep selection
  - 0: Deep sleep (recommended)
  - 1: Light sleep

A.14 Other Settings (0x11C0)

Bit	15	14	13	12	11	10	9	8
Description	Switch Enable	Switch Polarity	Prox Oscillator Adjustment		Reserved			

Bit	7	6	5	4	3	2	1	0
Description	Main Oscillator Selection		LP2 Auto-Prox Enable	LP1 Auto-Prox Enable	LP2 Auto-Prox Cycles		LP1 Auto-Prox Cycles	

- Bit 15: **Switch Enable** - Enable switch input
  - 0: Switch disabled
  - 1: Switch enabled
- Bit 14: **Switch Polarity** - Switch polarity selection
  - 0: Active-low
  - 1: Active-high
- Bit 13-12: **Prox Oscillator Adjustment** - Adjust Prox oscillator frequency
  - 00: Nominal (16 MHz)
  - 01: -10% (Main Osc 14 MHz) / -20% (Main Osc 20 MHz/24 MHz)
  - 10: -20% (Main Osc 14 MHz) / -30% (Main Osc 20 MHz/24 MHz)
  - 11: -30% (Main Osc 14 MHz) / -40% (Main Osc 20 MHz/24 MHz)
- Bit 11-8: Unused
- Bit 7-6: **Main Oscillator Selection** - Main oscillator frequency selection
  - 00: 14 MHz
  - 01: 20 MHz
  - 10: 24 MHz
- Bit 5: **LP2 Auto-Prox Enable** - Enable or disable LP2 Auto-Prox
  - 0: LP2 Auto-Prox disabled
  - 1: LP2 Auto-Prox enabled
- Bit 4: **LP1 Auto-Prox Enable** - Enable or disable LP1 Auto-Prox
  - 0: LP1 Auto-Prox disabled
  - 1: LP1 Auto-Prox enabled
- Bit 3-2: **LP2 Auto-Prox Cycles** - Number of LP2 auto-prox cycles
  - 00: 16
  - 01: 32
  - 10: 64
  - 11: 256
- Bit 1-0: **LP1 Auto-Prox Cycles** - Number of LP1 auto-prox cycles
  - 00: 16
  - 01: 32
  - 10: 64
  - 11: 256

A.15 ALP Setup (0x11C2)

Bit	31	30	29	28	27	26	25	24
Description	ALP Enable	ALP Count Filter Enable	ALP Sensing Method	Active Tx Shield Enable	Reserved		Rx25 Enable[a]	Rx24 Enable[a]

Bit	23	22	21	20	19	18	17	16
Description	Rx23 Enable[a]	Rx22 Enable[a]	Rx21 Enable[a]	Rx20 Enable[a]	Rx19 Enable[a]	Rx18 Enable[a]	Rx17 Enable[a]	Rx16 Enable[a]

Bit	15	14	13	12	11	10	9	8
Description	Rx15 Enable[a]	Rx14 Enable[a]	Rx13 Enable[a]	Rx12 Enable	Rx11 Enable	Rx10 Enable	Rx9 Enable	Rx8 Enable

Bit	7	6	5	4	3	2	1	0
Description	Rx7 Enable	Rx6 Enable	Rx5 Enable	Rx4 Enable	Rx3 Enable	Rx2 Enable	Rx1 Enable	Rx0 Enable

[a] Not applicable to IQS9151. Keep set to '0'.

- Bit 31: **ALP Enable** - Enable ALP channel
  - 0: ALP channel is disabled, trackpad channels active in LP1 and LP2
  - 1: ALP channel is enabled, ALP channel active in LP1 and LP2
- Bit 30: **ALP Count Filter Enable** - ALP count filter
  - 0: ALP channel count is unfiltered
  - 1: ALP count filter enabled
- Bit 29: **ALP Sensing Method** - ALP sensing method
  - 0: ALP is setup for self-capacitive sensing
  - 1: ALP is setup for mutual-capacitive sensing
- Bit 28: **Active Tx Shield Enable** - Configure Tx behaviour for self cap ALP setup
  - 0: All unused electrodes are grounded
  - 1: All ALP enabled Txs mimic Rx signal to reduce parasitic capacitance to GND
- Bit 27-26: Unused
- Bit 25-0: **Rx Enable** - ALP Rx electrodes
  - 0: Rx disabled (not used for ALP)
  - 1: Rx enabled (forms part of ALP sensor)

