



MICROCHIP

**AR1000 Series Resistive Touch Screen
Controller**

Contact Information
Microchip Technology, Inc.
mTouch Touchscreen Controller Products
9055 N. 51st Street, Suite H
Brown Deer, WI 53223
www.Microchip.com
Phone: 414-355-4675
Fax: 414-355-4775

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in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates.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.

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
== ISO/TS 16949:2002 ==



Trademarks

The Microchip name and logo, the Microchip logo, dsPIC, KEELOQ, KEELOQ logo, MPLAB, PIC, PICmicro, PICSTART, rfPIC and UNI/O are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

FilterLab, Hampshire, HI-TECH C, Linear Active Thermistor, MXDEV, MXLAB, SEEVAL and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Analog-for-the-Digital Age, Application Maestro, CodeGuard, dsPICDEM, dsPICDEM.net, dsPICworks, dsSPEAK, ECAN, ECONOMONITOR, FanSense, HI-TIDE, In-Circuit Serial Programming, ICSP, ICEPIC, Mindi, MiWi, MPASM, MPLAB Certified logo, MPLIB, MPLINK, mTouch, Omniscient Code Generation, PICC, PICC-18, PICkit, PICDEM, PICDEM.net, PICtail, PIC32 logo, REAL ICE, rLAB, Select Mode, Total Endurance, TSHARC, WiperLock 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.

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

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

Printed on recycled paper.

Microchip received ISO/TS-16949:2002 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.

TABLE OF CONTENTS

1	GENERAL.....	1
2	FEATURES.....	1
3	APPLICATIONS.....	2
4	ORDERING.....	2
5	BLOCK DIAGRAM.....	2
6	PIN DIAGRAM.....	3
7	FUNCTIONAL DESCRIPTION.....	4
7.1	MAIN SCHEMATIC.....	4
7.2	BILL OF MATERIALS.....	4
7.3	4, 5, 8-WIRE SENSOR SELECTION.....	5
7.4	4-WIRE TOUCH SENSOR INTERFACE.....	5
7.5	5-WIRE TOUCH SENSOR INTERFACE.....	6
7.6	8-WIRE TOUCH SENSOR INTERFACE.....	7
7.7	COMMUNICATION INTERFACE.....	8
7.7.1	I ² C.....	8
7.7.2	SPI.....	9
7.7.3	UART.....	10
7.8	STATUS LED.....	10
7.9	ESD CONSIDERATIONS.....	11
7.10	NOISE CONSIDERATIONS.....	11
7.11	TOUCH REPORTING PROTOCOL.....	11
7.12	CONFIGURATION REGISTERS.....	12
7.12.1	TouchThreshold Register (offset 0x02).....	12
7.12.2	SensitivityFilter Register (offset 0x03).....	12
7.12.3	SamplingFast Register (offset 0x04).....	13
7.12.4	SamplingSlow Register (offset 0x05).....	13
7.12.5	AccuracyFilterFast Register (offset 0x06).....	13
7.12.6	AccuracyFilterSlow Register (offset 0x07).....	13
7.12.7	SpeedThreshold Register (offset 0x08).....	13
7.12.8	SleepDelay Register (offset 0x0A).....	14
7.12.9	PenUpDelay Register (offset 0x0B).....	14
7.12.10	TouchMode Register (offset 0x0C).....	15
7.12.11	TouchOptions Register (offset 0x0D).....	16
7.12.12	CalibrationInset Register (offset 0x0E).....	16
7.12.13	PenStateReportDelay Register (offset 0x0F).....	17
7.12.14	TouchReportDelay Register (offset 0x11).....	17
7.12.15	User Configuration – Spiking.....	17
7.13	COMMAND FORMAT.....	17
7.14	COMMAND RESPONSE.....	18
7.15	COMMANDS.....	18
7.15.1	Enable Touch - 0x12.....	18
7.15.2	Disable Touch - 0x13.....	18
7.15.3	Calibrate - 0x14.....	19
7.15.3.1	How the Calibration Data is Encoded and Stored in EEPROM.....	20
7.15.4	Register Read - 0x20.....	21
7.15.5	Register Write - 0x21.....	21
7.15.6	Register Start Address Request - 0x22.....	21
7.15.7	Registers Write to EEPROM - 0x23.....	21
7.15.8	EEPROM Read - 0x28.....	22
7.15.9	EEPROM Write - 0x29.....	22
7.15.10	EEPROM Write to Registers - 0x2B.....	22
7.16	CALIBRATION OF TOUCH SENSOR WITH CONTROLLER.....	23

8	BASICS OF RESISTIVE SENSORS	25
8.1	TERMINOLOGY	25
8.2	GENERAL	25
8.3	4-WIRE SENSOR	26
8.4	8-WIRE SENSOR	27
8.5	5-WIRE SENSOR	28
8.6	SUMMARY	28
9	ELECTRICAL SPECIFICATIONS	29
9.1	ABSOLUTE MAXIMUM RATINGS(†)	29
10	PACKAGING	30
10.1	20 LEAD SOIC	30
10.2	20 LEAD SSOP	31
10.3	20 LEAD QFN (ML)	32

FIGURES

Figure 1:	Block Diagram	2
Figure 2 :	Pin Diagrams – SOIC / SSOP and QFN Packages	3
Figure 3 :	Main Schematic (SOIC, SSOP package pin out)	4
Figure 4 :	4-wire Touch Sensor Interface	5
Figure 5 :	5-wire Touch Sensor Interface	6
Figure 6 :	8-wire Touch Sensor Interface	7
Figure 7 :	I ² C Timing Diagram – Receive Data	8
Figure 8 :	I ² C Timing Diagram – Transmit Data	8
Figure 9 :	I ² C Timing Diagram – Touch Report Protocol	9
Figure 10 :	I ² C Timing Diagram – Command Protocol	9
Figure 11 :	SPI Timing Diagram – Bit Timing	9
Figure 12 :	SPI Timing Diagram – Touch Report Protocol	10
Figure 13 :	SPI Timing Diagram – Command Protocol	10
Figure 14:	4-Wire Decoding	26
Figure 15:	8-Wire Decoding	27
Figure 16:	5-Wire Decoding	28
Figure 17 :	20 Lead SOIC Package	30
Figure 18 :	20 Lead SSOP Package	31
Figure 19 :	20 Lead QFN Package	32

TABLES

Table 1:	Ordering Part Numbers	2
Table 2:	Pin Descriptions	3
Table 3 :	Bill of Materials	4
Table 4 :	4/8-wire vs 5-wire Selection	5
Table 5 :	Communication Selection	8
Table 6 :	Communication Pins	8
Table 7 :	Touch Coordinate Reporting Protocol	11
Table 8 :	Configuration Registers	12
Table 9 :	Command Set Summary	18
Table 10 :	Sensor Comparison	25

1 GENERAL

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

The AR1000 Series is designed for high volume, small form factor touch solutions with quick time to market requirements.

Developed by touch experts with over 15 years experience, the AR1000 Series has sophisticated proprietary touch screen decoding algorithms allowing it to send your application fully processed and reliable touch coordinates.

