Table of Contents

1.0 Device Overview 3
2.0 Basics of Resistive Sensors 5
3.0 Hardware 9
4.0 I^2C Communications 14
5.0 SPI Communications 18
6.0 UART Communications 22
7.0 Touch Reporting Protocol 23
8.0 Configuration Registers 24
9.0 Commands 30
10.0 Application Notes 39
11.0 Electrical Specifications 45
12.0 Packaging Information 47
Appendix A: Data Sheet Revision History 57
Appendix B: Device Differences 58
The Microchip Website 59
Customer Change Notification Service 59
Customer Support 59

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.

© 2009-2016 Microchip Technology Inc.

Downloaded from Arrow.com.

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

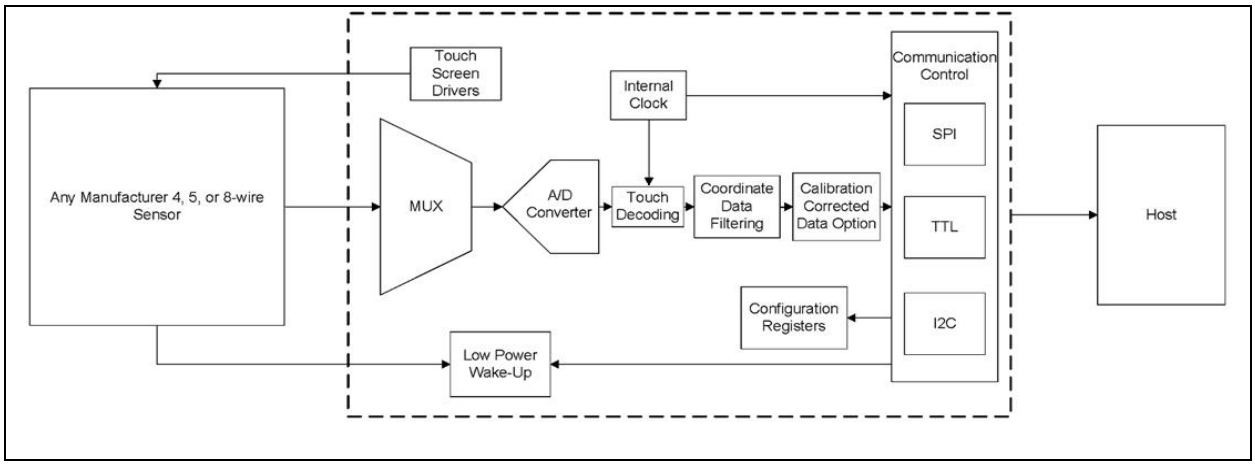
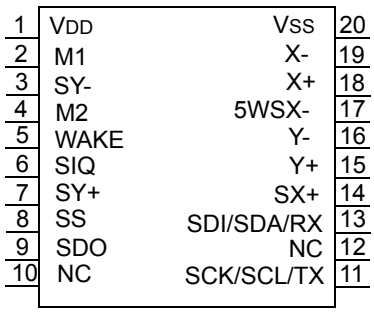
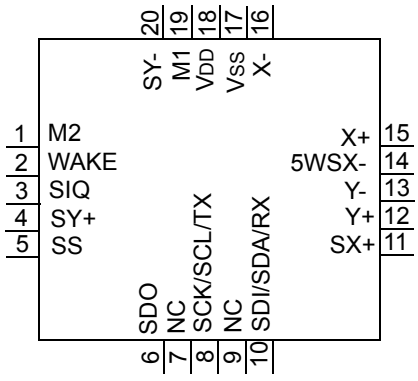


FIGURE 1-2: PIN DIAGRAM

AR1000 Series (SSOP, SOIC)



AR1000 Series (QFN)



Downloaded from Arrow.com.

TABLE 1-1: PIN DESCRIPTIONS

Pin		Function	Description/Comments
SSOP, SOIC	QFN		
1	18	V_{DD}	Supply Voltage
2	19	M1	Communication Selection
3	20	SY-	Sense Y- (8-wire). Tie to V_{SS}, 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 V_{SS}, if not used.
8	5	SS	Slave Select (SPI). Tie to V_{SS}, 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 V_{SS} or V_{DD}.
11	8	SCK/SCL/TX	SPI/I^2C Serial Clock/UART Transmit
12	9	NC	No connection. No connect or tie to V_{SS} or V_{DD}.
13	10	SDI/SDA/RX	I^2C Serial Data/SPI Serial Data Input/UART Receive
14	11	SX+	Sense X+ (8-wire). Tie to V_{SS}, 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 V_{SS}, if not used.
18	15	X+	X+ Drive
19	16	X-	X- Drive
20	17	V_{SS}	Supply Voltage Ground

© 2009-2016 Microchip Technology Inc.

Downloaded from Arrow.com.

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.

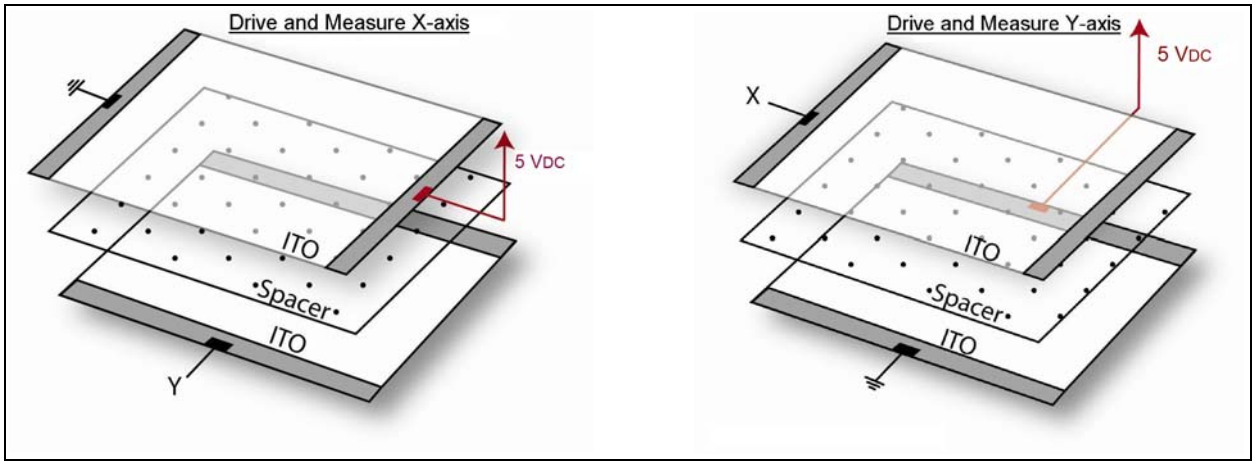
Downloaded from Arrow.com.

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



© 2009-2016 Microchip Technology Inc.

Downloaded from Arrow.com.

2.4 8-Wire Sensor

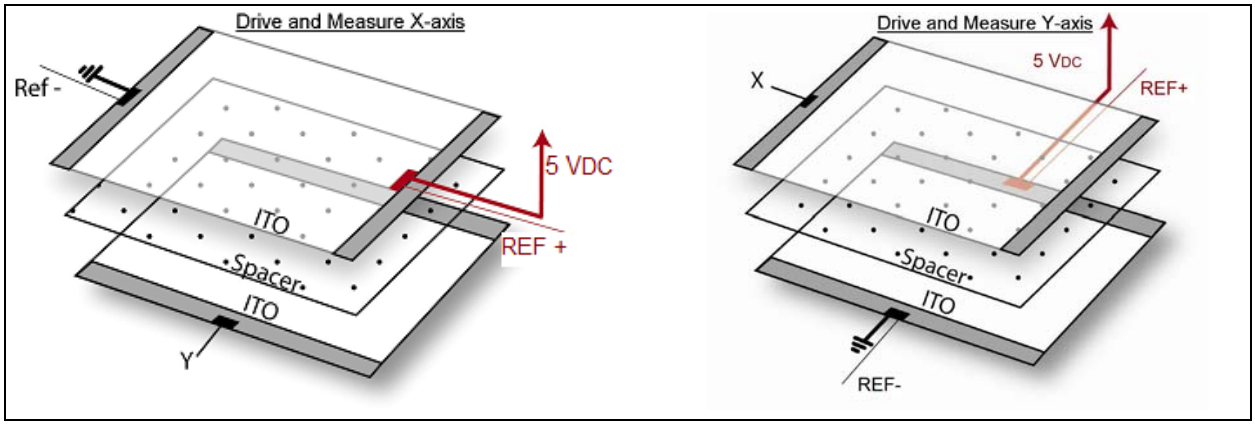
An 8-wire resistive touch sensor consists of a stable and flex layer, electrically separated by spacer dots. The layers are assembled perpendicular to each other. The touch position is determined by first applying a voltage gradient across the flex layer and using the stable layer to measure the flex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



Downloaded from Arrow.com.

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.

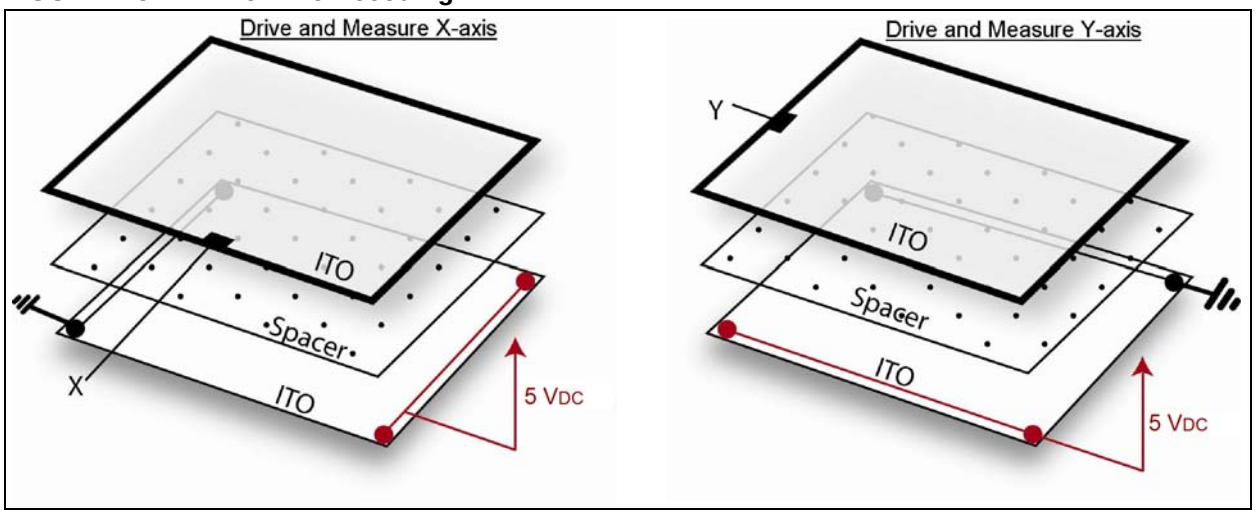
The voltage is not directly applied to the edges of the active layer, as it is for 4-wire and 8-wire sensors. The voltage is applied to the corners of a 5-wire sensor.

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



© 2009-2016 Microchip Technology Inc.

Downloaded from Arrow.com.

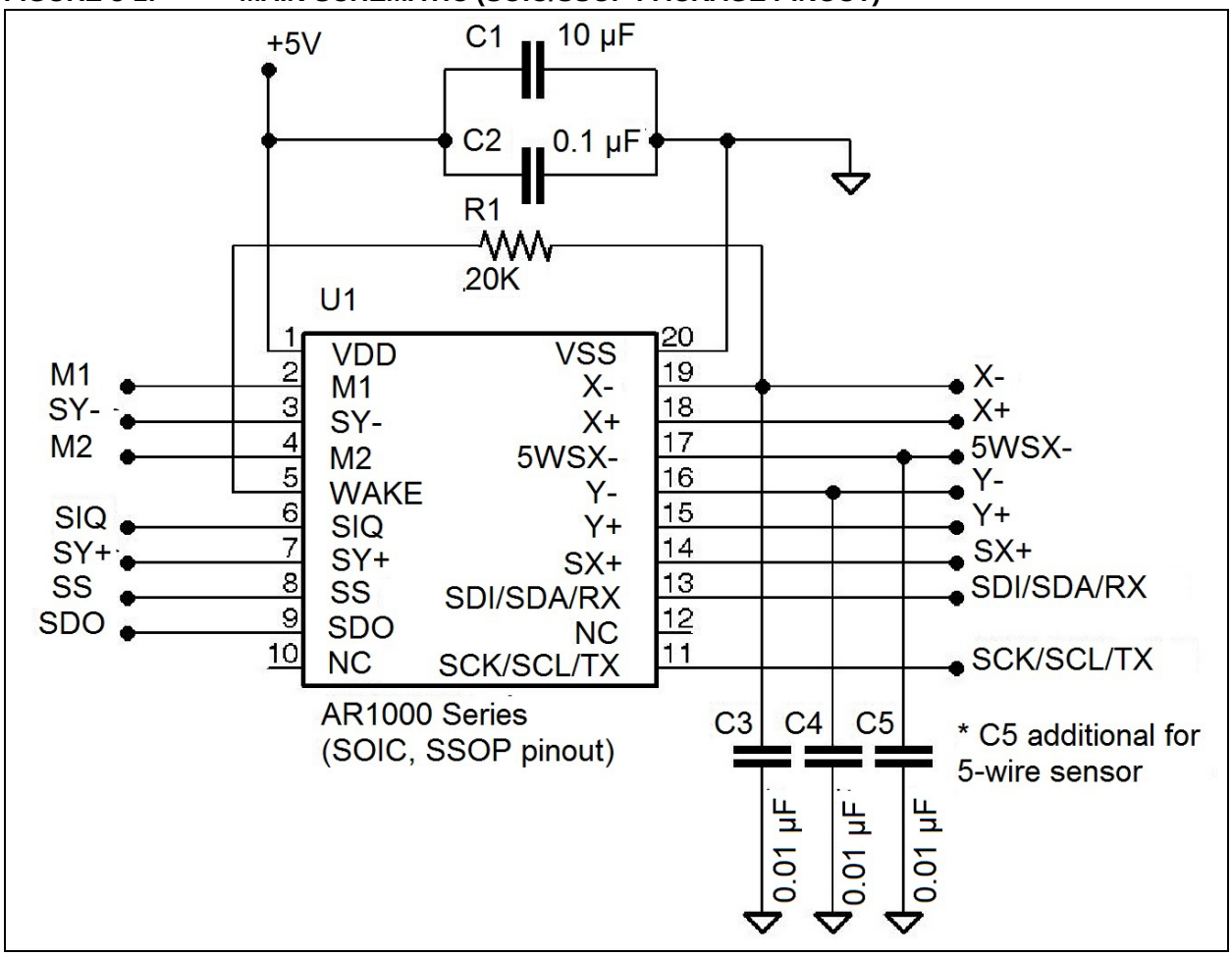
3.0 HARDWARE

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)



Downloaded from Arrow.com.

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.

