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

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

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

Contact us

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

AR1100 Resistive USB and RS-232 Touch Screen Controller

Data Sheet

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.

Trademarks

The Microchip name and logo, the Microchip logo, dsPIC, KEELOQ, KEELOQ logo, MPLAB, PIC, PICmicro, PICSTART, PIC^{32} logo, 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, chipKIT, chipKIT logo, CodeGuard, dsPICDEM, dsPICDEM.net, dsPICworks, dsSPEAK, ECAN, ECONOMONITOR, FanSense, HI-TIDE, In-Circuit Serial Programming, ICSP, Mindi, MiWi, MPASM, MPLAB Certified logo, MPLIB, MPLINK, mTouch, Omniscient Code Generation, PICC, PICC-18, PICDEM, PICDEM.net, PICkit, PICtail, REAL ICE, rfLAB, Select Mode, Total Endurance, TSHARC, UniWinDriver, 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.

Printed on recycled paper.

ISBN: 978-1-61341-833-8

QUALITY MANAGEMENT SYSTEM CERTIFIED BY DNV

ISO/TS 16949:2009

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.

AR1100 Resistive USB and RS-232 Touch Screen Controller

Special Features:

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

Power Requirements:

- Operating Voltage: 3.3-5.0V +/- 5%
- Standby Current:
  - <10 uA (UART)
  - <325 uA (USB)
- Operating Current:
  - <17 mA (no touch)
  - <25 mA (touch) (see **Note** below)

Note:	Results vary slightly with sensor.

Touch Modes:

- Off, Stream, Down and Up

Touch Sensor Support:

- 4-wire, 5-wire and 8-wire Analog Resistive
- Lead-to-Lead Resistance: 50-2000 Ohm
- Layer-to-Layer Capacitance: 0-0.5 uF

Touch Resolution:

- 10-bit Resolution (maximum)

Touch Coordinate Report Rate:

- 150 Reports Per Second (typ.) (see **Note** below)

Note:	Actual report rate is dynamically/automatically maximized according to the electrical characteristics of the sensor in use.

Communication:

- Automatic Detection/Selection
- UART, 9600 BAUD
- USB V2.0 Compliant, Full Speed
  - HID-GENERIC
  - HID-MOUSE
  - HID-DIGITIZER

Package Types

The device will be offered in the following packages:

- 20-Lead QFN (5 x 5 mm)
- 20-Lead SOIC
- 20-Lead SSOP

Table of Contents

TO OUR VALUED CUSTOMERS

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

If you have any questions or comments regarding this publication, please contact the Marketing Communications Department via E-mail at **docerrors@microchip.com** or fax the **Reader Response Form** in the back of this data sheet to (480) 792-4150. We welcome your feedback.

Most Current Data Sheet

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

http://www.microchip.com

You can determine the version of a data sheet by examining its literature number found on the bottom outside corner of any page. The last character of the literature number is the version number, (e.g., DS30000A is version A of document DS30000).

Errata

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

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

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

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

Customer Notification System

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

1.0 OVERVIEW

The Microchip mTouch™ AR1100 Analog Resistive USB and RS-232 Touch Screen Controller represents a feature-rich, fully-integrated universal touch screen controller solution. The AR1100 automatically selects between USB and RS-232 communication protocols, as well as supports 4, 5 or 8-wire analog resistive touch screens from any of a variety of touch screen manufacturers. The AR1100 dynamically adapts to the various touch screen electrical characteristics such as sensitivity, contact resistance, and capacitance to provide optimal performance with minimal design time.

Building on the AR1000 series, the new AR1100 offers customers an easy-to-integrate solution for low-cost, high-performing resistive touch with the advantages of USB plug and play, support for USB mouse or digitizer, advanced touch response and accuracy, field flash updatability, and free drivers for most operating systems to enable low risk designs for a wide variety of touch sensing requirements.

The AR1100 supports large displays like industrial controls, self-service kiosks, and POS terminals, as well as smaller tablet displays, handheld consumer devices, and medical devices.

