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

AR1000 SERIES RESISTIVE TOUCH SCREEN CONTROLLER

AR1000 Series Resistive Touch Screen Controller

Special Features

- RoHS Compliant
- Power-Saving Sleep mode
- Industrial Temperature Range
- Built-in Drift Compensation Algorithm
- 128 Bytes of User EEPROM

Power Requirements

- Operating Voltage: 2.5-5.0V ±5%
- Standby Current:
  - 5V: 85 µA, typical; 125 µA (maximum)
  - 2.5V: 40 µA, typical; 60 µA (maximum)
- Operating “No touch” Current:
  - 3.0 mA (typical)
- Operating “Touch” Current:
  - 17 mA, typical, with a touch sensor having 200Ω layers
  - Actual current is dependent on the touch sensor used
- AR1011/AR1021 Brown-Out Detection (BOR) set to 2.2V

Touch Modes

- Off, Stream, Down, Up and more.

Touch Sensor Support

- 4-Wire, 5-Wire and 8-Wire Analog Resistive
- Lead-to-Lead Resistance: 50-2,000Ω (typical)
- Layer-to-Layer Capacitance: 0-0.5 µF
- Touch Sensor Time Constant: 500 µs (maximum)

Touch Resolution

- 10-bit Resolution (maximum)

Touch Coordinate Report Rate

- 140 Reports Per Second (typical) with a Touch Sensor of 0.02 µF with 200Ω Layers
- Actual Report Rate is dependent on the Touch Sensor used

Communications

- SPI, Slave mode, p/n AR1021
- I^2C, Slave mode, p/n, AR1021
- UART, 9600 Baud Rate, p/n AR1011

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 Website 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 Website; **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 website at **www.microchip.com** to receive the most current information on all of our products.

1.0 DEVICE OVERVIEW

The Microchip mTouch® AR1000 Series Resistive Touch Screen Controller is a complete, easy to integrate, cost-effective and universal touch screen controller chip.

The AR1000 Series has sophisticated proprietary touch screen decoding algorithms to process all touch data, saving the host from the processing overhead. Providing filtering capabilities beyond that of other low-cost devices, the AR1000 delivers reliable, validated, and calibrated touch coordinates.

Using the on-board EEPROM, the AR1000 can store and independently apply the calibration to the touch coordinates before sending them to the host. This unique combination of features makes the AR1000 the most resource-efficient touch screen controller for system designs, including embedded system integrations.

1.1 Applications

The AR1000 Series is designed for high volume, small form factor touch solutions with quick time to market requirements – including, but not limited to:

- Mobile communication devices
- Personal Digital Assistants (PDA)
- Global Positioning Systems (GPS)
- Touch Screen Monitors
- KIOSK
- Media Players
- Portable Instruments
- Point of Sale Terminals

FIGURE 1-1: BLOCK DIAGRAM

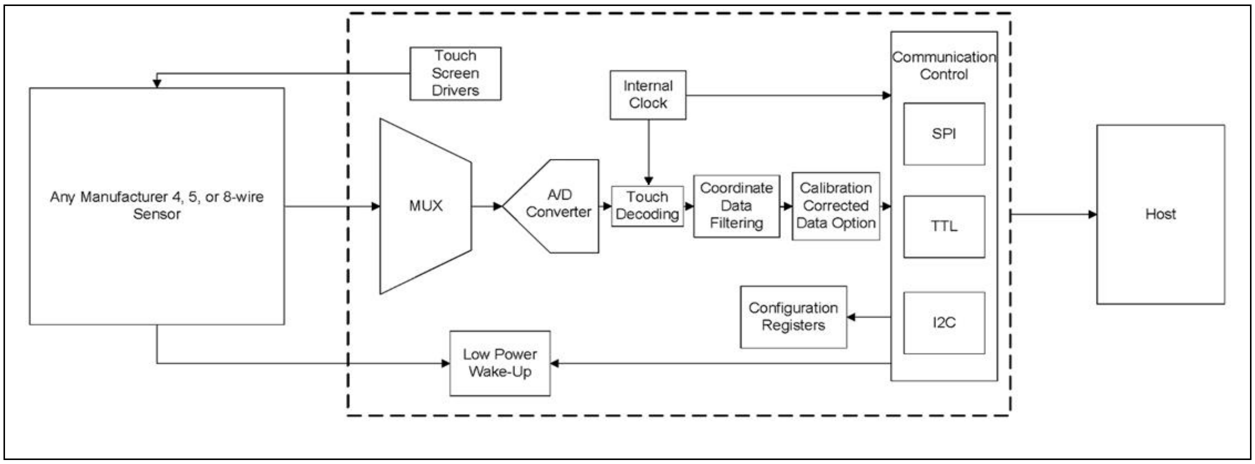


FIGURE 1-2: PIN DIAGRAM

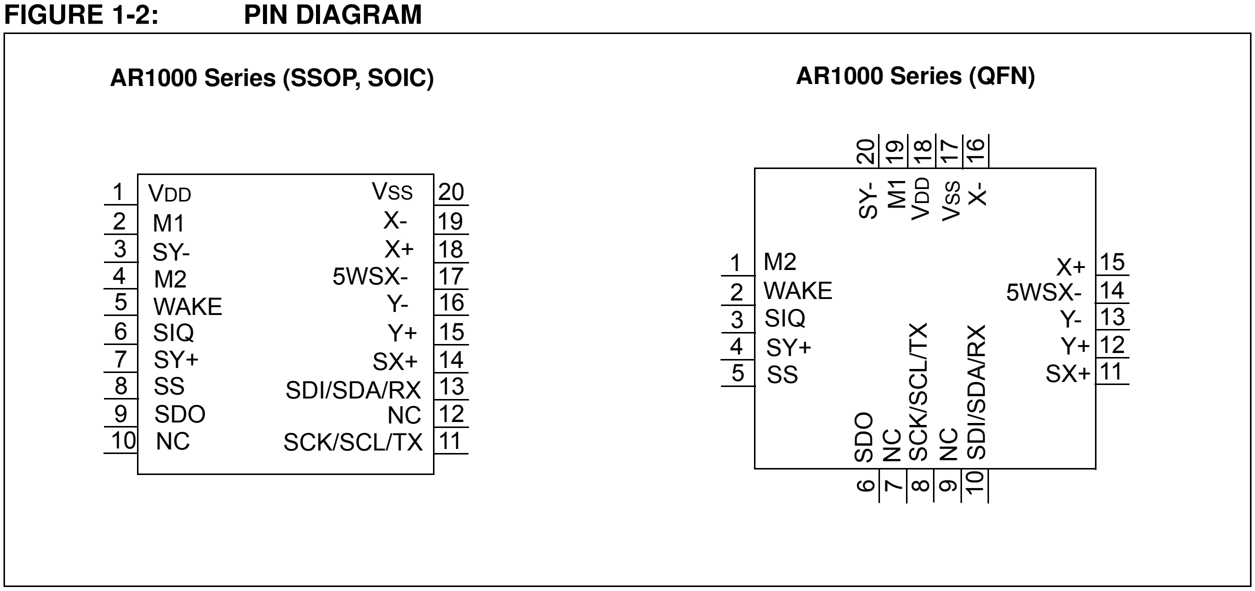


TABLE 1-1: PIN DESCRIPTIONS

Pin		Function	Description/Comments
SSOP, SOIC	QFN		
1	18	VDD	Supply Voltage
2	19	M1	Communication Selection
3	20	SY-	Sense Y- (8-wire). Tie to VSS, if not used.
4	1	M2	4/8-wire or 5-wire Sensor Selection
5	2	WAKE	Touch Wake-up/Touch Detection
6	3	SIQ	LED Drive/SPI Interrupt. No connect, if not used.
7	4	SY+	Sense Y+ (8-wire). Tie to VSS, if not used.
8	5	SS	Slave Select (SPI). Tie to VSS, if not used.
9	6	SDO	SPI Serial Data Output/I^2C Interrupt. Tie to Vss, if UART.
10	7	NC	No connection. No connect or tie to VSS or VDD.
11	8	SCK/SCL/TX	SPI/I^2C Serial Clock/UART Transmit
12	9	NC	No connection. No connect or tie to VSS or VDD.
13	10	SDI/SDA/RX	I^2C Serial Data/SPI Serial Data Input/UART Receive
14	11	SX+	Sense X+ (8-wire). Tie to VSS, if not used.
15	12	Y+	Y+ Drive
16	13	Y-	Y- Drive
17	14	5WSX-	5W Sense (5-wire)/Sense X- (8-wire). Tie to VSS, if not used.
18	15	X+	X+ Drive
19	16	X-	X- Drive
20	17	Vss	Supply Voltage Ground

2.0 BASICS OF RESISTIVE SENSORS

2.1 Terminology

ITO (Indium Tin Oxide) is the resistive coating that makes up the active area of the touch sensor. ITO is a transparent semiconductor that is sputtered onto the touch sensor layers.

Flex or Film or Topsheet is the top sensor layer that a user touches. *Flex* refers to the fact that the top layer physically flexes from the pressure of a touch.

Stable or Glass is the bottom sensor layer that interfaces against the display.

Spacer Adhesive is a frame of adhesive that connects the flex and stable layers together around the perimeter of the sensor.

