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

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

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

Contact us

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

MTCH6102

MTCH6102 Low-Power Projected Capacitive Touch Controller

Description:

Microchip's MTCH6102 is a turnkey projected capacitive touch controller that simplifies adding gestures to touch interface designs with industry-leading low-power performance. It utilizes up to 15 channels to support taps, swipes, and scrolling on XY touch pads and touch screens. MTCH6102 allows designers to quickly and easily integrate projected capacitive touch into their cost-sensitive, low-power application. MTCH6102 provides developers with a flexible touch-sensing solution to optimize common constraints of size, power and cost that are critical to applications such as wearable devices, remote controls, gaming devices and track pads.

Applications:

- Wearable Devices such as Headphones, Watches, Fitness Wristbands
- Track Pads and Computer Peripherals
- Input Devices with Configurable Button, Keypad or Scrolling Functions
- Any Interface with Single-Finger Gestures to Swipe, Scroll, or Doubletap Controls
- Home Automation Control Panels
- Security Control Keypads
- Automotive Center Stack Controls
- Gaming Devices
- Remote Control Touch Pads

Touch Sensor Support:

- Up to 15 Channels
- Sensor Sizes up to 120 mm (4.7")
- Individual Channel Tuning for Optimal Sensitivity
- Works with Printed Circuit Board (PCB) Sensors, Film, Glass and Flexible Printed Circuit (FPC) Sensors

Cover Layer Support:

- Plastic: up to 3 mm
- Glass: up to 5 mm

Touch Performance:

- >200 Reports per Second (configurable)
- 12-Bit Resolution Coordinate Reporting

Touch Features:

- Gesture Detection and Reporting
- Self-Capacitance Signal Acquisition
- Multiple Built-in Filtering Options

Power Management:

- Configurable Sleep/Idle Frame Rates
- Standby mode <500 nA (typical)
- Active mode <12 uA possible

Communication Interface:

- I^2C™ (up to 400 kbps)
- Both Polling and Interrupt Schemes Supported
- Sync Signal Allows for Host Frame Detection
- Field Upgradeable over I^2C

Operating Conditions:

- 1.8V to 3.6V, -40°C to +85°C

Package Types:

- 28-Pin SSOP
- 28-Pin UQFN

Table of Contents

TO OUR VALUED CUSTOMERS

It is our intention to provide our valued customers with the best documentation possible to ensure successful use of your Microchip products. To this end, we will continue to improve our publications to better suit your needs. Our publications will be refined and enhanced as new volumes and updates are introduced.

If you have any questions or comments regarding this publication, please contact the Marketing Communications Department via E-mail at **docerrors@microchip.com**. We welcome your feedback.

Most Current Data Sheet

To obtain the most up-to-date version of this data sheet, please register at our Worldwide Web site at:

http://www.microchip.com

You can determine the version of a data sheet by examining its literature number found on the bottom outside corner of any page. The last character of the literature number is the version number, (e.g., DS30000000A is version A of document DS30000000).

Errata

An errata sheet, describing minor operational differences from the data sheet and recommended workarounds, may exist for current devices. As device/documentation issues become known to us, we will publish an errata sheet. The errata will specify the revision of silicon and revision of document to which it applies.

To determine if an errata sheet exists for a particular device, please check with one of the following:

- Microchip's Worldwide Web site; **http://www.microchip.com**
- Your local Microchip sales office (see last page)

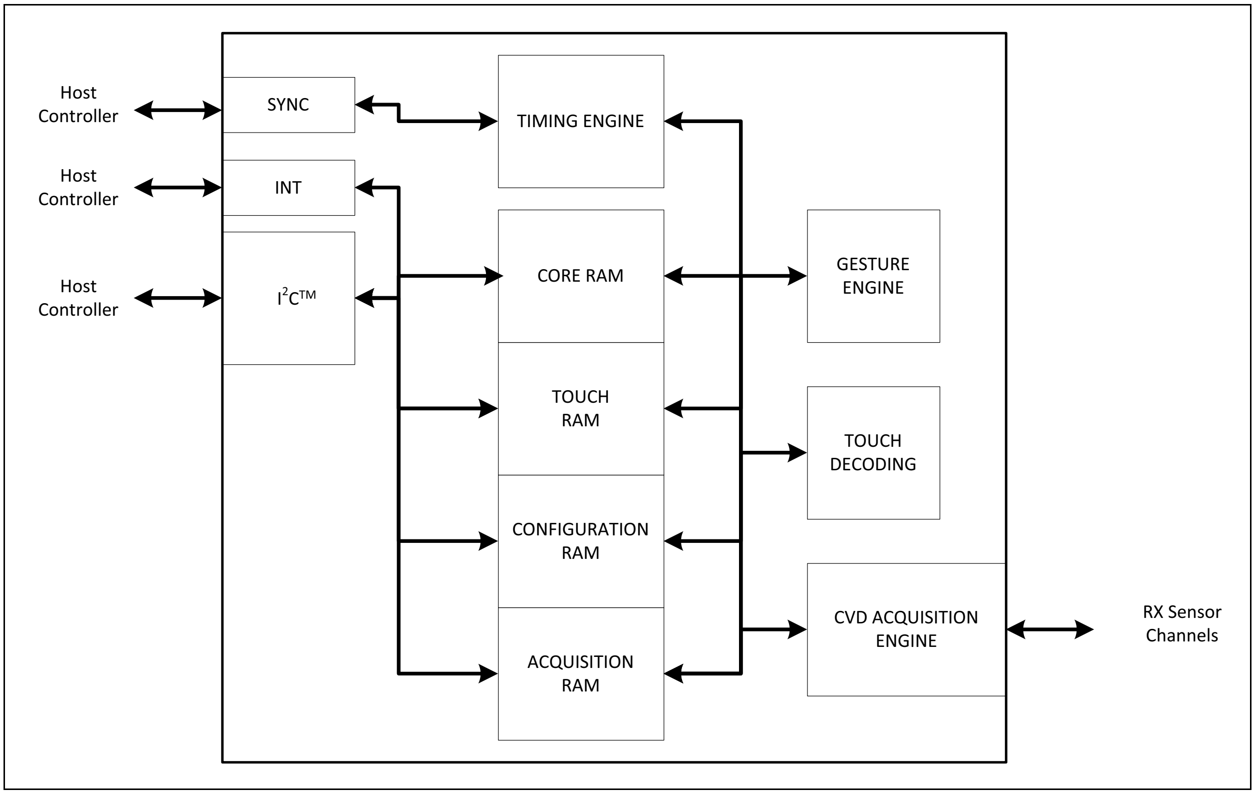
When contacting a sales office, please specify which device, revision of silicon and data sheet (include literature number) you are using.

Customer Notification System

Register on our web site at **www.microchip.com** to receive the most current information on all of our products.

