Features

- Chip Set Configuration
  - One master mXT1386 device
  - Three slave mXT154 devices
- maXTouch™ Touchscreen
  - True 12-bit multiple touch reporting and real-time XY tracking for up to 16 concurrent touches per touchscreen
- Number of Channels
  - Electrode grid configurations of up to 33 X and 42 Y lines supported
  - Touchscreens up to 1386 channels (subject to o
  - ther configurations)
  - Up to 64 channels can be allocated as fixed keys (subject to other configurations)
- Signal Processing
  - Advanced digital filtering using both hardware engine and firmware
  - Self-calibration
  - Auto drift compensation
  - Adjacent Key Suppression® (AKS®) technology
  - Grip suppression
  - Palm suppression
  - Reports one-touch and two-touch gestures
  - Down-scaling and clipping support to match LCD resolution
  - Ultra-fast start-up and calibration for best user experience
  - Supports axis flipping and axis switch-over for portrait and landscape modes
- Scan Speed
  - Maximum single touch 150Hz, subject to configuration
  - Configurable to allow power/speed optimization
  - Programmable timeout for automatic transition from active to idle states
- Response Times
  - Initial latency <25 ms for first touch from idle, subject to configuration
- Sensors
  - Works with PET or glass sensors, including curved profiles
  - Works with all proprietary sensor patterns recommended by Atmel®
  - Works with a passive stylus
- Panel Thickness
  - Glass up to 2.5 mm, screen size dependent
  - Plastic up to 1.2 mm, screen size dependent
- Interfaces
  - I^2C-compatible slave mode, 400 kHz
  - USB 2.0-compliant composite device, full speed (12 Mbps)
- Master Package
  - 64-pin QFN 9 x 9 x 1 mm, 0.5 mm pin pitch
- Slave Packages
  - 49-ball VFBGA 5 x 5 x 1 mm, 0.65 ball pitch
  - 48-pin QFN 6 x 6 x 0.6 mm, 0.4 mm pin pitch