Spacer Dots maintain physical and electrical separation between the flex and stable layers. The dots are typically printed onto the stable layer.

Bus Bars or Silver Frit electrically connect the ITO on the flex and stable layers to the sensor's interface tail. Bus bars are typically screen printed silver ink. They are typically much lower in resistivity than the ITO.

X-Axis is the left and right direction on the touch sensor.

Y-Axis is the top and bottom direction on the touch sensor.

Drive Lines supply a voltage gradient across the sensor.

2.2 General

Resistive 4, 5, and 8-wire touch sensors consist of two facing conductive layers, held in physical separation from each other. The force of a touch causes the top layer to deflect and make electrical contact with the bottom layer.

Touch position measurements are made by applying a voltage gradient across a layer or axis of the touch sensor. The touch position voltage for the axis can be measured using the opposing layer.

A comparison of typical sensor constructions is shown below in Table 2-1.

TABLE 2-1: SENSOR COMPARISON

Sensor	Comments
4-Wire	Less expensive than 5-wire or 8-wire Lower power than 5-wire More linear (without correction) than 5-wire Touch inaccuracies occur from flex layer damage or resistance changes
5-Wire	Maintains touch accuracy with flex layer damage Inherent nonlinearity often requires touch data correction Touch inaccuracies occur from resistance changes
8-Wire	More expensive than 4-wire Lower power than 5-wire More linear (without correction) than 5-wire Touch inaccuracies occur from flex layer damaged Maintains touch accuracy with resistance changes

The AR1000 Series Resistive Touch Screen Controllers will work with any manufacturers of analog resistive 4, 5 and 8-wire touch screens. The communications and decoding are included, allowing the user the quickest simplest method of interfacing analog resistive touch screens into their applications.

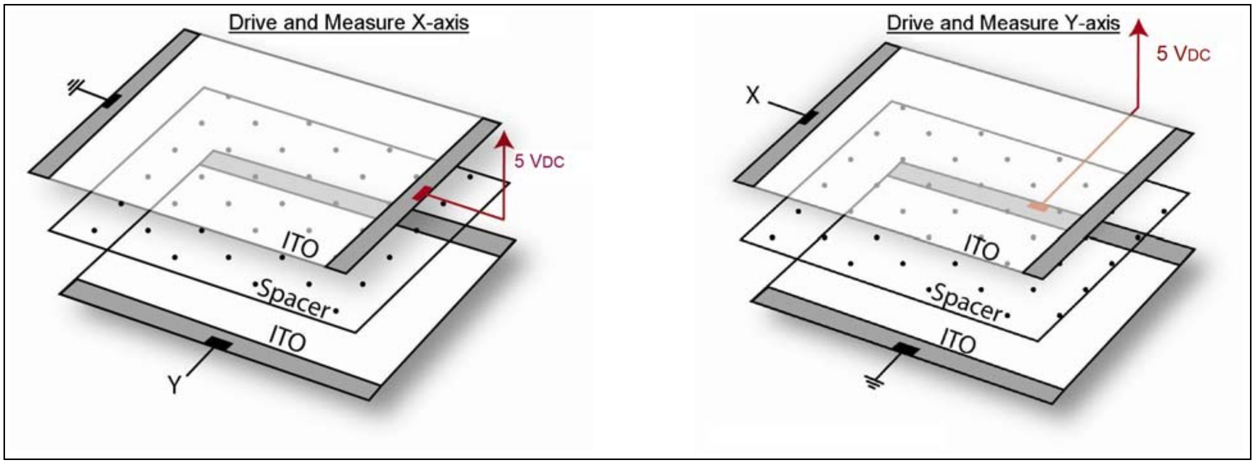
The AR1000 Series was designed with an understanding of the materials and processes that make up resistive touch screens. The AR1000 Series Touch Controller is not only reliable, but can enhance the reliability and longevity of the resistive touch screen, due to its advanced filtering algorithms and wide range of operation.

2.3 4-Wire Sensor

A 4-wire resistive touch sensor consists of a stable and flex layer, electrically separated by spacer dots. The layers are assembled perpendicular to each other. The touch position is determined by first applying a voltage gradient across the flex layer and using the stable layer to measure the flex layer's touch position voltage. The second step is applying a voltage gradient across the stable layer and using the flex layer to measure the stable layer's touch position voltage.

The measured voltage at any position across a driven axis is predictable. A touch moving in the direction of the driven axis will yield a linearly changing voltage. A touch moving perpendicular to the driven axis will yield a relatively unchanging voltage (See Figure 2-1).

FIGURE 2-1: 4-WIRE DECODING



2.4 8-Wire Sensor

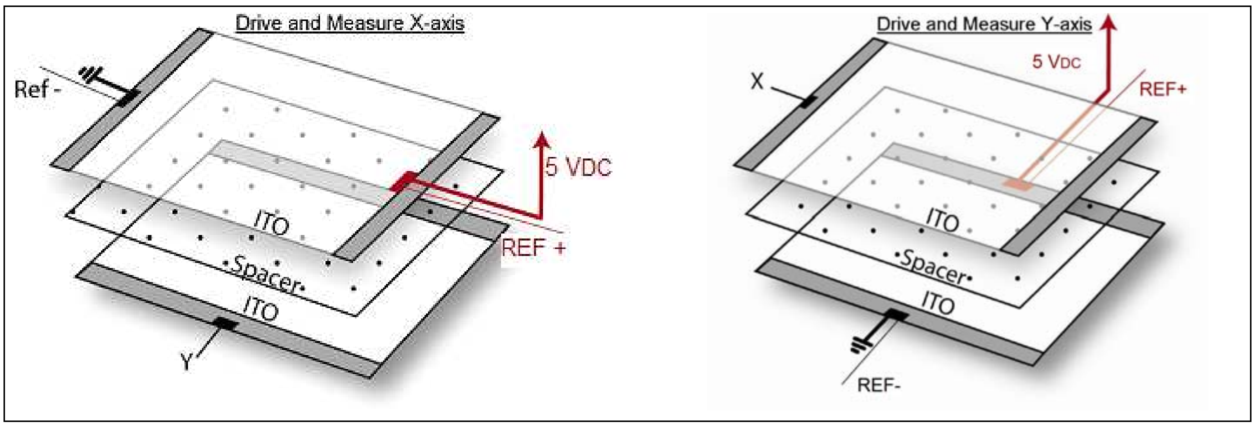
An 8-wire resistive touch sensor consists of a stable and flex layer, electrically separated by spacer dots. The layers are assembled perpendicular to each other. The touch position is determined by first applying a voltage gradient across the flex layer and using the stable layer to measure the flex layer's touch position voltage. The second step is applying a voltage gradient across the stable layer and using the flex layer to measure the stable layer's touch position voltage.

The measured voltage at any position across a driven axis is predictable. A touch moving in the direction of the driven axis will yield a linearly changing voltage. A touch moving perpendicular to the driven axis will yield a relatively unchanging voltage.

The basic decoding of an 8-wire sensor is similar to a 4-wire. The difference is that an 8-wire sensor has four additional interconnects used to reference sensor voltage back to the controller.

A touch system may experience voltage losses due to resistance changes in the bus bars and connection between the controller and sensor. The losses can vary with product use, temperature, and humidity. In a 4-wire sensor, variations in the losses manifest themselves as error or drift in the reported touch location. The four additional sense lines found on 8-wire sensors are added to dynamically reference the voltage to correct for this fluctuation during use (See Figure 2-2).

FIGURE 2-2: 8-WIRE DECODING



2.5 5-Wire Sensor

A 5-wire resistive touch sensor consists of a flex and stable layer, electrically separated by spacer dots. The touch position is determined by first applying a voltage gradient across the stable layer in the X-axis direction and using the flex layer to measure the axis touch position voltage. The second step is applying a voltage gradient across the stable layer in the Y-axis direction and using the flex layer to measure the axis touch position voltage.

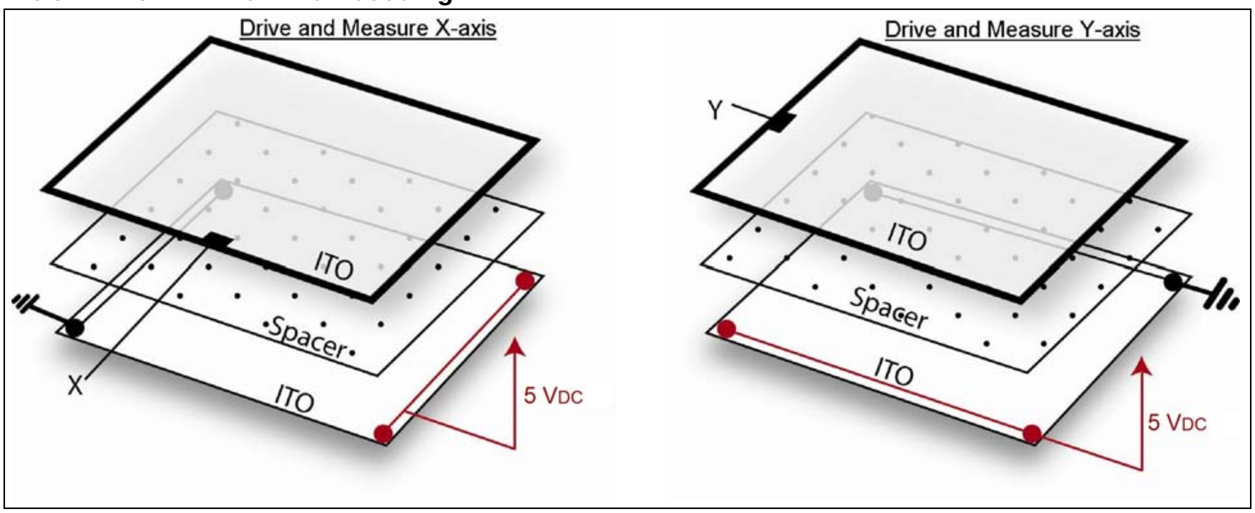
The voltage is not directly applied to the edges of the active layer, as it is for 4-wire and 8-wire sensors. The voltage is applied to the corners of a 5-wire sensor.