A.16 ALP Tx Enable (0x11C6)

Bit	47	46	45	44	43	42	41	40
Description	Reserved		Tx45 Enable	Reserved	Tx43 Enable	Tx42 Enable	Tx41 Enable	Tx40 Enable

Bit	39	38	37	36	35	34	33	32
Description	Tx39 Enable	Tx38 Enable	Tx37 Enable	Tx36 Enable	Tx35 Enable	Tx34 Enable	Tx33 Enable	Tx32 Enable[a]

Bit	31	30	29	28	27	26	25	24
Description	Tx31 Enable[a]	Tx30 Enable[a]	Tx29 Enable[a]	Tx28 Enable[a]	Tx27 Enable[a]	Tx26 Enable[a]	Tx25 Enable[a]	Tx24 Enable[a]

Bit	23	22	21	20	19	18	17	16
Description	Tx23 Enable[a]	Tx22 Enable[a]	Tx21 Enable[a]	Tx20 Enable[a]	Tx19 Enable[a]	Tx18 Enable[a]	Tx17 Enable[a]	Tx16 Enable[a]

Bit	15	14	13	12	11	10	9	8
Description	Tx15 Enable[a]	Tx14 Enable[a]	Tx13 Enable[a]	Tx12 Enable	Tx11 Enable	Tx10 Enable	Tx9 Enable	Tx8 Enable

Bit	7	6	5	4	3	2	1	0
Description	Tx7 Enable	Tx6 Enable	Tx5 Enable	Tx4 Enable	Tx3 Enable	Tx2 Enable	Tx1 Enable	Tx0 Enable

[a] Not applicable to IQS9151. Keep set to '0'.

- Bit 47-46: Unused
- Bit 44: Do not use, keep 0
- Bit 45, 43-0: **Tx Enable** - ALP Tx electrodes selection
  - 0: Tx disabled (not used for ALP)
  - 1: Tx enabled (forms part of ALP sensor)

A.17 Trackpad and ALP Conversion Frequency (0x11D8 and 0x11DB)

Address	Length	Description
0x11D8	1	Trackpad Fraction Value
0x11D9	1	Trackpad Period1 Value
0x11DA	1	Trackpad Period2 Value
0x11DB	1	ALP Fraction Value
0x11DC	1	ALP Period1 Value
0x11DD	1	ALP Period2 Value

- Please refer to Table A.7 below for the values to configure the desired conversion frequency.

Table A.7: Conversion Frequency Selections

Conversion Frequency (MHz)	Fraction Value	Period1 Value	Period1 Value
0.25	4	31	31
0.50	8	15	15
0.75	12	9	10
1.00	16	7	7
1.25	20	5	5
1.50	24	4	4
1.75	28	3	4
2.00	32	3	3
2.25	36	2	3
2.50	40	2	2
2.75	44	1	2
3.00	48	1	2
3.25	52	1	1
3.50	56	1	1
4.00	64	1	1

A.18 Trackpad and ALP Hardware Settings (0x11DE and 0x11E0)

Bit	15	14	13	12	11	10	9	8
Description	Initial Cycle Delay		SH Bias			Count Upper Limit		

Bit	7	6	5	4	3	2	1	0
Description	Cs Dis-charge Voltage	RF Filters	NM Out Static	NM In Static	Global SH Offset			

- Bit 15-14: **Initial Cycle Delay** - Initial cycles delay
  - 00: 16
  - 01: 32
  - 10: 64
  - 11: 256
- Bit 13-11: **SH Bias** - Sample-and-hold opamp bias current
  - 000: 5 µA
  - 001: 10 µA
  - 010: 15 µA
  - 011: 20 µA
  - 100: 15 µA
  - 101: 20 µA
  - 110: 25 µA
  - 111: 30 µA
- Bit 10-8: **Count Upper Limit** - Count upper limit (count value stops conversion after reaching this)
  - 000: 383
  - 001: 511
  - 010: 767
  - 011: 1023
  - 100: 2047