1.0 MTCH6102 BLOCK DIAGRAM

FIGURE 1-1: MTCH6102 BLOCK DIAGRAM



2.0 PIN DIAGRAMS

FIGURE 2-1: 28-PIN UQFN (4X4)

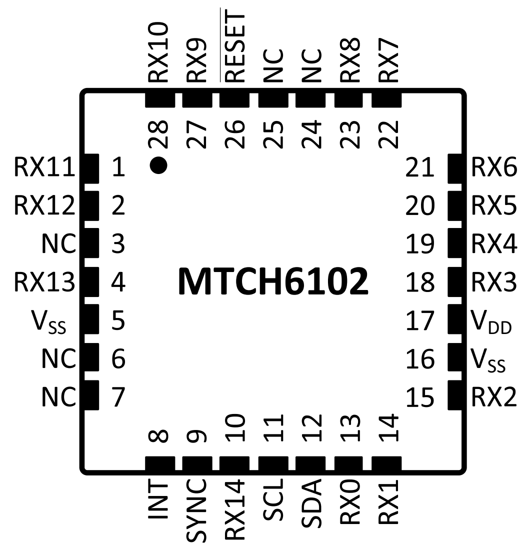
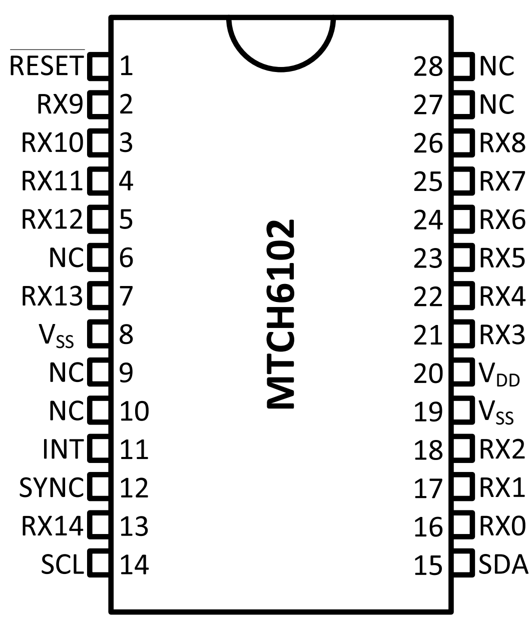


FIGURE 2-2: 28-PIN SSOP



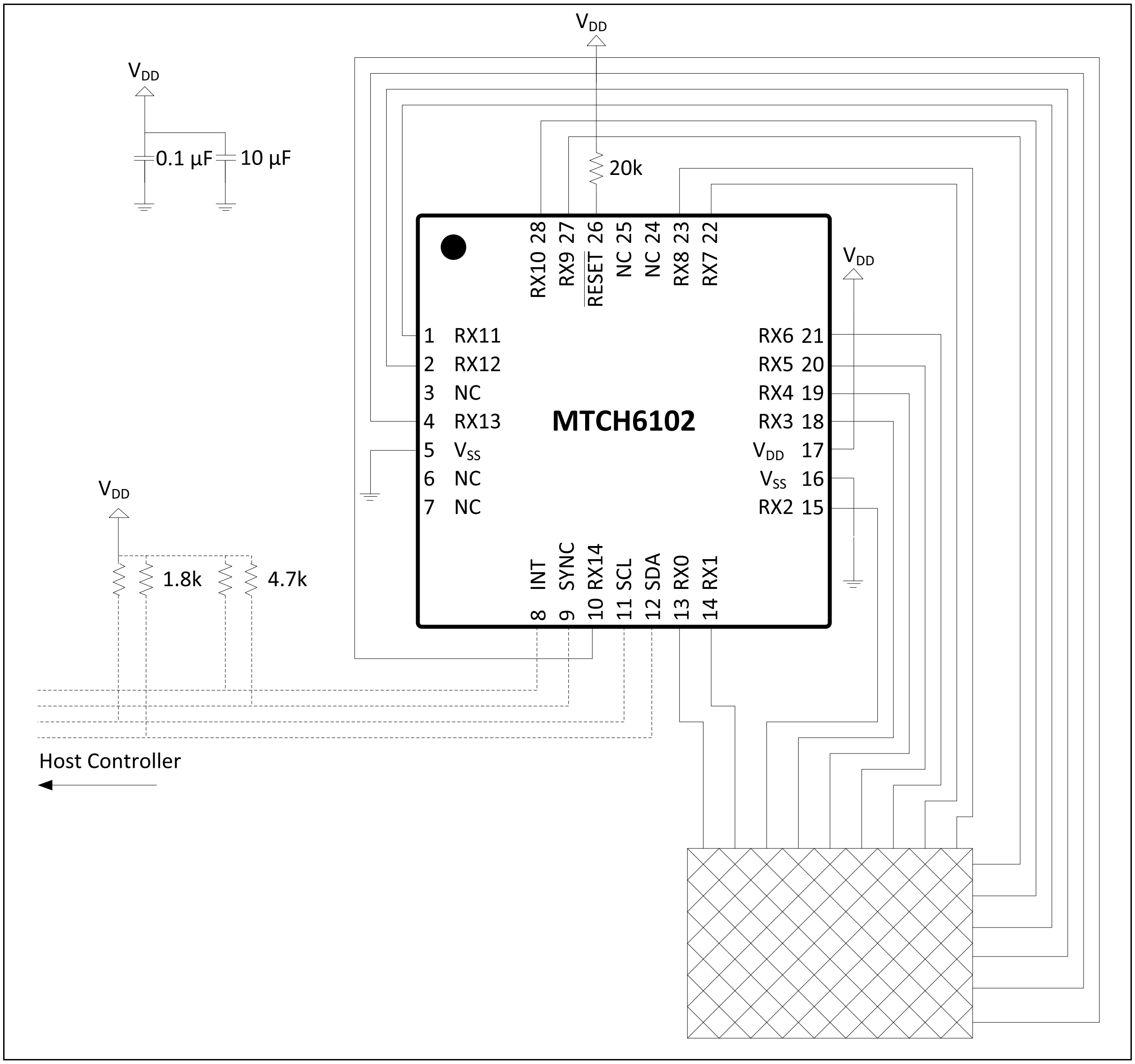
3.0 MTCH6102 PINOUT DESCRIPTION

TABLE 3-1: MTCH6102 PINOUT DESCRIPTION

Pin Name	UQFN Pin	SSOP Pin	Pin Type	Description
$\overline{RESET}$	26	1	I	Master Reset with Internal Pull-up
SCL	11	14	I/O	I^{2}C™ Clock
SDA	12	15	I/O	I^{2}C Data Input/Output
INT	8	11	O	Interrupt Request Output
SYNC	9	12	O	Synchronous Frame Output
RX0	13	16	I/O	Touch Sensor Channel Input
RX1	14	17	I/O	
RX2	15	18	I/O	
RX3	18	21	I/O	
RX4	19	22	I/O	
RX5	20	23	I/O	
RX6	21	24	I/O	
RX7	22	25	I/O	
RX8	23	26	I/O	
RX9	27	2	I/O	
RX10	28	3	I/O	
RX11	1	4	I/O	
RX12	2	5	I/O	
RX13	4	7	I/O	
RX14	10	13	I/O	
VDD	17	20	Power	Positive Supply
VSS	5,16	8,19	Power	Ground Reference
N/C	3, 6, 7, 24, 25	6, 9, 10, 27, 28	N/C	No Connect

4.0 LAYOUT

FIGURE 4-1: TYPICAL APPLICATION CIRCUIT



4.1 Decoupling Capacitors

The use of decoupling capacitors on power-supply pins, such as V_{DD} and V_{SS}, is required. Consider the following criteria when using decoupling capacitors:

1. Value and type of capacitor:

A value of 0.1 µF (100 nF), 10-20V is recommended. The capacitor should be a low Equivalent Series Resistance (low ESR) capacitor and have resonance frequency in the range of 20 MHz and higher. It is further recommended that ceramic capacitors be used.