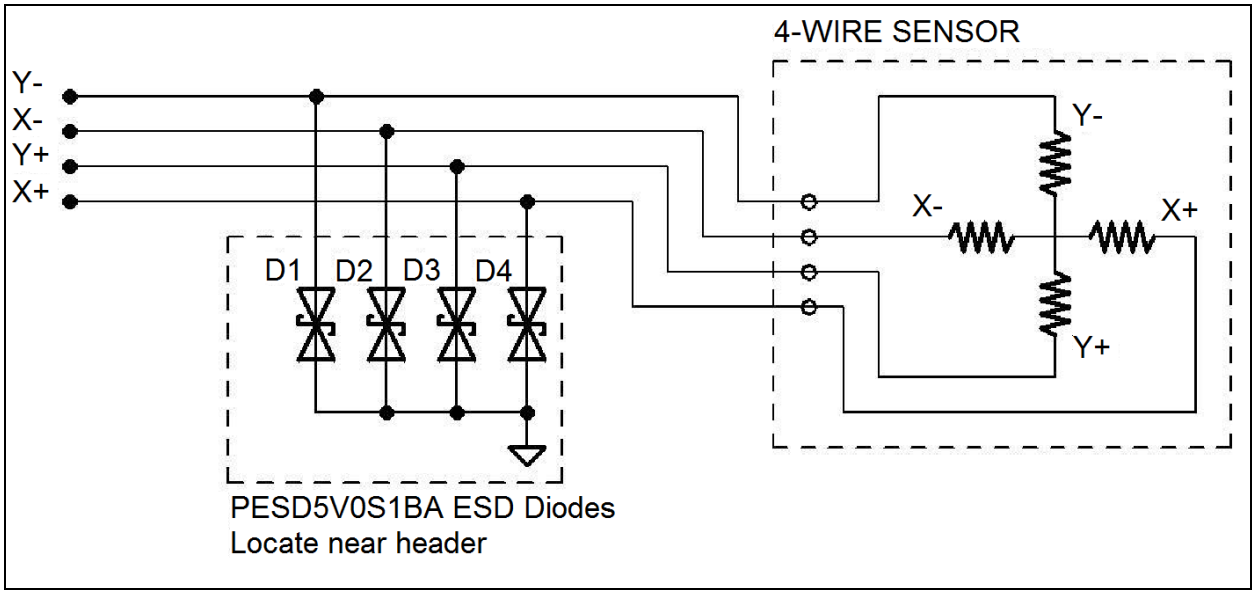
When 4/8-wire is hardware-selected, the controller defaults to 4-wire operation. If 8-wire operation is desired, then the TouchOptions Configuration register must be changed.

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.

© 2009-2016 Microchip Technology Inc.

Downloaded from Arrow.com.

3.4 5-Wire Touch Sensor Interface

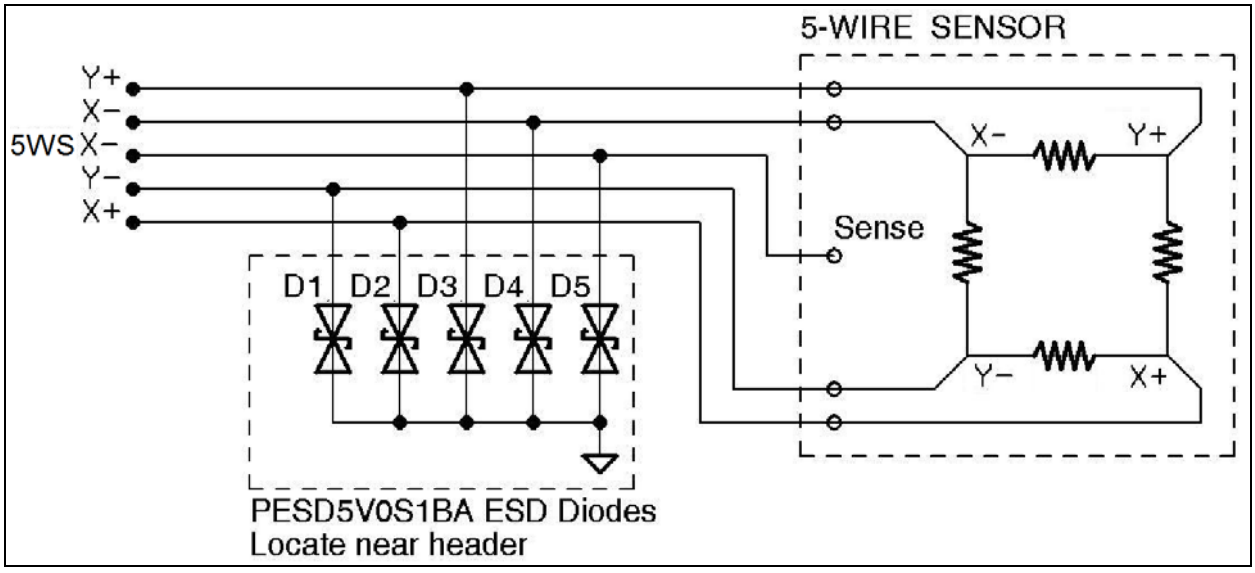
Sensor tail pinouts can vary by manufacturer and part number. Ensure sensor tail pins for one pair of diagonally related sensor corners are connected to the controller's X-/X+ pins and the tail pins for the other pair of diagonally related corners are connected to the controller's Y-/Y+ pins.

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.

Downloaded from Arrow.com.

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.

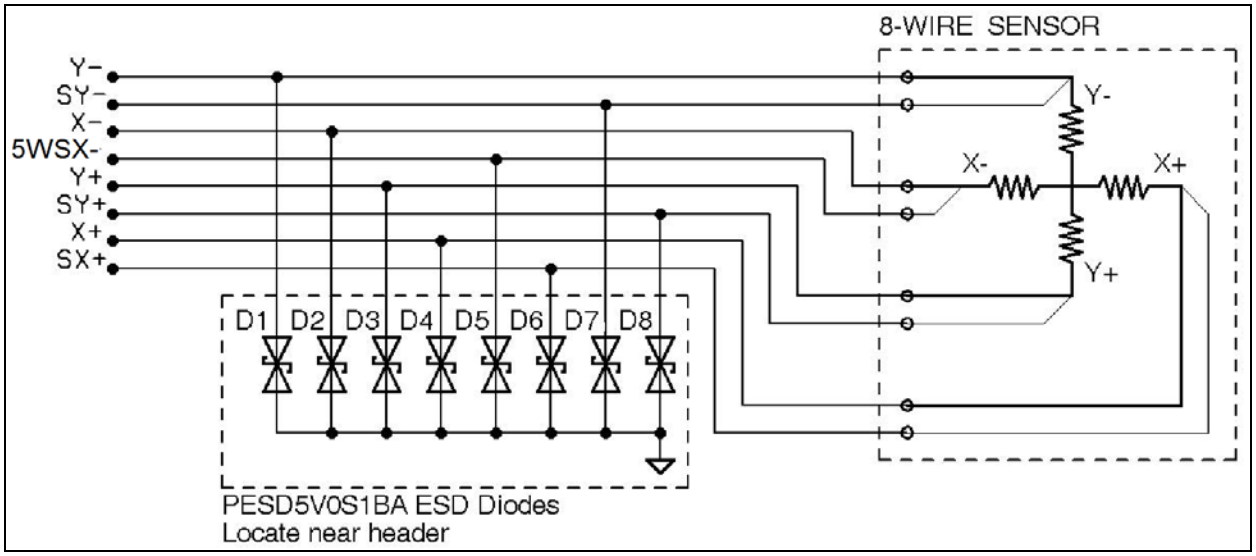
Consult with the sensor manufacturer's specification to determine which member of each edge connected pair is the special 8-wire "sense" connection. Incorrectly connecting the sense and excite lines to the controller will adversely affect performance.

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.

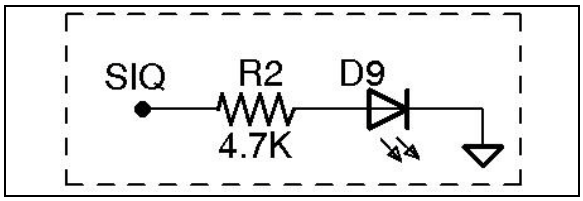
© 2009-2016 Microchip Technology Inc.

Downloaded from Arrow.com.

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 V_{DD} or V_{SS}.

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.

Downloaded from Arrow.com.

4.0 I²C COMMUNICATIONS

The AR1021 is an I²C 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 VSS to select I²C communications
SCL	Serial Clock
SDA	Serial Data
SDO	Data ready interrupt output to master

M1 Pin

- The M1 pin must be connected to VSS to configure the AR1021 for I²C 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 VDD.
- 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 VDD.
- 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.

© 2009-2016 Microchip Technology Inc.

Downloaded from Arrow.com.

4.2 I²C Pin Voltage Level Characteristics

TABLE 4-2: I²C PIN VOLTAGE LEVEL CHARACTERISTICS

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

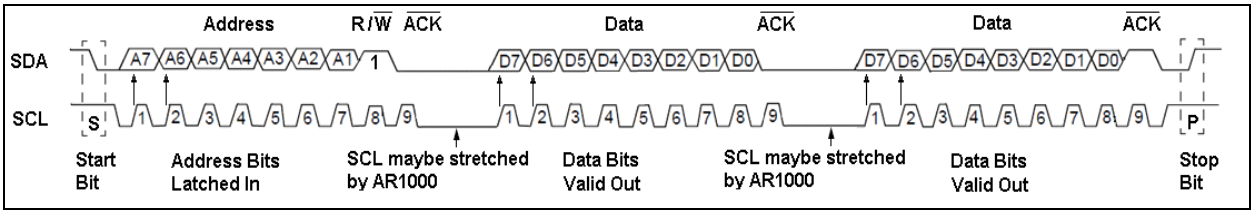
A7	A6	A5	A4	A3	A2	A1	A0	
1	0	0	1	1	0	1	1	0x9B

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.

Downloaded from Arrow.com.

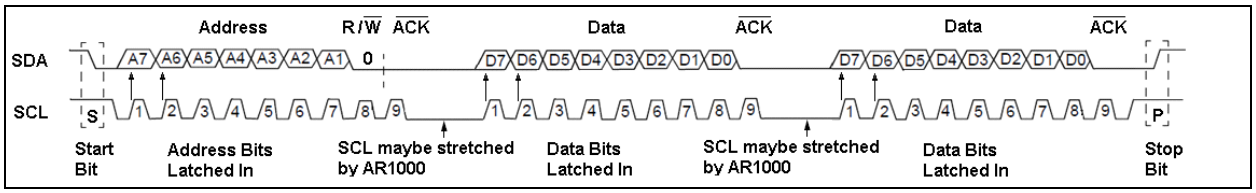
6. Master receives eight data bits (MSb first) presented on SDA by the AR1021, at eight sequential master clock (SCL) cycles. The data is latched out on SCL falling edges to ensure it is valid during the subsequent SCL high time.
7. If data transfer is not complete, then:
   - Master acknowledges (ACK) 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.

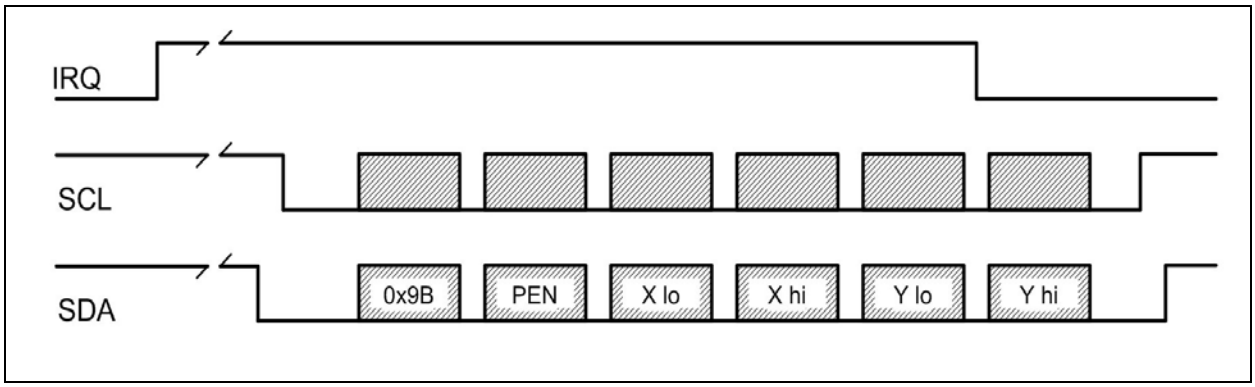
© 2009-2016 Microchip Technology Inc.

Downloaded from Arrow.com.

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



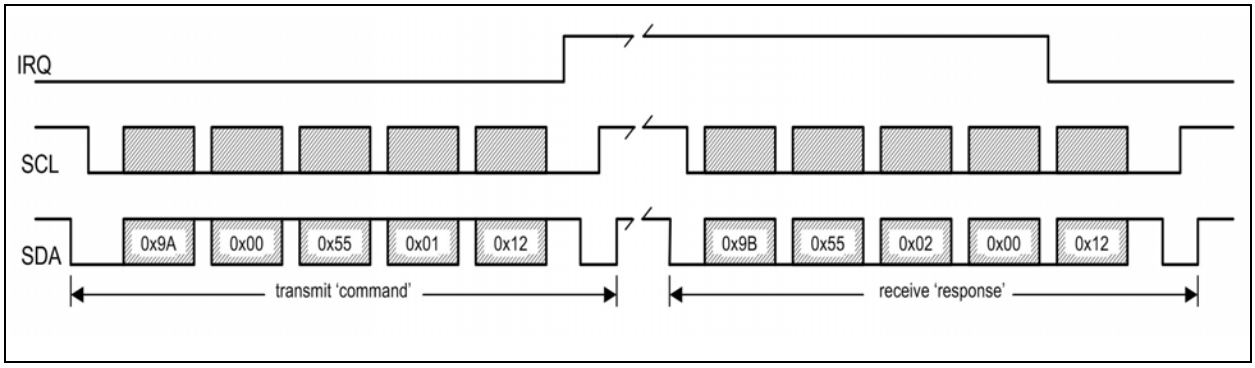
Note that the IRQ signal shown above occurs on the SDO pin of the AR1021.

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.

Downloaded from Arrow.com.

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.

© 2009-2016 Microchip Technology Inc.

Downloaded from Arrow.com.