2 FEATURES

<p>Special Features:</p> <ul style="list-style-type: none">• RoHS Compliant• Power-Saving Sleep mode• Industrial Temperature Range• Built in drift compensation algorithm• 128 Bytes of user EEPROM• 4 x 4 mm QFN package <p>Power Requirements:</p> <ul style="list-style-type: none">• Operating Voltage: 3.3 to 5.0 volts $\pm 5\%$• Standby Sleep Current: $<1\mu\text{A}$• Operating "No touch" Current: 3.0mA typ.• Operating "Touch" Current: 17mA typ with a touch sensor having 200Ω layers. - Actual current is dependent on the touch sensor used. <p>Touch Modes:</p> <ul style="list-style-type: none">• Off, Stream, Down, Up, and more.	<p>Touch Sensor Support:</p> <ul style="list-style-type: none">• 4 wire, 5 wire, and 8 wire analog resistive• Lead to Lead Resistance: 50Ω to 2,000Ω• Layer to Layer Capacitance: 0 to 0.5μF• Touch Sensor Time Constant: 500us maximum <p>Touch Resolution:</p> <ul style="list-style-type: none">• 10-bit resolution maximum <p>Touch Coordinate Report Rate:</p> <ul style="list-style-type: none">• 140 reports per second typ. with a touch sensor of 0.02μF with 200Ω layers. - Actual report rate is dependent on the touch sensor used. <p>Communications:</p> <ul style="list-style-type: none">• SPI, slave mode, p/n AR1020• I²C™, slave mode, p/n AR1020• UART, 9600 bps, p/n AR1010
--	--

Sensor Support

All 4, 5, and 8-wire sensors are supported, regardless of manufacturer or construction. See *BASICS OF RESISTIVE SENSORS* section.

Low-Power Wake-Up

Wake-up from power saving sleep mode, via a touch or communication input.

Configuration Registers

Configuration registers provide user configuration of controller features. See *Configuration Registers* section.

Touch Algorithms

Algorithms provide for good calibration-corrected coordinates, with no additional development or code.

- Touch decoding
- Coordinate data filtering
- Calibration corrected coordinate option
- Built-in touch modes

Communication Control

AR1010 supports UART communication. AR1020 supports I2C and SPI communication. See *Communication Interface* section.

3 APPLICATIONS

The AR1000 is suitable for any application that requires fast and reliable integration of touch in the design, 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

4 ORDERING

Part Number	Communication Type	Temp Range	Pin Package	Packing
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 ² C/SPI	-40°C to +85°C	QFN, 20 pin	Tube
AR1020-I/SO	I ² C/SPI	-40°C to +85°C	SOIC, 20 pin	Tube
AR1020-I/SS	I ² C/SPI	-40°C to +85°C	SSOP, 20 pin	Tube
AR1020T-I/ML	I ² C/SPI	-40°C to +85°C	QFN, 20 pin	T/R
AR1020T-I/SO	I ² C/SPI	-40°C to +85°C	SOIC, 20 pin	T/R
AR1020T-I/SS	I ² C/SPI	-40°C to +85°C	SSOP, 20 pin	T/R

Table 1: Ordering Part Numbers

5 BLOCK DIAGRAM

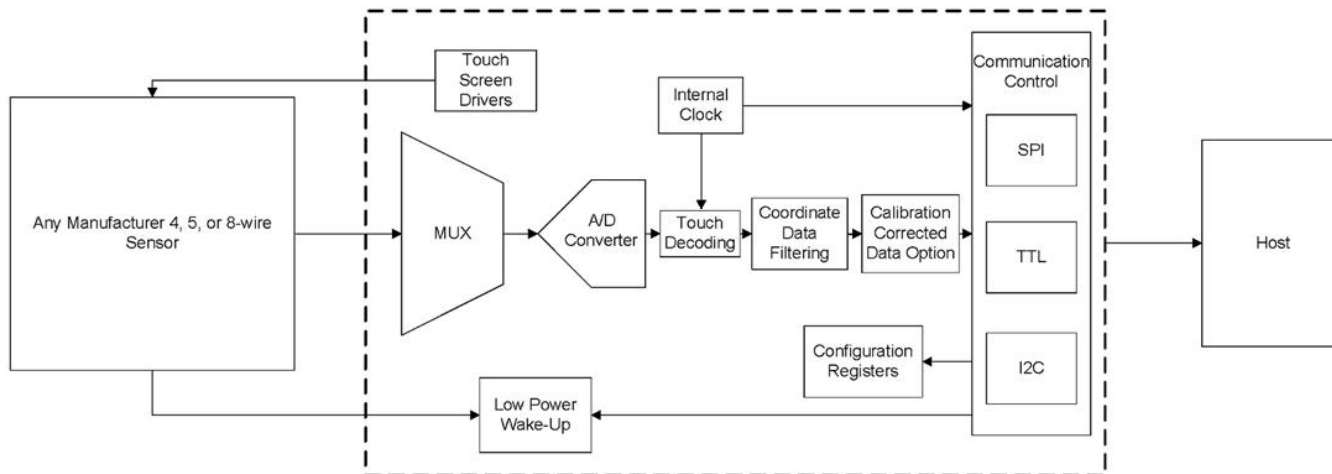


Figure 1: Block Diagram

6 PIN DIAGRAM

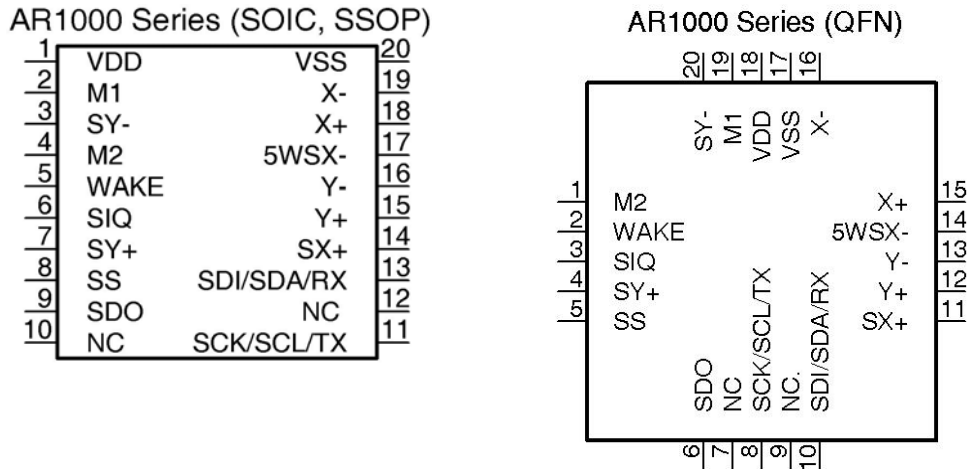


Figure 2 : Pin Diagrams – SOIC / SSOP and QFN Packages

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

Table 2: Pin Descriptions

7 FUNCTIONAL DESCRIPTION

7.1 Main Schematic

A main application schematic is shown below for the SOIC/SSOP package pinout. See the *Pin Diagram* section for the QFN package pinout.

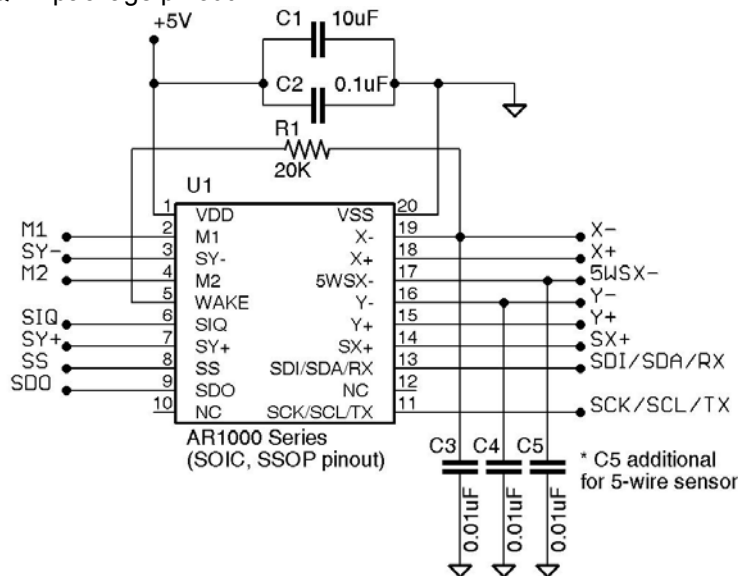


Figure 3 : Main Schematic (SOIC, SSOP package pin out)

7.2 Bill of Materials

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.