maXTouch™ 1386-channel Touchscreen Controller

mXT1386

Firmware 1.x

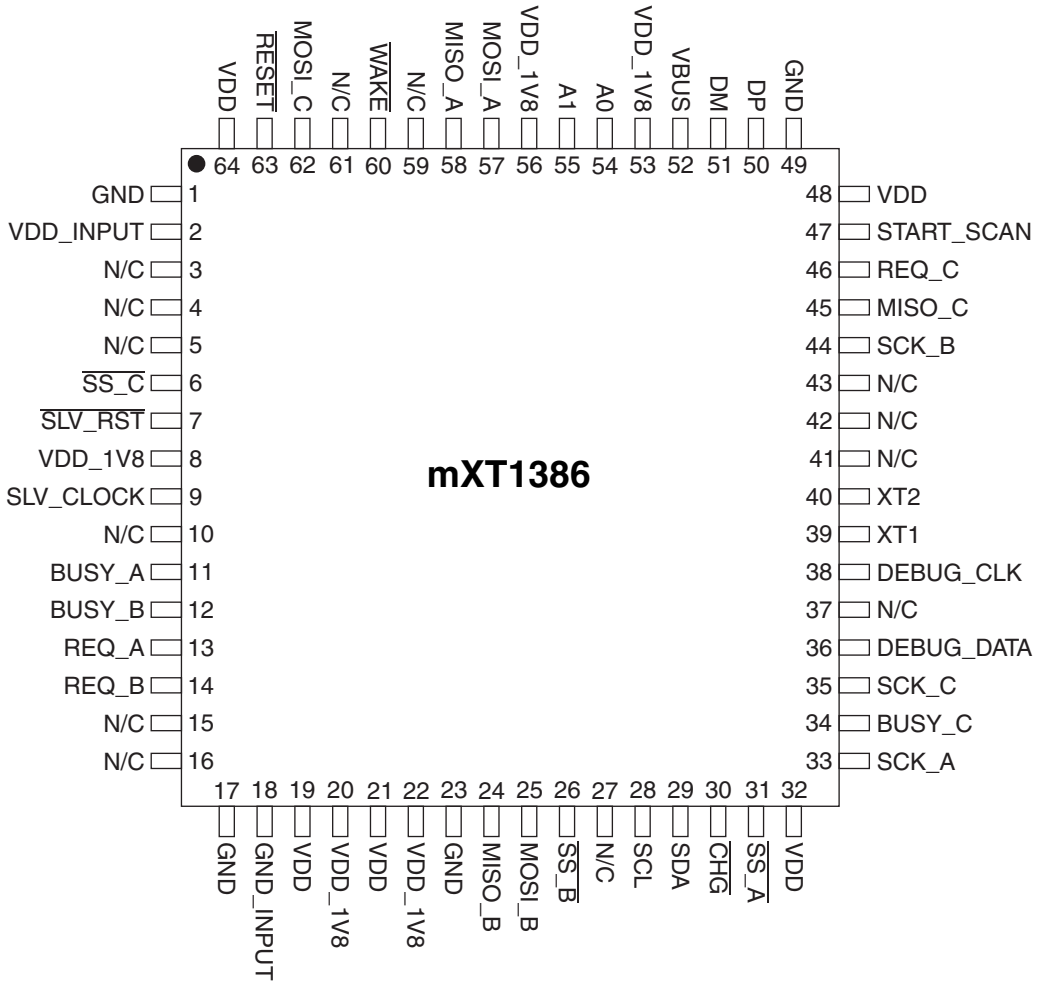
maXTouch™

9584LX–AT42–07/11

1. Pinout and Block Diagram

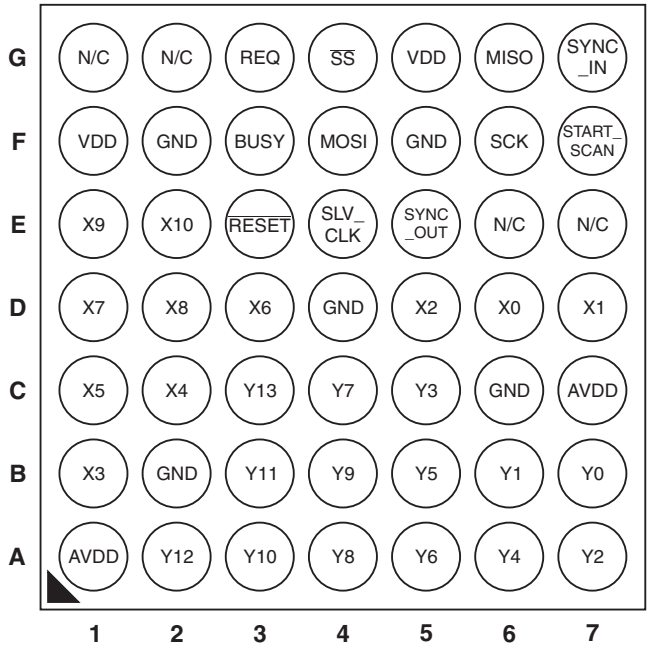
1.1 Pinout Configuration

1.1.1 Master mXT1386 – 64-pin QFN



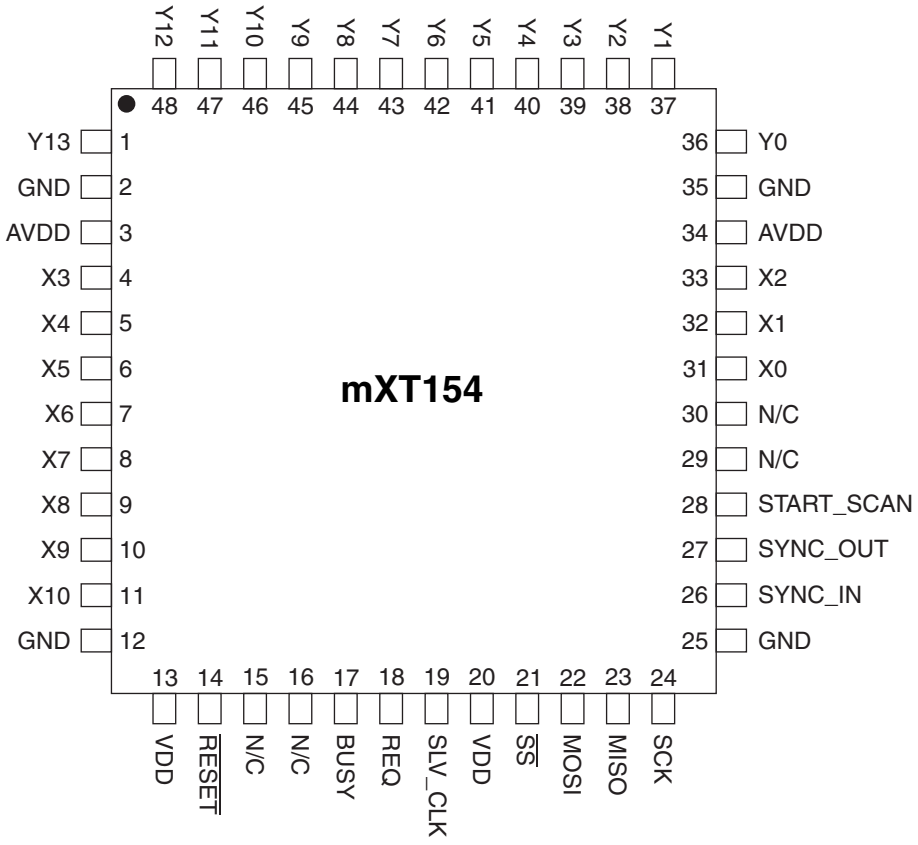
Top View

1.1.2 Slave mXT154 – 49-ball VFBGA



Bottom View

1.1.3 Slave mXT154 – 48-pin QFN



Top View

1.2 Pinout Descriptions

1.2.1 Master mXT1386 – 64-pin QFN

Table 1-1. Pin Listing

Pin	Name	Type	Comments	If Unused, Connect To...
1	GND	P	Ground	–
2	VDD_INPUT	I	Inter-chip signal; for factory use only	–
3	N/C	–	No connection	Leave open
4	N/C	–	No connection	Leave open
5	N/C	–	No connection	Leave open
6	$\overline{\text{SS\_C}}$	I	Inter-chip signal	–
7	$\overline{\text{SLV\_RST}}$	O	Inter-chip signal	–
8	VDD_1V8 (1)	P	Inter-chip signal	–