- Bit 7: **Cs Discharge Voltage** - Select internal Cs discharge voltage
  - 0: Discharge to 0 V (recommended for most cases)
  - 1: Discharge to 0.5 V
- Bit 6: **RF Filter** - Internal RF filters
  - 0: RF filters disabled
  - 1: RF filters enabled
- Bit 5: **NM Out Static** - NM out static
  - 0: Disabled (recommended)
  - 1: Enabled
- Bit 4: **NM In Static** - NM in static
  - 0: Disabled (recommended)
  - 1: Enabled
- Bit 3-0: **Global SH Offset** - Global SH offset
  - 0000: 0 mV
  - 0001: -2 mV
  - 0010: -4 mV
  - 0011: -6 mV
  - 0100: -8 mV
  - 0101: -10 mV
  - 0110: -12 mV
  - 0111: -14 mV
  - 1001: +2 mV
  - 1010: +4 mV
  - 1011: +6 mV
  - 1100: +8 mV
  - 1101: +10 mV
  - 1110: +12 mV
  - 1111: +14 mV

A.19 Trackpad Settings (0x11E2)

Bit	7	6	5	4	3	2	1	0
Description	Reserved	Area Filter Disable	Jitter Filter	IIR Static	IIR Filter	Switch XY Axis	Flip Y	Flip X

- Bit 7: Unused
- Bit 6: **Area Filter Disable** - Disable area filter
  - 0: Area filter on touch position enabled
  - 1: Area filter on touch position disabled
- Bit 5: **Jitter Filter** - Enable jitter filter
  - 0: XY jitter filter on touch position disabled
  - 1: XY jitter filter on touch position enabled
- Bit 4: **IIR Static** - IIR filtering method for the XY data points
  - 0: Damping factor for IIR filter is dynamically adjusted relative to XY movement (recommended)
  - 1: Damping factor for IIR filter is fixed
- Bit 3: **IIR Filter** - IIR filter
  - 0: XY IIR filter disabled
  - 1: XY IIR filter enabled (recommended)
- Bit 2: **Switch XY Axis** - Switch X and Y axes
  - 0: Rxs are arranged in trackpad columns (X), and Txs in rows (Y)
  - 1: Txs are arranged in trackpad columns (X), and Rxs in rows (Y)
- Bit 1: **Flip Y** - Flip Y output values
  - 0: Keep default Y values
  - 1: Invert Y output values
- Bit 0: **Flip X** - Flip X output values
  - 0: Keep default X values
  - 1: Invert X output values

A.20 Single Finger Gesture Enable (0x11F6)

Bit	15	14	13	12	11	10	9	8
Description	Swipe and Hold Y-	Swipe and Hold Y+	Swipe and Hold X-	Swipe and Hold X+	Swipe Y-	Swipe Y+	Swipe X-	Swipe X+

Bit	7	6	5	4	3	2	1	0
Description	Reserved			Palm Gesture	Press-and-Hold	Triple Tap	Double Tap	Single Tap

- Bit 15: **Swipe and Hold Y-** - Swipe and hold in negative Y direction
  - 0: Gesture disabled
  - 1: Gesture enabled
- Bit 14: **Swipe and Hold Y+** - Swipe and hold in positive Y direction
  - 0: Gesture disabled
  - 1: Gesture enabled
- Bit 13: **Swipe and Hold X-** - Swipe and hold in negative X direction
  - 0: Gesture disabled
  - 1: Gesture enabled
- Bit 12: **Swipe and Hold X+** - Swipe and hold in positive X direction
  - 0: Gesture disabled
  - 1: Gesture enabled
- Bit 11: **Swipe Y-** - Swipe in negative Y direction
  - 0: Gesture disabled
  - 1: Gesture enabled
- Bit 10: **Swipe Y+** - Swipe in positive Y direction
  - 0: Gesture disabled
  - 1: Gesture enabled
- Bit 9: **Swipe X-** - Swipe in negative X direction
  - 0: Gesture disabled
  - 1: Gesture enabled
- Bit 8: **Swipe X+** - Swipe in positive X direction
  - 0: Gesture disabled
  - 1: Gesture enabled
- Bit 7-5: Unused
- Bit 4: **Palm Gesture** - Palm gesture
  - 0: Gesture disabled
  - 1: Gesture enabled
- Bit 3: **Press-and-Hold** - Press-and-Hold gesture
  - 0: Gesture disabled
  - 1: Gesture enabled
- Bit 2: **Triple Tap** - Triple tap gesture
  - 0: Gesture disabled
  - 1: Gesture enabled
- Bit 1: **Double Tap** - Double tap gesture
  - 0: Gesture disabled
  - 1: Gesture enabled
- Bit 0: **Single Tap** - Single tap gesture
  - 0: Gesture disabled
  - 1: Gesture enabled