Label	Qty	Value	Description	Manufacturer	Part Number
C1	1	10 uF	Capacitor – Ceramic, 10uF, 20%, 6.3V, X7R, 0603	AVX	06036D106MAT2A
C2	1	0.1 uF	Capacitor – Ceramic, 0.1uF, 10%, 16V, X7R, 0603	AVX	0603YC104KAT2A
C3, C4, C5 ¹	2-3	0.01 uF	Capacitor – Ceramic, 0.01uF, 10%, 50V, X7R, 0603	AVX	06035C103KAT2A
D1-D8 ²	4-8	130 W	Diode – Bi-dir., 130W, ESD Protection, SOD323	NXP	PESD5V0S1BA
R1	1	20K ohm	Resistor - 20K ohm, 1/10W, 5%, 0603	Yageo America	RC0603JR-0720KL
U1	1	Na	Touch controller IC	Microchip	AR1010 or AR1020

Table 3 : Bill of Materials

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

Note 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

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 35pF. 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.

7.3 4, 5, 8-Wire Sensor Selection

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

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

Table 4 : 4/8-wire vs 5-wire Selection

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.

7.4 4-wire Touch Sensor Interface

Sensor tail pin-outs 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.

Connections to a 4-wire touch sensor are as follows.

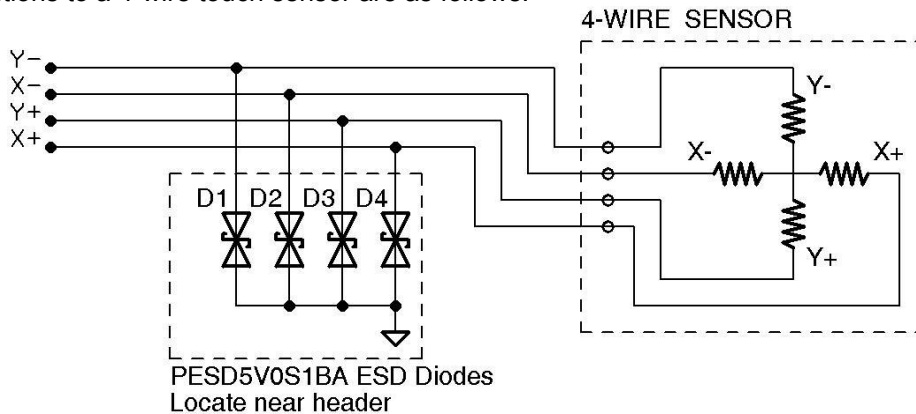


Figure 4 : 4-wire Touch Sensor Interface

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 35pF. 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.

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

7.5 5-wire Touch Sensor Interface

Sensor tail pin-outs 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 topsheet must be connected to the controller's 5WSX- pin.

Connections to a 5-wire touch sensor are as follows.

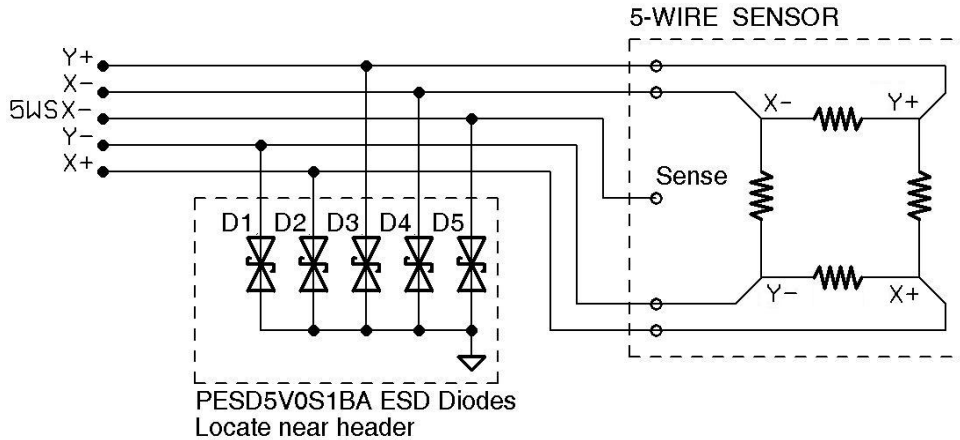


Figure 5 : 5-wire Touch Sensor Interface

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 35pF. 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.

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

7.6 8-wire Touch Sensor Interface

Sensor tail pin-outs 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. 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 as follows.

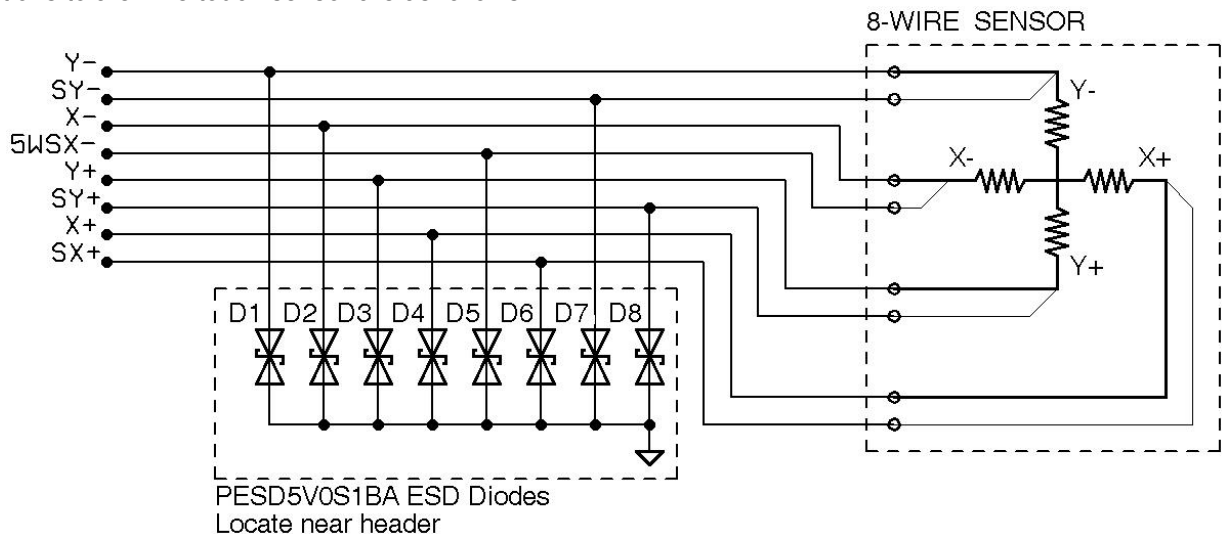


Figure 6 : 8-wire Touch Sensor Interface

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 35pF. 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.

7.7 Communication Interface

The desired communication type is hardware selectable by chip part number and using pin M1.

Type	Chip P/N	M1 pin
I ² C	AR1020	VSS
SPI	AR1020	VDD
UART	AR1010	VDD

Table 5 : Communication Selection

The communication interface pins are shown below for each communication type.

Type	Communication Pins
UART	RX, TX, VSS
I ² C	SCL, SDA, SDO (I _{rq}), VSS
SPI	SCK, SDI, SDO, SS, SIQ (I _{rq}), VSS

Table 6 : Communication Pins

7.7.1 I²C

I²C operates in slave mode with 7-bit address of 0x9A.

The SDO pin functions as an interrupt output. It is asserted high when data is available.

Timing diagrams are shown below.