To measure the X-axis, the left edge of the layer is driven with 0V (ground), using connections to the upper left and lower left sensor corners. The right edge is driven with +5 VDC, using connections to the upper right and lower right sensor corners.

To measure the Y-axis, the top edge of the layer is driven with 0V (ground), using connections to the upper left and upper right sensor corners. The bottom edge is driven with +5 VDC, using connections to the lower left and lower right sensor corners.

The measured voltage at any position across a driven axis is predictable. A touch moving in the direction of the driven axis will yield a linearly changing voltage. A touch moving perpendicular to the driven axis will yield a relatively unchanging voltage (See Figure 2-3).

FIGURE 2-3: 5-Wire Decoding



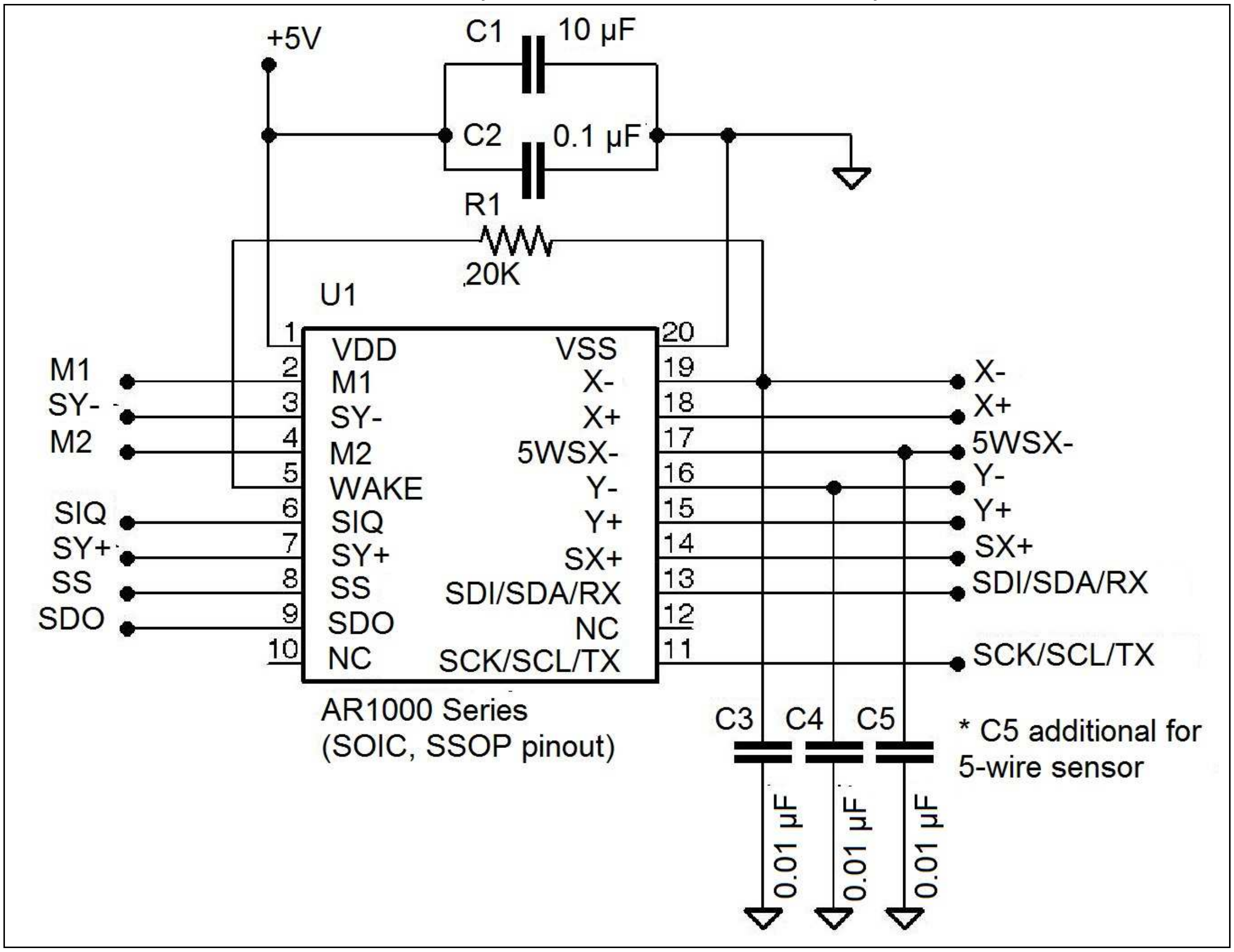
3.0 HARDWARE

3.1 Main Schematic

A main application schematic for the SOIC/SSOP package pinout is shown in Figure 3-1.

See Figure 1-2 for the QFN package pinout.

FIGURE 3-1: MAIN SCHEMATIC (SOIC/SSOP PACKAGE PINOUT)



3.2 4, 5, 8-Wire Sensor Selection

The desired sensor type of 4/8-wire or 5-wire is hardware selectable using pin M2.

TABLE 3-1: 4/8-WIRE vs. 5-WIRE SELECTION

Type	M2 pin
4/8-wire	VSS
5-wire	VDD

If 4/8-wire has been hardware-selected, then the choice of 4-wire or 8-wire is software-selectable via the TouchOptions Configuration register.

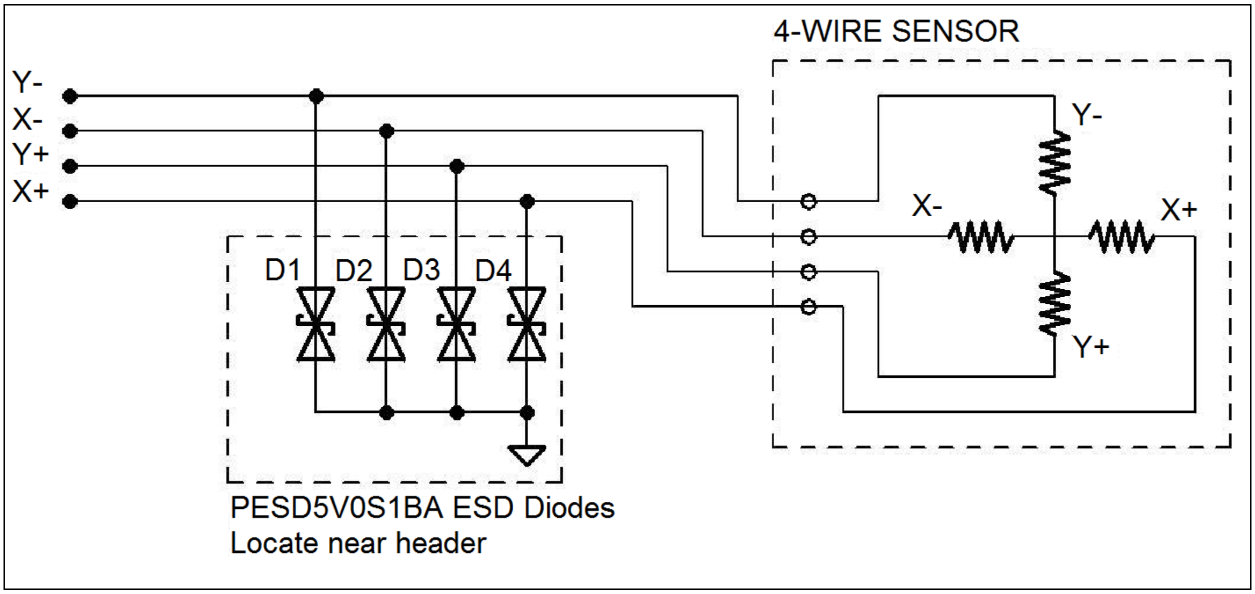
When 4/8-wire is hardware-selected, the controller defaults to 4-wire operation. If 8-wire operation is desired, then the TouchOptions Configuration register must be changed.

3.3 4-Wire Touch Sensor Interface

Sensor tail pinouts can vary by manufacturer and part number. Ensure that both sensor tail pins for one sensor axis (layer) are connected to the controller's X-/X+ pins and the tail pins for the other sensor axis (layer) are connected to the controller's Y-/Y+ pins. The controller's X-/X+ and Y-/Y+ pin pairs do not need to connect to a specific sensor axis. The orientation of controller pins X- and X+ to the two sides of a given sensor axis is not important. Likewise, the orientation of controller pins Y- and Y+ to the two sides of the other sensor axis is not important.