2. Placement on the Printed Circuit Board:

The decoupling capacitors should be placed as close to the pins as possible. It is recommended that the capacitors be placed on the same side of the board as the device. If space is constricted, the capacitor can be placed on another layer on the PCB using a via; however, ensure that the trace length from the pin to the capacitor is within one-quarter inch (6 mm) in length.

3. Handling high-frequency noise:

If the board is experiencing high-frequency noise, upward of tens of MHz, add a second ceramic-type capacitor in parallel to the above-described decoupling capacitor. The value of the second capacitor can be in the range of 0.01 µF to 0.001 µF. Place this second capacitor next to the primary decoupling capacitor. In high-speed circuit designs, consider implementing a decade pair of capacitances as close to the power and ground pins as possible (for example, 0.1 µF in parallel with 0.001 µF).

4. Maximizing performance:

On the board layout from the power supply circuit, run the power and return traces to the decoupling capacitors first, and then to the device pins. This ensures that the decoupling capacitors are first in the power chain. It is equally important to keep the trace length between the capacitor and the power pins to a minimum, thereby reducing PCB track inductance.

4.2 Bulk Capacitors

The use of a bulk capacitor is recommended to improve power-supply stability. Typical values range from 4.7 µF to 47 µF. This capacitor should be located as close to the device as possible.

5.0 COMMUNICATION

5.1 I²C Pin Specification

5.1.1 DESCRIPTION

The MTCH6102 low-power projected capacitive touch controller uses a standard register-based read/write I²C protocol based upon the memory map. This protocol is similar to many other devices such as temperature sensors and serial EEPROMs. Although data can be read at any time (polling), an interrupt pin (INT) is provided for flexible integration options.

5.1.2 READING/WRITING REGISTERS

To access memory (both to read or write), the I²C transaction must start by addressing the chip with the Write bit set, then writing out a single byte of data representing the memory address to be operated on. After that, the host can choose to do either of the following (see Figure 5-1):

1. To write memory, continue writing [n] data bytes (see Figure 5-2).
2. To read memory, restart the I²C transaction (via either a Stop-Start or Restart), then address the chip with the Read bit set. Continue to read in [n] data bytes (see Figure 5-3).

During either of these transactions, multiple bytes within the same block may be read or written due to the device's address auto-increment feature. See **Section 17.0 "Memory Map"** for block separation.

FIGURE 5-1: I²C™ TRANSACTION DIAGRAM

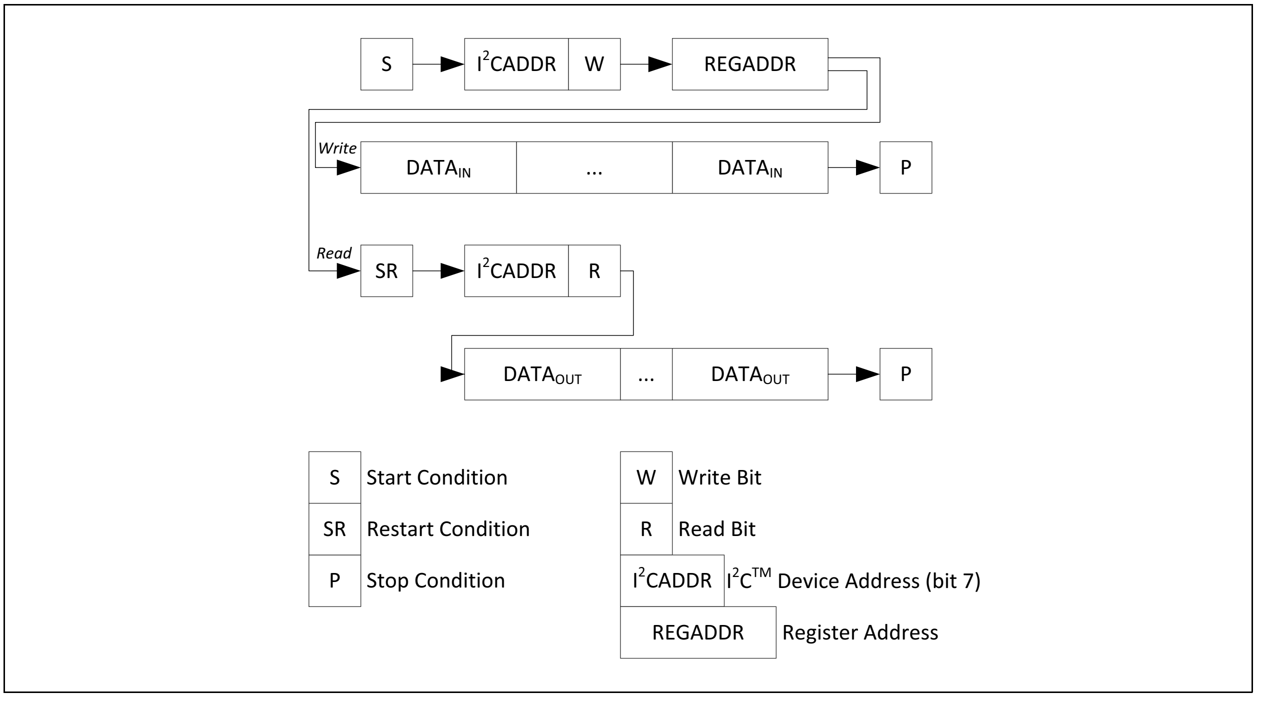


FIGURE 5-2: EXAMPLE I²C™ WRITE TRANSACTION

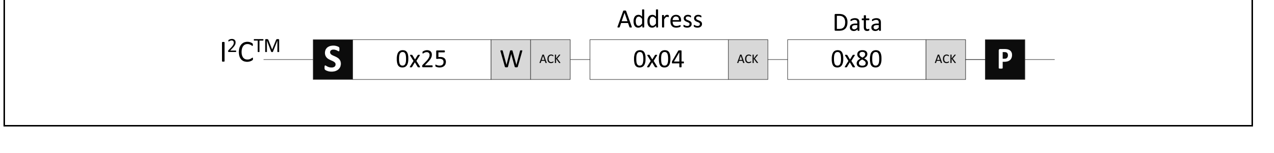
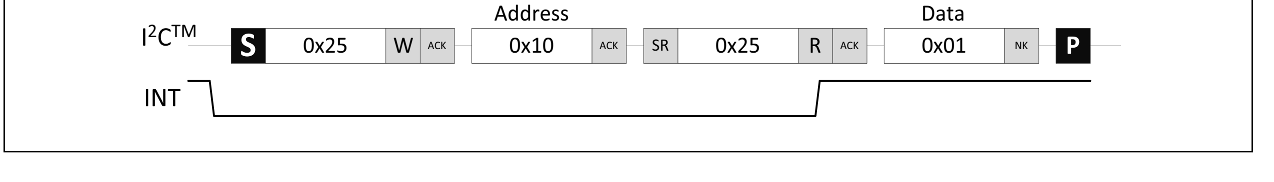