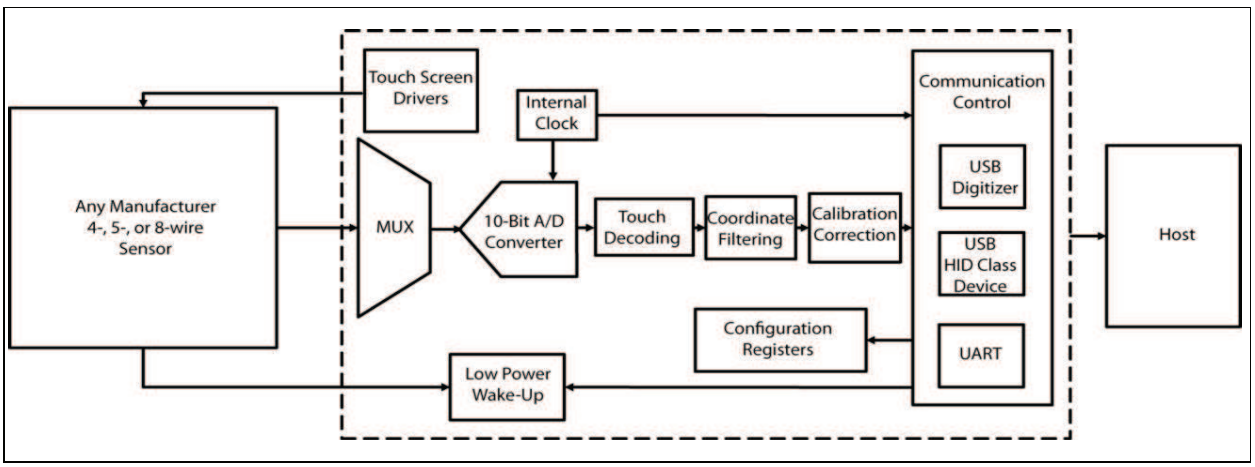
Resistive touch provides the advantages of easy integration, low total system cost and acceptance of finger, glove or stylus input, and USB communication is the industry standard for attaching peripherals to a computer. The AR1100 is an easy-to-integrate touch screen controller that meets all of these needs in a single-chip solution or production ready-board product. The device comes with free drivers for most major operating systems, making it easy for designers to quickly create low-risk touch interface solutions.

1.1 Applications

The AR1100 is suitable for any application that requires fast, accurate and reliable integration of touch – 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



NOTES:

2.0 IMPLEMENTATION – QUICK START

The AR1100 is designed to be a fully-functioning touch controller on power-up – no configuration is necessary and only minimal hardware support is needed to create a universal controller board (refer to simplified schematic).

The hard-coded defaults for the operational parameters are suitable for all but the most unique circumstances. A jumper on the MODE pin easily selects the sensor type (5-Wire or 4/8-Wire) and the Communication mode (USB or UART) is automatically detected and selected by the device. If USB is detected, the AR1100 automatically defaults to a HID-MOUSE, compatible with intrinsic drivers of standard operating systems. Any desired modifications to either the operating parameters or USB device type can be easily saved to internal nonvolatile memory to override the defaults.

NOTES:

3.0 HARDWARE

3.1 Pin Assignments

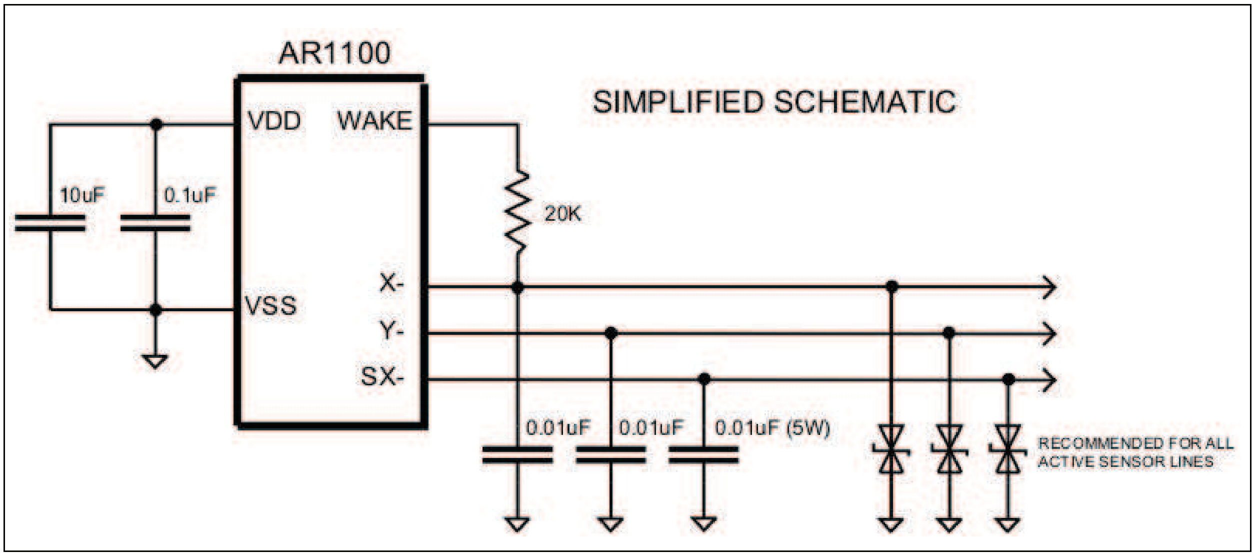
TABLE 3-1: PIN ASSIGNMENTS

Pin		Function	Description/Comments
SSOP, SOIC	QFN		
1	18	VDD	Power
2	19	OSC1	Oscillator
3	20	OSC2	Oscillator
4	1	MODE	GND: 5-Wire Open: 4-/8-Wire
5	2	LED	Led control
6	3	(Y+)	Sensor connection 4W: n/a 5W: n/a 8W: Y+
7	4	X+	Sensor connection
8	5	SY-	Sensor connection
9	6	SX+	Sensor connection
10	7	UART-TX	UART Transmit Data
11	8	WAKE	Wake pin
12	9	UART-RX	UART Receive Data
13	10	SX-	Sensor connection 4W: n/a 5W: WSX- 8W: SX-
14	11	X-	Sensor connection
15	12	Y+(SY+)	Sensor connection 4W: Y+ 5W: Y+ 8W: SY+
16	13	Y-	Sensor connection
17	14	VUSB	USB Internal Voltage Reference
18	15	USB-D-	USB data I/O
19	16	USB-D+	USB data I/O
20	17	VSS	GND

3.2 Schematic

A simplified schematic is provided below. A detailed schematic and BOM is given in **FIGURE B-1: "Schematic"** and **FIGURE B-2: "Bill of Materials"**.

FIGURE 3-1: SIMPLIFIED SCHEMATIC

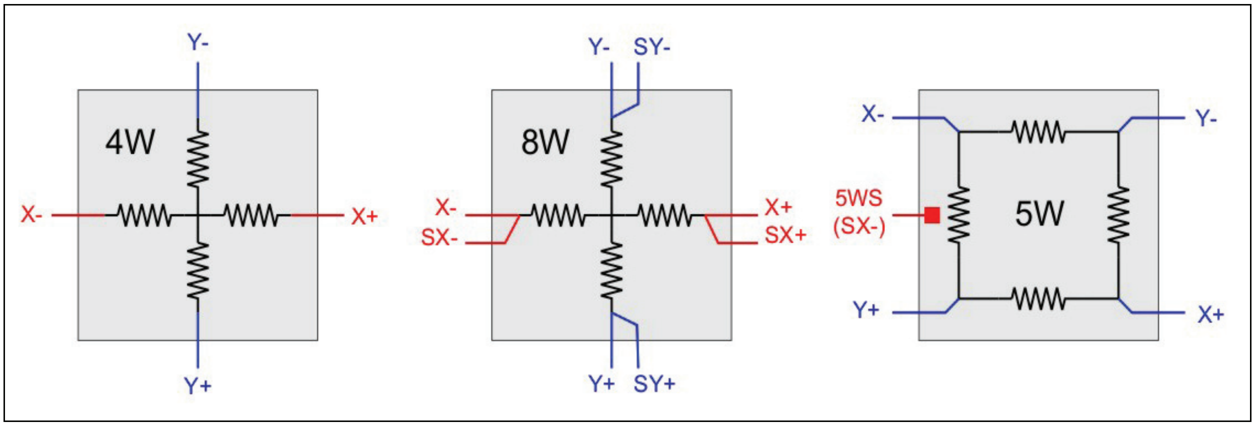


Note:	Unused SENSOR pins should be grounded.

3.3 Sensor Attachment

AR1100 connections to the various sensor types are described graphically in Figure 3-2.

FIGURE 3-2: SENSOR EQUIVALENT CIRCUIT MODELS



3.4 ESD Considerations

Suggested ESD protection is shown on the reference schematic (See **FIGURE B-1: "Schematic"**). Additional/alternate ESD countermeasures may be employed to meet application-specific requirements. Test to ensure the selected ESD protection does not degrade touch performance.

> **Note:** ESD protection diodes are recommended for all active sensor lines but care should be taken to minimize capacitance. As an example, PESD5V0S1BA is recommended and used on reference designs due to its nominally-low 35 pF.

3.5 Noise Considerations

Touch sensor filtering capacitors are included in the reference design schematic (See **FIGURE B-1: "Schematic"**).

> **Note:** Changing the value of the sensor filter capacitors may adversely affect touch performance.

NOTES:

4.0 COMMUNICATION

4.1 Physical

The AR1100 supports UART and USB communication and will automatically detect the active mode between the two. Additionally, USB mode will enumerate as one of three ‘devices’. The default USB device type is defined (and can be changed) by the configuration command and is saved in nonvolatile memory. The factory default is HID-MOUSE.

1. UART/Serial
2. HID-GENERIC
3. HID-MOUSE
4. HID-DIGITIZER

4.1.1 MODE DETECT/SELECT

To support auto-detection – the firmware and hardware resources for UART and USB are both functional at power-up until the active mode is determined by one of the following events.