A.21 Two Finger Gesture Enable (0x11F8)

Bit	15	14	13	12	11	10	9	8
Description	Reserved							

Bit	7	6	5	4	3	2	1	0
Description	Horizontal Scroll	Vertical Scroll	Zoom Out	Zoom In	Press-and-Hold	Triple Tap	Double Tap	Single Tap

- Bit 15-8: Unused
- Bit 7: **Horizontal Scroll** - Indicates a horizontal scroll gesture
  - 0: Gesture disabled
  - 1: Gesture enabled
- Bit 6: **Vertical Scroll** - Indicates a vertical scroll gesture
  - 0: Gesture disabled
  - 1: Gesture enabled
- Bit 5: **Zoom Out** - Indicates a zoom out gesture
  - 0: Gesture disabled
  - 1: Gesture enabled
- Bit 4: **Zoom In** - Indicates a zoom in gesture
  - 0: Gesture disabled
  - 1: Gesture enabled
- Bit 3: **Press-and-Hold** - Indicates a Press-and-Hold gesture
  - 0: Gesture disabled
  - 1: Gesture enabled
- Bit 2: **Triple Tap** - Indicates a triple tap gesture
  - 0: Gesture disabled
  - 1: Gesture enabled
- Bit 1: **Double Tap** - Indicates a double tap gesture
  - 0: Gesture disabled
  - 1: Gesture enabled
- Bit 0: **Single Tap** - Indicates a single tap gesture
  - 0: Gesture disabled
  - 1: Gesture enabled

A.22 RxTx Mapping (0x1218)

Address	Length	Description
0x1218	1	RxTx Mapping 0
0x1219	1	RxTx Mapping 1
0x121A	1	RxTx Mapping 2
0x121B	1	RxTx Mapping 3
0x121C	1	RxTx Mapping 4
0x121D	1	RxTx Mapping 5
0x121E	1	RxTx Mapping 6
0x121F	1	RxTx Mapping 7
0x1220	1	RxTx Mapping 8
0x1221	1	RxTx Mapping 9
0x1222	1	RxTx Mapping 10
0x1223	1	RxTx Mapping 11
0x1224	1	RxTx Mapping 12
0x1225	1	RxTx Mapping 13
0x1226	1	RxTx Mapping 14

Continued on next page...

0x1227	1	RxTx Mapping 15
0x1228	1	RxTx Mapping 16
0x1229	1	RxTx Mapping 17
0x122A	1	RxTx Mapping 18
0x122B	1	RxTx Mapping 19
0x122C	1	RxTx Mapping 20
0x122D	1	RxTx Mapping 21
0x122E	1	RxTx Mapping 22
0x122F	1	RxTx Mapping 23
0x1230	1	RxTx Mapping 24
0x1231	1	RxTx Mapping 25
0x1232	1	RxTx Mapping 26
0x1233	1	RxTx Mapping 27
0x1234	1	RxTx Mapping 28
0x1235	1	RxTx Mapping 29
0x1236	1	RxTx Mapping 30
0x1237	1	RxTx Mapping 31
0x1238	1	RxTx Mapping 32
0x1239	1	RxTx Mapping 33
0x123A	1	RxTx Mapping 34
0x123B	1	RxTx Mapping 35
0x123C	1	RxTx Mapping 36
0x123D	1	RxTx Mapping 37
0x123E	1	RxTx Mapping 38
0x123F	1	RxTx Mapping 39
0x1240	1	RxTx Mapping 40
0x1241	1	RxTx Mapping 41
0x1242	1	RxTx Mapping 42
0x1243	1	RxTx Mapping 43
0x1244	1	RxTx Mapping 44
0x1245	1	Reserved (0x00)

- Byte 44-0: **RxTxMapping** - Trackpad Rx and Tx mapping, see Section 7.1.4
- **Note:** The value 44 (0x2C) may not be written to any of the registers

A.23 Individual Touch Threshold Adjustments (0x12F6)

Address	Length	Description
0x12F6	1	CH0 Touch Threshold Adjustment
0x12F7	1	CH1 Touch Threshold Adjustment
0x12F8	1	CH2 Touch Threshold Adjustment
⋮	500	⋮
0x14ED	1	CH503 Touch Threshold Adjustment
0x14EE	1	CH504 Touch Threshold Adjustment
0x14EF	1	CH505 Touch Threshold Adjustment