Connections to a 4-wire touch sensor are as follows (See Figure 3-2).

FIGURE 3-2: 4-WIRE TOUCH SENSOR INTERFACE



Tie unused controller pins 5WSX-, SX+, SY-, and SY+ to VSS.

See **Section 3.8 "ESD Considerations"** and **Section 3.9 "Noise Considerations"** for important information regarding the capacitance of the controller schematic hardware.

3.4 5-Wire Touch Sensor Interface

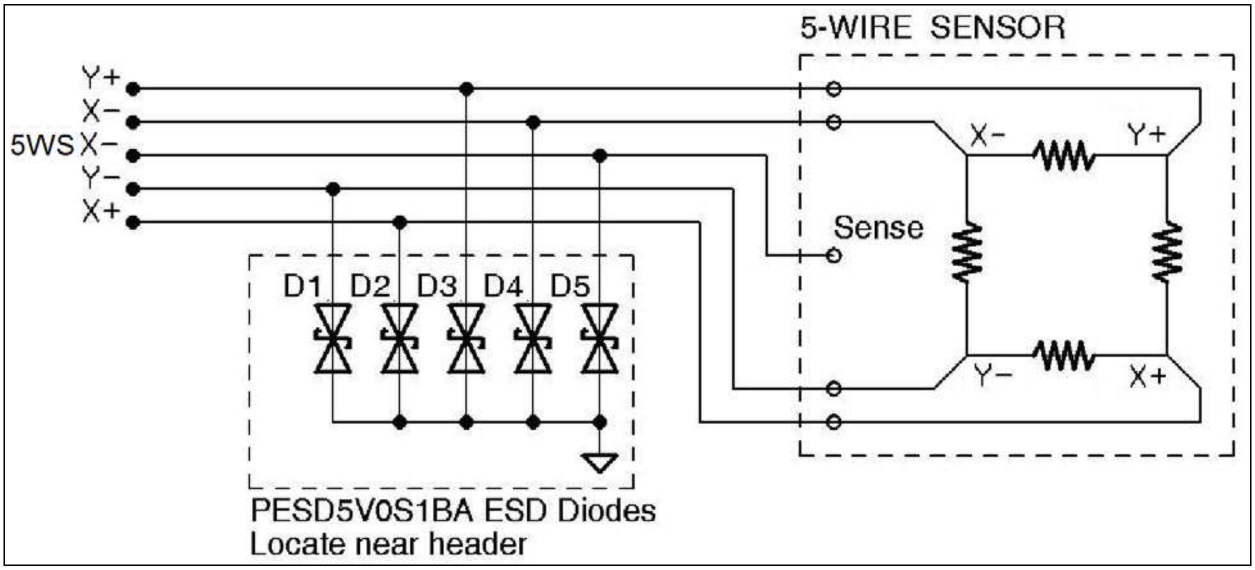
Sensor tail pinouts can vary by manufacturer and part number. Ensure sensor tail pins for one pair of diagonally related sensor corners are connected to the controller's X-/X+ pins and the tail pins for the other pair of diagonally related corners are connected to the controller's Y-/Y+ pins.

The controller's X-/X+ and Y-/Y+ pin pairs do not need to connect to a specific sensor axis. The orientation of controller pins X- and X+ to the two selected diagonal sensor corners is not important.

Likewise, the orientation of controller pins Y- and Y+ to the other two selected diagonal sensor corners is not important. The sensor tail pin connected to its top layer must be connected to the controller's 5WSX- pin.

Connections to a 5-wire touch sensor are shown in Figure 3-3 below.

FIGURE 3-3: 5-WIRE TOUCH SENSOR INTERFACE



Tie unused controller pins SX+, SY-, and SY+ to Vss.

See "**Section 3.8 "ESD Considerations"**" and **Section 3.9 "Noise Considerations"** for important information regarding the capacitance of the controller schematic hardware.

3.5 8-Wire Touch Sensor Interface

Sensor tail pinouts can vary by manufacturer and part number. Ensure both sensor tail pins for one sensor axis (layer) are connected to the controller's X-/X+ pins and the tail pins for the other sensor axis (layer) are connected to the controller's Y-/Y+ pins.

The controller's X-/X+ and Y-/Y+ pin pairs do not need to connect to a specific sensor axis. The orientation of controller pins X- and X+ to the two sides of a given sensor axis is not important. Likewise, the orientation of controller pins Y- and Y+ to the two sides of the other sensor axis is not important.

The 8-wire sensor differs from a 4-wire sensor in that each edge of an 8-wire sensor has a secondary connection brought to the sensor's tail. These secondary connections are referred to as "sense" lines. The controller pins associated with the sense line for an 8-wire sensor contain an 'S' prefix in their respective names. For example, the SY- pin is the sense line connection associated with the main Y- pin connection.

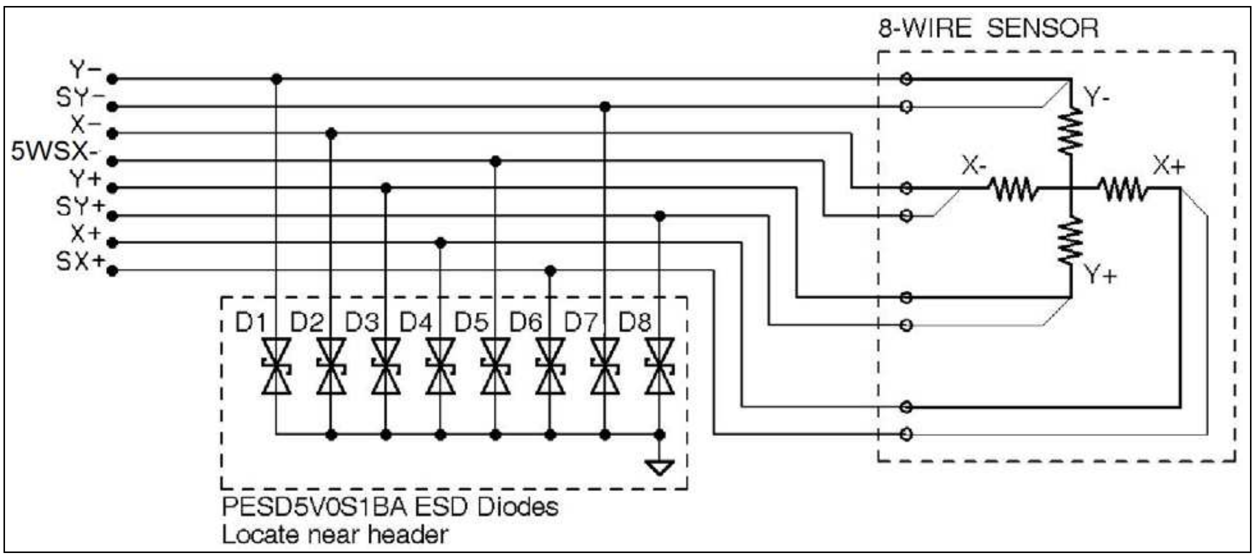
Consult with the sensor manufacturer's specification to determine which member of each edge connected pair is the special 8-wire "sense" connection. Incorrectly connecting the sense and excite lines to the controller will adversely affect performance.

The controller requires that the main and "sense" tail pin pairs for sensor edges be connected to controller pin pairs as follows:

- Y- and SY-
- Y+ and SY+
- X- and 5WSX-
- X+ and SX+

Connections to a 8-wire touch sensor are shown in Figure 3-4 below.

FIGURE 3-4: 8-WIRE TOUCH SENSOR INTERFACE

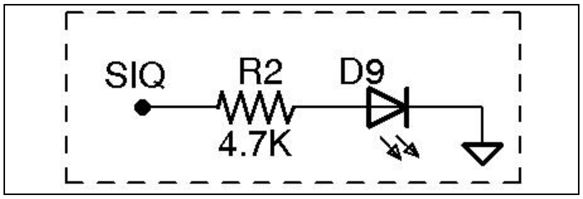


See **Section 3.8 "ESD Considerations"** and **Section 3.9 "Noise Considerations"** for important information regarding the capacitance of the controller schematic hardware.

3.6 Status LED

The LED and associated resistor are optional.

FIGURE 3-5: LED SCHEMATIC



The LED serves as a status indicator that the controller is functioning. It will slow flash when the controller is running with no touch in progress. It will flicker quickly (mid-level on) when a touch is in progress.

If the LED is used with SPI communication, then the LED will be off with no touch and flicker quickly (mid-level on) when a touch is in progress.

Note:	If the SIQ pin is not used, it must be left as a No Connect and NOT tied to circuit VDD or VSS.

3.7 WAKE Pin

The AR1000's WAKE pin is described as "Touch Wake-Up/Touch Detection". It serves the following three roles in the controller's functionality:

- Wake-up from touch
- Touch detection
- Measure sensor capacitance

The application circuit shows a 20 KΩ resistor connected between the WAKE pin and the X- pin on the controller chip. The resistor is required for product operation, based on all three of the above roles.

3.8 ESD Considerations

ESD protection is shown on the 4-wire, 5-wire, and 8-wire interface applications schematics.