5.2 SPI Pin Voltage Level Characteristics

TABLE 5-2: SPI PIN VOLTAGE CHARACTERISTICS

Operating Voltage: 2.5V ≤ V_{DD} ≤ 5.25V			
Function	**Pin**	**Input**	**Output**
SCK	SCL/SCK/TX	V_{SS} ≤ **V_{IL}** ≤ 0.2*V_{DD} 0.8*V_{DD} ≤ **V_{IH}** ≤ V_{DD}	—
SDI	SDI/SDA/RX	V_{SS} ≤ **V_{IL}** ≤ 0.2*V_{DD} 0.8*V_{DD} ≤ **VIH** ≤ V_{DD}	—
SDO	SDO	—	V_{SS} ≤ **V_{OL}**(1) ≤ (1.2V – 0.15*V_{DD})(2) (1.25*V_{DD} – 2.25V)(3) ≤ **V_{OH}**(1) ≤ V_{DD}
SIQ	SIQ	—	V_{SS} ≤ **V_{OL}**(1) ≤ (1.2V – 0.15*V_{DD})(2) (1.25*V_{DD} – 2.25V)(3) ≤ **V_{OH}**(1) ≤ V_{DD}
SS	SS	V_{SS} ≤ **V_{IL}** ≤ 0.2*V_{DD} 0.8*V_{DD} ≤ **V_{IH}** ≤ 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.

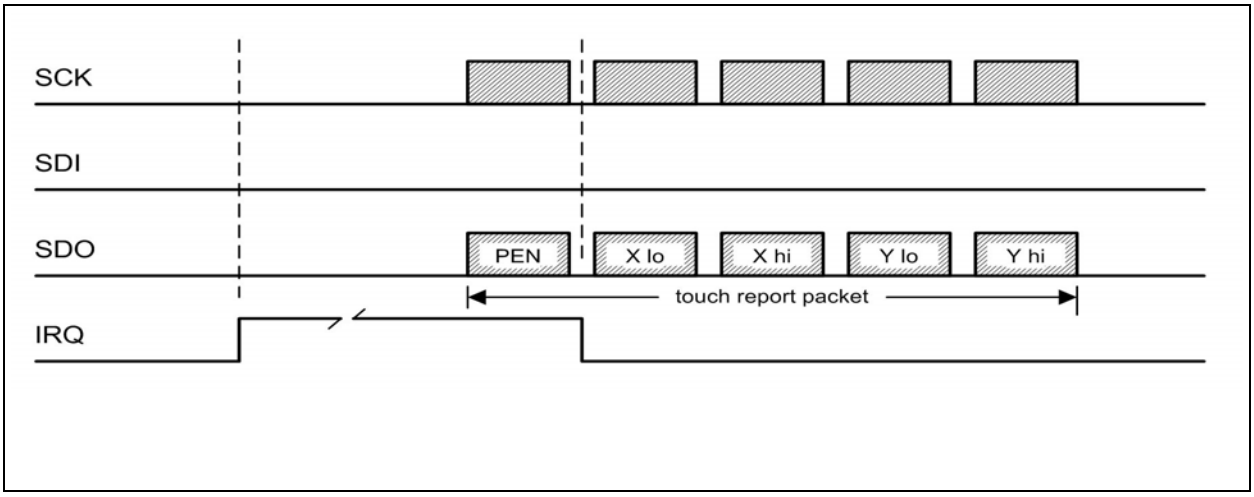
5.4 Touch Report Protocol

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



Downloaded from Arrow.com.

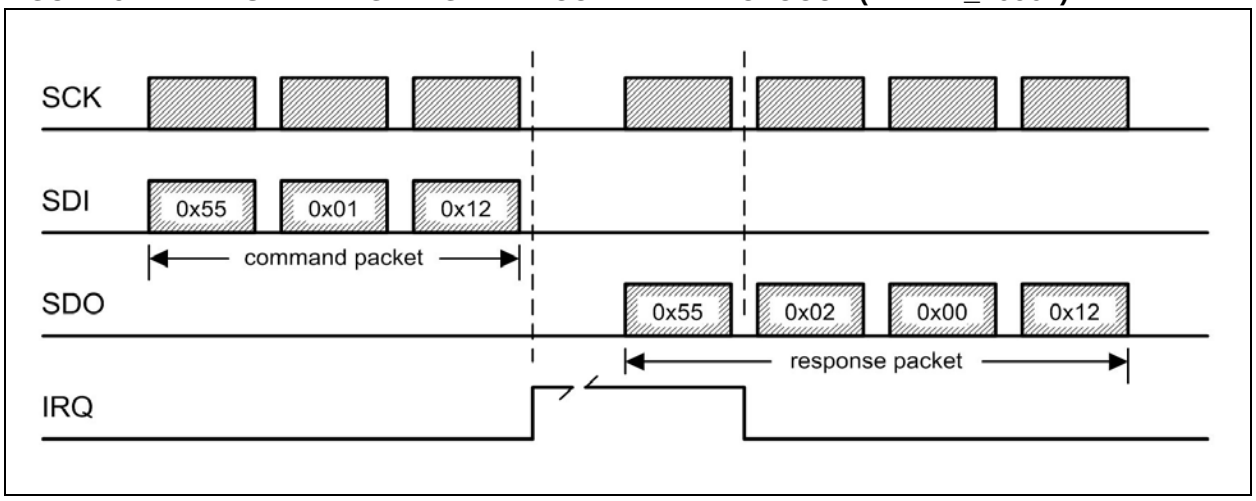
5.5 Command Protocol

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.

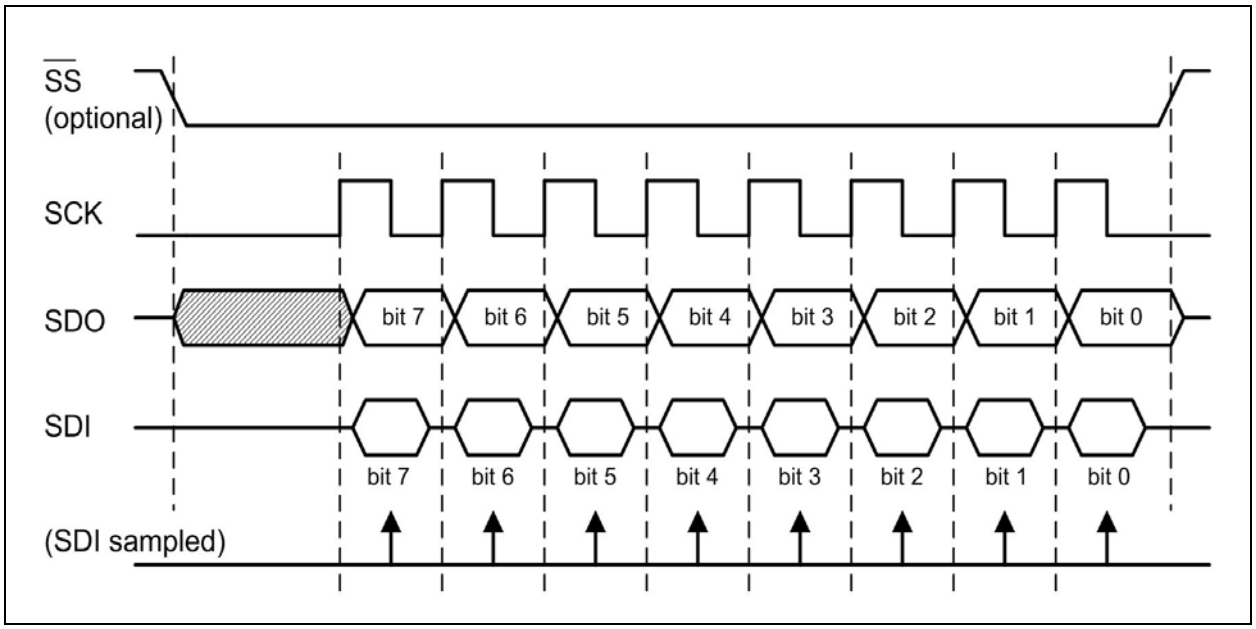
FIGURE 5-2: SPI TIMING DIAGRAM – COMMAND PROTOCOL (ENABLE_TOUCH)



5.6 SPI Bit Timing – General

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

FIGURE 5-3: SPI GENERAL BIT TIMING WAVEFORM



© 2009-2016 Microchip Technology Inc.

Downloaded from Arrow.com.

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

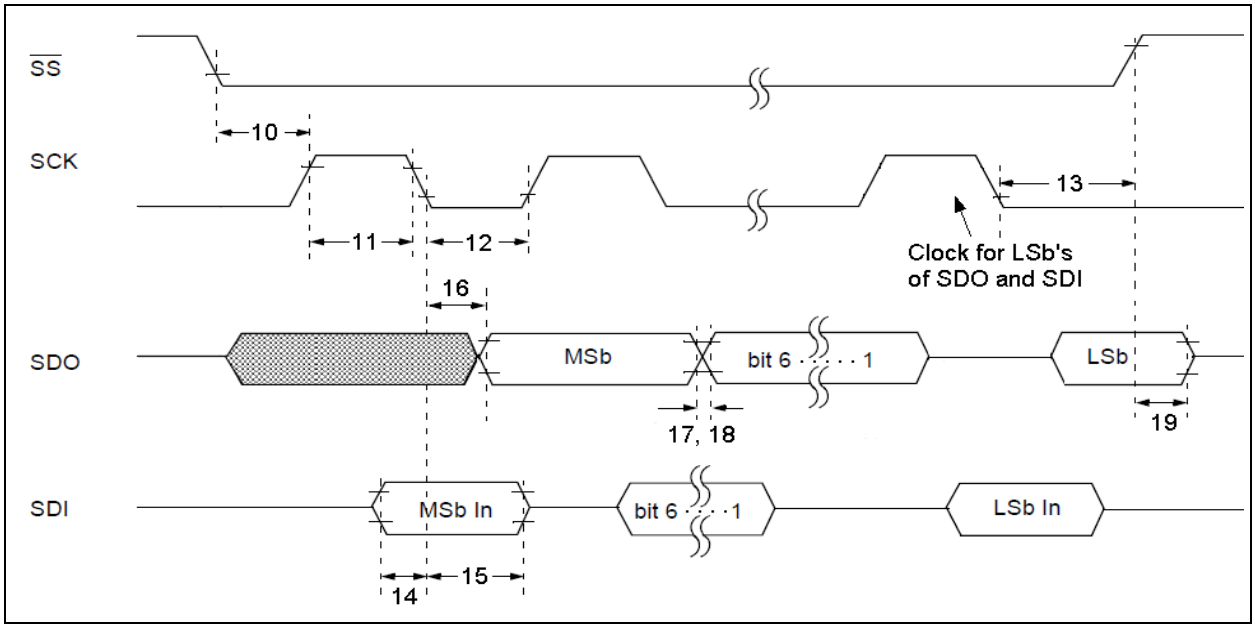


TABLE 5-3: SPI BIT TIMING MIN. AND MAX. VALUES

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.

Downloaded from Arrow.com.

6.0 UART COMMUNICATIONS

TABLE 6-1: UART HARDWARE INTERFACE

AR1011 Pin	Description
M1	Connect M1 to VDD to select UART communications
TX	Transmit to host
RX	Receive from host
SDO	Connect SDO to VSS

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.

© 2009-2016 Microchip Technology Inc.

Downloaded from Arrow.com.

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

Downloaded from Arrow.com.

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.

© 2009-2016 Microchip Technology Inc.

Downloaded from Arrow.com.

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

Downloaded from Arrow.com.

8.2.8 SleepDelay Register (OFFSET 0x0A)

The SleepDelay register sets the time duration with no touch or command activity that will cause the controller to enter a low-power Sleep mode. Valid values are as follows:

0 ≤ SleepDelay ≤ 255

Sleep Delay Time = SleepDelay * 100 ms; when Sleep-Delay > 0

A value of zero disables the Sleep mode, such that the controller will never enter low-power Sleep mode.

A touch event will wake the controller from low-power Sleep mode and start sending touch reports. Communications sent to the controller will wake it from the low-power Sleep mode and initiate action to the command.

8.2.9 PenUpDelay Register (OFFSET 0x0B)

The PenUpDelay register sets the duration of a pen-up event that the controller will allow, without sending a pen-up report for the event. The delay time is started upon detecting a pen-up condition.

If a pen down is reestablished before the delay time expires, then pen-down reports will continue without a pen up being sent. This effectively debounces a touch event in process.

A lower value will make the controller more responsive to pen ups, but will cause more touch drop outs with a lighter touch. A higher value will make the controller less responsive to pen ups, but will reduce the number of touch drop outs with a lighter touch. Valid values are as follows:

0 ≤ PenUpDelay ≤ 255

Pen-up Delay Time ≈ PenUpDelay * 240 µs

8.2.10 TouchMode Register (OFFSET 0x0C)

The TouchMode register configures the action taken for various touch states.

There are three states of touch for the controller's touch reporting action which can be independently controlled.

Touch States:

1. Pen Down (initial touch)

   User defined 0-3 touch reports, with selectable pen states.

2. Pen Movement (touch movement after initial touch)

   User defined no-touch reports or streaming touch reports, with selectable pen states.

3. Pen Up (touch release)

   User defined 0-3 touch reports, with selectable pen states.

Every touch report includes a "P" (Pen) bit that indicates the pen state.

- Pen Down: P = `1`
- Pen Up: P = `0`

© 2009-2016 Microchip Technology Inc.

Downloaded from Arrow.com.

REGISTER 8-1: TouchMode REGISTER FORMAT

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PD2	PD1	PD0	PM1	PM0	PU2	PU1	PU0
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'

bit 7-5 **PD<2:0>:** Pen-Down State bits (action taken upon pen down).
`000` = No touch report
`001` = Touch report with P=`0`
`010` = Touch report with P=`1`
`011` = Touch report with P=`1`, then touch report with P=`0`
`100` = Touch report with P=`0`, then touch report with P=`1`, then touch report with P=`0`
`101` = Touch report with P=`0`, then touch report with P=`1`

bit 4-3 **PM<1:0>:** Pen Movement State bits (action taken upon pen movement).
`00` = No touch report
`01` = Touch report with P=`0`
`10` = Touch report with P=`1`

bit 2-0 **PU<2:0>:** Pen-Up State bits (action taken upon pen up).
`000` = No touch report
`001` = Touch report with P=`0`
`010` = Touch report with P=`1`
`011` = Touch report with P=`1`, then touch report with P=`0`
`100` = Touch report with P=`0`, then touch report with P=`1`, then touch report with P=`0`
`101` = Touch report with P=`0`, then touch report with P=`1`

A couple of typical setup examples for the TouchMode are as follows:

- Report a pen down P=`1` on initial touch, followed by reporting a stream of pen downs P=`1` during the touch, followed by a final pen up P=`0` on touch release. TouchMode = `0b01010001` = 0x51
- Report a pen up P=`0` then a pen down P=`1` on initial touch, followed by reporting a stream of pen downs P=`1` during the touch, followed by a final pen up P=`0` on touch release. TouchMode = `0b10110001` = 0xB1

Downloaded from Arrow.com.

8.2.11 TouchOptions Register (OFFSET 0x0D)

The TouchOptions register contains various “touch” related option bits.

REGISTER 8-2: TouchOptions REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	R/W	R/W
—	—	—	—	—	—	48W	CCE
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’

bit 7-2 **Unimplemented**: Read as ‘0’

bit 1 **48W:** 4-Wire or 8-Wire Sensor Selection bit
1 = Selects 8-wire Sensor Operating mode
0 = Selects 4-wire Sensor Operating mode

bit 0 **CCE:** Calibrated Coordinates Enable bit
1 = Enables calibrated coordinates, if the controller has been calibrated
0 = Disables calibrated coordinates

Note: A 4-wire touch sensor will not work if the 48W Configuration bit is incorrectly defined as 1, which selects 8-wire.