- **CH Touch Threshold Adjustment** - Signed 8-bit values, see Section 5.6.1
  - 0000 0000: Threshold Multiplier + 0
  - 0000 0001: Threshold Multiplier + 1
  - 0000 0010: Threshold Multiplier + 2
  - 0000 0011: Threshold Multiplier + 3

  ⋮
  - 0111 1111: Threshold Multiplier + 127
  - 1000 0000: Threshold Multiplier - 128
  - 1000 0001: Threshold Multiplier - 127

  ⋮
  - 1111 1101: Threshold Multiplier - 3
  - 1111 1110: Threshold Multiplier - 2
  - 1111 1111: Threshold Multiplier - 1

A.24 Number Of Virtual Sensors Enabled (0x14F0)

Bit	15	14	13	12	11	10	9	8
Description	Number of Wheels				Number of Sliders			

Bit	7	6	5	4	3	2	1	0
Description	Number of Buttons							

- Bit 15-12: **Number of Wheels** - Number of virtual wheels enabled, see Section 9.1
- Bit 11-8: **Number of Sliders** - Number of virtual sliders enabled
- Bit 7-0: **Number of Buttons** - Number of virtual buttons enabled

A.25 Engineering Configuration (0x2000)

Bit	15	14	13	12	11	10	9	8
Description	Reserved					Main Osc Status		Reserved

Bit	7	6	5	4	3	2	1	0
Description	Reserved							Main Osc Stepping Enable

- Bit 15-11: Unused
- Bit 10-9: **Main Oscillator Status**
- Bit 8-1: Unused
- Bit 0: **Main Oscillator Stepping Enable**

A.26 Trackpad ATI Compensation (0xD000)

Address	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	Trackpad Compensation Divider					Trackpad Compensation									
0xD000	CH0															
0xD002	CH1															
0xD004	CH2															
⋮	⋮															

- Bit 15: Unused
- Bit 14-10: **Trackpad Compensation Divider**
- Bit 9-0: **Trackpad Compensation**

A.27 Count / Delta / Reference Data

For the count, delta, and reference values (2 bytes per channel), the structure is defined as shown in the table below. The data in the table is in the format of Count/Delta/Reference Value [*Row/Tx*][*Column/Rx*]. Table A.10 is valid for a 26 Rx by 19 Tx trackpad.

Table A.10: Count / Delta / Reference Value Bytes for a 26 Rx by 19 Tx Trackpad

Byte Number	Data	Description
X	Count/Delta/Reference Value[0][0] - Low Byte	Count, delta or reference at 1st Tx, and 1st Rx (thus top left)
X+1	Count/Delta/Reference Value[0][0] - High Byte	
X+2	Count/Delta/Reference Value[0][1] - Low Byte	Count, delta or reference at 1st Tx, and 2nd Rx
X+3	Count/Delta/Reference Value[0][1] - High Byte	
⋮	⋮	
X+986	Count/Delta/Reference Value[18][25] - Low Byte	Count, delta or reference at 19th Tx, and 26th Rx (thus bottom right)
X+987	Count/Delta/Reference Value[18][25] - High Byte	

For a trackpad with fewer than 26 Rxs, the values are densely packed based on the setting for *Total Rxs*. Consequently, the subsequent values become available immediately after reaching the specified *Total Rxs* value. For instance, in a 4 Rx by 2 Tx trackpad configuration, the values are packed from address X to X+15, as illustrated in Table A.11 below:

Table A.11: Count / Delta / Reference Value Bytes for a 4 Rx by 2 Tx Trackpad

Byte Number	Data	Description
X	Count/Delta/Reference Value[0][0] - Low Byte	Count, delta or reference at 1st Tx, and 1st Rx (thus top left)
X+1	Count/Delta/Reference Value[0][0] - High Byte	
X+2	Count/Delta/Reference Value[0][1] - Low Byte	Count, delta or reference at 1st Tx, and 2nd Rx
X+3	Count/Delta/Reference Value[0][1] - High Byte	
X+4	Count/Delta/Reference Value[0][2] - Low Byte	Count, delta or reference at 1st Tx, and 3rd Rx
X+5	Count/Delta/Reference Value[0][2] - High Byte	
X+6	Count/Delta/Reference Value[0][3] - Low Byte	Count, delta or reference at 1st Tx, and 4th Rx
X+7	Count/Delta/Reference Value[0][3] - High Byte	
Step to next Row/Tx		

Continued on next page...