The capacitance of alternate ESD diodes may adversely affect touch performance. A lower capacitance is better. The PESD5V0S1BA parts shown in the reference design have a typical capacitance of 35 pF. Test to ensure that selected ESD protection does not degrade touch performance.

ESD protection is shown in the reference design, but acceptable protection is dependent on your specific application. Ensure your ESD solution meets your design requirements.

3.9 Noise Considerations

Touch sensor filtering capacitors are included in the reference design.

Warning: Changing the value of the capacitors may adversely affect performance of the touch system.

4.0 I²C COMMUNICATIONS

The AR1021 is an I^2C slave device with a 7-bit address of 0x4D, supporting up to 400 kHz bit rate.

A master (host) device interfaces with the AR1021.

4.1 I²C Hardware Interface

A summary of the hardware interface pins is shown below in Table 4-1.

TABLE 4-1: I²C HARDWARE INTERFACE

AR1021 Pin	Description
M1	Connect to V_{SS} to select I^2C communications
SCL	Serial Clock
SDA	Serial Data
SDO	Data ready interrupt output to master

M1 Pin

- The M1 pin must be connected to V_{SS} to configure the AR1021 for I^2C communications.

SCL Pin

- The SCL (Serial Clock) pin is electrically open-drain and requires a pull-up resistor, typically 2.2 KΩ to 10 KΩ, from SCL to V_{DD}.
- SCL Idle state is high.

SDA Pin

- The SDA (Serial Data) pin is electrically open-drain and requires a pull-up resistor, typically 2.2 KΩ to 10 KΩ, from SDA to V_{DD}.
- SDA Idle state is high.
- Master write data is latched in on SCL rising edges.
- Master read data is latched out on SCL falling edges to ensure it is valid during the subsequent SCL high time.

SDO Pin

- The SDO pin is a driven output interrupt to the master.
- SDO Idle state is low.
- SDO will be asserted high when the AR1021 has data ready (touch report or command response) for the master to read.

4.2 I²C Pin Voltage Level Characteristics

TABLE 4-2: I²C PIN VOLTAGE LEVEL CHARACTERISTICS

Function	Pin	Input	Output
SCL/SCK	SCL/SCK/TX	$V_{SS} \le V_{IL} \le 0.2 \cdot V_{DD}$ $0.8 \cdot V_{DD} \le V_{IH} \le V_{DD}$	—
SDO	SDO	—	$V_{SS} \le V_{OL}$(1) $\le (1.2V - 0.15 \cdot V_{DD})$(2) $(1.25 \cdot V_{DD} - 2.25V)$(3) $\le V_{OH}$(1) $\le V_{DD}$
SDA	SDI/SDA/RX	$V_{SS} \le V_{IL} \le 0.2 \cdot V_{DD}$ $0.8 \cdot V_{DD} \le V_{IH} \le V_{DD}$	Open-drain

Note 1: These parameters are characterized but not tested.
2: At 10 mA.
3: At –4 mA.

4.3 Addressing

The AR1021's device ID 7-bit address is: 0x4D (`0b1001101`)

TABLE 4-3: I²C DEVICE ID ADDRESS

Device ID Address, 7-bit						
A7	A6	A5	A4	A3	A2	A1
1	0	0	1	1	0	1

TABLE 4-4: I²C DEVICE WRITE ID ADDRESS

A7	A6	A5	A4	A3	A2	A1	A0	
1	0	0	1	1	0	1	0	0x9A

TABLE 4-5: I²C DEVICE READ ID ADDRESS

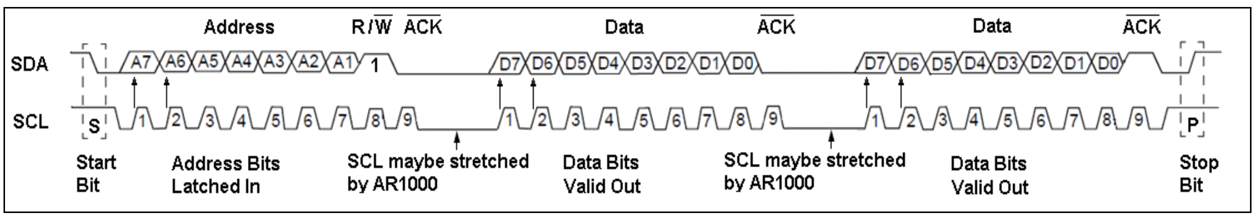
A7	A6	A5	A4	A3	A2	A1	A0	
1	0	0	1	1	0	1	1	0x9B

4.4 Master Read Bit Timing

Master read is to receive touch reports and command responses from the AR1021.

- Address bits are latched into the AR1021 on the rising edges of SCL.
- Data bits are latched out of the AR1021 on the rising edges of SCL.
- ACK is presented (by AR1021 for address, by master for data) on the ninth clock.
- The master must monitor the SCL pin prior to asserting another clock pulse, as the AR1021 may be holding off the master by stretching the clock.

FIGURE 4-1: I²C MASTER READ BIT TIMING DIAGRAM



Steps

1. SCL and SDA lines are Idle high.
2. Master presents "Start" bit to the AR1021 by taking SDA high-to-low, followed by taking SCL high-to-low.
3. Master presents 7-bit Address, followed by a R/W = `1` (Read mode) bit to the AR1021 on SDA, at the rising edge of eight master clock (SCL) cycles.
4. AR1021 compares the received address to its device ID. If they match, the AR1021 acknowledges (ACK) the master sent address by presenting a low on SDA, followed by a low-high-low on SCL.
5. Master monitors SCL, as the AR1021 may be "clock stretching", holding SCL low to indicate that the master should wait.

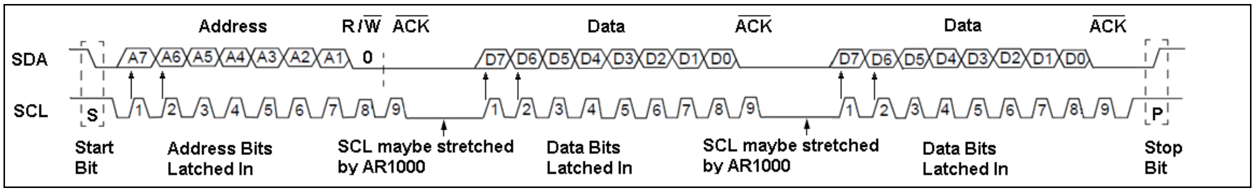
6. Master receives eight data bits (MSb first) presented on SDA by the AR1021, at eight sequential master clock (SCL) cycles. The data is latched out on SCL falling edges to ensure it is valid during the subsequent SCL high time.
7. If data transfer is not complete, then:
   - Master acknowledges (ACK) reception of the eight data bits by presenting a low on SDA, followed by a low-high-low on SCL.
   - Go to step 5.
8. If data transfer is complete, then:
   - Master acknowledges (ACK) reception of the eight data bits and a completed data transfer by presenting a high on SDA, followed by a low-high-low on SCL.
9. Master presents a "Stop" bit to the AR1021 by taking SCL low-high, followed by taking SDA low-to-high.

4.5 Master Write Bit Timing

Master write is to send supported commands to the AR1021.

- Address bits are latched into the AR1021 on the rising edges of SCL.
- Data bits are latched into the AR1021 on the rising edges of SCL.
- ACK is presented by AR1021 on the ninth clock.
- The master must monitor the SCL pin prior to asserting another clock pulse, as the AR1021 may be holding off the master by stretching the clock.

FIGURE 4-2: I²C MASTER WRITE BIT TIMING DIAGRAM



Steps

1. SCL and SDA lines are Idle high.
2. Master presents "Start" bit to the AR1021 by taking SDA high-to-low, followed by taking SCL high-to-low.
3. Master presents 7-bit Address, followed by a R/W = 0 (Write mode) bit to the AR1021 on SDA, at the rising edge of eight master clock (SCL) cycles.
4. AR1021 compares the received address to its device ID. If they match, the AR1021 acknowledges (ACK) the master sent address by presenting a low on SDA, followed by a low-high-low on SCL.
5. Master monitors SCL, as the AR1021 may be "clock stretching", holding SCL low to indicate the master should wait.
6. Master presents eight data bits (MSb first) to the AR1021 on SDA, at the rising edge of eight master clock (SCL) cycles.
7. AR1021 acknowledges (ACK) receipt of the eight data bits by presenting a low on SDA, followed by a low-high-low on SCL.
8. If data transfer is not complete, then go to step 5.
9. Master presents a "Stop" bit to the AR1021 by taking SCL low-high, followed by taking SDA low-to-high.

4.6 Clock Stretching

The master normally controls the clock line SCL. Clock stretching is when the slave device holds the SCL line low, indicating to the master that it is not ready to continue the communications.

During communications, the AR1021 may hold off the master by stretching the clock with a low on SCL.

The master must monitor the slave SCL pin to ensure the AR1021 is not holding it low, prior to asserting another clock pulse for transmitting or receiving.