FIGURE 5-3: EXAMPLE I²C™ READ TRANSACTION



5.1.3 DEVICE ADDRESSING

The MTCH6102 default 7-bit base address is 0x25. Every transmission must be prefixed with this address, as well as a bit signifying whether the transmission is a master write ('0') or master read ('1'). After appending this Read/Write bit to the base address, this first byte becomes either 0x4A (write) or 0x4B (read).

This address can be modified (see I^2CADDR), but this requires initially communicating with the device under the default address. If this is not feasible in the user's application, contact Microchip support for additional options.

5.2 Interrupt Pin

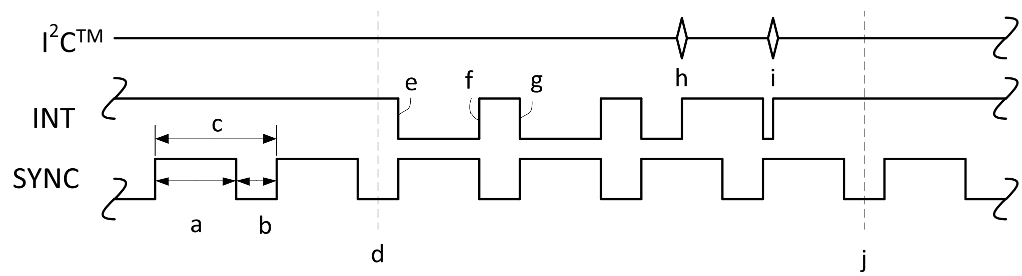
MTCH6102 provides an open-collector active-low Interrupt pin (INT) that will be asserted any time new data is available. INT is automatically released under two conditions:

1. A read is performed of *any* register within the device.
2. The next frame of decoding has started.

5.3 SYNC Output Pin

MTCH6102 provides an active-high sync signal that correlates with the current touch frame status. The SYNC pin is low while the device is sleeping (between frames) and high while touch sensing/decoding is occurring. A common use of this pin includes a host that makes use of data on every frame (such as raw-acquisition data), for host-side decoding (see Figure 5-4).

FIGURE 5-4: EXAMPLE INT/SYNC LOGIC

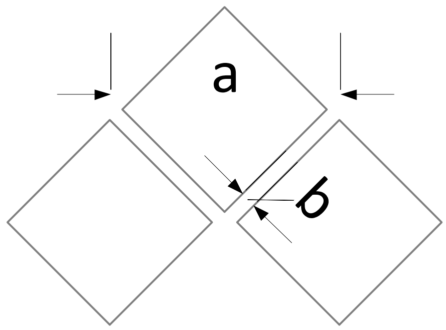


a	Controller sleeping		f	INT is released as controller goes to Sleep
b	Controller decoding		g	INT is asserted due to (d)
c	Touch frame		h	INT is released due to I^2C™ read
d	Finger is present		i	INT is released due to I^2C™ read
e	INT asserted due to (d)		j	Finger is not present

6.0 SENSOR DESIGN CONSIDERATIONS

6.1 General Guidelines

FIGURE 6-1: DIAMOND DIMENSION GUIDELINES



Dim.	Typ.	Min.	Max.
a	6 mm	4 mm	10 mm
b	0.2 mm	—	0.5 mm

6.1.1 PROTOTYPING DESIGNS

Touch sensor designs typically require a thorough debugging phase to ensure a reliable product. If possible, it is suggested that flexible prototyping hardware be created with this in mind. A common example is providing external access to the communication lines for quick test and tuning while in-circuit.

6.1.2 SENSOR OVERLAY MATERIAL

To prevent saturation of sensor levels, a minimum overlay of 0.5 mm plastic or glass is required for proper operation of the device, even during a prototyping phase, even if this value is different than the final design.

Note:	At no time should the device be expected to respond correctly to a user touching a bare PCB sensor.

6.1.3 OPERATION WITH AN LCD

MTCH6102 has integrated algorithms to detect and minimize the effects of noise, but proper care should always be taken in selecting an LCD and support components with a focus on reducing noise as much as possible. Since the interaction between the touch sensor and display is highly dependent upon the physical arrangement of the components, proper testing should always be executed with a fully integrated device. Please reference the appropriate projected capacitive touch screen manufacturer's integration guide for additional design considerations.

6.2 Sensor Layout Configuration

TABLE 6-1: REGISTERS ASSOCIATED WITH SENSOR LAYOUT CONFIGURATION

Address	Name	Description
0x20	NUMBEROFXCHANNELS	Number of channels used for X axis
0x21	NUMBEROFYCHANNELS	Number of channels used for Y axis

MTCH6102 is designed to work with sensors with a minimum of 3x3 sensor channels, and a total maximum of 15 channels. The number of channels on each axis is governed by the registers in Table 6-1. For all sensor configurations, the following conditions must be met:

1. Channel layout must start at RX0.
2. Each axis must have the associated channels in either ascending or descending order.
3. No unconnected channel pins are allowed in the middle of a layout.

Table 6-2 shows an example of each rule being broken by a 6x5 sensor layout, followed by the correct layout in the last column.