An 8-wire touch sensor will provide basic operation if the 48W Configuration bit is incorrectly defined as 0, which selects 4-wire. However, the benefit of the 8-wire sensor will only be realized if the 48W Configuration bit is correctly defined as 1, selecting 8-wire.

8.2.12 CalibrationInset Register (OFFSET 0x0E)

The CalibrationInset register defines the expected position of the calibration points, inset from the perimeter of the touch sensor’s active area, by a percentage of the full scale dimension.

This allows for the calibration targets to be placed inset from edge to make it easier for a user to touch them.

The CalibrationInset register value is only used when the CALIBRATION_MODE command is issued to the controller. In Calibration mode, the controller will extrapolate the calibration point touch report values by the defined CalibrationInset percentage to achieve full scale.

A software application that issues the CALIBRATION_MODE command must present the displayed calibration targets at the same inset percentage as defined in this CalibrationInset register.

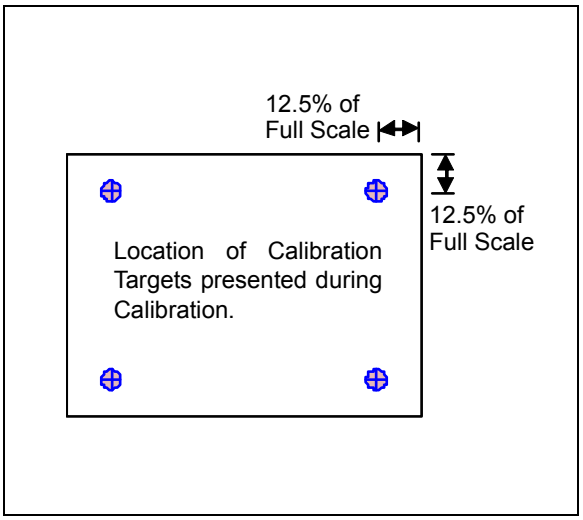
Valid values are as follows:

0 ≤ CalibrationInset ≤ 40

Calibration Inset = (CalibrationInset/2) %, Range of 0-20% with 0.5% resolution

For example, CalibrationInset = 25 (0x19) yields a calibration inset of (25/2) or 12.5%. During the calibration procedure, the controller will internally extrapolate the calibration point touch values in Calibration mode by 12.5% to achieve full scale.

FIGURE 8-1: CALIBRATION TARGET EXAMPLE



© 2009-2016 Microchip Technology Inc.

Downloaded from Arrow.com.

8.2.13 PenStateReportDelay Register (OFFSET 0x0F)

The PenStateReportDelay register sets the delay time between sending of sequential touch reports for the “Pen-Down” and “Pen-Up” Touch mode states. See **Section 8.2.10 “TouchMode Register (offset 0x0C)”** for touch modes.

For example, if “Pen-Up” state of the TouchMode register is configured to send a touch report with P=1, followed by a touch report with P=0, then this delay occurs between the two touch reports. This provides some timing flexibility between the two touch reports that may be desired in certain applications. Valid values are as follows.

0 ≤ PenStateReportDelay ≤ 255

Pen State Report Delay Time = PenStateReportDelay * 50 µs

8.2.14 TouchReportDelay Register (OFFSET 0x11)

The TouchReportDelay register sets a forced delay time between successive touch report packets. This allows slowing down of the touch report rate, if desirable for a given application. For example, a given application may not need a high rate of touch reports and may want to reduce the overhead used to service all of the touch reports being sent. In this situation, increasing the value of this register will reduce the rate at which the controller sends touch reports. Valid values are as follows:

0 ≤ TouchReportDelay ≤ 255

Touch Report Delay Time ≈ TouchReportDelay * 500 µs

Downloaded from Arrow.com.

9.0 COMMANDS

9.1 Sending Commands

9.1.1 COMMAND SEND FORMAT

The controller supports application-specific configuration commands as shown in Table 9-1, below.

TABLE 9-1: COMMAND SEND FORMAT

Byte #	Name	Value	Description
1	Header	0x55	Header (mark beginning of command packet)
2	Size	0x<>	Size, # of bytes following this byte
3	Command	0x<>	Command ID
4	Data	0x<>	Data, if applicable for the command
:	Data	0x<>	Data, if applicable for the command

To ensure command communication is not interrupted by touch activity, it is recommended that the controller touch is disabled, prior to other commands. This can be done as follows:

1. Send `DISABLE_TOUCH` command
2. Wait 50 ms
3. Send desired commands
4. Send `ENABLE_TOUCH` command

9.1.2 COMMAND RESPONSE

A received command will be responded to as seen in Table 9-2 below.

TABLE 9-2: COMMAND RESPONSE FORMAT

Byte #	Name	Value	Description
1	Header	0x55	Header (mark beginning of command packet)
2	Size	0x<>	Size, # of bytes following this byte
3	Status	0x<>	Status
4	Command	0x<>	Command ID
5	Data	0x<>	Data, if applicable for the command
:	Data	0x<>	Data, if applicable for the command

The “Status” value within the response packet should be one of the following (See Table 9-3):

TABLE 9-3: COMMAND RESPONSE STATUS VALUES

Status Value	Description
0x00	Success
0x01	Command Unrecognized
0x03	Header Unrecognized
0x04	Command Time Out (exceeded ~100 ms)
0xFC	Cancel Calibration mode

© 2009-2016 Microchip Technology Inc.

Downloaded from Arrow.com.

9.1.3 DISABLE TOUCH BEFORE SENDING SUBSEQUENT COMMANDS

The AR1000 does not support full duplex communications. It cannot send touch reports to the host simultaneously with receiving commands from the host.

Disable AR1000 touch reporting prior to sending any other command(s), then re-enable touch reporting when complete with executing other commands.

1. Send the `DISABLE_TOUCH` command.

   Check for expected command response.

2. Send a desired command.

   Check for expected command response.

3. Repeat at step 2 if another command is to be sent.
4. Send the `ENABLE_TOUCH` command.

   Check for expected command response.

9.1.4 CONFIRM COMMAND IS SENT

Confirm each command sent to the AR1000, prior to issuing another command, to ensure it is executed. This is accomplished by evaluating the AR1000 response to a command that has been sent to it.

Check for each of the following five conditions to be met (See Table 9-4).

TABLE 9-4: COMMAND RESPONSE ERROR CONDITIONS

Condition	Response Byte	Description
Header	1	Header 0x55 value is expected
Size	2	Size 0x<> value to match what is expected for command sent
Status	3	Status 0x00 “success” value is expected
ID	4	Command ID 0x<> value to match what is expected (ID of sent command)
Data	5 to end	Data byte count to match what is expected for command sent

0x<> represents a value that is dependent on the command.

An error has occurred if no response is received at all or if any of the above conditions are not met in the response from the AR1000. If an error condition occurs, delay for a period of ~50 ms then send the same command again.

Downloaded from Arrow.com.

9.2 AR1000 Commands

TABLE 9-5: COMMAND SET SUMMARY

Command Value	Command Description
0x10	GET_VERSION
0x12	ENABLE_TOUCH
0x13	DISABLE_TOUCH
0x14	CALIBRATE_MODE
0x20	REGISTER_READ
0x21	REGISTER_WRITE
0x22	REGISTER_START_ADDRESS_REQUEST
0x23	REGISTERS_WRITE_TO_EEPROM
0x28	EEPROM_READ
0x29	EEPROM_WRITE
0x2B	EEPROM_WRITE_TO_REGISTERS

9.3 AR1000 Command Descriptions

9.3.1 GET_VERSION – 0x10

Controller will return version number and type.

Send: <0x55><0x01><0x10>

Receive: <0x55><0x05><Response><0x10><Version High><Version Low><Type>

where <Type>

REGISTER 9-1: GET_VERSION <TYPE> FORMAT

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RS1	RS0	TP5	TP4	TP3	TP2	TP1	TP0
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'

bit 7-6 **RS<1:0>:** Resolution of Touch Coordinates bits

00 = 8-bit

01 = 10-bit

10 = 12-bit

bit 5-0 **TP<5:0>:** Type of Controller bits
001010 = ARA10

© 2009-2016 Microchip Technology Inc.

Downloaded from Arrow.com.

9.3.2 ENABLE_TOUCH – 0x12

Controller will send touch coordinate reports for valid touch conditions.

Send: `<0x55><0x01><0x12>`

Receive: `<0x55><0x02><Response><0x12>`

9.3.3 DISABLE_TOUCH – 0x13

Controller will not send any touch coordinate reports. A touch will, however, still wake-up the controller if asleep.

Send: `<0x55><0x01><0x13>`

Receive: `<0x55><0x02><Response><0x13>`

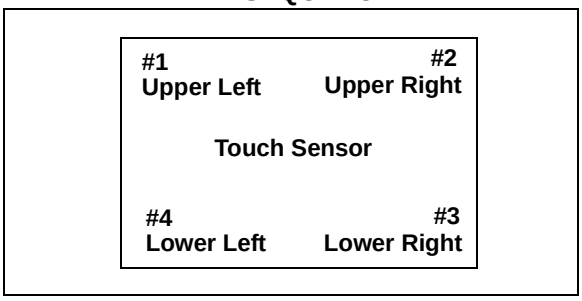
9.3.4 CALIBRATE – 0x14

Enter Calibration mode. This instructs the controller to enter a mode of accepting the next four touches as the calibration point coordinates. See **Section 10.1 "Calibration of Touch Sensor with Controller"** for an example.

Completion of Calibration mode will automatically store the calibration point coordinates in on-board controller memory and set (to 1) the CCE bit of the TouchOptions register. This bit enables the controller to report touch coordinates that have been processed with the previously collected calibration data.

To provide for proper touch orientation, the four sequential calibration touches must be input in the physical order on the touch sensor, as shown in Figure 9-1.

FIGURE 9-1: CALIBRATION ROUTINE SEQUENCE



Upon completion, the controller's register values and calibration data are stored to the EEPROM.

The Calibration mode will be canceled by sending any command before the mode has been completed. If the calibration is canceled, the controller response may appear incorrect or incomplete. This is expected behavior.

Downloaded from Arrow.com.

9.3.4.1 AR1010/AR1020 Calibrate Response

Send:	<0x55><0x02><0x14><Calibration Type>	
	Calibration Type	Description
	0x04	4 point
Receive:	<0x55><0x02><0x00><0x14>	for initial command response
	<0x55><0x02><0x00><0x14>	Response for touch of Calibration point #1
	<0x55><0x02><0x00><0x14>	Response for touch of Calibration point #2
	<0x55><0x02><0x00><0x14>	Response for touch of Calibration point #3
	<0x55><0x02><0x00><0x14>	Response for touch of Calibration point #4

A successful CALIBRATE command results in five response packets being sent to the host.

Once the response has been received for the completed 4th target, a delay of one second must be implemented prior to sending any commands to the controller. This one second delay insures all data has been completely written to the EEPROM.

9.3.4.2 AR1011/AR1021 Calibrate Response

Send:	<0x55><0x02><0x14><Calibration Type>	
	Calibration Type	Description
	0x04	4 point
Receive:	<0x55><0x02><0x00><0x14>	for initial command response
	<0x55><0x02><0x00><0x14>	Response for touch of Calibration point #1
	<0x55><0x02><0x00><0x14>	Response for touch of Calibration point #2
	<0x55><0x02><0x00><0x14>	Response for touch of Calibration point #3
	<0x55><0x02><0x00><0x14>	Response for touch of Calibration point #4
	<0x55><0x02><0x00><0x14>	Response after EEPROM has been written

A successful CALIBRATE command results in six response packets being sent to the host.

© 2009-2016 Microchip Technology Inc.

Downloaded from Arrow.com.

9.3.4.3 Calibration Data Encoded and Stored in EEPROM

System integrators may prefer to preload a calibration into their design. This allows the user to properly navigate to the calibration routine icon or shortcut without the use of a mouse. This also addresses the need to calibrate each system individually before deploying it to the field.

The raw touch coordinates, decoded by the controller, for each of the four calibration touches are extrapolated if CalibrationInset was non-zero. The four coordinate pairs are then re-oriented, if required, such that the upper left corner is the minimum (X,Y) “origin” value pair and the lower right corner is the maximum (X,Y) value pair.

Coordinates are 10-bit significant values, scaled to 16-bit and stored in a High (Hi) and Low (Lo) byte pair.

Separator	Upper Left (Node 1)				Upper Right (Node 2)				Lower Right (Node 3)				Lower Left (Node 4)				Flip State
	X		Y		X		Y		X		Y		X		Y		
	Lo	Hi	Lo	Hi	Lo	Hi	Lo	Hi	Lo	Hi	Lo	Hi	Lo	Hi	Lo	Hi	

Decode the above data to as follows:

1. Swap the order of stored low and high bytes for a given coordinate.
2. Convert the 16-bit value (stored high and low bytes) from hexadecimal to decimal.
3. Divide the result by 64 to properly rescale the 16-bit stored value back to a 10-bit significant coordinate.

Example of Low = 0x40 and High = 0xF3:

Swap: 0xF340

Hex to Decimal: 62272

Divide by 64: 973

REGISTER 9-2: FLIP STATE BYTE

U-0	U-0	U-0	U-0	U-0	R/W	R/W	R/W
—	—	—	—	—	XYFLIP	XFLIP	YFLIP
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’

bit 7-3 **Unimplemented**: Read as ‘0’

bit 2 **XYFLIP:** X and Y Axis Flip bit
1 = X and Y axis are flipped
0 = X an Y axis are not flipped

bit 1 **XFLIP:** X-Axis Flip bit
1 = X-axis flipped
0 = X-axis not flipped

bit 0 **YFLIP:**Y-Axis Flip bit
1 = Y-axis flipped
0 = Y-axis not flipped

For storing desired calibration values to the EEPROM:

- AR1010/AR1020 (See **Section 9.3.12 “EEPROM Map”**).
- AR1011/AR1021 (See **Section 9.3.12 “EEPROM Map”** and **Section 10.2 “AR1011/AR1021 Storing Default Calibration Values to EEPROM”**).

Downloaded from Arrow.com.

9.3.5 REGISTER_READ – 0x20

Reads a value from a controller register location. This can be used to determine a controller configuration setting.

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

1. Issue the REGISTER_START_ADDRESS_REQUEST command to obtain the Start address for the register group.
2. Calculate the desired register's absolute address by adding the register's Offset value to Start address for the register group.
3. Issue this REGISTER_READ command, as follows, using the calculated register's absolute address:

Send: <0x55><0x04><0x20><Register Address High byte><Register Address Low byte><# of Registers to Read>

Register Address High byte: 0x00

# of Registers to Read: 0x01 thru 0x08

Receive: <0x55><0x02 + # of Registers Read><Response><0x20><Register value>…<Register value>

The AR1000 controller will ignore the value entered for the Register Address High Byte. However, 0x00 is recommended to safeguard against any possible future product development.