1. USB successfully enumerates – result: USB is active
2. Valid communication is received via UART – result: UART is active
3. The Sleep timer has expired and USB has not yet enumerated – result: UART is active (by default)

Note:	Immediately after Reset, the AR1100 will not attempt to ‘Sleep’ for at least 45 seconds to allow time for USB enumeration – no matter the setting of the Sleep timer parameter.

Once the active communication mode is determined, the ‘inactive’ mode is decommissioned to minimize power. The active communication mode will remain in force until the AR1100 is reset.

4.1.2 UART MODE

In UART mode, the AR1100 supports a simple, 2-wire (transmit/receive) asynchronous serial communication. The device does not support hardware handshaking but does employ a data protocol handshake described in the device command section. The host should be configured for 9600 BAUD, 8 data bits and 1 Stop bit.

4.1.3 USB MODE

The USB can enumerate as one of three ‘devices’ (or device types) identified by a byte in EEPROM. See Table 4-1.

TABLE 4-1: USB IDs

NAME	DESC	CLASS	VENDOR ID	PRODUCT ID	SPEED
HID-GENERIC	Proprietary (AR1000-style)	HID	x04D8	x0C01	FULL
HID-MOUSE	Mouse, absolute coordinates 0-4095	HID	x04D8	x0C02	FULL
HID-DIGITIZER	Single-input digitizer	HID	x04D8	x0C03	FULL

The HID-MOUSE and HID-DIGITIZER types are recognized by many host operating systems and will provide cursor movement with a touch.

The HID-GENERIC type is a proprietary style, which would require a custom software driver to support.

The controller defaults to the HID-MOUSE device type, unless it is commanded to enumerate as one of the other supported types.

Once enumerated, the USB device can be signaled/commanded to re-enumerate as the same device or to one of the other two. In processing the command, the AR1100 writes the desired USB device type to EEPROM prior to detaching from the bus and executing a Reset. The `SET_FEATURE` control transfer or a WRITE standard data transfer (via the Interrupt end point) is used to convey the command (described in Section 5.0, Commands).

Note:	The HID-MOUSE requires `SET_FEATURE` and does not support a data WRITE.

4.2 Data Protocol

Data protocol utilizes multi-byte packet transfers in two categories/formats:

1. Touch reports
2. Command packets

4.2.1 TOUCH REPORTS

Touch reports always originate from the AR1100 and are transmitted in response to touch detection. The format of the touch report is mode-dependent.

The measurement resolution for touch coordinates is 10-bit. The measured values are shifted (multiplied by 4) and reported in a 12-bit format. In the reporting protocol, the Least Significant coordinate bits X1:X0 and Y1:Y0 will be zeros. The resulting full-scale range for reported touch coordinates is 0 to 4095.

4.2.1.1 Mode: UART, HID-GENERIC

The ‘standard’, 5-byte touch report is formatted as in Table 4-2:

TABLE 4-2: TOUCH REPORT FORMAT – GENERIC

BYTE	BIT 7	6	5	4	3	2	1	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

P	Pen state - 1: Pen down - 0: Pen up
R	(Reserved)
X	X ordinate of touch location (12 bits)
Y	Y ordinate of touch location (12 bits)

Up to three touch reports are sent in response to each touch ‘event’ (events are defined as: pen down, pen up and pen move). A behavior is defined per event by the ‘Touch mode’ configuration parameter and described in Table 4-3 below.

TABLE 4-3: TOUCH MODE OPTIONS – GENERIC

MODE	SUPPORTED EVENT PD	PU	PM	BEHAVIOR
0	X	X	X	NO REPORT
1	X	X	X	REPORT (P=0)
2	X	X	X	REPORT (P=1)
3	X	X	X	REPORT (P=1), REPORT (P=0)
4	X	X		REPORT (P=0), REPORT (P=1), REPORT (P=0)
5	X	X		REPORT (P=0), REPORT (P=1)

PD	Pen down
PU	Pen up
PM	Pen move
Report	Touch report

4.2.1.2 Mode: HID-MOUSE

Touch report format:

TABLE 4-4: TOUCH REPORT FORMAT – MOUSE

BYTE	BIT							
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
1	0	0	0	0	0	B3	B2	B1
2	X7	X6	X5	X4	X3	X2	X1	X0
3	0	0	0	0	X11	X10	X9	X8
4	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0
5	0	0	0	0	Y11	Y10	Y9	Y8

B1	Button 1 depressed
B2	Button 2 depressed
B3	Button 3 depressed
X	X ordinate of touch location (12 bits)
Y	X ordinate of touch location (12 bits)

4.2.1.3 Mode: HID-DIGITIZER

Touch report format:

TABLE 4-5: TOUCH REPORT FORMAT – DIGITIZER

BYTE	BIT							
	7	6	5	4	3	2	1	0
1	0	0	0	0	0	0	P	T
2	0	0	0	0	0	0	0	0
3	X7	X6	X5	X4	X3	X2	X1	X0
4	0	0	0	0	X11	X10	X9	X8
5	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0
6	0	0	0	0	Y11	Y10	Y9	Y8

T	Tip switch
P	Proximity (in range) – always 1
X	X ordinate of touch location (12 bits)
Y	X ordinate of touch location (12 bits)

For flexibility, the value and behavior of the 'tip switch' data entity ("T") and touch reporting react to and is defined by the 'Touch mode' parameter (similar to 'pen state' bit in HID-GENERIC or UART).