4.7 AR1020 Write Conditions

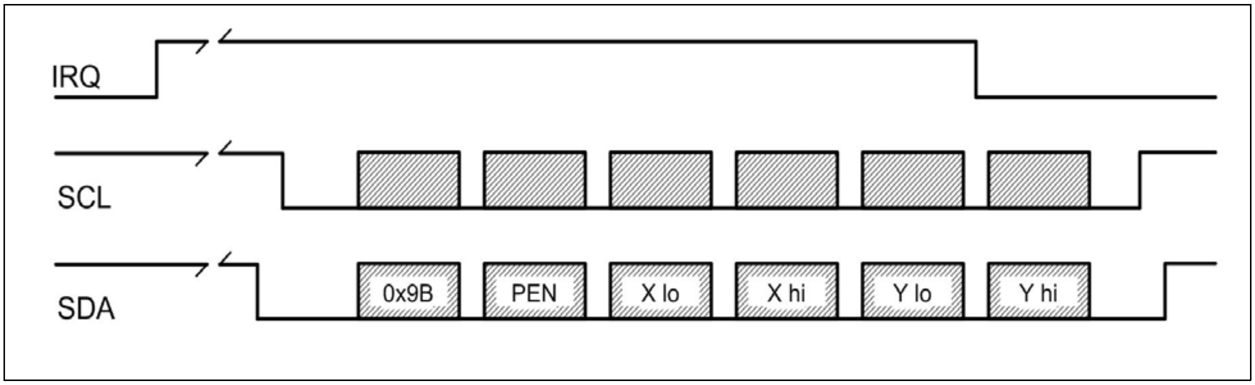
The AR1020 part does not implement clock stretching on write conditions.

A 50 us delay is needed before the Stop bit, when clocking a command to the AR1020.

4.8 Touch Report Protocol

Touch coordinates, when available, are provided to the master by the AR1021 in the following protocol (See Figure 4-3).

FIGURE 4-3: I²C TOUCH REPORT PROTOCOL



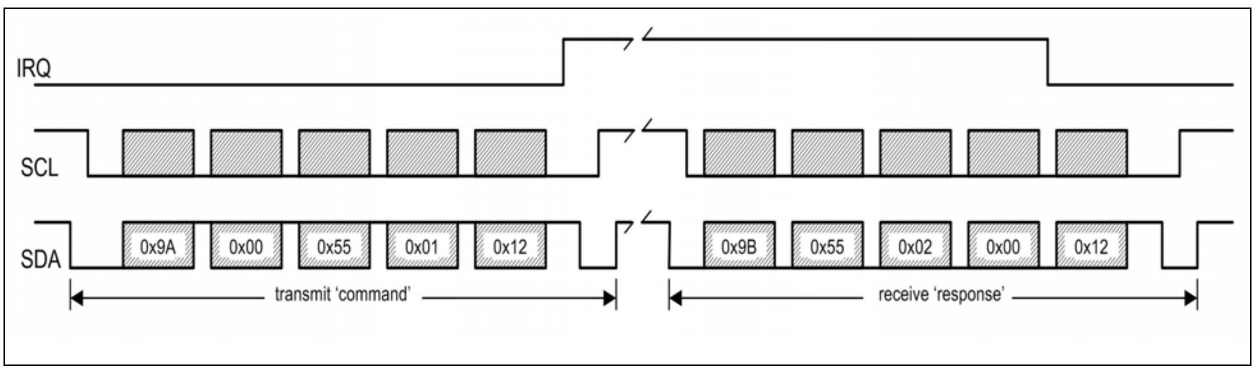
Note that the IRQ signal shown above occurs on the SDO pin of the AR1021.

4.9 Command Protocol

The master issues supported commands to the AR1021 in the following protocol.

Below is an example of the `ENABLE_TOUCH` command (see Figure 4-4).

FIGURE 4-4: I²C COMMAND PROTOCOL



Note that the IRQ shown above occurs on the SDO pin.

- 0x9A AR1021 Device ID address
- 0x00 Protocol command byte (send 0x00 for the protocol command register)
- 0x55 Header
- 0x01 Data size
- 0x12 Command

4.10 Sleep State

Pending communications are not maintained through a sleep/wake cycle.

If the SDO pin is asserted for a pending touch report or command response, and the AR1021 enters a Sleep state, prior to the master performing a read on the data, then the data is lost.

5.0 SPI COMMUNICATIONS

SPI operates in Slave mode with an Idle low SCK and data transmitted on the SCK falling edge.

5.1 SPI Hardware Interface

A summary of the hardware interface pins is shown below in Table 5-1.

TABLE 5-1: SPI HARDWARE INTERFACE

AR1021 Pin	Description
M1	Connect to VDD to select SPI communications
SDI	Serial data sent from master
SCK	Serial clock from master
SDO	Serial data to master SPI
SIQ	Interrupt output to master (optional)
SS	Slave Select (optional)

SCK Pin

- The AR1021 controller's SCL/SCK/TX pin receives Serial Clock (SCK), controlled by the host.
- The Idle state of the SCK should be low.
- Data is transmitted on the falling edge of SCK.

SDI Pin

- The AR1021 controller's SDI/SDA/RX pin reads Serial Data Input (SDI), sent by the host.

SDO Pin

- The AR1021 controller's SDO pin presents Serial Data Output (SDO) to the host.

SIQ Pin

- The AR1021 controller's SIQ pin provides an optional interrupt output from the controller to the host.
- The SIQ pin is asserted high when the controller has data available (a touch report or a command response) for the host.
- The SIQ pin is deasserted after the host clocks out the first byte of the data packet.

Note:	The AR1000 Development kit PICkit™ Serial Pin 1 is designated for the SIQ interrupt pin after the firmware updated is executed for the PICkit.

SS Pin

- The AR1021 controller's SS pin provides optional "slave select" functionality.

SS Pin Level	AR1021 Select
VSS	Active
VDD	Inactive

In the 'inactive' state, the controller's SDO pin presents a high-impedance in order to prevent bus contention with another device on the SPI bus.

5.2 SPI Pin Voltage Level Characteristics

TABLE 5-2: SPI PIN VOLTAGE CHARACTERISTICS

Operating Voltage: $2.5V \le V_{DD} \le 5.25V$			
Function	**Pin**	**Input**	**Output**
SCK	SCL/SCK/TX	$V_{SS} \le \mathbf{V_{IL}} \le 0.2*V_{DD}$ $0.8*V_{DD} \le \mathbf{V_{IH}} \le V_{DD}$	—
SDI	SDI/SDA/RX	$V_{SS} \le \mathbf{V_{IL}} \le 0.2*V_{DD}$ $0.8*V_{DD} \le \mathbf{V_{IH}} \le V_{DD}$	—
SDO	SDO	—	$V_{SS} \le \mathbf{V_{OL}}$[1] $\le (1.2V - 0.15*V_{DD})$[2] $(1.25*V_{DD} - 2.25V)$[3] $\le \mathbf{V_{OH}}$[1] $\le V_{DD}$
SIQ	SIQ	—	$V_{SS} \le \mathbf{V_{OL}}$[1] $\le (1.2V - 0.15*V_{DD})$[2] $(1.25*V_{DD} - 2.25V)$[3] $\le \mathbf{V_{OH}}$[1] $\le V_{DD}$
SS	SS	$V_{SS} \le \mathbf{V_{IL}} \le 0.2*V_{DD}$ $0.8*V_{DD} \le \mathbf{V_{IH}} \le V_{DD}$	

Note 1: These parameters are characterized but not tested.
2: At 10 mA.
3: At -4 mA.

5.3 Data Flow

SPI data is transferred by the host clocking the AR1021 controller's Serial Clock (SCK) pin.

Each host driven clock cycle simultaneously shifts a bit of data into and out from the AR1021 controller:

- Out from the AR1021 controller's Serial Data Out (SDO) line.
- Into the AR1021 controller's Serial Data In (SDI) line.

The data is shifted Most Significant bit (MSb) first.

If the host clocks data out from the AR1021 controller when no valid data is available, then a byte value of 0x4d will be presented by the controller.

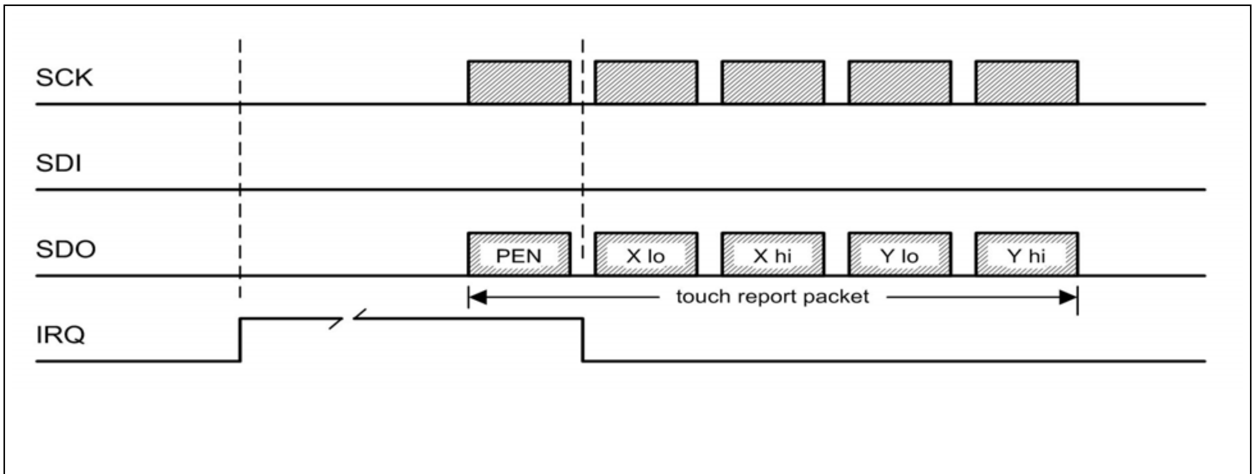
5.4 Touch Report Protocol

The AR1021 controller's touch reporting is interrupt driven:

- The AR1021 controller asserts the SIQ interrupt pin high when it has a touch report ready.
- The host clocks out the bytes of the touch report packet from the AR1021 controller.
- The AR1021 controller clears the SIQ interrupt pin low, after the first byte of the touch report packet has been clocked out by the host.