TABLE 6-2: EXAMPLE OF INCORRECT 6X5 SENSOR CONNECTIONS

	(1)	(2)	(3)	Correct
RX0		X0	X0	X0
RX1		X1	X1	X1
RX2		X2	X2	X2
RX3		X4	X3	X3
RX4	X0	X3	X4	X4
RX5	X1	X5	X5	X5
RX6	X2	Y0		Y0
RX7	X3	Y2		Y1
RX8	X4	Y1	Y0	Y2
RX9	X5	Y3	Y1	Y3
RX10	Y0	Y4	Y2	Y4
RX11	Y1		Y3	
RX12	Y2			
RX13	Y3		Y4	
RX14	Y4			

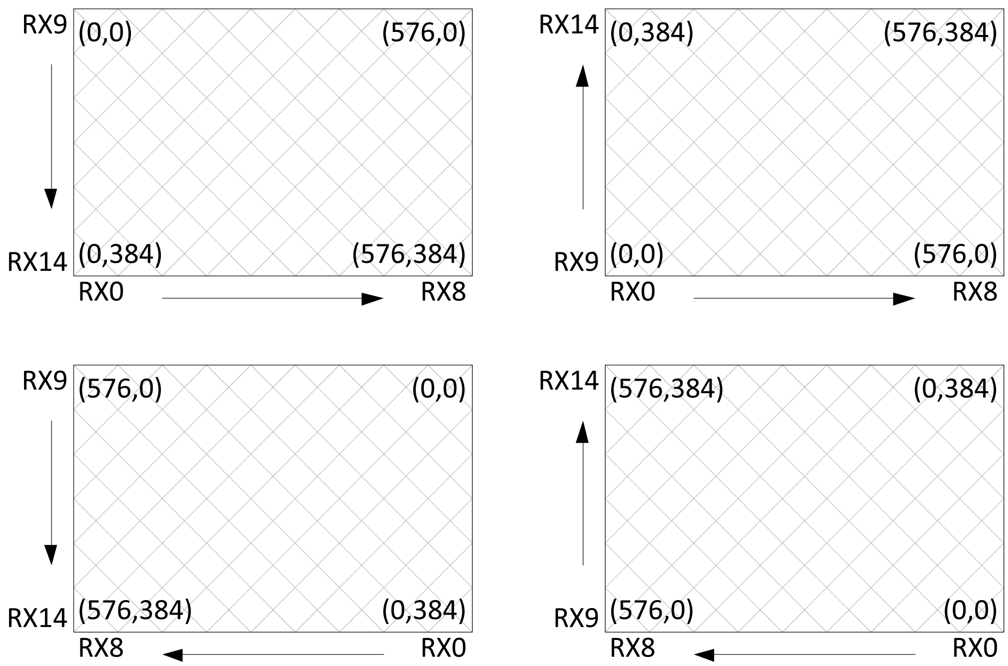
6.3 Sensor Output Resolution

MTCH6102 interpolates 64 discrete points between each channel and 32 points past the centerline of each edge. As a result, the maximum value in the TOUCHX and TOUCHY registers will be (64xNUMBEROFCHANNELS) on each axis. For the default 9x6 sensor, this results in a maximum resolution of 576x384.

6.4 Sensor Orientation

To aid in PCB layout, the sensor can be oriented in any direction, have either axis reversed, or even have the axes swapped. The host controller must take into account the X/Y output and gesture orientation based on Figure 6-2.

FIGURE 6-2: SENSOR ORIENTATION EXAMPLES



7.0 OPERATING MODES

MTCH6102 operates in multiple modes (see Table 7-1) governed by the MODE register (see Register 7-1).

TABLE 7-1: OPERATING MODE DESCRIPTIONS

Mode Name	Description	INT Behavior
Full	Full X/Y and gesture decoding occurs (Default mode)	Asserted if touch is present or if a change in touch status or a gesture have occurred
Touch	Full X/Y decoding only	Asserted if touch is present or if a change in touch status occurs
Gesture	Full X/Y and gesture decoding occurs, but INT is no longer asserted for touch data	Asserted for gestures only(1)
Raw	Raw-capacitance signals are stored in RAWADC registers, no decoding done. Channel selection and type of measurement is governed by the MODECON register	None
Standby	Device is no longer sensing or performing baseline tasks	None

Note 1: Data in TOUCH registers is still valid.

REGISTER 7-1: MODE: TOUCH DECODE MODE REGISTER

U-x	U-x	U-x	U-x	R/W-0	R/W-0	R/W-1	R/W-1
—	—	—	—	MODE<3:0>			
bit 7							bit 0

Legend:			
R = Readable bit	'1' = Bit is set	x = Bit is unknown	-n = Value after initialization (default)
W = Writable bit	'0' = Bit is cleared	U = Unimplemented bit	q = Conditional

bit 7-4 **Unimplemented:** Read as '0'

bit 3-0 **MODE<3:0>:** Touch Decoding mode bits

- 0000 = Standby
- 0001 = Gesture
- 0010 = Touch only
- 0011 = Full (touch and gesture)
- 01XX = Raw ADC

REGISTER 7-2: MODECON: RAWADC MODE CONTROL REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
TYPE<3:0>				CH<3:0>			
bit 7							bit 0

Legend:			
R = Readable bit	‘1’ = Bit is set	x = Bit is unknown	-n = Value after initialization (default)
W = Writable bit	‘0’ = Bit is cleared	U = Unimplemented bit	q = Conditional

bit 7-4 **TYPE<3:0>:** CVD Result Arithmetic bits

`0000` = (1023 – Result1) + Result 2
`0001` = Result 1 only
`0010` = Result 2 only

bit 3-0 **CH<3:0>:** RX Sense Channel bits

`0000` = RX0
..
..
`1110` = RX14
`1111` = Reserved, do not use

8.0 CONTROLLER COMMANDS

Various controller commands can be initiated by writing a '1' to the appropriate bit in the CMD register (Register 8-1). This bit will automatically be cleared after the command has been completed.

REGISTER 8-1: CMD: COMMAND REGISTER

R/W-0	R/W-0	R/W-0	U-x	R/W-0	U-x	U-x	R/W-0
NV	DEF	CFG	—	MFG	—	—	BS
bit 7							bit 0

Legend:			
R = Readable bit	'1' = Bit is set	x = Bit is unknown	-n = Value after initialization (default)
W = Writable bit	'0' = Bit is cleared	U = Unimplemented bit	q = Conditional

bit 7	**NV:** Nonvolatile Storage Write bit
bit 6	**DEF:** Restore Controller to Default Configuration Values bit
bit 5	**CFG:** Configure Controller bit (after parameters have been changed)
bit 4	**Unimplemented:** Read as '0'
bit 3	**MFG:** Execute Manufacturing Test bit
bit 2-1	**Unimplemented:** Read as '0'
bit 0	**BS:** Force Baseline bit (recalibration) to occur

9.0 TOUCH FRAME CONTROL

Touch decoding is based around the concept of a touch *frame* that begins with acquisition, followed by decoding of the acquired values, and lastly a Sleep phase for power savings. The duration of the touch frame is governed by the current touch state, as well as the timing registers outlined in this section (see Table 9-1). Figure 9-1 shows the interaction between these registers during a typical touch cycle.

TABLE 9-1: REGISTERS ASSOCIATED WITH TOUCH FRAME CONTROL

Address	Name	Description
0x25	ACTIVEPERIODL	Active Period
0x26	ACTIVEPERIODH	
0x27	IDLEPERIODL	Idle Period
0x28	IDLEPERIODH	
0x29	IDLETIMEOUT	Idle Timeout
0x2B	DEBOUNCEUP	Liftoff Debounce
0x2C	DEBOUNCEDOWN	Touch Down Debounce

Both active and idle period calculations are as shown in Equation 9-1.