TABLE 4-6: TOUCH MODE OPTIONS – DIGITIZER

MODE	SUPPORTED EVENT			BEHAVIOR
	PD	PU	PM	
0	X	X	X	NO REPORT
1	X	X	X	REPORT (T=0)
2	X	X	X	REPORT (T=1)
3	X	X	X	REPORT (T=1), REPORT (T=0)
4	X	X		REPORT (T=0), REPORT (T=1), REPORT (T=0)
5	X	X		REPORT (T=0), REPORT (T=1)

PD	Pen down
PU	Pen up
PM	Pen move
Report	Touch report

4.2.2 COMMAND PACKETS

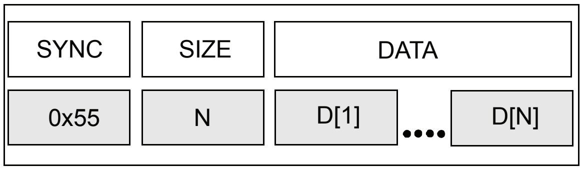
PACKETs are used for all communications, other than touch reports (i.e., configuration/control). COMMAND packets (issued by the host) and RESPONSE packets (issued by the device) have identical framework but differ slightly in format, as described below. In standard operation, communication is initiated by the host then acknowledged by the device. In some diagnostic scenarios (not discussed here) – a COMMAND packet does not necessarily dictate a response from the device and, in other cases, a RESPONSE packet may be issued by the device unsolicited.

4.2.2.1 Construction

GENERAL

The generic framework for all packets (Figure 4-1) is comprised of a SYNC byte, a SIZE byte and a DATA section. The DATA section has a maximum size of 255 total bytes.

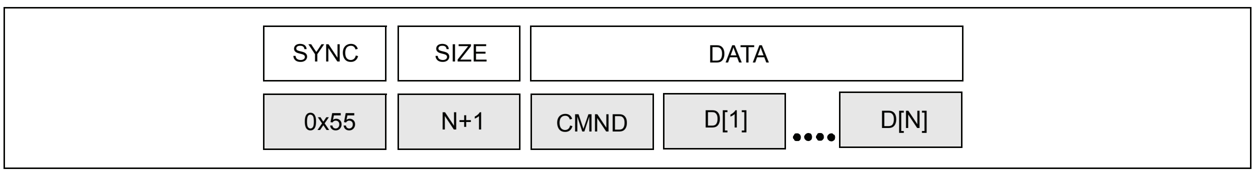
FIGURE 4-1: PACKET FORMAT – GENERAL



COMMAND

A COMMAND packet has a minimum of 3 bytes defined as SYNC, SIZE and CMND. The DATA section is command-dependant and can include up to 254 associated data bytes (D[1] – D[N]). See Figure 4-2.

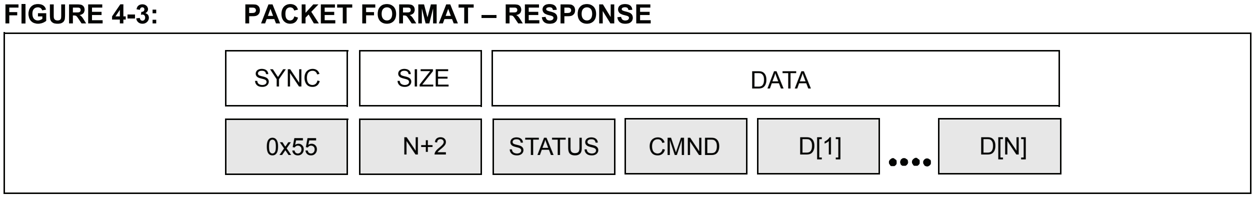
FIGURE 4-2: PACKET FORMAT – COMMAND



RESPONSE

A RESPONSE packet has a minimum of 4 bytes defined as SYNC, SIZE, STATUS and CMND. As with the COMMAND packet, the RESPONSE packet is command-dependant and can include up to 253 associated data bytes. In cases where the RESPONSE packet is in direct response to a COMMAND packet, the CMND byte is the same in both.