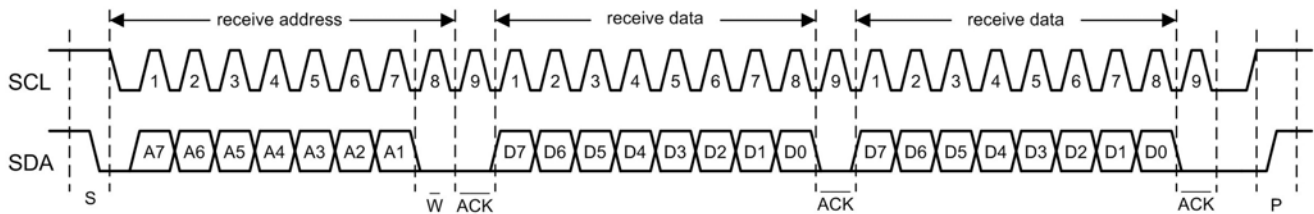


Figure 7 : I²C Timing Diagram – Receive Data

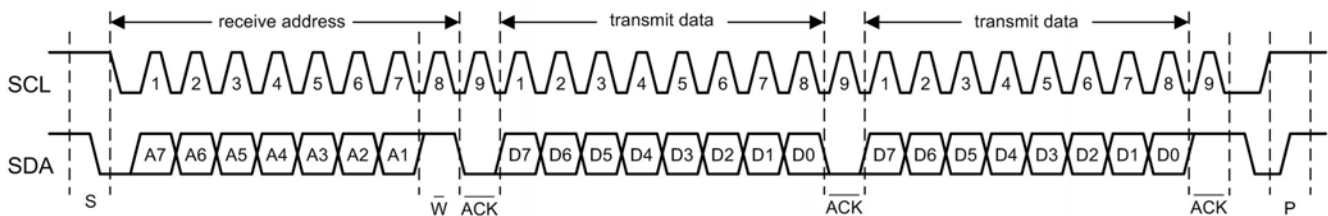


Figure 8 : I²C Timing Diagram – Transmit Data

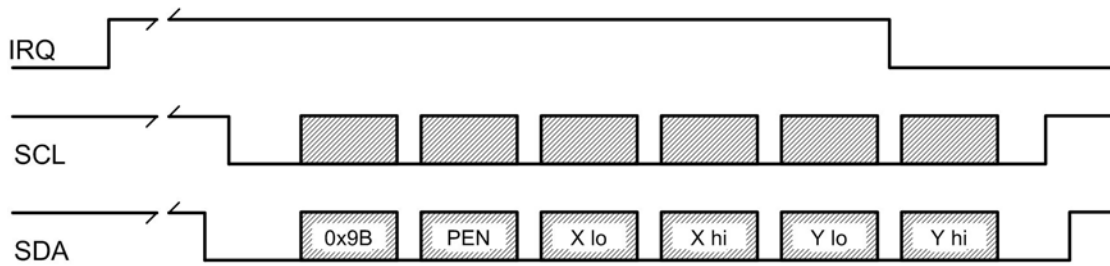


Figure 9 : I²C Timing Diagram – Touch Report Protocol

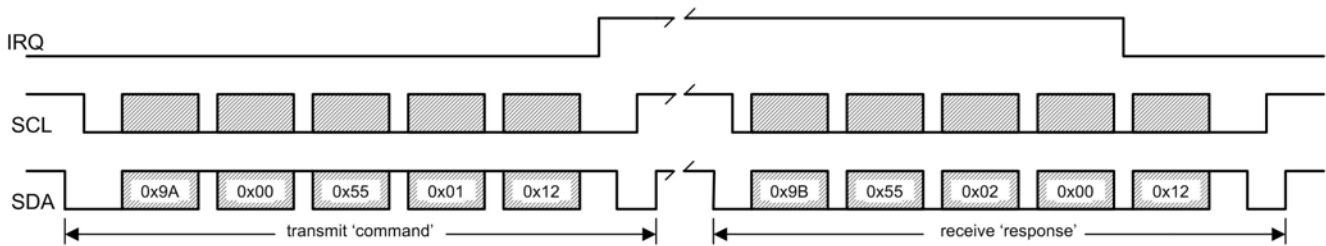


Figure 10 : I²C Timing Diagram – Command Protocol

7.7.2 SPI

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

The SIQ pin functions as an interrupt output. It is asserted high when data is available. The pin has a dual purpose with driving an optional LED.

The SS pin is optional for “Slave Select” functionality. It is active (selects) when pulled low to VSS. Tie to VSS if not used.

If data is clocked out when the controller has no valid data to provide, then 0x4d (ASCII "M") will be presented.

Timing diagrams are shown below.