The communication protocol for the AR1021 controller reporting touches to the host as shown below in Figure 5-1.

FIGURE 5-1: SPI TOUCH REPORT PROTOCOL



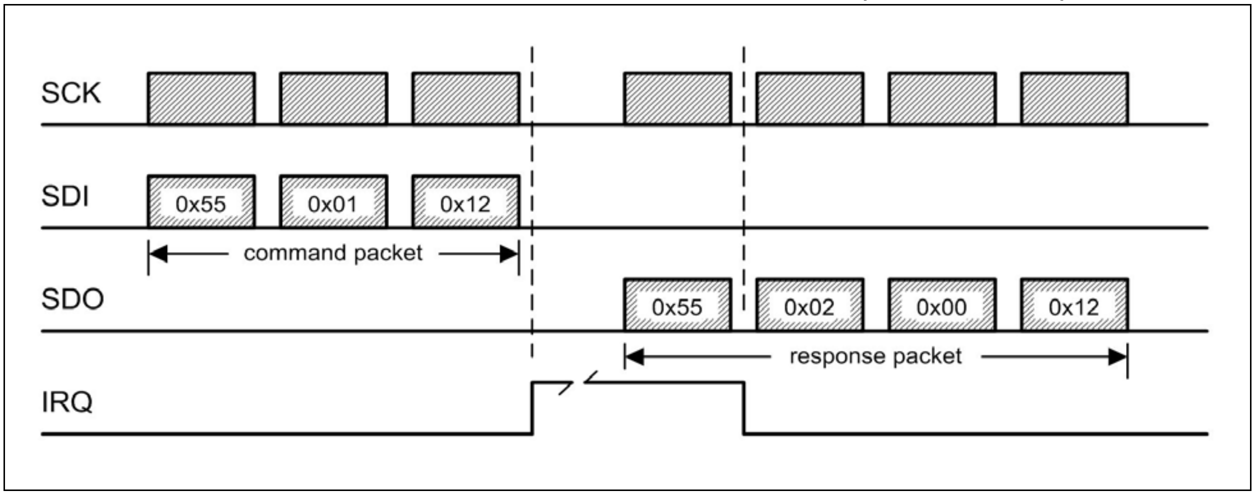
5.5 Command Protocol

The AR1021 controller receives commands from the host as follows:

- The host clocks the bytes of a command to the AR1021 controller.
- The AR1021 controller asserts the SIQ interrupt pin high when it is ready with a response to the command sent by the host.
- The host clocks out the bytes of the command response from the AR1021 controller.
- The AR1021 controller clears the SIQ interrupt pin low, after the first byte of the command response has been clocked out by the host.

The communication protocol for the host sending the `ENABLE_TOUCH` command to the AR1021 controller is shown below in Figure 5-2.

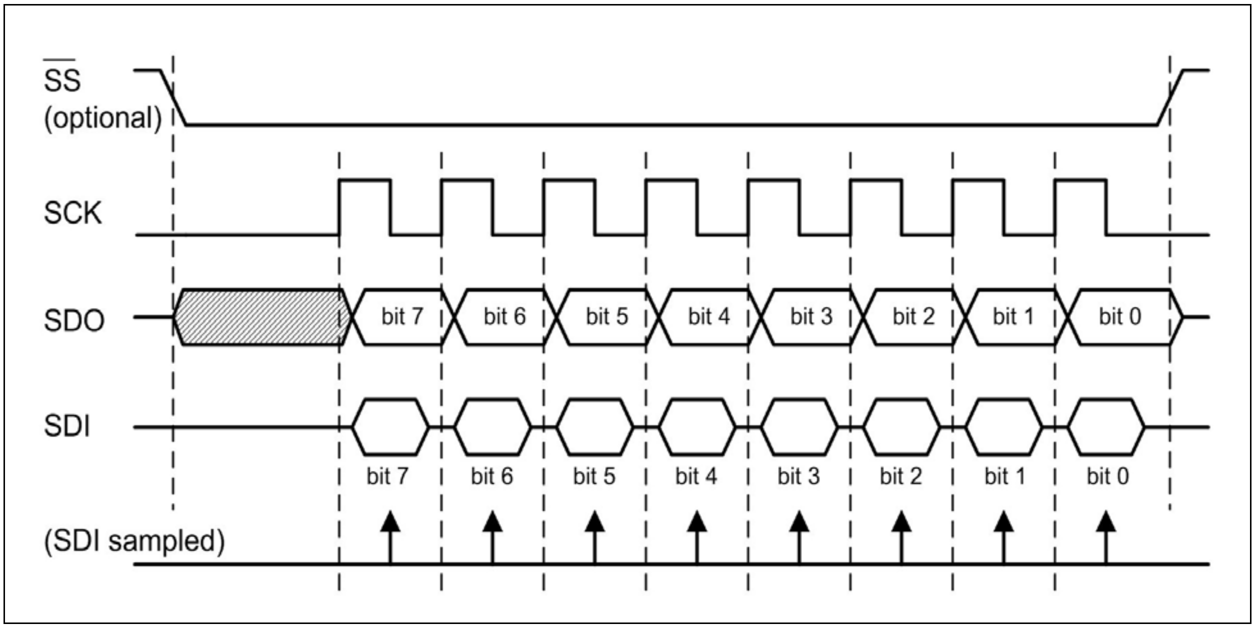
FIGURE 5-2: SPI TIMING DIAGRAM – COMMAND PROTOCOL (`ENABLE_TOUCH`)



5.6 SPI Bit Timing – General

General timing waveforms are shown below in Figure 5-3.

FIGURE 5-3: SPI GENERAL BIT TIMING WAVEFORM



5.7 Timing – Bit Details

5.7.1 BIT RATE

The SPI standard does not specify a maximum data rate for the serial bus. In general, SPI data rates can be in MHz. Peripherals devices, such as the AR1021 controller, specify their own unique maximum SPI data rates.

The maximum SPI bit rate for the AR1021 controller is ~900 kHz.

Characterization has been performed at bit rates of ~39 kHz and ~156 kHz.

5.7.2 INTER-BYTE DELAY

The AR1021 controller requires an inter-byte delay of ~50 us. This means the host should wait ~50 us between the end of clocking a given byte and the start of clocking the next byte.

5.7.3 BIT TIMING – DETAIL

Characterized timing details are shown below, in Figure 5-4.

FIGURE 5-4: SPI BIT TIMING – DETAIL

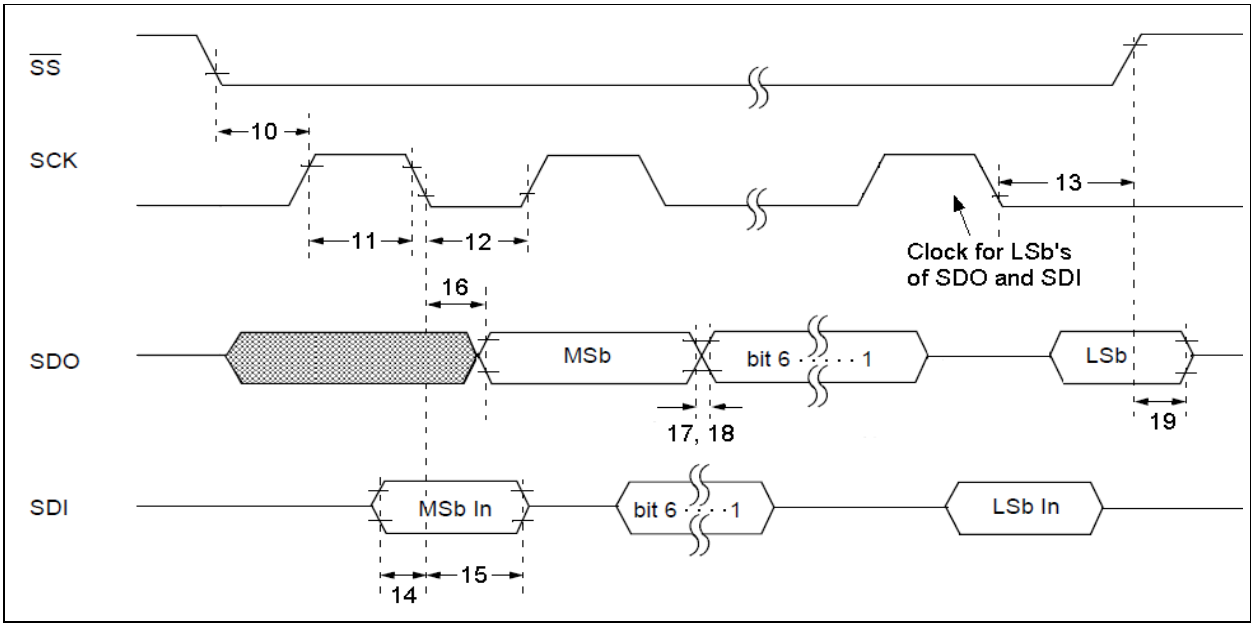


TABLE 5-3: SPI BIT TIMING MIN. AND MAX. VALUES

Parameter Number[1]	Parameter Description	Min.	Max.	Units
10	SS↓ (select) to SCK↑ (initial)	500	—	ns
11	SCK high	550	—	ns
12	SCK low	550	—	ns
13	SCK↓ (last) to SS↑ (deselect)	800	—	ns
14	SDI setup before SCK↓	100	—	ns
15	SDI hold after SCK↓	100	—	ns
16	SDO valid after SCK↓	—	150	ns
17	SDO↑ rise	—	50	ns
18	SDO↓ fall	—	50	ns
19	SS↑ (deselect) to SDO High-z	10	50	ns

Note 1: Parameters are characterized, but not tested.