9.3.6 REGISTER_WRITE – 0x21

Write a value to a controller register location. This can be used to change a controller configuration setting.

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

1. Issue the REGISTER_START_ADDRESS_REQUEST command to obtain the *Start* address for the register group.
2. Calculate the desired register's absolute address by adding the register's Offset value to Start address for the register group.
3. Issue this REGISTER_WRITE command, as follows, using the calculated register's absolute address:

Send: <0x55><0x04 + # Registers to Write><0x21><Register Address High byte><Register Address Low byte>

<# of Registers to Write><Data>…<Data>

Register Address High byte: 0x00

# of Registers to Read: 0x01 thru 0x08

Receive: <0x55><0x02><Response><0x21>

The AR1000 controller will ignore the value entered for the Register Address High Byte. However, 0x00 is recommended to safeguard against any possible future product development.

9.3.7 REGISTER_START_ADDRESS_REQUEST – 0x22

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

Send: <0x55><0x01><0x22>

Receive:

<0x55><0x03><Response><0x22><Register Start Address>

9.3.8 REGISTERS_WRITE_TO_EEPROM – 0x23

Save Configuration register values to EEPROM. This allows the controller to remember configurations settings through controller power cycles.

Send: <0x55><0x01><0x23>

Receive: <0x55><0x02><Response><0x23>

9.3.9 EEPROM_READ – 0X28

The controller has 256 bytes of on-board EEPROM.

- The first 128 bytes (address range 0x00-0x7F) are reserved by the controller for the Configuration register settings and calibration data.
- The second 128 bytes (address range 0x80-0xFF) are provided for the user's application, if desired.

This command provides a means to read values from the EEPROM.

Send: <0x55><0x04><0x28><EEPROM Address High byte><EEPROM Address Low byte><# of EEPROM to Read>

Register Address High byte: 0x00

# of Registers to Read: 0x01 thru 0x08

Receive: <0x55><0x02 + # EEPROM Read><Response><0x28><EEPROM value>…<EEPROM value>

The AR1000 controller will ignore the value entered for the EEPROM Address High Byte. However, 0x00 is recommended to safeguard against any possible future product development.

© 2009-2016 Microchip Technology Inc.

Downloaded from Arrow.com.

9.3.10 EEPROM_WRITE – 0x29

The controller has 256 bytes of on-board EEPROM.

This command provides a means to write values to the user space within the EEPROM.

- The first 128 bytes (address range 0x00-0x7F) are reserved by the controller for the Configuration register settings and calibration data. Only the Register Write to EEPROM command should be used to write Configuration registers to EEPROM. Failure to use the Register Write command to save Configuration registers to EEPROM may result in failures or reverting to previously stored Configuration register values.
- The second 128 bytes (address range 0x80-0xFF) are provided for the user's application, if desired.

Warning: ONLY write to user EEPROM addresses of 0x80-0xFF.

One of the following actions is required for EEPROM changes to be used by the controller:

- The controller power must be cycled from OFF to ON or
- Issue the EEPROM_WRITE_TO_REGISTERS command.

Write to EEPROM as follows:

Send: <0x55><0x04 + # EEPROM to Write><0x29><EEPROM Address High byte><EEPROM Address Low byte>

<# of EEPROM to Write><Data>…<Data>

Register Address High byte: 0x00

# of Registers to Read: 0x01 thru 0x08

Receive: <0x55><0x02><Response><0x29>

The AR1000 controller will ignore the value entered for the EEPROM Address High Byte. However, 0x00 is recommended to safeguard against any possible future product development.

9.3.11 EEPROM_WRITE_TO_REGISTERS – 0x2B

Write applicable EEPROM data to Configuration registers. This will cause the controller to immediately begin using changes made to EEPROM stored Configuration register values. A power cycle of the controller will automatically cause the controller to use changes made to the EEPROM stored Configuration register values, without the need for issuing this command. This command eliminates the need for the power cycle.

Send: <0x55><0x01><0x2B>

Receive: <0x55><0x02><Response><0x2B>

9.3.12 EEPROM MAP

The first 128 bytes in address range 0x00:0x7F are reserved by the controller for the Configuration register settings and calibration data. The mapping of data in this reserved controller space of the EEPROM may change over different revisions within the product lifetime.

The EEPROM_WRITE command must not be used to write directly to the lower 128 bytes of the controller EEPROM space of 0x00:0x7F.

The second 128 bytes in address range 0x80:0xFF are provided for the user's application, if desired.

TABLE 9-6: AR1010/AR1020 EEPROM AND REGISTER MAP

EEPROM Address	Function
0x00	<Special Use>
0x01	<Special Use>
0x02	<Special Use>
0x03	Touch Threshold
0x04	Sensitivity Filter
0x05	Sampling Fast
0x06	Sampling Slow
0x07	Accuracy Filter Fast
0x08	Accuracy Filter Slow
0x09	Speed Threshold
0x0A	<Special Use>
0x0B	Sleep Delay
0x0C	Pen-Up Delay
0x0D	Touch Mode
0x0E	Touch Options
0x0F	Calibration Inset
0x10	Pen State Report Delay
0x11	<Reserved>
0x12	Touch Report Delay
0x13	<Special Use>
0x14	Data Block Separator
0x15	Calibration UL X-low
0x16	Calibration UL X-high
0x17	Calibration UL Y-low
0x18	Calibration UL Y-high
0x19	Calibration UR X-low
0x1A	Calibration UR X-high
0x1B	Calibration UR Y-low
0x1C	Calibration UR Y-high
0x1D	Calibration LR X-low
0x1E	Calibration LR X-high
0x1F	Calibration LR Y-low

Downloaded from Arrow.com.

TABLE 9-6: AR1010/AR1020 EEPROM AND REGISTER MAP

EEPROM Address	Function
0x20	Calibration LR Y-high
0x21	Calibration LL X-low
0x22	Calibration LL X-high
0x23	Calibration LL Y-low
0x24	Calibration LL Y-high
0x25	Calibration Flip State
0x26:0x7E	<Special Use>
0x7F	End of Controller Space
0x80:0xFF	User Space

TABLE 9-7: AR1011/AR1021 EEPROM AND REGISTER MAP

EEPROM Address	Function
0x00	Not used
0x01	Configuration Registers – Block Key
0x02	<Special Use>
0x03	<Special Use>
0x04	Touch Threshold
0x05	Sensitivity Filter
0x06	Sampling Fast
0x07	Sampling Slow
0x08	Accuracy Filter Fast
0x09	Accuracy Filter Slow
0x0A	Speed Threshold
0x0B	<Special Use>
0x0C	Sleep Delay
0x0D	Pen-Up Delay
0x0E	Touch Mode
0x0F	Touch Options
0x10	Calibration Inset
0x11	Pen State Report Delay
0x12	<Special Use>
0x13	Touch Report Delay
0x14	<Special Use>
0x15	Configuration Registers – Checksum
0x16	Calibration - Block Key
0x17	Calibration UL X-low
0x18	Calibration UL X-high
0x19	Calibration UL Y-low
0x1A	Calibration UL Y-high
0x1B	Calibration UR X-low
0x1C	Calibration UR X-high

TABLE 9-7: AR1011/AR1021 EEPROM AND REGISTER MAP

EEPROM Address	Function
0x1D	Calibration UR Y-low
0x1E	Calibration UR Y-high
0x1F	Calibration LR X-low
0x20	Calibration LR X-high
0x21	Calibration LR Y-low
0x22	Calibration LR Y-high
0x23	Calibration LL X-low
0x24	Calibration LL X-high
0x25	Calibration LL Y-low
0x26	Calibration LL Y-high
0x27	Calibration Flip State
0x28	Calibration – Checksum
0x29:0x50	<Special Use>
0x51:0x7F	<Reserved>
0x80:0xFF	User Space

© 2009-2016 Microchip Technology Inc.

Downloaded from Arrow.com.

10.0 APPLICATION NOTES

10.1 Calibration of Touch Sensor with Controller

The reported coordinates from a touch screen controller are typically calibrated to the application's video display. The task is often left up to the host to perform. This controller provides a feature for it to send coordinates that have already been calibrated, rather than the host needing to perform this task. If enabled, the feature will apply pre-collected 4-point calibration data to the reported touch coordinates. Calibration only accounts for X and Y directional scaling. It does not correct for angular errors due to rotation of the touch sensor on the video display.

The calibration process can be canceled at anytime by sending a command to the controller.

Upon completion of the calibration process, the calibration data is automatically stored to the EEPROM and "Calibrated Coordinates" is enabled.

The process of "calibration" with the controller is described below.

1. Disable touch reporting by issuing *<Disable Touch>* command.

Send: `<0x55><0x01><0x13>`

Receive: `<0x55><0x02><Response><0x13>`

2. Get register group Start address by issuing `REGISTER_START_ADDRESS_REQUEST` command.

A register Start address of 0x20 is used below, for this example.

Send: `<0x55><0x01><0x22>`

Receive: `<0x55><0x03><0x00><0x22><0x20>`

3. Calculate the CalibrationInset register's address by adding its offset value of 0x0E to the register group Start address of 0x20.

Register Address = Register Start Address + CalibratioInset Register Offset = 0x20 + 0x0E = 0x2E

4. Calculate the desired value for the CalibrationInset register.

A Calibration Inset of 12.5% is used below for this example.

CalibrationInset = 2 * Desire Calibration Inset % = 2 * 12.5 = 25 = 0x19

5. Set the Calibration Inset by writing the desired value to the CalibrationInset register.

Send: `<0x55><0x05><0x21><0x00><0x2E><0x01><0x19>`

Receive: `<0x55><0x02><0x00><0x21>`

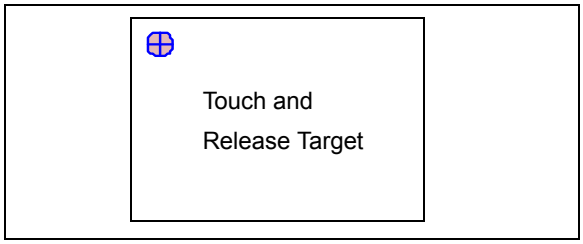
6. Issue the `CALIBRATE_MODE` command.

Send: `<0x55><0x02><0x14><0x04>`

Receive: `<0x55><0x02><0x00><0x14>`

7. Software must display the first calibration point target in the upper left quadrant of the display and prompt the user to touch and release the target.

FIGURE 10-1: SUGGESTED TEXT FOR FIRST CALIBRATION TARGET



8. Wait for the user to touch and release the first calibration point target. Do this by looking for a controller response of:
`<0x55><0x02><0x00> <0x14>`

9. Software must display the second calibration point target in the upper right quadrant of the display and prompt the user to touch and release the target.

FIGURE 10-2: SUGGESTED TEXT FOR SECOND CALIBRATION TARGET

10. Wait for the user to touch and release the second calibration point target. Do this by looking for a controller response of: `<0x55><0x02><0x00><0x14>`

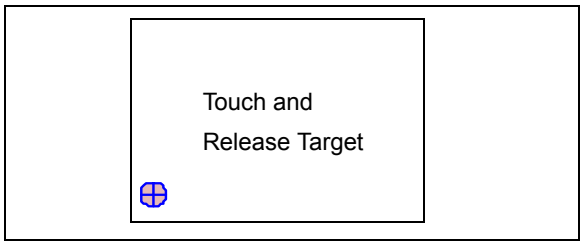
Downloaded from Arrow.com.

11. Software must display the third calibration point target in the lower right quadrant of the display and prompt the user to touch and release the target.

FIGURE 10-3: SUGGESTED TEXT FOR THIRD CALIBRATION TARGET

12. Wait for the user to touch and release the third calibration point target. Do this by looking for a controller response of:
`<0x55><0x02><0x00><0x14>`
13. Software must display the fourth calibration point target in the lower left quadrant of the display and prompt the user to touch and release the target.

FIGURE 10-4: SUGGESTED TEXT FOR FOURTH CALIBRATION TARGET



14. Wait for the user to touch and release the fourth calibration point target. Do this by looking for a controller response of:
`<0x55><0x02><0x00><0x14>`
15. Wait for the controller to correctly write calibration data into EEPROM
    - AR1010/AR1020: Wait one second for data to be stored into EEPROM
    - AR1011/AR1021: Wait for a controller response of `<0x55><0x02><0x00><0x14>`
16. Enable touch reporting by issuing `ENABLE_TOUCH` command.

Send: `<0x55><0x01><0x12>`

Receive: `<0x55><0x02><Response><0x12>`

© 2009-2016 Microchip Technology Inc.

Downloaded from Arrow.com.

10.2 AR1011/AR1021 Storing Default Calibration Values to EEPROM

If you wish to implement fixed calibration values, preloaded into the AR1000 EEPROM, then the following procedure must be followed (See **Section 10.2.1 "Preparation for Fixed Calibration Values"**).

10.2.1 PREPARATION FOR FIXED CALIBRATION VALUES

Determine if fixed calibration values are suitable for your application and determine your desired values.

Calculate a checksum for your custom data set. See **Section 9.3.4.3 "Calibration Data Encoded and Stored in EEPROM"** for additional details regarding calibration data format.

An example of calculating the checksum is shown below (See Table 10-1).

TABLE 10-1: CHECKSUM CALCULATION EXAMPLE

Description			Value	Operation	Checksum Result
Seed			0x45	n/a	0x45
Block Key			0x55	0x45 + 0x55 =	0x9A
Upper Left	X	Low byte	0x06	0x9A + 0x06 =	0xA0
Upper Left	X	High byte	0x1B	0xA0 + 0x1B =	0xBB
Upper Left	Y	Low byte	0xA5	0xBB + 0xA5 =	0x60
Upper Left	Y	High byte	0x08	0x60 + 0x08 =	0x68
Upper Right	X	Low byte	0x13	0x68 + 0x13 =	0x7B
Upper Right	X	High byte	0xDF	0x7B + 0xDF =	0x5A
Upper Right	Y	Low byte	0xF4	0x5A + 0xF4 =	0x4E
Upper Right	Y	High byte	0x0B	0x4E + 0x0B =	0x59
Lower Right	X	Low byte	0x98	0x59 + 0x98 =	0xF1
Lower Right	X	High byte	0xE4	0xF1 + 0xE4 =	0xD5
Lower Right	Y	Low byte	0x1E	0xD5 + 0x1E =	0xF3
Lower Right	Y	High byte	0xEC	0xF3 + 0xEC =	0xDF
Lower Left	X	Low byte	0xBF	0xDF + 0xBF =	0x9E
Lower Left	X	High byte	0x1A	0x9E + 0x1A =	0xB8
Lower Left	Y	Low byte	0x32	0xB8 + 0x32 =	0xEA
Lower Left	Y	High byte	0xE7	0xEA + 0xE7 =	0xD1
Flip State			0x01	0xD1 + 0x01 =	0xD2
Checksum					**0xD2**

The Checksum is an 8-bit value calculated by successive additions with overflow ignored, as shown below.