FIGURE 4-3: PACKET FORMAT – RESPONSE



4.2.2.2 Mode: UART

In UART communication mode a complete COMMAND packet must be delivered before the packet 'time-out' timer expires (~250 ms). A packet 'time out' will cause any partial packet to be discarded and the packet parsing state controller to reset. Using this mechanism, the host can always ensure (re-establish) 'SYNC' by pausing for 250+ ms before sending another packet.

4.2.2.3 Mode: USB

Typically, a packet arrives (and is delivered) in a single, 64-byte (max.) USB buffer. Theoretically, a COMMAND packet can span multiple, physical USB buffers but will be subject to the packet time-out criteria described in the UART section. A RESPONSE packet will always be delivered in a single buffer and a buffer will never contain more than one.

4.2.2.4 Mode Capabilities

Only two of the four AR1100 communication modes, UART and HID-GENERIC, support 'low-level' operations such as:

1. Configuration
2. Calibration
3. Boot loading (field re-programming)

The remaining two communication modes, HID-MOUSE and HID-DIGITIZER, only support output of TOUCH REPORT(s) and only receive (respond to) mode change command(s). They can be supported by intrinsic operating system driver(s). To configure, calibrate and/or reprogram these two devices, the host must cause them to re-enumerate as HID-GENERIC. Once the low-level operations are complete, the host can then re-configure back to the desired device.

NOTES:

5.0 COMMANDS

In normal operation, the AR1100 automatically returns 'touch reports' in response to a touch – no 'prompting' is required from the host. The following command set can be used to configure the parameters used to 'fine-tune' the operation. To prevent touch reports from interfering with these commands, it is recommended that `TOUCH_DISABLE` be executed prior to any other command(s) and the `TOUCH_ENABLE` command be used as the last step to return the AR1100 to normal operation.

5.1 Summary

Table 5-1 summarizes the standard command set.

TABLE 5-1: COMMAND SUMMARY

CMND	NAME	DESCRIPTION
0x12	`TOUCH_ENABLE`	Enable TOUCH reporting
0x13	`TOUCH_DISABLE`	Disable TOUCH reporting
0x14	`CALIBRATE`	Execute Calibrate routine
0x20	`REG_READ`	Read register(s)
0x21	`REG_WRITE`	Write register(s)
0x28	`EE_READ`	Read EE location(s)
0x29	`EE_WRITE`	Write EE location(s)
0x2B	`EE_READ_PARAMS`	Read parameter set (from EE to RAM)
0x23	`EE_WRITE_PARAMS`	Write parameter set (from RAM to EE)

Table 5-2 summarizes the STATUS byte in the RESPONSE packet.

TABLE 5-2: STATUS SUMMARY

STATUS	NAME	DESCRIPTION
0x00	`OK`	No error
0x01	`UNRECOGNIZED`	Unrecognized command
0x04	`TIMEOUT`	Packet time out
0x05	`EEPARAMS_ERR`	Error reading EEPROM parameters
0xFC	`CAL_CANCEL`	Calibration sequence cancelled

5.2 Command: TOUCH_ENABLE

Enable touch reporting.

COMMAND PACKET:

TABLE 5-3: COMMAND: TOUCH_ENABLE

BYTE#	VALUE	DESCRIPTION
1	0x55	SYNC
2	0x01	SIZE
3	0x12	COMMAND

RESPONSE PACKET:

TABLE 5-4: RESPONSE: TOUCH_ENABLE

BYTE#	VALUE	DESCRIPTION
1	0x55	SYNC
2	0x02	SIZE
3	STATUS	STATUS
4	0x12	COMMAND

5.3 Command: TOUCH_DISABLE

Disable touch reporting.

COMMAND PACKET:

TABLE 5-5: COMMAND: TOUCH_DISABLE

BYTE#	VALUE	DESCRIPTION
1	0x55	SYNC
2	0x01	SIZE
3	0x13	COMMAND

RESPONSE PACKET:

TABLE 5-6: RESPONSE: TOUCH_DISABLE

BYTE#	VALUE	DESCRIPTION
1	0x55	SYNC
2	0x02	SIZE
3	STATUS	STATUS
4	0x13	COMMAND

5.4 Command: CALIBRATE

The CALIBRATE command initiates the controller-based calibration sequence. A RESPONSE packet is returned for each calibration point touched.

COMMAND PACKET

TABLE 5-7: COMMAND: CALIBRATE

BYTE#	VALUE	DESCRIPTION
1	0x55	SYNC
2	0x02	BYTE COUNT
3	0x14	COMMAND
4	TYPE	0x01: 4-Point – Full interpolation 0x02: 9-Point 0x03: 25-Point 0x04: 4-Point (AR1000 style)

RESPONSE PACKET (for each calibration point touched, then released):

TABLE 5-8: RESPONSE: CALIBRATE

BYTE#	VALUE	DESCRIPTION
1	0x55	SYNC
2	0x02	SIZE
3	STATUS	STATUS
4	0x14	COMMAND

5.5 Command: REG_READ

Read one or more operational registers.