9	SLV_CLOCK	O	Inter-chip signal	–
10	N/C	–	No connection	Leave open
11	BUSY_A	I	Inter-chip signal	–
12	BUSY_B	I	Inter-chip signal	–
13	REQ_A	O	Inter-chip signal	–
14	REQ_B	O	Inter-chip signal	–
15	N/C	–	No connection	Leave open
16	N/C	–	No connection	Leave open
17	GND	P	Ground	–
18	GND_INPUT	I	Inter-chip signal; for factory use only	–
19	VDD	P	3.3V power	–
20	VDD_1V8 (1)	P	Inter-chip signal	–
21	VDD	P	3.3V power	–
22	VDD_1V8 (1)	P	Inter-chip signal	–
23	GND	P	Ground	–
24	MISO_B	O	Inter-chip signal	–
25	MOSI_B	I	Inter-chip signal	–
26	$\overline{\text{SS\_B}}$	I	Inter-chip signal	–
27	N/C	–	No connection	Leave open
28	SCL (2)	OD	Serial Interface Clock	Leave open
29	SDA (2)	OD	Serial Interface Data	Leave open
30	$\overline{\text{CHG}}$ (3)	OD	State change interrupt	Leave open
31	$\overline{\text{SS\_A}}$	I	Inter-chip signal	–
32	VDD	P	3.3V power	–
33	SCK_A	I	Inter-chip signal	–