EQUATION 9-1:

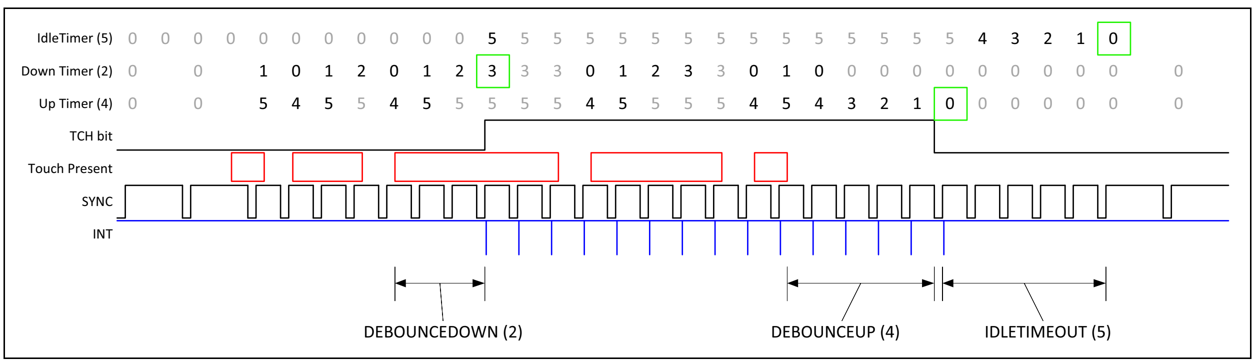
$$\left(\frac{Duration(ms) \times 1000}{31}\right) + 1 = PERIOD$$

Typical frame rates have been computed for the user's convenience and are shown in Table 9-2.

TABLE 9-2: EXAMPLE FRAME RATE PERIOD CALCULATIONS

Desired Rate (ms)	Period
10	0x0142
20	0x0284
50	0x064C
100	0x0C99

FIGURE 9-1: TOUCH FRAME TIMING



10.0 TOUCH DATA REGISTERS

REGISTER 10-1: TOUCHSTATE: CURRENT TOUCH STATE REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	U-x	R/W-0	R/W-0	R/W-0
FRAME<3:0>				—	LRG	GES	TCH
bit 7							bit 0

Legend:			
R = Readable bit	‘1’ = Bit is set	x = Bit is unknown	-n = Value after initialization (default)
W = Writable bit	‘0’ = Bit is cleared	U = Unimplemented bit	q = Conditional

bit 7-4	**FRAME<3:0>:** Increments on Every Touch Frame
bit 3	**Unimplemented:** Read as ‘0’
bit 2	**LRG:** Large Activation is Present
bit 1	**GES:** Gesture is Present
bit 0	**TCH:** Touch is Present

REGISTER 10-2: TOUCHLSB REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
TOUCHX<3:0>				TOUCHY<3:0>			
bit 7							bit 0

Legend:			
R = Readable bit	‘1’ = Bit is set	x = Bit is unknown	-n = Value after initialization (default)
W = Writable bit	‘0’ = Bit is cleared	U = Unimplemented bit	q = Conditional

bit 7-4	**TOUCHX<3:0>:** Current X Position (Least Significant bits)
bit 3-0	**TOUCHY<3:0>:** Current Y Position (Least Significant bits)

TABLE 10-1: SUMMARY OF REGISTERS ASSOCIATED WITH TOUCH DATA

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x10	TOUCHSTATE	FRAME<3:0>				—	LRG	GES	TCH
0x11	TOUCHX	TOUCHX<11:4>							
0x12	TOUCHY	TOUCHY<11:4>							
0x13	TOUCHLSB	TOUCHX<3:0>				TOUCHY<3:0>			

11.0 ACQUISITION AND TOUCH PARAMETERS

TABLE 11-1: REGISTERS ASSOCIATED WITH ACQUISITION AND TOUCH PARAMETERS

Address	Name	Default
0x22	SCANCOUNT	6
0x23	TOUCHTHRESHX	55
0x24	TOUCHTHRESHY	40
0x2A	HYSTERESIS	4
0x31	FILTERTYPE	2
0x32	FILTERSTRENGTH	1
0x35	LARGEACTIVATIONTHRESHL	0
0x36	LARGEACTIVATIONTHRESHH	0

11.1 SCANCOUNT

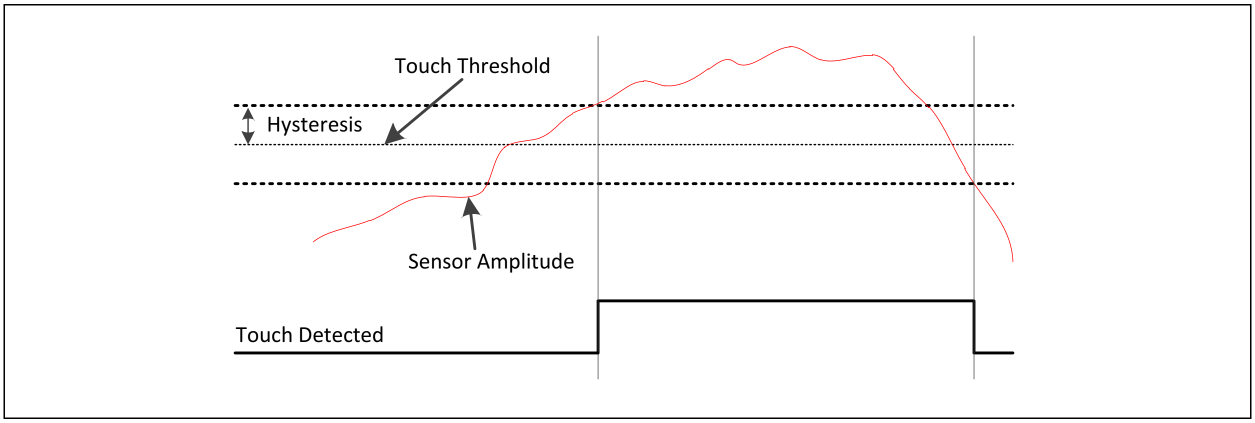
Every time a channel is scanned, it is scanned multiple times (SCANCOUNT) and summed. Increasing this number will give an inherent averaging effect, but at the cost of time and subsequently increased power consumption.

11.2 TOUCHTHRESHX/ TOUCHTHRESHY and HYSTERESIS

The presence of a touch is determined by the sensor channel's current value compared to the touch thresholds set by TOUCHTHRESHX (or TOUCHTHRESHY if the channel is on the Y axis).

The HYSTERESIS register contains a threshold modifier that acts as a dynamic threshold modifier depending on the state of the touch (higher without a touch). A single channel of touch is shown in Figure 11-1.