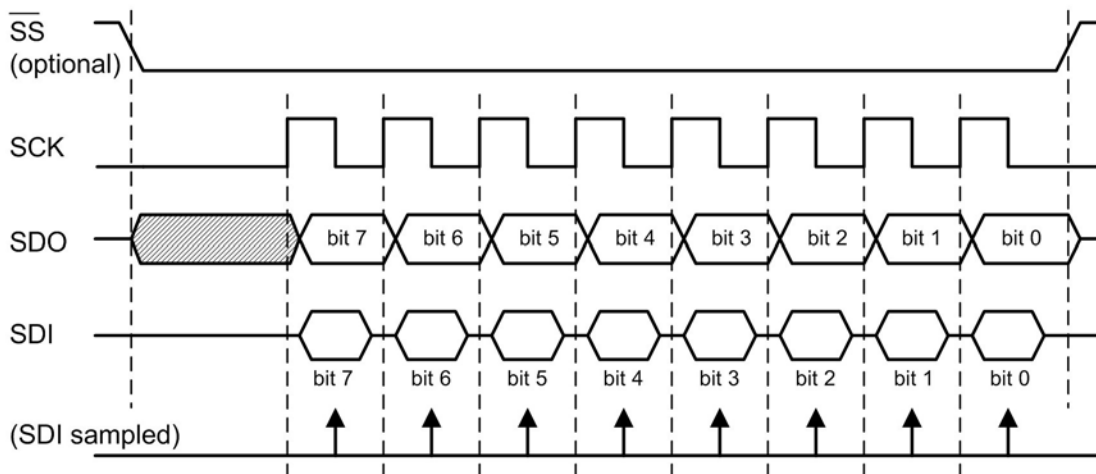


Figure 11 : SPI Timing Diagram – Bit Timing

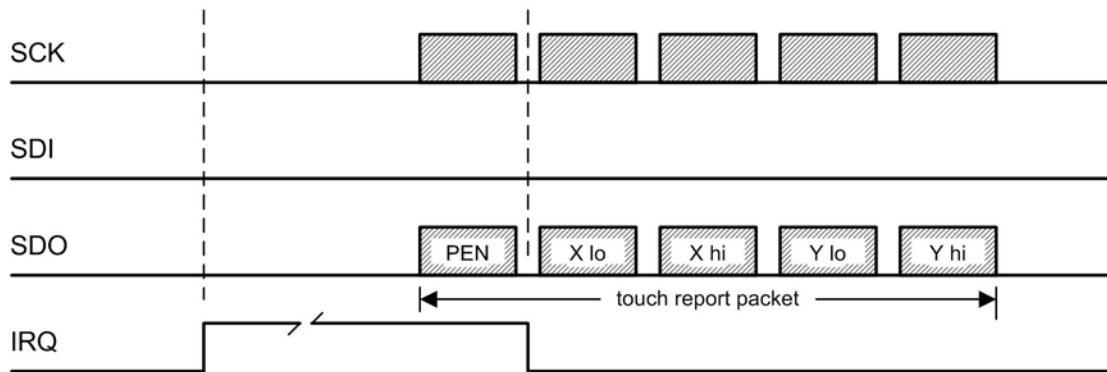


Figure 12 : SPI Timing Diagram – Touch Report Protocol

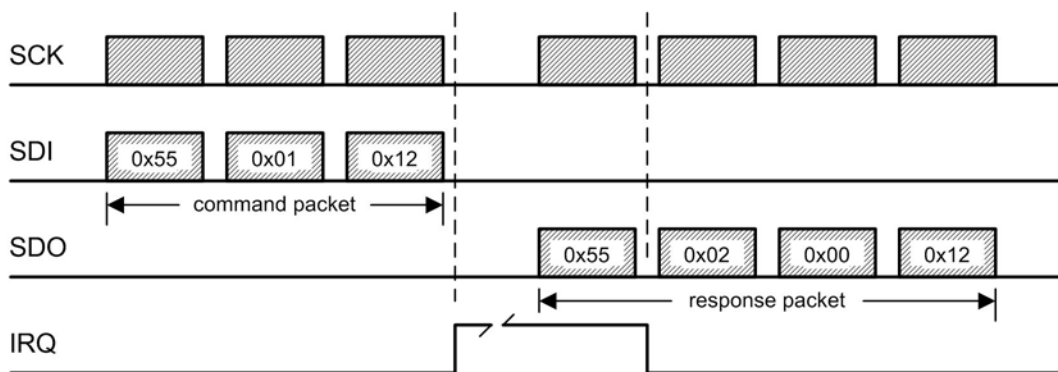


Figure 13 : SPI Timing Diagram – Command Protocol

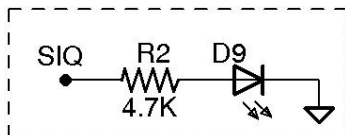
7.7.3 UART

The SDO pin is not used and must be left as a “no connect”. Do NOT tie it directly to VSS or VDD.

UART communication is at a fixed 9600 baud rate.

7.8 Status LED

The LED and associated resistor are optional.



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.

If the SIQ pin is not used, it must be left as a no connect and NOT tied to circuit VDD or VSS.

7.9 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 35pF. 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.

7.10 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.

7.11 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, as this product currently defines, in a non-calibrated state, bits X11:X0 and Y11:Y0 as 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.

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	X5	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

Table 7 : Touch Coordinate Reporting Protocol

where:

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

7.12 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.

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.

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

Table 8 : Configuration Registers

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.

- 1) Issue the *<Register Start Address Request>* command to obtain the *Start* address for the register group.
- 2) Calculated the desired register's absolute address by adding the register's *Offset* value to *Start* address for the register group.
- 3) Issue the *<Register Read>* or *<Register Write>* command, using the calculated register's absolute address.

Warning: Use of invalid register value will yield unpredictable results!

7.12.1 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 touches conditions. Valid values are as follows.

$$0 \leq \text{TouchThreshold} \leq 255$$

7.12.2 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.

$$0 \leq \text{SensitivityFilter} \leq 255$$

7.12.3 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, 2, 4, 8, 16, 32, 64, 128

7.12.4 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

7.12.5 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.

$1 \leq \text{AccuracyFilterFast} \leq 8$

7.12.6 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.

$1 \leq \text{AccuracyFilterSlow} \leq 8$

7.12.7 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.

$0 \leq \text{SpeedThreshhold} \leq 255$

7.12.8 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.

$$0 \leq \text{SleepDelay} \leq 255$$

$$\text{Sleep Delay Time} = \text{SleepDelay} * 100\text{ms}$$

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.

7.12.9 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.

$$0 \leq \text{PenUpDelay} \leq 255$$

$$\text{Pen Up Delay Time} \approx \text{PenUpDelay} * 240\mu\text{s}$$

7.12.10 TouchMode Register (offset 0x0C)

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

There are three states of touch for which the controller touch reporting action 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 include a “P” (Pen) bit that indicates the pen state.

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

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

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	

bit 7-5 **PD<7:5>**: 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<4:3>**: 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

7.12.11 TouchOptions Register (offset 0x0D)

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

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

U-0	U-0	U-0	U-0	U-0	U-0	R/W	R/W
-	-	-	-	-	-	48W	CCE
Bit 7						Bit 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

A 4-wire touch sensor will not work if the *48W* configuration bit is incorrectly defined as one, which selects 8-wire.

An 8-wire touch sensor will provide basic operation if the *48W* configuration bit is incorrectly defined as zero, 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 one, selecting 8-wire.

7.12.12 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.

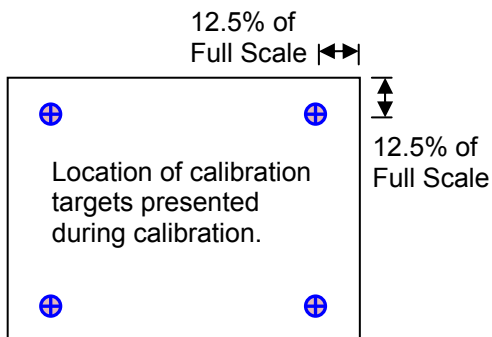
A software application that issues the *Calibration Mode* command must present the displayed calibration targets at the same inset percentage as defined in this CalibrationInset register.

Valid values are as follows.

$$0 \leq \text{CalibrationInset} \leq 40$$

$$\text{Calibration Inset} = (\text{CalibrationInset} / 2) \% , \text{ Range of } 0\text{--}20\% \text{ with } 0.5\% \text{ resolution}$$

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.



7.12.13 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 *TouchMode Register* section 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.

$$0 \leq \text{PenStateReportDelay} \leq 255$$

$$\text{Pen State Report Delay Time} = \text{PenStateReportDelay} * 50\mu\text{s}$$

7.12.14 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, increased the value of this register will reduce the rate at which the controller sends touch reports. Valid values are as follows.

$$0 \leq \text{TouchReportDelay} \leq 255$$

$$\text{Touch Report Delay Time} \approx \text{TouchReportDelay} * 500\mu\text{s}$$

7.12.15 User Configuration – Spiking

If the reported touch position contains some spiking position errors, then this can be improved by changing the following *Configuration Register* values.

- SamplingFast (offset 0x04) = 8
- AccuracyFilterFast (offset 0x06) = 8

The trade off with the above change is that the touch report rate (points per second) will be decreased. This is fine for many applications.

7.13 Command Format

The controller supports application specific configuration commands.

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 50ms
- 3) Send desired command/s
- 4) Send *Enable Touch* command

The format for sending commands is: <Header><DataSize><Command><Data>...<Data>

<Header> is defined as a value of 0x55 and is used to mark the beginning of the command packet.

<DataSize> is the number of bytes being sent in the command packet, after <DataSize>.

<Command> is a value assigned to a specific command.

<Data> represent from zero to eight bytes of command specific data.

7.14 Command Response

A received command will be responded to as follows.

<Header><DataSize><Response><Command><Data>...<Data>

<Header> is defined as a value of 0x55 and is used to mark the beginning of the response packet.

<DataSize> is the number of bytes being sent in the response packet, after <DataSize>.

<Response> is a returned status to the received command.

Value	Description
0x00	Success
0x01	Unrecognized Command
0x03	Unrecognized Header
0x04	Command Timeout (entire command not received within 100ms)
0xFC	Calibration Cancel

<Command> is a confirmation of the command being responded to.

<Data> represent from zero to eight bytes of returned data.

7.15 Commands

Command Value	Command Description
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

Table 9 : Command Set Summary

7.15.1 Enable Touch - 0x12

Controller will send touch coordinate reports for valid touch conditions.

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

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

7.15.2 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>

7.15.3 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 *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 pre-collected calibration data.

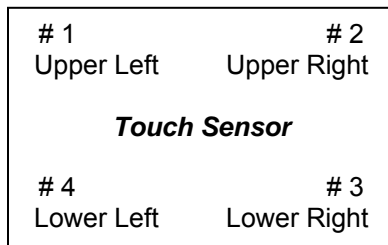
The “calibration” mode will be cancelled by sending any command before the mode has been completed.