Checksum = 0x45

For each of the 18 calibration values, starting at the Block Key and ending with the Flip State

Checksum += Calibration value

Next Calibration value

Downloaded from Arrow.com.

10.2.2 EXECUTION OF FIXED CALIBRATION VALUE LOADING

Follow error checking practices by checking the AR1000 responses to issued commands.

1. Send the AR1000 `DISABLE_TOUCH` command.
2. Use the AR1000 `EEPROM_WRITE` command multiple times to write the following to the AR1000 EEPROM.
   a. Block Key 0x55 to address 0x16
   b. Data set to addresses 0x17:0x27. See **Section 9.3.4.3 "Calibration Data Encoded and Stored in EEPROM"** and **Section 9.3.12 "EEPROM Map"**.
   c. Checksum for the data block to address 0x28
   d. Mirror image of a, b and c from above to address 0x3E:0x50
3. Set the CCE bit of the TouchOptions register. This will enable the controller to use the calibration data on the next power boot. See **Section 10.2.3 "Configuring the CCE bit to Use Fixed Calibration Values"** for additional details on the CCE bit.
4. Send the AR1000 `ENABLE_TOUCH` (0x12) command.

10.2.3 CONFIGURING THE CCE BIT TO USE FIXED CALIBRATION VALUES

The CCE bit of the TouchOptions Register (offset 0x0D) must be set to '1' to enable the usage of the stored calibration values in EEPROM.

This should be completed before re-enabling the controller via the `ENABLE_TOUCH` command.

REGISTER 10-1: CCE BIT FORMAT

U-0	U-0	U-0	U-0	U-0	U-0	R/W	R/W
—	—	—	—	—	—	48W	CCE
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'

bit 7-2 **Unimplemented**: Read as '0'

bit 1 **48W:** 4-Wire or 8-Wire Sensor Selection bit
1 = Selects 8-wire Sensor Operating mode
0 = Selects 4-wire Sensor Operating mode

bit 0 **CCE:** Calibrated Coordinates Enable bit
1 = Enables calibrated coordinates, if the controller has been calibrated
0 = Disables calibrated coordinates

© 2009-2016 Microchip Technology Inc.

Downloaded from Arrow.com.

1. Send the DISABLE_TOUCH (0x13) command.
2. Send the REGISTER_START_ADDRESS_REQUEST (0x22) to determine the absolute address for TouchOptions Register.
3. Send the REGISTER_WRITE (0x21) command to set the CCE bit of the TouchOptions Register.
4. Send REGISTERS_WRITE_TO_EEPROM (0x23) command to have all current registers stored into EEPROM.
5. Send the AR1000 ENABLE_TOUCH (0x12) command.

The controller will use the stored calibration data after cycling power to the controller.

10.2.4 EEPROM_WRITE COMMAND TO STORE DEFAULT CALIBRATION

The EEPROM_WRITE command is shown in this section. See **Section 9.0 "Commands"** for more command details.

<> = application specific value

Send to AR1000:

0x55	Header
0x<>	Number of bytes to follow this one
0x29	Command ID
0x00	Desired EEPROM address to write high byte. Always 0x00
0x<>	Desired EEPROM address to write low byte
0x<>	Number of consecutive EEPROM addresses to write (supports 0x01 to 0x08)
0x<>	Value # 1 to write
0x<>	Value # 2 to write, if applicable
0x<>	Value # 3 to write, if applicable
0x<>	Value # 4 to write, if applicable
0x<>	Value # 5 to write, if applicable
0x<>	Value # 6 to write, if applicable
0x<>	Value # 7 to write, if applicable
0x<>	Value # 8 to write, if applicable

Response from AR1000:

0x55	Header
0x02	Number of bytes to follow this one
0x00	Success response
0x29	Command ID

10.2.5 QUALITY TEST

Although not required, a level of quality assurance can be added to the process by the application issuing multiple EEPROM_READ commands to the AR1000.

The response data from the EEPROM_READ commands would be tested by the application against the application's desired data as a quality check.

10.2.6 EXAMPLE COMMAND SEQUENCE

An example eight command sequence for the entire process is shown below.

All values shown are in hexadecimal.

Calibration values are applications specific and have been symbolically represented as follows:

ULxL = **U**pper **L**eft corner **x**-coordinate **L**ow byte

:

LLyH = **L**ower **L**eft corner **y**-coordinate **H**igh byte

DISABLE_TOUCH

Downloaded from Arrow.com.

Disable Touch

Command:	55	01	13	
Response:	55	02	00	13

Write Calibration to EEPROM Image # 1

Command:	55	0C	29	00	16	08	55	ULxL	ULxH	ULyL	ULyH	URxL	URxH	URyL
Response:	55	02	00	29										
Command:	55	0C	29	00	1E	08	URyH	LRxL	LRxH	LRyL	LRyH	LLxL	LLxH	LLyL
Response:	55	02	00	29										
Command:	55	07	29	00	26	03	LLyH	FlipS	Chksm					
Response:	55	02	00	29										

Write Calibration to EEPROM Image # 2

Command:	55	0C	29	00	3E	08	55	ULxL	ULxH	ULyL	ULyH	URxL	URxH	URyL
Response:	55	02	00	29										
Command:	55	0C	29	00	46	08	URyH	LRxL	LRxH	LRyL	LRyH	LLxL	LLxH	LLyL
Response:	55	02	00	29										
Command:	55	07	29	00	4E	03	LLyH	FlipS	Chksm					
Response:	55	02	00	29										

Enable Use of Calibrated Data

Command:	55	01	22				
Response:	55	03	00	22	<Start Address>		
Command:							
4/8-Wire	55	05	21	00	<Start Address + 0x0D>	01	01
5-Wire	55	05	21	00	<Start Address + 0x0D>	01	03
Response:	55	02	00	21			

Enable Touch

Command:	55	01	12	
Response:	55	02	00	12

© 2009-2016 Microchip Technology Inc.

Downloaded from Arrow.com.

11.0 ELECTRICAL SPECIFICATIONS

Absolute Maximum Ratings(†)

Ambient temperature under bias ..... -40°C to +125°C
Storage temperature ..... -65°C to +150°C
Voltage on V_{DD} with respect to V_{SS} ..... -0.3V to +6.5V
Voltage on all other pins with respect to V_{SS} ..... -0.3V to (V_{DD} + 0.3V)
Total power dissipation ..... 800 mW
Maximum current out of V_{SS} pin ..... 300 mA
Maximum current into V_{DD} pin ..... 250 mA
Input clamp current (V_I < 0 or V_I > V_{DD}) ..... ± 20 mA
Maximum output current sunk by any I/O pin ..... 25 mA
Maximum output current sourced by any I/O pin ..... 25 mA

† NOTICE: Stresses above those listed under “Absolute Maximum Ratings” may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those or any other conditions above those indicated in the operation listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

† NOTICE: This device is sensitive to ESD damage and must be handled appropriately. Failure to properly handle and protect the device in an application may cause partial to complete failure of the device.

Downloaded from Arrow.com.

11.1 Minimum Operating Voltage

The AR1000 series controller will operate down to 2.5V ± 5%. Touch performance will be optimized by using the highest allowable voltage for the design.

The PICkit™ Serial included in the AR1000 Development kit supports 3V-5V range of operation.

11.2 AR1000 Electrical Characteristics

Operating Voltage: 2.5 ≤ V_{DD} ≤ 5.25V			
Function	**Pin**	**Input**	**Output**
M1	M1	V_{SS} ≤ **V_{IL}** ≤ 0.15*V_{DD} (0.25*V_{DD} + 0.9V) ≤ **V_{IH}** ≤ V_{DD}	—
M2	M2	V_{SS} ≤ **V_{IL}** ≤ 0.15*V_{DD} (0.25*V_{DD} + 0.9V) ≤ **V_{IH}** ≤ V_{DD}	—
SCL/SCK	SCL/SCK/TX	V_{SS} ≤ **V_{IL}** ≤ 0.2*V_{DD} 0.8*V_{DD} ≤ **V_{IH}** ≤ V_{DD}	—
TX	SCL/SCK/TX	—	V_{SS} ≤ **V_{OL}**(1) ≤ (1.2V – 0.15*V_{DD})(2) (1.25*V_{DD} – 2.25V)(3) ≤ **V_{OH}**(1) ≤ V_{DD}
SDI	SDI/SDA/RX	V_{SS} ≤ **V_{IL}** ≤ 0.2*V_{DD} 0.8*V_{DD} ≤ **VIH** ≤ V_{DD}	—
SDO	SDO	—	V_{SS} ≤ **V_{OL}**(1) ≤ (1.2V – 0.15*V_{DD})(2) (1.25*V_{DD} – 2.25V)(3) ≤ **V_{OH}**(1) ≤ V_{DD}
SIQ	SIQ	—	V_{SS} ≤ **V_{OL}**(1) ≤ (1.2V – 0.15*V_{DD})(2) (1.25*V_{DD} – 2.25V)(3) ≤ **V_{OH}**(1) ≤ V_{DD}
SDA	SDI/SDA/RX	V_{SS} ≤ **V_{IL}** ≤ 0.2*V_{DD} 0.8*V_{DD} ≤ **VIH** ≤ V_{DD}	Open-drain
RX	SDI/SDA/RX	V_{SS} ≤ **V_{IL}** ≤ 0.2*V_{DD} 0.8*V_{DD} ≤ **VIH** ≤ V_{DD}	—
SS	SS	V_{SS} ≤ **V_{IL}** ≤ 0.2*V_{DD} 0.8*V_{DD} ≤ **V_{IH}** ≤ V_{DD}	—

Note 1: These parameters are characterized but not tested.

2: At 10 mA.

3: At -4 mA.

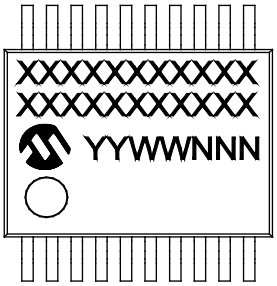
© 2009-2016 Microchip Technology Inc.

Downloaded from Arrow.com.

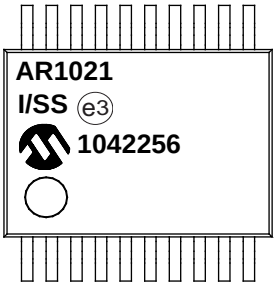
12.0 PACKAGING INFORMATION

12.1 Package Marking Information

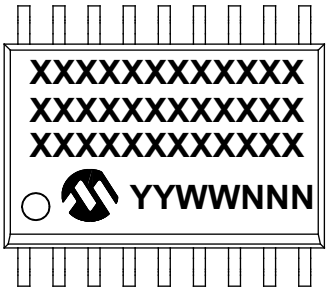
20-Lead SSOP (5.30 mm)



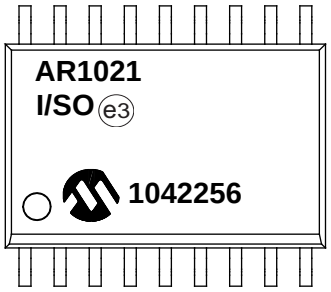
Example



20-Lead SOIC (7.50 mm)



Example



Legend:	XX...X	Customer-specific information
	Y	Year code (last digit of calendar year)
	YY	Year code (last 2 digits of calendar year)
	WW	Week code (week of January 1 is week ‘01’)
	NNN	Alphanumeric traceability code
	(e3)	Pb-free JEDEC designator for Matte Tin (Sn)
	*	This package is Pb-free. The Pb-free JEDEC® designator (e3)) can be found on the outer packaging for this package.

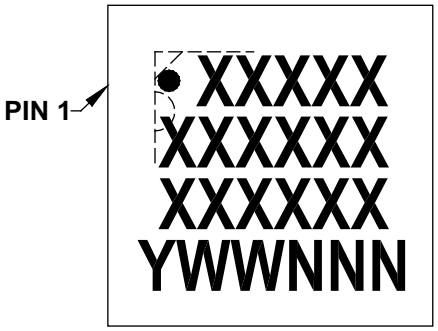
Note: In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line, thus limiting the number of available characters for customer-specific information.

***** Standard PICmicro® device marking consists of Microchip part number, year code, week code and traceability code. For PICmicro device marking beyond this, certain price adders apply. Please check with your Microchip Sales Office. For QTP devices, any special marking adders are included in QTP price.

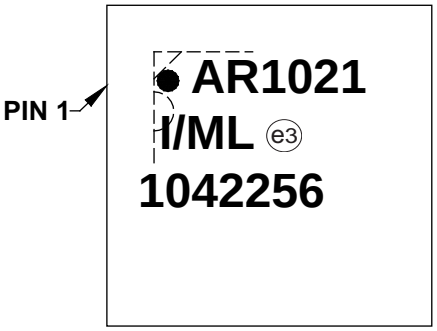
Downloaded from Arrow.com.

12.2 Package Marking Information (Continued)

20-Lead QFN (4x4x0.9 mm)



Example



Legend:	XX...X	Customer-specific information
	Y	Year code (last digit of calendar year)
	YY	Year code (last 2 digits of calendar year)
	WW	Week code (week of January 1 is week '01')
	NNN	Alphanumeric traceability code
	(e3)	Pb-free JEDEC designator for Matte Tin (Sn)
	*	This package is Pb-free. The Pb-free JEDEC® designator (e3)) can be found on the outer packaging for this package.

Note: In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line, thus limiting the number of available characters for customer-specific information.

***** Standard PICmicro® device marking consists of Microchip part number, year code, week code and traceability code. For PICmicro device marking beyond this, certain price adders apply. Please check with your Microchip Sales Office. For QTP devices, any special marking adders are included in QTP price.

© 2009-2016 Microchip Technology Inc.

Downloaded from Arrow.com.

12.3 Ordering

Note: The AR1011/AR1021 are recommended for new designs. The AR1010/AR1020 are still supported and available, but are not recommended for new designs.