6.0 UART COMMUNICATIONS

TABLE 6-1: UART HARDWARE INTERFACE

AR1011 Pin	Description
M1	Connect M1 to V_{DD} to select UART communications
TX	Transmit to host
RX	Receive from host
SDO	Connect SDO to V_{SS}

UART communication is fixed at 9600 baud rate, 8N1 format.

Sleep mode will cause the TX line to drop low, which may appear as a `0x00` byte sent from the controller.

7.0 TOUCH REPORTING PROTOCOL

Touch coordinates are sent from the controller to the host system in a 5-byte data packet, which contains the X-axis coordinate, Y-axis coordinate, and a “Pen-Up/ Down” touch status.

The range for X-axis and Y-axis coordinates is from 0-4095 (12-bit). The realized resolution is 1024, and bits X1:X0 and Y1:Y0 are zeros.

It is recommended that applications be developed to read the 12-bit coordinates from the packet and use them in a 12-bit format. This enhances the application robustness, as it will work with either 10 or 12 bits of coordinate information.

The touch coordinate reporting protocol is shown below in Table 7-1.

TABLE 7-1: TOUCH COORDINATE REPORTING PROTOCOL

Byte #	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	1	R	R	R	R	R	R	P
2	0	X6	X65	X4	X3	X2	X1	X0
3	0	0	0	X11	X10	X9	X8	X7
4	0	Y6	Y5	Y4	Y3	Y2	Y1	Y0
5	0	0	0	Y11	Y10	Y9	Y8	Y7

where:

- P: 0 Pen Up, 1 Pen Down
- R: Reserved
- X11-X0: X-axis coordinate
- Y11-Y0: Y-axis coordinate

8.0 CONFIGURATION REGISTERS

The Configuration registers allow application-specific customization of the controller. The default values have been optimized for most applications and are automatically used, unless you choose to change them.

Unique sensors and/or product applications may benefit from adjustment of Configuration registers.

> **Note:** Although most registers can be configured for a value ranging from 0 to 255, using a value outside the specified range for the specific register may negatively impact performance.

8.1 Restoring Default Parameters

- AR1010/AR1020

The factory default settings for the Configuration registers can be recovered by writing a value of 0xFF to address 0x00 of the EEPROM, then cycling power.

- AR1011/AR1021

The factory default settings for the Configuration registers can be recovered by writing a value of 0xFF to addresses 0x01 and 0x29 of the EEPROM, then cycling power.

TABLE 8-1: CONFIGURATION REGISTERS

Register Name	Address Offset	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	AR1010/ AR1020 Default	AR1011/ AR1021 Default
<Special Use>	0x00	<Non-Configurable>								0x58	0x58
<Special Use>	0x01	<Non-Configurable>								0x01	0x01
TouchThreshold	0x02	Value of: 0-255								0xC5	0xC5
SensitivityFilter	0x03	Value of: 0-255								0x04	0x04
SamplingFast	0x04	Value of: 1, 2, 4, 8, 16, 32, 64, 128								0x04	0x04
SamplingSlow	0x05	Value of: 1, 2, 4, 8, 16, 32, 64, 128								0x10	0x10
AccuracyFilterFast	0x06	Value of: 1-8								0x02	0x04
AccuracyFilterSlow	0x07	Value of: 1-8								0x08	0x08
SpeedThreshold	0x08	Value of: 0-255								0x04	0x04
<Special Use>	0x09	<Non-Configurable>								0x23	0x23
SleepDelay	0x0A	Value of: 0-255								0x64	0x64
PenUpDelay	0x0B	Value of: 0-255								0x80	0x80
TouchMode	0x0C	PD2	PD1	PD0	PM1	PM0	PU2	PU1	PU0	0xB1	0xB1
TouchOptions	0x0D	—	—	—	—	—	—	48W	CCE	0x00	0x00
CalibrationInset	0x0E									0x19	0x19
PenStateReportDelay	0x0F	Value of: 0-40								0xC8	0xC8
<Special Use>	0x10	Value of: 0-255								0x03	0x03
TouchReportDelay	0x11	<Non-Configurable>								0x00	0x00
<Special Use>	0x12	Value of: 0-255								0x00	0x00

Configuration registers are defined as an Offset value from the Start address for the register group.

To read or write to a register, do the following:

- Issue the `REGISTER_START_ADDRESS_REQUEST` command to obtain the Start address for the register group.
- Calculate the desired register's absolute address by adding the register's Offset value to Start address for the register group.
- Issue the `REGISTER_READ` or `REGISTER_WRITE` command, using the calculated register's absolute address.

> **Warning:** Use of invalid register values will yield unpredictable results.

8.2 Register Descriptions

8.2.1 TouchThreshold Register (OFFSET 0x02)

The TouchThreshold register sets the threshold for a touch condition to be detected as a touch. A touch is detected if it is below the TouchThreshold setting. Too small of a value might prevent the controller from accepting a real touch, while too large of a value might allow the controller to accept very light or false touch conditions. Valid values are as follows:

0 ≤ TouchThreshold ≤ 255

8.2.2 SensitivityFilter Register (OFFSET 0x03)

The SensitivityFilter register sets the level of touch sensitivity. A higher value is more sensitive to a touch (accepts a lighter touch), but may exhibit a less stable touch position. A lower value is less sensitive to a touch (requires a harder touch), but will provide a more stable touch position. Valid values are as follows:

0 ≤ SensitivityFilter ≤ 10

8.2.3 SamplingFast Register (OFFSET 0x04)

The SamplingFast register sets the level of touch measurement sample averaging, when touch movement is determined to be fast. See the SpeedThreshold register for information on the touch movement threshold. A lower value will provide for a higher touch coordinate reporting rate when touch movement is fast, but may exhibit more high-frequency random noise error in the touch position. A higher value will reduce the touch coordinate reporting rate when touch movement is fast, but will reduce high-frequency random noise error in the touch position. Valid values are as follows:

SamplingFast: <1, 4, 8, 16, 32, 64, 128>

Recommended Values: <4, 8, 16>

Higher values may improve accuracy with some sensors.

8.2.4 SamplingSlow Register (OFFSET 0x05)

The SamplingSlow register sets the level of touch measurement sample averaging, when touch movement is slow. See the SpeedThreshold register for information on the touch movement threshold. A lower value will increase the touch coordinate reporting rate when the touch motion is slow, but may exhibit a less stable more jittery touch position. A higher value will decrease the touch coordinate reporting rate when the touch motion is slow, but will provide a more stable touch position. Valid values are as follows:

SamplingSlow: 1, 2, 4, 8, 16, 32, 64, 128

8.2.5 AccuracyFilterFast Register (OFFSET 0x06)

The AccuracyFilterFast register sets the level of an accuracy enhancement filter, used when the touch movement is fast. See the SpeedThreshold register for information on the touch movement threshold. A lower value will provide better touch coordinate resolution when the touch motion is fast, but may exhibit more low-frequency noise error in the touch position. A higher value will reduce touch coordinate resolution when the touch motion is fast, but will reduce low-frequency random noise error in the touch position. Valid values are as follows:

1 ≤ AccuracyFilterFast ≤ 8

Higher values may improve accuracy with some sensors.

8.2.6 AccuracyFilterSlow Register (OFFSET 0x07)

The AccuracyFilterSlow register sets the level of an accuracy enhancement filter, used when the touch movement is slow. See the SpeedThreshold register for information on the touch movement threshold. A lower value will provide better touch coordinate resolution when the touch motion is slow, but may exhibit more low-frequency noise error in the touch position. A higher value will reduce touch coordinate resolution when the touch motion is slow, but will reduce low-frequency random noise error in the touch position. Valid values are as follows:

1 ≤ AccuracyFilterSlow ≤ 8

8.2.7 SpeedThreshold Register (OFFSET 0x08)

The SpeedThreshold register sets the threshold for touch movement to be considered as slow or fast. A lower value reduces the touch movement speed that will be considered as fast. A higher value increases the touch movement speed that will be considered as fast. Valid values are as follows:

0 ≤ SpeedThreshhold ≤ 255