FIGURE 11-1: TOUCH THRESHOLD AND HYSTERESIS FUNCTIONALITY



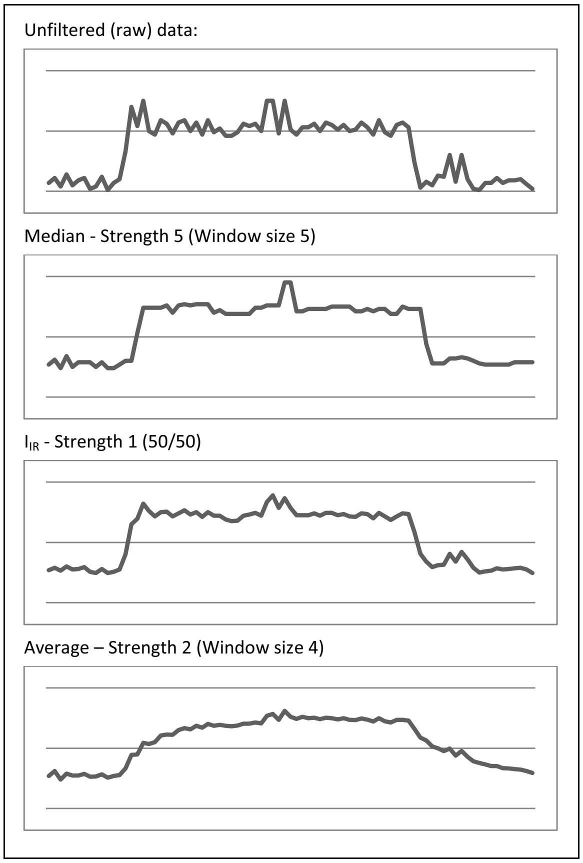
11.3 FILTERTYPE/FILTERSTRENGTH

As new sensor values are acquired, they are filtered based on the settings of the FILTERTYPE/ FILTERSTRENGTH registers (see Table 11-2). Examples of the effects of each filter type are shown in Figure 11-2.

TABLE 11-2: FILTERTYPE AND FILTERSTRENGTH DEFINITIONS

FILTERTYPE	FILTERSTRENGTH	Valid Values
0 – No Filter	N/A	—
1 – Median	Size of median window	3, 5, 7, 9
2 – IIR	Weighting of previous to current value	1, 2, 3 (1/2, 1/4 and 1/8 weighting accordingly)
3 – Average	Size of average window	1, 2, 3 (2, 4 and 8 accordingly)

FIGURE 11-2: FILTER EXAMPLES



Choosing the correct filtering option for the user's application depends on the environment and sensor. Note that while the median filter has good characteristics, it is not the most efficient and will consume more power than other filters.

11.4 Large Activation

The LARGEACTIVATIONTHRESH registers provide a way to do simple rejection of signals that are too large to interpret. The amplitude of all sensor channels are added together and compared to this threshold. If greater, the LRG bit of the TOUCHSTATE register (Register 10-1) will be set.

Note that this does not affect touch decoding. In other words, even if the large activation threshold is breached, the controller will still decode the touch position as normal. The LRG bit merely serves to inform the host that the large activation threshold has been reached.

If this functionality is not intended to be used, this register should be set to zero, which will disable the large activation routines from running.

12.0 COMPENSATION RAM

It is very common for a typical touch sensor to have non-uniform capacitive properties. To equalize the sensor, a series of coefficients can be written to the compensation RAM block. These coefficients represent a ratio that is applied to the individual channel in post-acquisition, before touch decoding occurs.

EQUATION 12-1: COMPENSATION RAM CALCULATION

$$\frac{[RAW\ VALUE] \times SENSORCOMP}{64} = [FINAL\ VALUE]$$

To obtain the correct compensation RAM values, the following procedure should be used:

1. Set all SENSORVALUES registers to zero (if necessary).
2. Record the peak values that occur in the SENSORVALUES registers when using the sensor under normal conditions (column A of Table 12-1).
3. Pick a commonly occurring value to represent the median of the set (‘125’).
4. Calculate the ratio of the peak value by dividing the median value by the peak (column B).
5. Multiply this value by 64 and truncate (column C). These are the compensation values that should be written to the SENSORCOMP registers. Please note that, if no compensation is required (value of ‘64’, ratio of ‘1’), the register should be set to ‘0’, to save time running compensation routines for that channel.
6. To see the expected output from the compensation values, follow Equation 12-1 (result in column D).

TABLE 12-1: COMPENSATION RAM EXAMPLE CALCULATION

CH	A	B	C	D
0	102	1.225	78	124
1	113	1.106	71	125
2	118	1.059	68	125
3	125	1	64 (0)	125
4	125	1	64 (0)	125
5	128	0.977	63	126
6	132	0.947	61	126
7	160	0.781	50	125

13.0 BASELINE

Capacitive touch principles rely on analyzing a change in capacitance from a previously-stored baseline value (sometimes referred to as a calibration value). Baseline routines and behavior can be tweaked using the registers listed in Table 13-1.

TABLE 13-1: REGISTERS ASSOCIATED WITH BASELINE

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Default
0x04	NV	DEF	CFG	-	MFG	-	-	BS	0
0x2D	BASEINTERVALL								10
0x2E	BASEINTERVALH								0
0x2F	BASEPOSFILTER								20
0x30	BASENEGFILTER								20
0x33	BASEFILTERTYPE								2
0x34	BASEFILTERSTRENGTH								1

13.1 BS Bit (CMD Register)

The BS bit forces the current sensor values to be stored as the baseline values, disregarding the constraints of BASEPOSFILTER and BASENEGFILTER.

13.2 BASEINTERVAL

It represents the number of touch frames between baseline sampling. Data that is sampled will be applied at the next baseline interval, provided that a touch has not occurred between the two.

If at any point, the touch threshold is breached, the baseline counter is reset, and a full interval without a touch must occur before baselining resumes.

Note that this value is specified in terms of the number of touch frames, so any changes in frame rate should take this into consideration by raising or lowering this interval accordingly.

13.3 BASEPOSFILTER/BASENEGFILTER

The positive and negative filters act as slew-rate limiters for a new baseline being applied. For example, if the new baseline value is larger than the previous by a value of 35, and the BASEPOSFILTER is set to 20 (default), the new baseline will only be increased by 20.

Use of these registers helps prevent unwanted spikes in the baseline value.

13.4 BASEFILTERTYPE/ BASEFILTERSTRENGTH

Baseline acquisition frames follow the same filter type and strength parameters as normal acquisition filters, defined in **Section 11.3 "FILTERTYPE/FILTER-STRENGTH"**.

14.0 GESTURE FEATURES AND PARAMETERS

Gesture detection and reporting is governed by the registers outlined in Table 14-1.