TABLE 12-1: ORDERING PART NUMBERS

Part Number	Communication Type	Temp. Range	Pin Package	Packing
AR1011-I/ML	UART	-40°C to + 85°C	QFN, 20 pin	Tube
AR1011-I/SO	UART	-40°C to + 85°C	SOIC, 20 pin	Tube
AR1011-I/SS	UART	-40°C to + 85°C	SSOP, 20 pin	Tube
AR1011T-I/ML	UART	-40°C to + 85°C	QFN, 20 pin	T/R
AR1011T-I/SO	UART	-40°C to + 85°C	SOIC, 20 pin	T/R
AR1011T-I/SS	UART	-40°C to + 85°C	SSOP, 20 pin	T/R
AR1021-I/ML	I^2C/SPI	-40°C to + 85°C	QFN, 20 pin	Tube
AR1021-I/SO	I^2C/SPI	-40°C to + 85°C	SOIC, 20 pin	Tube
AR1021-I/SS	I^2C/SPI	-40°C to + 85°C	SSOP, 20 pin	Tube
AR1021T-I/ML	I^2C/SPI	-40°C to + 85°C	QFN, 20 pin	T/R
AR1021T-I/SO	I^2C/SPI	-40°C to + 85°C	SOIC, 20 pin	T/R
AR1021T-I/SS	I^2C/SPI	-40°C to + 85°C	SSOP, 20 pin	T/R
AR1010-I/ML	UART	-40°C to + 85°C	QFN, 20 pin	Tube
AR1010-I/SO	UART	-40°C to + 85°C	SOIC, 20 pin	Tube
AR1010-I/SS	UART	-40°C to + 85°C	SSOP, 20 pin	Tube
AR1010T-I/ML	UART	-40°C to + 85°C	QFN, 20 pin	T/R
AR1010T-I/SO	UART	-40°C to + 85°C	SOIC, 20 pin	T/R
AR1010T-I/SS	UART	-40°C to + 85°C	SSOP, 20 pin	T/R
AR1020-I/ML	I^2C/SPI	-40°C to + 85°C	QFN, 20 pin	Tube
AR1020-I/SO	I^2C/SPI	-40°C to + 85°C	SOIC, 20 pin	Tube
AR1020-I/SS	I^2C/SPI	-40°C to + 85°C	SSOP, 20 pin	Tube
AR1020T-I/ML	I^2C/SPI	-40°C to + 85°C	QFN, 20 pin	T/R
AR1020T-I/SO	I^2C/SPI	-40°C to + 85°C	SOIC, 20 pin	T/R
AR1020T-I/SS	I^2C/SPI	-40°C to + 85°C	SSOP, 20 pin	T/R

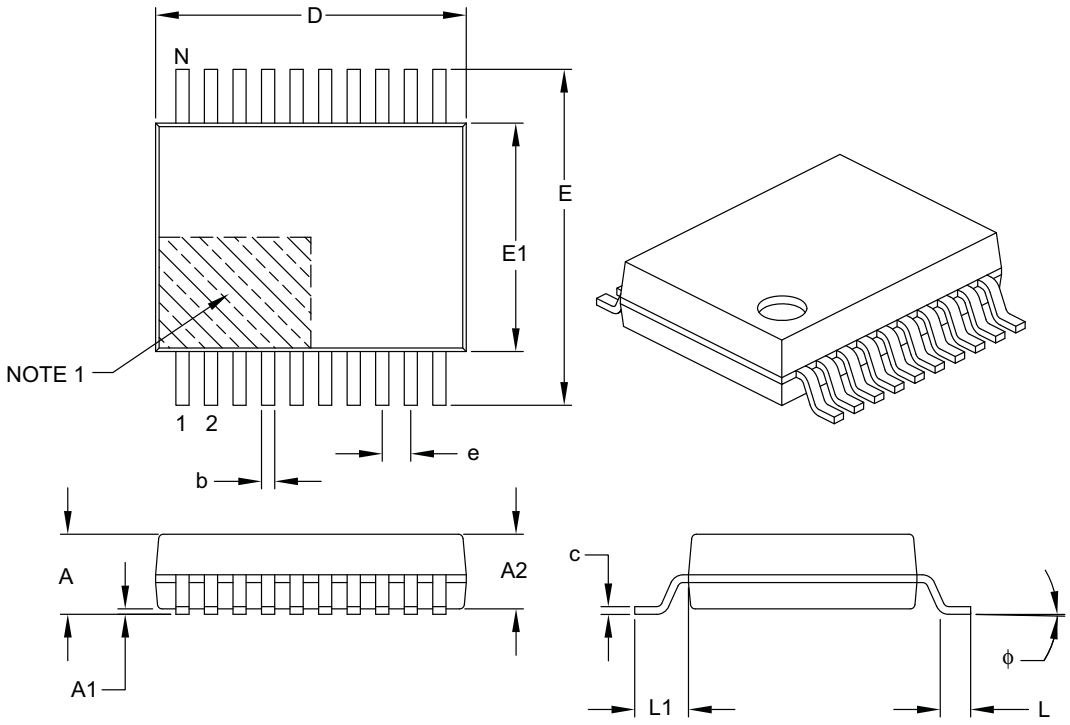
Downloaded from Arrow.com.

12.4 Package Details

The following sections give the technical details of the packages.

20-Lead Plastic Shrink Small Outline (SS) – 5.30 mm Body [SSOP]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N		20	
Pitch	e		0.65 BSC	
Overall Height	A	–	–	2.00
Molded Package Thickness	A2	1.65	1.75	1.85
Standoff	A1	0.05	–	–
Overall Width	E	7.40	7.80	8.20
Molded Package Width	E1	5.00	5.30	5.60
Overall Length	D	6.90	7.20	7.50
Foot Length	L	0.55	0.75	0.95
Footprint	L1		1.25 REF	
Lead Thickness	c	0.09	–	0.25
Foot Angle	ϕ	0°	4°	8°
Lead Width	b	0.22	–	0.38

Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.20 mm per side.
3. Dimensioning and tolerancing per ASME Y14.5M.

BSC: Basic Dimension. Theoretically exact value shown without tolerances.
REF: Reference Dimension, usually without tolerance, for information purposes only.

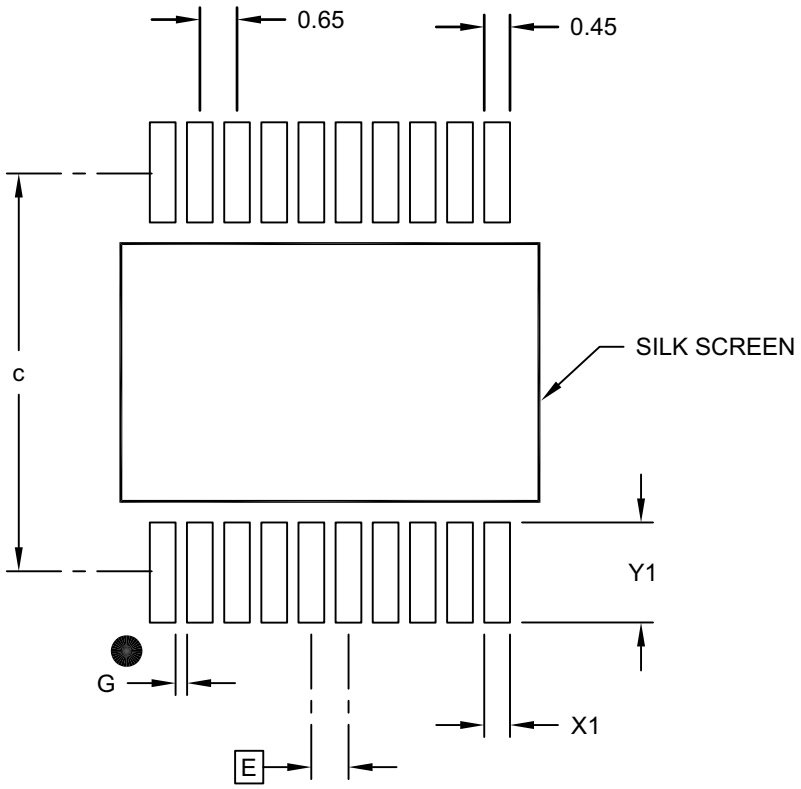
Microchip Technology Drawing C04-072B

© 2009-2016 Microchip Technology Inc.

Downloaded from Arrow.com.

20-Lead Plastic Shrink Small Outline (SS) - 5.30 mm Body [SSOP]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E	0.65 BSC		
Contact Pad Spacing	C		7.20	
Contact Pad Width (X20)	X1			0.45
Contact Pad Length (X20)	Y1			1.75
Distance Between Pads	G	0.20		

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

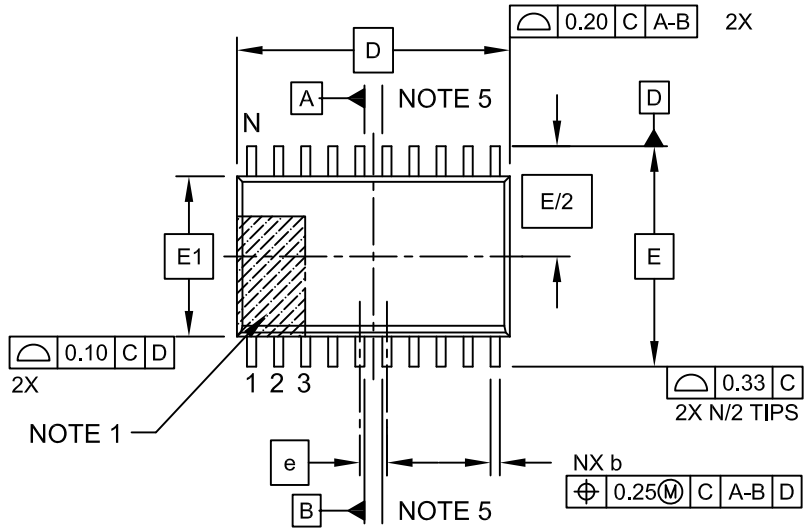
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2072B

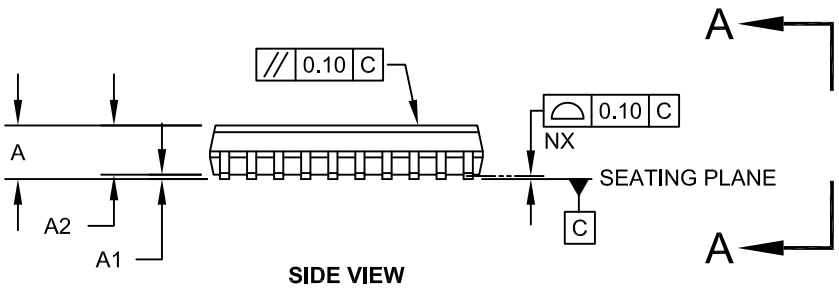
Downloaded from Arrow.com.

20-Lead Plastic Small Outline (SO) - Wide, 7.50 mm Body [SOIC]

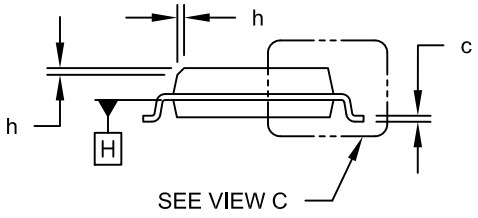
Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



TOP VIEW



SIDE VIEW



VIEW A-A

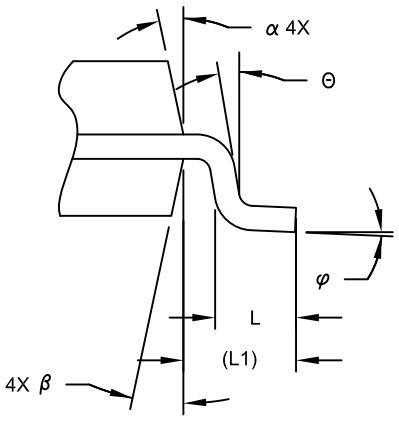
Microchip Technology Drawing C04-094C Sheet 1 of 2

© 2009-2016 Microchip Technology Inc.

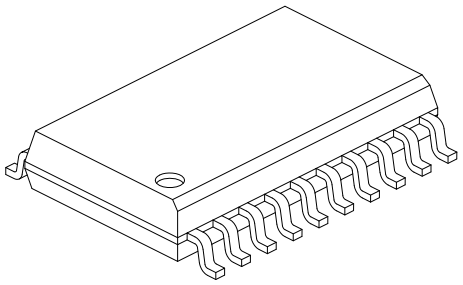
Downloaded from Arrow.com.

20-Lead Plastic Small Outline (SO) - Wide, 7.50 mm Body [SOIC]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



VIEW C



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N		20	
Pitch	e		1.27 BSC	
Overall Height	A	-	-	2.65
Molded Package Thickness	A2	2.05	-	-
Standoff §	A1	0.10	-	0.30
Overall Width	E		10.30 BSC	
Molded Package Width	E1		7.50 BSC	
Overall Length	D		12.80 BSC	
Chamfer (Optional)	h	0.25	-	0.75
Foot Length	L	0.40	-	1.27
Footprint	L1		1.40 REF	
Lead Angle	Θ	0°	-	-
Foot Angle	φ	0°	-	8°
Lead Thickness	c	0.20	-	0.33
Lead Width	b	0.31	-	0.51
Mold Draft Angle Top	α	5°	-	15°
Mold Draft Angle Bottom	β	5°	-	15°

Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. § Significant Characteristic
3. Dimension D does not include mold flash, protrusions or gate burrs, which shall not exceed 0.15 mm per end. Dimension E1 does not include interlead flash or protrusion, which shall not exceed 0.25 mm per side.
4. Dimensioning and tolerancing per ASME Y14.5M

   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

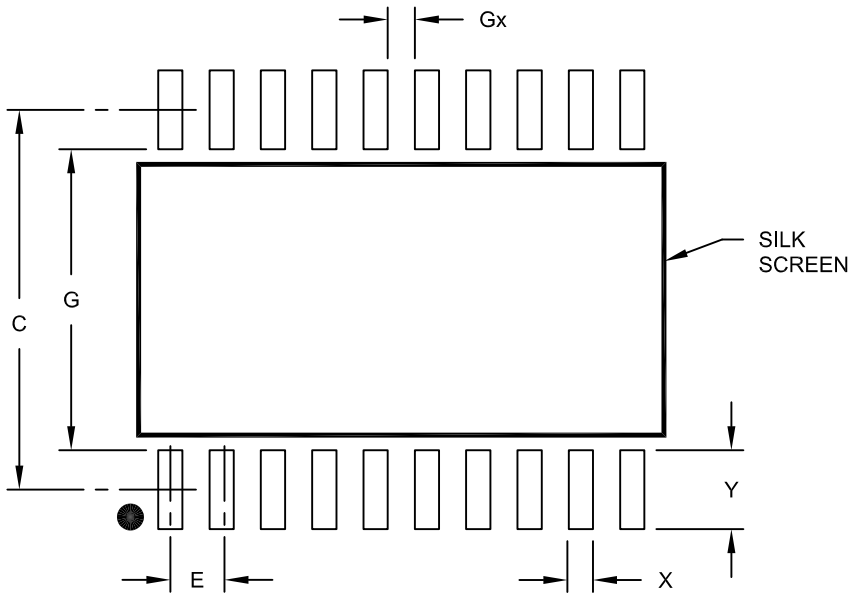
   REF: Reference Dimension, usually without tolerance, for information purposes only.
5. Datums A & B to be determined at Datum H.

Microchip Technology Drawing No. C04-094C Sheet 2 of 2

Downloaded from Arrow.com.