Send : <0x55><0x02><0x14><Calibration Type>

<u>Calibration Type</u>	<u>Description</u>
0x04	4 point

Receive : <0x55><0x02><Response><0x14> for initial command response
<0x55><0x02><Response><0x14> for response to 1st target touch release
<0x55><0x02><Response><0x14> for response to 2nd target touch release
<0x55><0x02><Response><0x14> for response to 3rd target touch release
<0x55><0x02><Response><0x14> for response to 4th target touch release

To provide for proper touch orientation, the four sequential calibration touches must be input in the following physical order on the touch sensor.



7.15.3.1 How the Calibration Data is Encoded and Stored in EEPROM

This section is informational only. No user action is required.

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.

The manipulated calibration values are stored in 17 address locations of the EEPROM. In the EEPROM, the calibration data follows the storage of the controller operating parameters. The controller operating parameters and the calibration data are each preceded by a separator value of 0x55. Coordinates are 10-bit significant values, scaled to 16-bit and stored in a Hi and Lo byte pair.

Address Value	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	
0x14	0x15	0x16	0x17	0x18	0x19	0x1A	0x1B	0x1C	0x1D	0x1E	0x1F	0x20	0x21	0x22	0x23	0x24	0x25	
0x55																		

Decode the above data to as follows:

- 1) Swap the order of stored Lo and Hi bytes for a given coordinate.
- 2) Convert the 16-bit value (stored Hi and Lo 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 Lo = 0x40 and Hi = 0xF3:

```
Swap      : 0xF340
Hex to Decimal : 62272
Divide by 64 : 973
```

Flip State Byte: R = Readable bit, W = Writable bit, U = Unimplemented bit read as '0'

U-0	U-0	U-0	U-0	U-0	R/W	R/W	R/W
-	-	-	-	-	XYFLIP	XFLIP	YFLIP
Bit 7							Bit 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 and 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

7.15.4 Register Read - 0x20

Read 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>

of Registers to Read : 0x01 thru 0x08
Register Address High byte : 0x00

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

7.15.5 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>

of Registers to Write : 0x01 thru 0x08
Register Address High byte : 0x00

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

7.15.6 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>

7.15.7 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>

7.15.8 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>

of EEPROM to Read : 0x01 thru 0x08
EEPROM Address High byte : 0x00

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

7.15.9 EEPROM Write - 0x29

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 write values to the EEPROM.

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>

of EEPROM to Write : 0x01 thru 0x08
EEPROM Address High byte : 0x00

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

7.15.10 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>

7.16 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 cancelled 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 0x07 is used below, for this example.

```
Send      : <0x55><0x01><0x22>
Receive   : <0x55><0x03><0x00><0x22><0x07>
```

- 3) Calculate the CalibrationInset register's address by adding it's offset value of 0x0E to the register group *Start* address of 0x07.

Register Address = Register Start Address + CalibrationInset Register Offset = 0x07 + 0x0E = 0x15

- 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

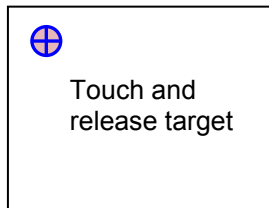
- 5) Set the Calibration Inset by writing the desired value to the CalibrationInset register.

```
Send      : <0x55><0x05><0x21><0x00><0x15><0x01><0x19>
Receive   : <0x55><0x02><0x00><0x21>
```

- 6) Issue the *<Calibrate Mode>* command.

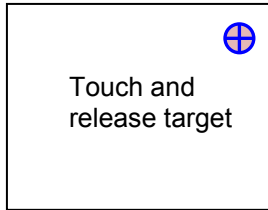
```
Send      : <0x55><0x02><0x14><0x04>
Receive   : <0x55><0x02><0x00><0x14>
```

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

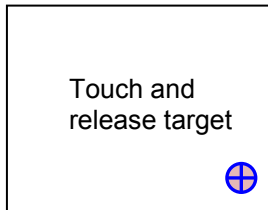


- 8) Wait for the user to touch and release the first calibration point target. Do this by looking for a controller response of: <0x55><0x02><0x00><0x14>

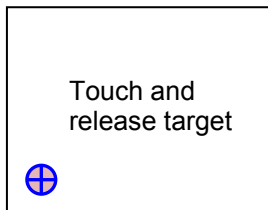
-
-
- 9) Software to display the second calibration point target in the upper right quadrant of the display and prompt the user to touch and release the target.



- 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>
- 11) Software to display the third calibration point target in the lower right quadrant of the display and prompt the user to touch and release the 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 to display the fourth calibration point target in the lower left quadrant of the display and prompt the user to touch and release the 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) Enable touch reporting by issuing <Enable Touch> command.

Send : <0x55><0x01><0x12>
Receive : <0x55><0x02><Response><0x12>

8 BASICS OF RESISTIVE SENSORS

8.1 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.

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 to the ITO on the Flex and Stable layers to the sensor's interface tail. Bus bars are typically screen printed silver ink. They are usually 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.

8.2 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 move and make electrical contact with the bottom layer.

Touch position measurements are typically made by applying a linear 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.

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 non-linearity 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

Table 10 : Sensor Comparison

8.3 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.

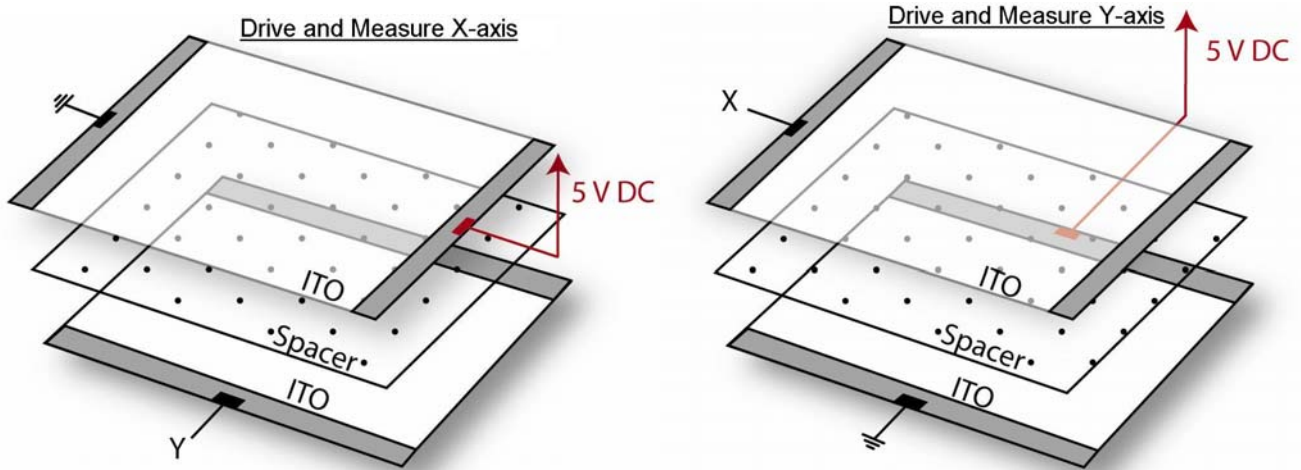


Figure 14: 4-Wire Decoding

8.4 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. An 8-wire touch sensor automatically adjusts for the changes, with the additional four reference lines. The reference lines allow the controller to know what the voltage is, at the touch sensor bus bars.