TABLE 14-1: SUMMARY OF REGISTERS ASSOCIATED WITH GESTURES

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Default
0x10	TOUCHSTATE	FRAME<3:0>				-	LRG	GES	TCH	N/A
0x14	GESTURESTATE	GESTURESTATE								0
0x15	GESTUREDIAG	GESTUREDIAG								0
0x37	HORIZONTALSWIPEDISTANCE	HORIZONTALSWIPEDISTANCE								64
0x38	VERTICALSWIPEDISTANCE	VERTICALSWIPEDISTANCE								64
0x39	SWIPEHOLDBOUNDARY	SWIPEHOLDBOUNDARY								25
0x3A	TAPDISTANCE	TAPDISTANCE								25
0x3B	DISTANCEBETWEENTAPS	DISTANCEBETWEENTAPS								64
0x3C	TAPHOLDTIME	TAPHOLDTIMEL								50
0x3D		TAPHOLDTIMEH								0
0x3E	GESTURECLICKTIME	GESTURECLICKTIME								12
0x3F	SWIPEHOLDTHRESH	SWIPEHOLDTHRESH								32
0x40	MINSWIPEVELOCITY	MINSWIPEVELOCITY								4
0x41	HORIZONTALGESTUREANGLE	HORIZONTALGESTUREANGLE								45
0x42	VERTICALGESTUREANGLE	VERTICALGESTUREANGLE								45

When a gesture is performed, the gesture ID will be placed in GESTURESTATE, and the GES bit of the TOUCHSTATE register will be set. Both of these items are cleared after reading the GESTURESTATE register. The GESTUREDIAG register contains a code explaining the logic behind the last operation of the gesture engine, primarily to help with debugging of the gesture parameters. These diagnostic codes are shown in Register 14-2.

REGISTER 14-1: GESTURESTATE: CURRENT GESTURE STATE REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
GESTURESTATE<7:0>							
bit 7							bit 0

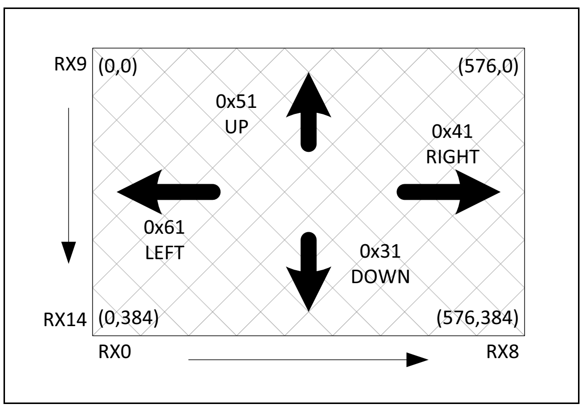
Legend:			
R = Readable bit	'1' = Bit is set	x = Bit is unknown	-n = Value after initialization (default)
W = Writable bit	'0' = Bit is cleared	U = Unimplemented bit	q = Conditional

bit 7-0 **GESTURESTATE<7:0>:**

0x00	No Gesture Present
0x10	Single Click
0x11	Click and Hold
0x20	Double Click
0x31	Down Swipe
0x32	Down Swipe and Hold
0x41	Right Swipe
0x42	Right Swipe and Hold
0x51	Up Swipe
0x52	Up Swipe and Hold
0x61	Left Swipe
0x62	Left Swipe and Hold

Please note that the gesture orientations listed in Register 14-1 are correct for a default layout, with *right* moving on increasing X-axis channels, and *down* moving on increasing Y-axis channels. These default orientations are shown in Figure 14-1. Depending on the application, the host may need to associate the gesture IDs differently.

FIGURE 14-1: GESTURE ORIENTATION



REGISTER 14-2: GESTUREDIAG: GESTURE DIAGNOSTICS REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
GESTUREDIAG<7:0>							
bit 7							bit 0

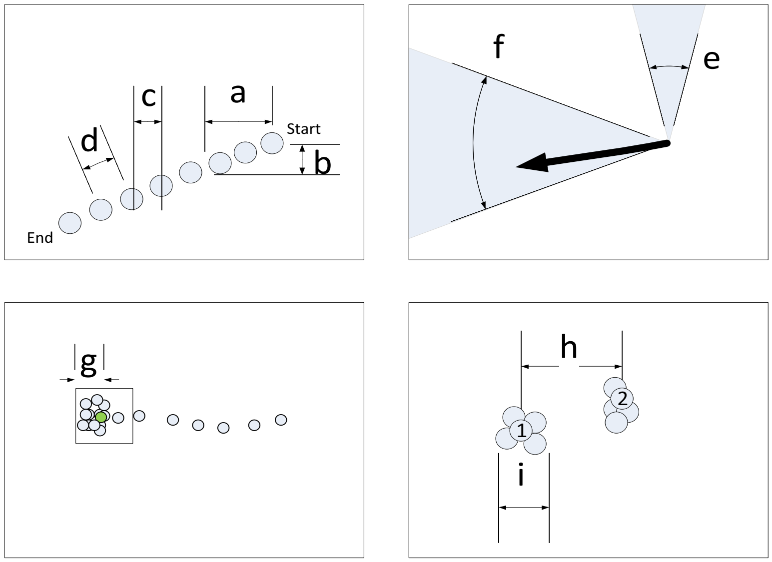
Legend:			
R = Readable bit	'1' = Bit is set	x = Bit is unknown	-n = Value after initialization (default)
W = Writable bit	'0' = Bit is cleared	U = Unimplemented bit	q = Conditional

bit 7-0 **GESTUREDIAG<7:0>:**

0x01	Click Timeout
0x02	Swipe Timeout
0x03	General Timeout
0x04	Click Threshold Exceeded
0x05	Swipe Threshold Exceeded
0x06	Swipe and Hold Threshold Exceeded
0x07	Swipe Opposite Direction Threshold Exceeded
0x08	Reserved
0x09	Swipe and Hold Value Exceeded
0x0A	Outside Swipe Angle

14.1 Gesture Tuning

FIGURE 14-2: GESTURE PARAMETER VISUALIZATION



Legend:

1. HORIZONTALSWIPEDISTANCE/VERTICALSWIPEDISTANCE (**a/b**): Distance that a touch must move from the initial touchdown to be considered a *swipe.*
2. SWIPEHOLDTHRESH (**c**): Allowed movement in the opposite direction before a swipe is canceled.
3. MINSWIPEVELOCITY (**d**): Once a movement is classified as a *swipe*, this distance must be kept frame to frame before the swipe is classified as a *hold* (direction of travel is not important).
4. HORITZONTALGESTUREANGLE/VERTICALGESTUREANGLE (**f/e**): Angle in degrees allowed on horizontal (X-axis) and vertical (Y-axis) swipe movements. Swipes outside this parameter are detected, but not reported.
5. SWIPEHOLDBOUNDARY (**g**): Once a swipe is classified as a *hold*, movement must not exceed this parameter in any direction.
6. DISTANCEBETWEENTAPS (**h**): Distance allowed between two taps to be considered a *double click*.
7. TAPDISTANCE (**i**): Even when a sensor is pressed for a very short time (click), multiple frames of touch data occur. This parameter governs how close those individual touch points must be for a *click*.
8. TAPHOLDTIME (not shown): Duration of time a click must be held for to be considered a *click and hold.*
9. GESTURECLICKTIME (not shown): Maximum time between two clicks to be considered a *double click.*