20-Lead Plastic Small Outline (SO) - Wide, 7.50 mm Body [SOIC]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E		1.27 BSC	
Contact Pad Spacing	C		9.40	
Contact Pad Width (X20)	X			0.60
Contact Pad Length (X20)	Y			1.95
Distance Between Pads	Gx	0.67		
Distance Between Pads	G	7.45		

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

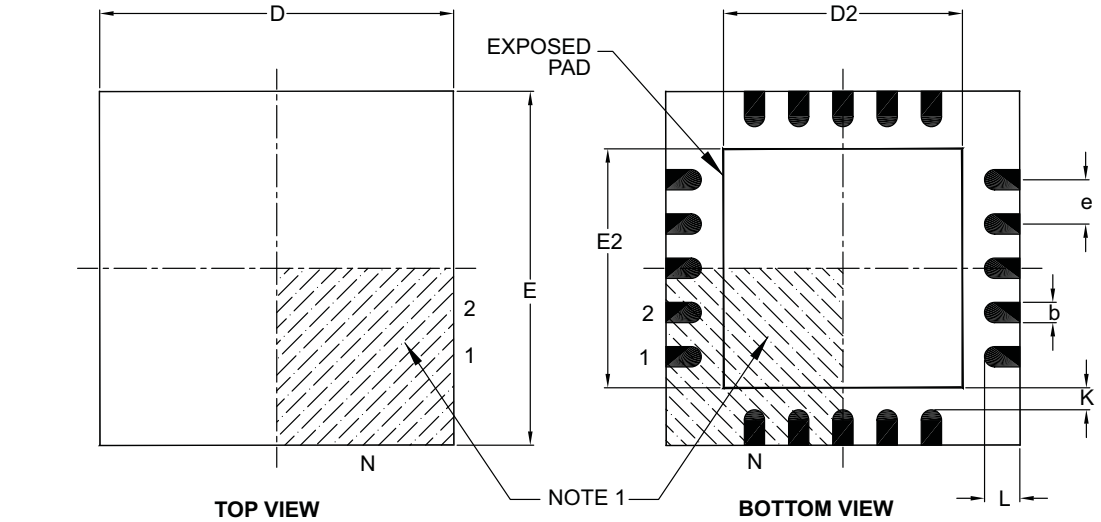
Microchip Technology Drawing No. C04-2094A

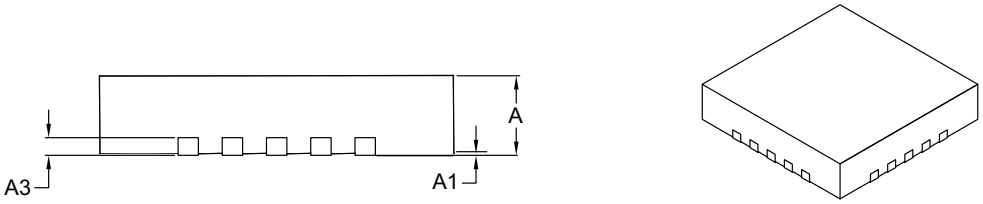
© 2009-2016 Microchip Technology Inc.

Downloaded from Arrow.com.

20-Lead Plastic Quad Flat, No Lead Package (ML) – 4x4x0.9 mm Body [QFN]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging





Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N		20	
Pitch	e		0.50 BSC	
Overall Height	A	0.80	0.90	1.00
Standoff	A1	0.00	0.02	0.05
Contact Thickness	A3		0.20 REF	
Overall Width	E		4.00 BSC	
Exposed Pad Width	E2	2.60	2.70	2.80
Overall Length	D		4.00 BSC	
Exposed Pad Length	D2	2.60	2.70	2.80
Contact Width	b	0.18	0.25	0.30
Contact Length	L	0.30	0.40	0.50
Contact-to-Exposed Pad	K	0.20	–	–

Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Package is saw singulated.
3. Dimensioning and tolerancing per ASME Y14.5M.

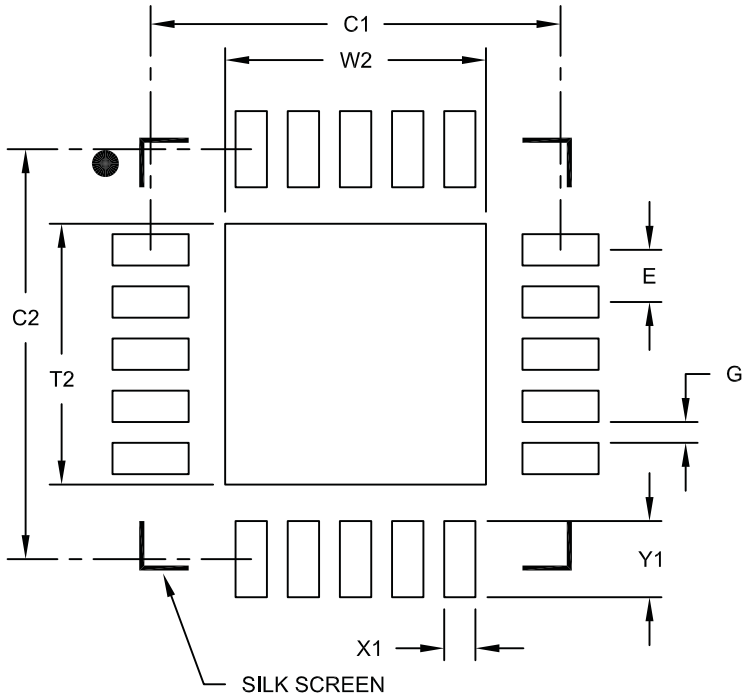
BSC: Basic Dimension. Theoretically exact value shown without tolerances.
REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-126B

Downloaded from Arrow.com.

20-Lead Plastic Quad Flat, No Lead Package (ML) - 4x4 mm Body [QFN]
With 0.40 mm Contact Length

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E		0.50 BSC	
Optional Center Pad Width	W2			2.50
Optional Center Pad Length	T2			2.50
Contact Pad Spacing	C1		3.93	
Contact Pad Spacing	C2		3.93	
Contact Pad Width	X1			0.30
Contact Pad Length	Y1			0.73
Distance Between Pads	G	0.20		

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2126A

© 2009-2016 Microchip Technology Inc.

Downloaded from Arrow.com.

APPENDIX A: DATA SHEET REVISION HISTORY

Revision A (07/2009)

Original release of this data sheet.

Revision B (03/2012)

Updated data sheet.

Revision C (07/2016)

Updated Table 4-1 and Table 5-1. Other minor corrections.

Downloaded from Arrow.com.

APPENDIX B: DEVICE DIFFERENCES

Modifying, removing or adding components may adversely affect touch performance.

Specific manufacturers and part numbers are provided only as a guide. Equivalents can be used.

TABLE B-1: BILL OF MATERIALS

Label	Quantity	Value	Description	Manufacturer	Part Number
C1	1	10 uF	Capacitor – Ceramic, 10 uF, 20%, 6.3V, X7R, 0603	AVX	06036D106MAT2A
C2	1	0.1 uF	Capacitor – Ceramic, 0.1 uF, 10%, 16V, X7R, 0603	AVX	0603YC104KAT2A
C3, C4, C5(1)	2-3	0.01 uF	Capacitor – Ceramic, 0.01 uF, 10%, 50V, X7R, 0603	AVX	06035C103KAT2A
D1-D8(2)	4-8	130W	Diode – Bidirectional, 130W, ESD Protection, SOD323	NXP	PESD5V0S1BA
R1	1	20 KΩ	Resistor – 20 KΩ, 1/10W, 5%, 0603	Yageo America	RC0603JR-0720KL
U1	1	N/A	Touch controller IC	Microchip	AR1011 or AR1021

Note 1: C5 is only needed for 5-wire applications.

2: D1-D8 are for ESD protection.

- 4-wire touch screen, use D1-D4
- 5-wire touch screen, use D1-D5
- 8-wire touch screen, use D1-D8

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

© 2009-2016 Microchip Technology Inc.

Downloaded from Arrow.com.

THE MICROCHIP WEBSITE

Microchip provides online support via our website at www.microchip.com. This website is used as a means to make files and information easily available to customers. Accessible by using your favorite Internet browser, the website contains the following information:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQ), technical support requests, online discussion groups, Microchip consultant program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

CUSTOMER CHANGE NOTIFICATION SERVICE

Microchip's customer notification service helps keep customers current on Microchip products. Subscribers will receive e-mail notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, access the Microchip website at www.microchip.com. Under "Support", click on "Customer Change Notification" and follow the registration instructions.

CUSTOMER SUPPORT

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Field Application Engineer (FAE)
- Technical Support

Customers should contact their distributor, representative or Field Application Engineer (FAE) for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in the back of this document.

Technical support is available through the website at: http://microchip.com/support

Downloaded from Arrow.com.

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE**.** Microchip disclaims all liability arising from this information and its use. Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights unless otherwise stated.

Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
═ ISO/TS 16949 ═

Trademarks

The Microchip name and logo, the Microchip logo, AnyRate, dsPIC, FlashFlex, flexPWR, Heldo, JukeBlox, KeeLoq, KeeLoq logo, Kleer, LANCheck, LINK MD, MediaLB, MOST, MOST logo, MPLAB, OptoLyzer, PIC, PICSTART, PIC32 logo, RightTouch, SpyNIC, SST, SST Logo, SuperFlash and UNI/O are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

ClockWorks, The Embedded Control Solutions Company, ETHERSYNCH, Hyper Speed Control, HyperLight Load, IntelliMOS, mTouch, Precision Edge, and QUIET-WIRE are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Analog-for-the-Digital Age, Any Capacitor, AnyIn, AnyOut, BodyCom, chipKIT, chipKIT logo, CodeGuard, dsPICDEM, dsPICDEM.net, Dynamic Average Matching, DAM, ECAN, EtherGREEN, In-Circuit Serial Programming, ICSP, Inter-Chip Connectivity, JitterBlocker, KleerNet, KleerNet logo, MiWi, motorBench, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, MultiTRAK, NetDetach, Omniscient Code Generation, PICDEM, PICDEM.net, PICkit, PICtail, PureSilicon, RightTouch logo, REAL ICE, Ripple Blocker, Serial Quad I/O, SQI, SuperSwitcher, SuperSwitcher II, Total Endurance, TSHARC, USBCheck, VariSense, ViewSpan, WiperLock, Wireless DNA, and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

Silicon Storage Technology is a registered trademark of Microchip Technology Inc. in other countries.

GestIC is a registered trademark of Microchip Technology Germany II GmbH & Co. KG, a subsidiary of Microchip Technology Inc., in other countries.

All other trademarks mentioned herein are property of their respective companies.

© 2009-2016, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

ISBN: 978-1-5224-0761-4

© 2009-2016 Microchip Technology Inc.

Downloaded from Arrow.com.

Worldwide Sales and Service

AMERICAS

Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
Technical Support:
http://www.microchip.com/support
Web Address:
www.microchip.com

Atlanta
Duluth, GA
Tel: 678-957-9614
Fax: 678-957-1455

Austin, TX
Tel: 512-257-3370

Boston
Westborough, MA
Tel: 774-760-0087
Fax: 774-760-0088

Chicago
Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

Cleveland
Independence, OH
Tel: 216-447-0464
Fax: 216-447-0643

Dallas
Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

Detroit
Novi, MI
Tel: 248-848-4000

Houston, TX
Tel: 281-894-5983

Indianapolis
Noblesville, IN
Tel: 317-773-8323
Fax: 317-773-5453

Los Angeles
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608

New York, NY
Tel: 631-435-6000

San Jose, CA
Tel: 408-735-9110

Canada - Toronto
Tel: 905-695-1980
Fax: 905-695-2078

ASIA/PACIFIC

Asia Pacific Office
Suites 3707-14, 37th Floor
Tower 6, The Gateway
Harbour City, Kowloon

Hong Kong
Tel: 852-2943-5100
Fax: 852-2401-3431

Australia - Sydney
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

China - Beijing
Tel: 86-10-8569-7000
Fax: 86-10-8528-2104

China - Chengdu
Tel: 86-28-8665-5511
Fax: 86-28-8665-7889

China - Chongqing
Tel: 86-23-8980-9588
Fax: 86-23-8980-9500

China - Dongguan
Tel: 86-769-8702-9880

China - Guangzhou
Tel: 86-20-8755-8029

China - Hangzhou
Tel: 86-571-8792-8115
Fax: 86-571-8792-8116

China - Hong Kong SAR
Tel: 852-2943-5100
Fax: 852-2401-3431

China - Nanjing
Tel: 86-25-8473-2460
Fax: 86-25-8473-2470

China - Qingdao
Tel: 86-532-8502-7355
Fax: 86-532-8502-7205

China - Shanghai
Tel: 86-21-5407-5533
Fax: 86-21-5407-5066

China - Shenyang
Tel: 86-24-2334-2829
Fax: 86-24-2334-2393

China - Shenzhen
Tel: 86-755-8864-2200
Fax: 86-755-8203-1760

China - Wuhan
Tel: 86-27-5980-5300
Fax: 86-27-5980-5118

China - Xian
Tel: 86-29-8833-7252
Fax: 86-29-8833-7256

ASIA/PACIFIC

China - Xiamen
Tel: 86-592-2388138
Fax: 86-592-2388130

China - Zhuhai
Tel: 86-756-3210040
Fax: 86-756-3210049

India - Bangalore
Tel: 91-80-3090-4444
Fax: 91-80-3090-4123

India - New Delhi
Tel: 91-11-4160-8631
Fax: 91-11-4160-8632

India - Pune
Tel: 91-20-3019-1500

Japan - Osaka
Tel: 81-6-6152-7160
Fax: 81-6-6152-9310

Japan - Tokyo
Tel: 81-3-6880- 3770
Fax: 81-3-6880-3771

Korea - Daegu
Tel: 82-53-744-4301
Fax: 82-53-744-4302

Korea - Seoul
Tel: 82-2-554-7200
Fax: 82-2-558-5932 or
82-2-558-5934

Malaysia - Kuala Lumpur
Tel: 60-3-6201-9857
Fax: 60-3-6201-9859

Malaysia - Penang
Tel: 60-4-227-8870
Fax: 60-4-227-4068

Philippines - Manila
Tel: 63-2-634-9065
Fax: 63-2-634-9069

Singapore
Tel: 65-6334-8870
Fax: 65-6334-8850

Taiwan - Hsin Chu
Tel: 886-3-5778-366
Fax: 886-3-5770-955

Taiwan - Kaohsiung
Tel: 886-7-213-7828

Taiwan - Taipei
Tel: 886-2-2508-8600
Fax: 886-2-2508-0102

Thailand - Bangkok
Tel: 66-2-694-1351
Fax: 66-2-694-1350

EUROPE

Austria - Wels
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

Denmark - Copenhagen
Tel: 45-4450-2828
Fax: 45-4485-2829

France - Paris
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

Germany - Dusseldorf
Tel: 49-2129-3766400

Germany - Karlsruhe
Tel: 49-721-625370

Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

Italy - Venice
Tel: 39-049-7625286

Netherlands - Drunen
Tel: 31-416-690399
Fax: 31-416-690340

Poland - Warsaw
Tel: 48-22-3325737

Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

Sweden - Stockholm
Tel: 46-8-5090-4654

UK - Wokingham
Tel: 44-118-921-5800
Fax: 44-118-921-5820

06/23/16

Downloaded from Arrow.com.