COMMAND PACKET:

TABLE 5-9: COMMAND: REG_READ

BYTE#	VALUE	DESCRIPTION
1	0x55	SYNC
2	0x04	SIZE
3	0x20	COMMAND
4	0x00	REGISTER ADDRESS (MSB)
5	ADR	REGISTER ADDRESS (LSB)
6	N	NUMBER OF BYTES TO READ

RESPONSE PACKET:

TABLE 5-10: RESPONSE: REG-READ

BYTE#	VALUE	DESCRIPTION
1	0x55	SYNC
2	N+2	SIZE
3	STATUS	STATUS
4	0x20	COMMAND
5	REG[ADR]	REGISTER VALUE
...	...	REGISTER VALUE(S)
4+N	REG[ADR+N-1]	REGISTER VALUE

5.6 Command: `REG_WRITE`

Write one or more operational registers.

COMMAND PACKET:

TABLE 5-11: COMMAND: `REG_WRITE`

BYTE#	VALUE	DESCRIPTION
1	0x55	SYNC
2	N+2	SIZE (N = # of REGS to WRITE)
3	0x21	COMMAND
4	0x00	REGISTER ADDRESS (MSB)
5	ADR	REGISTER ADDRESS (LSB)
6	REG[ADR]	REGISTER VALUE
...	...	REGISTER VALUE(S)
5+N	REG[ADR+N-1]	REGISTER VALUE

RESPONSE PACKET:

TABLE 5-12: RESPONSE: `REG_WRITE`

BYTE#	VALUE	DESCRIPTION
1	0x55	SYNC
2	0x02	SIZE
3	STATUS	STATUS
4	0x21	COMMAND

5.7 Command: `EE_READ`

Read one or more bytes from EEPROM.

COMMAND PACKET:

TABLE 5-13: COMMAND: `EE_READ`

BYTE#	VALUE	DESCRIPTION
1	0x55	SYNC
2	0x04	SIZE
3	0x28	COMMAND
4	0x00	EE ADDRESS (MSB)
5	ADR	EE ADDRESS (LSB)
6	N	Number of bytes to read

RESPONSE PACKET:

TABLE 5-14: RESPONSE: `EE_READ`

BYTE#	VALUE	DESCRIPTION
1	0x55	SYNC
2	N+2	SIZE (N = # of BYTES to WRITE)
3	STATUS	STATUS
4	0x28	COMMAND
5	EE[ADR]	EE VALUE
...	...	EE VALUE VALUE(S)
4+N	EE[ADR+N-1]	EE VALUE

5.8 Command: `EE_WRITE`

Write one or more bytes to EEPROM.

COMMAND PACKET:

TABLE 5-15: COMMAND: `EE_WRITE`

BYTE#	VALUE	DESCRIPTION
1	0x55	SYNC
2	4+N	SIZE (N = # of REGS to WRITE)
3	0x29	COMMAND
4	0x00	EE ADDRESS (MSB)
5	ADR	EE ADDRESS (LSB)
6	EE[ADR]	EE VALUE
...	...	EE VALUE(S)
5+N	EE[ADR+N-1]	EE VALUE

RESPONSE PACKET:

TABLE 5-16: RESPONSE: `EE_WRITE`

BYTE#	VALUE	DESCRIPTION
1	0x55	SYNC
2	2	SIZE
3	STATUS	STATUS
4	0x29	COMMAND

5.9 Command: EE_READ_PARAMS

Read entire set of operational parameters from EEPROM to RAM.

COMMAND PACKET:

TABLE 5-17: COMMAND: EE_READ_PARAMS

BYTE#	VALUE	DESCRIPTION
1	0x55	SYNC
2	0x01	SIZE
3	0x23	COMMAND

RESPONSE PACKET:

TABLE 5-18: RESPONSE: EE_READ_PARAMS

BYTE#	VALUE	DESCRIPTION
1	0x55	SYNC
2	2	SIZE
3	STATUS	STATUS
4	0x23	COMMAND

5.10 Command: EE_WRITE_PARAMS

Write entire set of operational parameters to EEPROM from RAM.

COMMAND PACKET:

TABLE 5-19: COMMAND: EE_WRITE_PARAMS

BYTE#	VALUE	DESCRIPTION
1	0x55	SYNC
2	0x01	SIZE
3	0x2B	COMMAND

RESPONSE PACKET:

TABLE 5-20: RESPONSE: EE_WRITE_PARAMS

BYTE#	VALUE	DESCRIPTION
1	0x55	SYNC
2	2	SIZE
3	STATUS	STATUS
4	0x2B	COMMAND

5.11 Command: USB_MODE_GENERIC

Set default USB mode (device type) to “HID-GENERIC”. Mode HID-GENERIC is required for low-level configuration commands, calibration and boot load operations. No RESPONSE packet is returned.