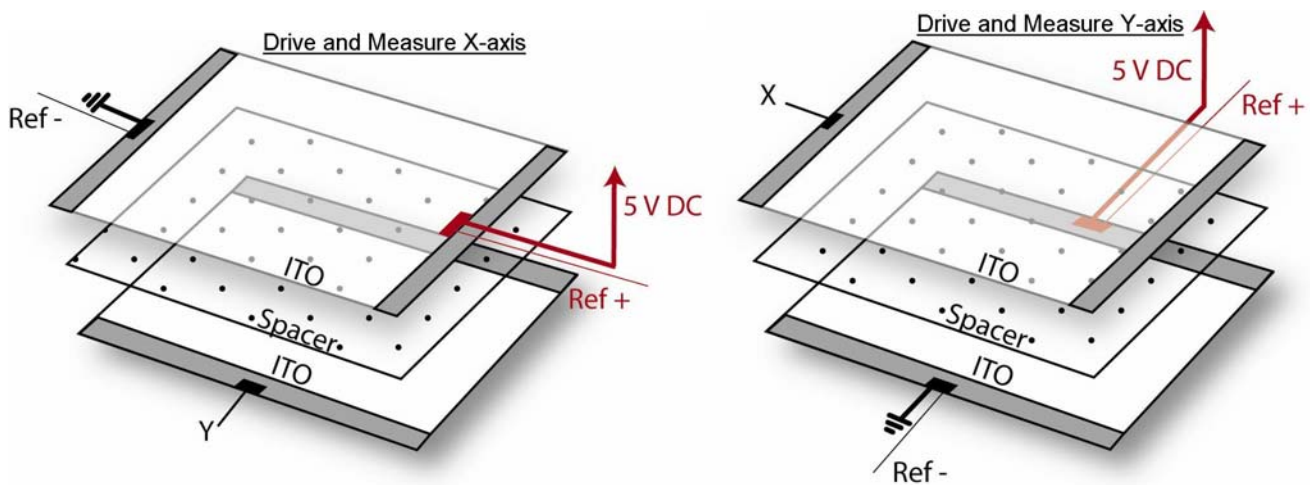


Figure 15: 8-Wire Decoding

8.5 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 stable 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 0 Volts (ground), using connections to the upper left and lower left sensor corners. The right edge is driven with +5Vdc, 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 0 Volts (ground), using connections to the upper left and upper right sensor corners. The bottom edge is driven with +5Vdc, 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.

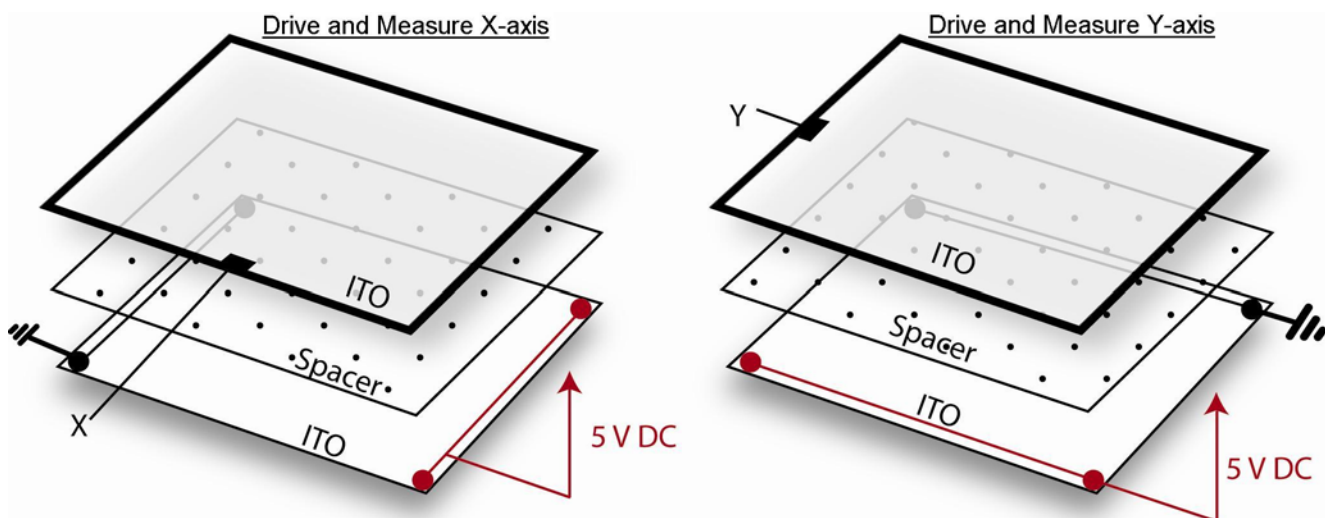


Figure 16: 5-Wire Decoding

8.6 Summary

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.

9 ELECTRICAL SPECIFICATIONS

9.1 Absolute Maximum Ratings(†)

Ambient temperature under bias.....	-40° to +125°C
Storage temperature	-65°C to +150°C
Voltage on VDD with respect to VSS	-0.3V to +6.5V
Voltage on all other pins with respect to VSS	-0.3V to (VDD + 0.3V)
Total power dissipation	800 mW
Maximum current out of VSS pin	300 mA
Maximum current into VDD pin	250 mA
Input clamp current (VI < 0 or VI > VDD).....	± 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.



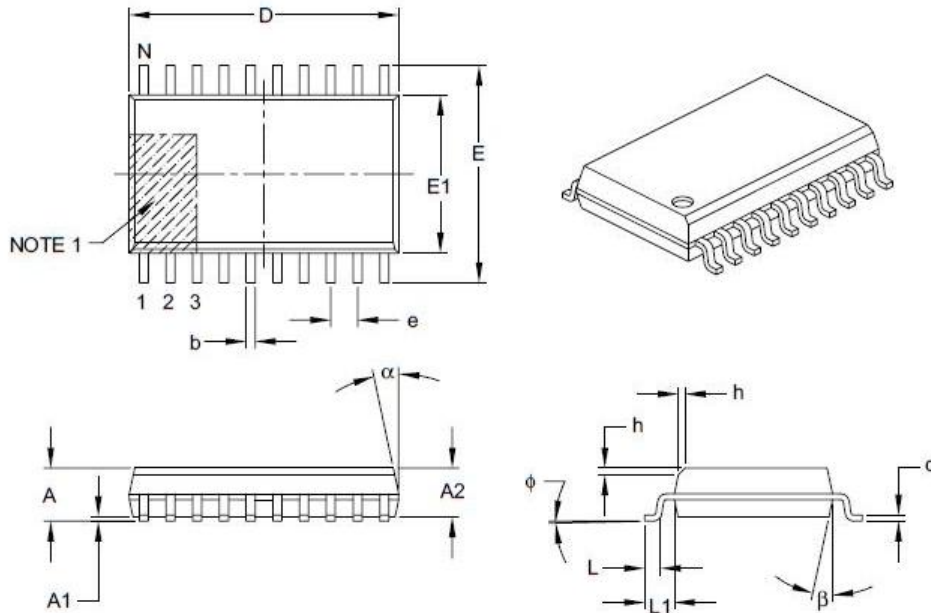
This device is sensitive to ESD damage and must be handled appropriately. Failure to properly handle and protect the device in an application may cause partial to complete failure of the device.

10 PACKAGING

10.1 20 Lead SOIC

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>



Dimension Limits	Units	MILLIMETERS		
		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		
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:

- Pin 1 visual index feature may vary, but must be located within the hatched area.
- § Significant Characteristic.
- Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.15 mm per side.
- 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.