X+8	Count/Delta/Reference Value[1][0] - Low Byte	Count, delta or reference at 2nd Tx, and 1st Rx
X+9	Count/Delta/Reference Value[1][0] - High Byte	
X+10	Count/Delta/Reference Value[1][1] - Low Byte	Count, delta or reference at 2nd Tx, and 2nd Rx
X+11	Count/Delta/Reference Value[1][1] - High Byte	
X+12	Count/Delta/Reference Value[1][2] - Low Byte	Count, delta or reference at 2nd Tx, and 3rd Rx
X+13	Count/Delta/Reference Value[1][2] - High Byte	
X+14	Count/Delta/Reference Value[1][3] - Low Byte	Count, delta or reference at 2nd Tx, and 4th Rx (thus bottom right)
X+15	Count/Delta/Reference Value[1][3] - High Byte	

A.28 Individual Channel Status / Config Bit Definitions

For all status outputs or configuration parameters where one bit relates to one channel, the structure is defined as shown in Table A.12 below. Each row has a 32-bit value where the status/config of each bit corresponds to the status/config of the corresponding column.

Table A.12: Status Bytes

Byte Number	Data	Description
X	Status/Config [Row0] - Byte 0	
X+1	Status/Config [Row0] - Byte 1	
X+2	Status/Config [Row0] - Byte 2	
X+3	Status/Config [Row0] - Byte 3	
X+4	Status/Config [Row1] - Byte 0	
X+5	Status/Config [Row1] - Byte 1	
X+6	Status/Config [Row1] - Byte 2	
X+7	Status/Config [Row1] - Byte 3	
⋮	⋮	
X+84	Status/Config [Row21] - Byte 0	
X+85	Status/Config [Row21] - Byte 1	
X+86	Status/Config [Row21] - Byte 2	
X+87	Status/Config [Row21] - Byte 3	

	Byte 3								Byte 2							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RowZ	-	-	-	-	-	-	Col25	Col24	Col23	Col22	Col21	Col20	Col19	Col18	Col17	Col16

	Byte 1								Byte 0							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RowZ	Col15	Col14	Col13	Col12	Col11	Col10	Col9	Col8	Col7	Col6	Col5	Col4	Col3	Col2	Col1	Col0

Note: If the XY axes are switched, these registers do NOT switch. This means that the bits will always link to Rxs, and the registers will always link to Txs.

Table A.13: Channel Status/Config Bit Definitions

Parameter	Bit = 0	Bit = 1
Touch Status	Channel does not have a touch	Channel does have a touch
Snap Status	Channel does not have a snap	Channel does have a snap
Channel Disable	Trackpad channel enabled	Trackpad channel disabled
Snap Enable	Snap feature disabled on channel	Snap feature enabled on channel

B Revision History

Release	Date	Comments
v1.0	2024/04/23	Initial document released
v1.1	2024/10/14	Added all IQS9151 product information Minor formatting and grammar changes IQ9150 FW updated from v1.0 to v1.2 (Table A.1) Updated IQS9150 ordering codes (Section 13.1)

Contact Information

	USA	China	Taiwan	South Africa
Physical Address	11940 Jollyville Suite 120-S Austin TX 78759 USA	Room 501A, Block A T-Share International Centre Taoyuan Road, Nanshan District Shenzhen, Guangdong, PRC	Xintai 5th Road, Sec. 1 No. 99, 9F-12C Xizhi District 221001 New Taipei City Taiwan	1 Bergsig Avenue Paarl 7646 South Africa
Postal Address	11940 Jollyville Suite 120-S Austin TX 78759 USA	Room 501A, Block A T-Share International Centre Taoyuan Road, Nanshan District Shenzhen, Guangdong, PRC	Xintai 5th Road, Sec. 1 No. 99, 9F-12C Xizhi District 221001 New Taipei City Taiwan	PO Box 3534 Paarl 7620 South Africa
Tel	+1 512 538 1995	+86 755 8303 5294 ext 808	+886 932 219 444	+27 21 863 0033
Email	info@azoteq.com	info@azoteq.com	info@azoteq.com	info@azoteq.com

Visit www.azoteq.com
for a list of distributors and worldwide representation.