COMMAND PACKET:

TABLE 5-21: COMMAND: USB_MODE_GENERIC

BYTE#	VALUE	DESCRIPTION
1	0x55	SYNC
2	0x01	SIZE
3	0x70	COMMAND

RESPONSE PACKET:

None.

5.12 Command: USB_MODE_MOUSE

Set default USB mode (device type) to ‘HID-MOUSE’.

COMMAND PACKET:

TABLE 5-22: COMMAND: USB_MODE_MOUSE

BYTE#	VALUE	DESCRIPTION
1	0x55	SYNC
2	0x01	SIZE
3	0x71	COMMAND

RESPONSE PACKET:

None.

5.13 Command: USB_MODE_DIGITIZER

Set default USB mode (device type) to ‘HID-DIGITIZER’.

COMMAND PACKET:

TABLE 5-23: COMMAND: USB_MODE_DIGITIZER

BYTE#	VALUE	DESCRIPTION
1	0x55	SYNC
2	0x01	SIZE
3	0x72	COMMAND

RESPONSE PACKET:

None.

6.0 CONFIGURATION REGISTERS

TABLE 6-1: CONFIGURATION REGISTER SUMMARY

ADDR	NAME	DECRIPTION	7	6	5	4	3	2	1	0	DFLT
0x00	RisetimeCapTime-out	*170 usec	Value of: 0-255								0x18
0x01	RisetimeQuick	*10 usec	Value of: 0-255								0x02
0x02	TouchThreshold	8-bit ADC – touch_check()	Value of: 0-255								0x80
0x03	SensitivityFilter	8-bit ADC	Value of: 0-255								0x04
0x04	SamplingFast	# of ADC samples/touch to average	Value of: 1, 2, 4, 8, 16, 32, 64, 128								0x04
0x05	SamplingSlow	# of ADC samples/touch to average	Value of: 1, 2, 4, 8, 16, 32, 64, 128								0x08
0x06	AccuracyFilterFast	# of touch positions to average	Value of: 1-8								0x08
0x07	AccuracyFilter-Slow	# of touch positions to average	Value of: 1-8								0x08
0x08	SpeedThreshold	8-bit ADC (raw touch coordinates)	Value of: 0-255								0x03
0x09	DitherFilter	size of anti-dithering win-dow 1/4096 of sensor dim.	Value of: 0-255								0x00
0x0A	SleepDelay	*250 msec	Value of: 0-255								0x00
0x0B	PenUpDelay	touch process loop count	Value of: 0-255								0x02
0x0C	TouchMode	**(Note 3)**	PD2	PD1	PD0	PM1	PM0	PU2	PU1	PU0	0xB1
0x0D	TouchOptions	**(Note 1)**	TEN	VCF	—	C	DRT	—	48W	CALE	0x89
0x0E	CalibrationInset	2x % sensor dimension - units: 1/256 (e.g. 64=25%)	Value of: 1-128								0x40
0x0F	PenStateReport-Delay	*170 usec	Value of: 0-255								0x04
0x10	Reserved	—	—								0x00
0x11	TouchReportDelay	*0.5 msec	Value of: 0-255								0x00
0x12	RisetimeDefault	*21 usec	Value of: 0-255								0x80
0x13	RisetimeModifier	(value-128) * 10 usec (i.e., 128 = 0, 127 = -1, 129 = +10)	Value of: 0-255								0x80
0x14	Status	**(Note 2)**	TCHK	EEV	CALV	JMP	DRT	8W	5W	4W	0x00
0x15	Debug	**(Note 4)**	—	—	—	RT	RTC	RTR	—	—	0x00

Note 1: OPTIONS/FLAGS: TEN (Touch Enable); VCF (Verbose Cal Feedback); DRT (Dynamic Risetime Enable); 48W (0 = 4-Wire, 1 = 8-Wire); CALE (Calibration Enable);

2: STATUS (READ ONLY): TCHK (Result of Touch CHK DIAG); EEV (Params in EEPROM used); CALV (Calibration valid); JMP (State of mode Jumper); DRT (DYN RISE TIME CAP MEAS'D); 8W (Configured for 8W Sensor); 5W (Configured for 5W Sensor); 4W (Configured for 4W Sensor);

3: TOUCH MODE(S) (specified for each event: PD, PM and PU): 0: no report(s) issued; 1: report (P = `0`); 2: report (P = `1`); 3: report (P = `1`), report (P = `0`); 4: report (P = `0`), report (P = `1`); report (P = `0`), 5: report (P = `0`), report (P = `1`);

4: DEBUG FLAGS (optional): RT (Risetime Report); RTC (Risetime CAP Report); RTR (Risetime RES Report).