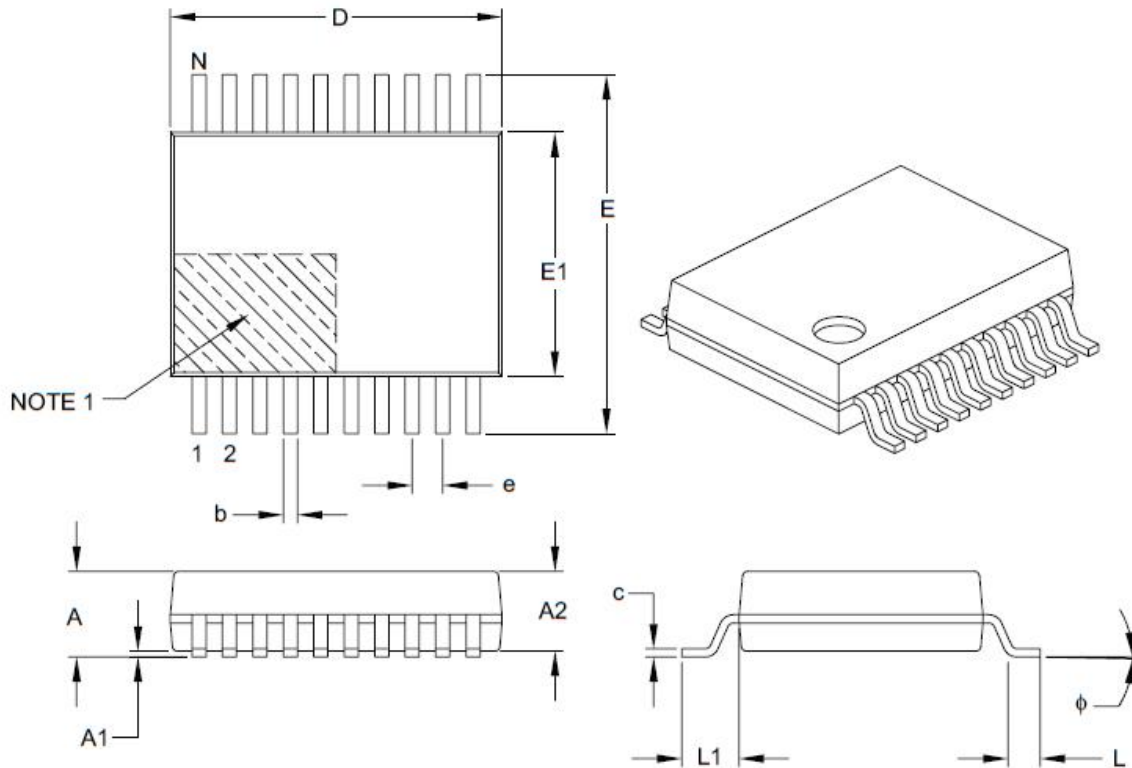
Microchip Technology Drawing C04-094B

Figure 17 : 20 Lead SOIC Package

10.2 20 Lead SSOP

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>



Dimension Limits	Units	MILLIMETERS		
		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:

- Pin 1 visual index feature may vary, but must be located within the hatched area.
- Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.20 mm per side.
- 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

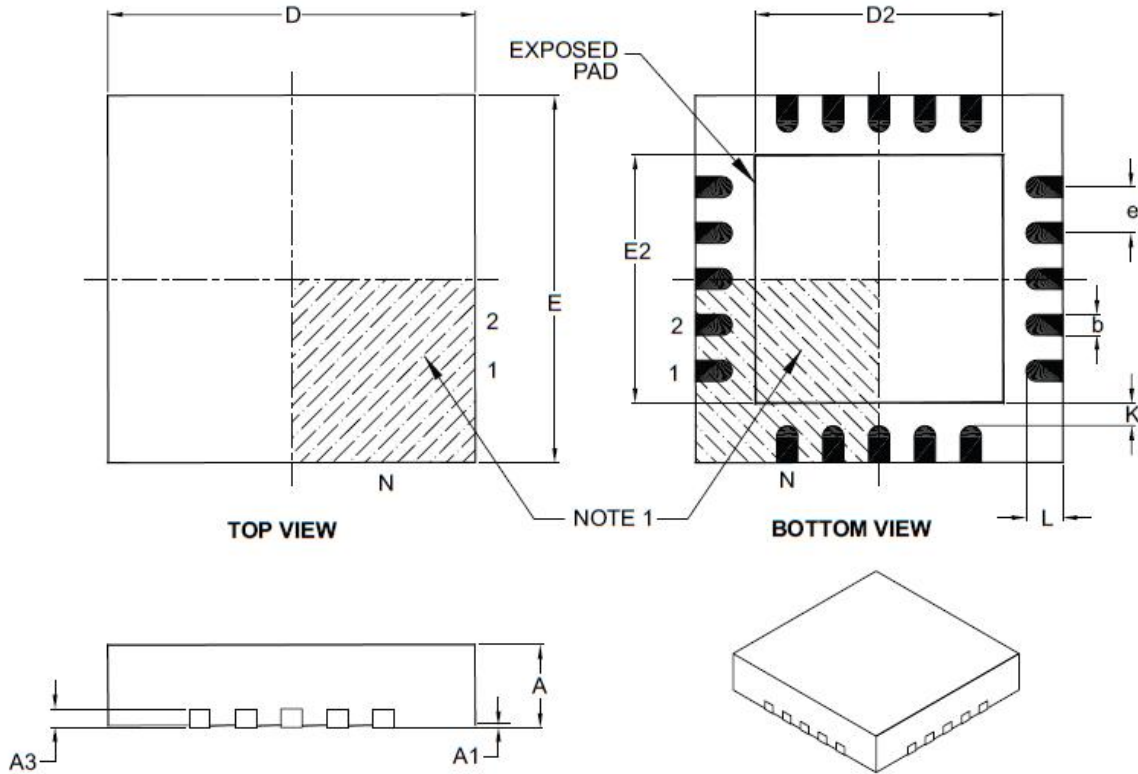
Figure 18 : 20 Lead SSOP Package

10.3 20 Lead QFN (ML)

The metal plate on the package bottom can be floated, but connecting it to VSS (circuit ground) is recommended.

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>



Dimension Limits	Units	MILLIMETERS		
		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:

- Pin 1 visual index feature may vary, but must be located within the hatched area.
- Package is saw singulated.
- 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

Figure 19 : 20 Lead QFN Package



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://support.microchip.com>
Web Address:
www.microchip.com

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

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
Farmington Hills, MI
Tel: 248-538-2250
Fax: 248-538-2260

Kokomo
Kokomo, IN
Tel: 765-864-8360
Fax: 765-864-8387

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

Santa Clara
Santa Clara, CA
Tel: 408-961-6444
Fax: 408-961-6445

Toronto
Mississauga, Ontario,
Canada
Tel: 905-673-0699
Fax: 905-673-6509

ASIA/PACIFIC

Asia Pacific Office
Suites 3707-14, 37th Floor
Tower 6, The Gateway
Harbour City, Kowloon
Hong Kong
Tel: 852-2401-1200
Fax: 852-2401-3431

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

China - Beijing
Tel: 86-10-8528-2100
Fax: 86-10-8528-2104

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

China - Hong Kong SAR
Tel: 852-2401-1200
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: 86-755-8203-2660
Fax: 86-755-8203-1760

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

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

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

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

ASIA/PACIFIC

India - Bangalore
Tel: 91-80-3090-4444
Fax: 91-80-3090-4080

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

India - Pune
Tel: 91-20-2566-1512
Fax: 91-20-2566-1513

Japan - Yokohama
Tel: 81-45-471-6166
Fax: 81-45-471-6122

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: 886-3-6578-300
Fax: 886-3-6578-370

Taiwan - Kaohsiung
Tel: 886-7-536-4818
Fax: 886-7-536-4803

Taiwan - Taipei
Tel: 886-2-2500-6610
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 - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

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

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

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

UK - Wokingham
Tel: 44-118-921-5869
Fax: 44-118-921-5820

Mouser Electronics

Authorized Distributor

Click to View Pricing, Inventory, Delivery & Lifecycle Information:

Microchip:

[DV102011](#) [AR1010-I/ML](#) [AR1010-I/SO](#) [AR1010-I/SS](#) [AR1020-I/ML](#) [AR1020-I/SO](#) [AR1020-I/SS](#) [AR1010T-I/ML](#)
[AR1010T-I/SO](#) [AR1010T-I/SS](#) [AR1020T-I/ML](#) [AR1020T-I/SO](#) [AR1020T-I/SS](#)