Table 1-1. Pin Listing (Continued)

Pin	Name	Type	Comments	If Unused, Connect To...
34	BUSY_C	I	Inter-chip signal	
35	SCK_C	I	Inter-chip signal	–
36	DEBUG_DATA	O	Debug port data (4)	Leave open
37	N/C	–	No connection	Leave open
38	DEBUG_CLK	O	Debug port clock (4)	Leave open
39	XT1	I	External oscillator – 8 MHz	–
40	XT2	O	External oscillator – 8 MHz	Leave open
41	N/C	–	No connection	Leave open
42	N/C	–	No connection	Leave open
43	N/C	–	No connection	Leave open
44	SCK_B	I	Inter-chip signal	–
45	MISO_C	O	Inter-chip signal	–
46	REQ_C	O	Inter-chip signal	–
47	START_SCAN	O	Inter-chip signal	–
48	VDD	P	3.3V power	–
49	GND	P	Ground	–
50	DP (2)	USB	USB device port data +	GND
51	DM (2)	USB	USB device port data -	GND
52	VBUS (2)	USB	USB VBUS monitor	GND
53	VDD_1V8 (1)	P	Inter-chip signal	–
54	A0	I	I^2C-compatible address select	Leave open
55	A1	I	I^2C-compatible address select	Leave open
56	VDD_1V8 (1)	P	Inter-chip signal	–
57	MOSI_A	I	Inter-chip signal	–
58	MISO_A	O	Inter-chip signal	–
59	N/C	–	No connection	Leave open
60	$\overline{\text{WAKE}}$	I	External wake-up; typically connected to SCL pin	Vdd if USB used
61	N/C	–	No connection	Leave open
62	MOSI_C	I	Inter-chip signal	–
63	$\overline{\text{RESET}}$	I	Reset low	Vdd (5)
64	VDD	P	3.3V power	–

1. The mXT1386 has an internal 1.8V regulator. The host system only needs to supply the VDD_3V3 rail.
2. Either I2C-compatible or USB interface can be used, but only one interface should be used in any one design.
3. $\overline{\text{CHG}}$ is momentarily set (approximately 100 ms) as an input after power-up or reset for diagnostic purposes.
4. See Section 4.8 on page 23 for additional information.
5. It is recommend that $\overline{\text{RESET}}$ is connected to the host system.

I	Input only	OD	Open drain output	O	Output only, push-pull
USB	USB communications	P	Ground or power		

1.2.2 Slave mXT154 – 49-ball VFBGA

Table 1-2. Pin Listing

Ball	Name	Type	Comments	If Unused, Connect To...
A1	AVDD	P	Analog power	–
A2	Y12	I	Y line connection	Leave open
A3	Y10	I	Y line connection	Leave open
A4	Y8	I	Y line connection	Leave open
A5	Y6	I	Y line connection	Leave open
A6	Y4	I	Y line connection	Leave open
A7	Y2	I	Y line connection	Leave open
B1	X3	O	X matrix drive line	Leave open
B2	GND	P	Ground	–
B3	Y11	I	Y line connection	Leave open
B4	Y9	I	Y line connection	Leave open
B5	Y5	I	Y line connection	Leave open
B6	Y1	I	Y line connection	Leave open
B7	Y0	I	Y line connection	Leave open
C1	X5	O	X matrix drive line	Leave open
C2	X4	O	X matrix drive line	Leave open
C3	Y13	I	Y line connection	Leave open
C4	Y7	I	Y line connection	Leave open
C5	Y3	I	Y line connection	Leave open
C6	GND	P	Ground	–
C7	AVDD	P	Analog power	–
D1	X7	O	X matrix drive line	Leave open
D2	X8	O	X matrix drive line	Leave open
D3	X6	O	X matrix drive line	Leave open
D4	GND	P	Ground	–
D5	X2	O	X matrix drive line	Leave open
D6	X0	O	X matrix drive line	Leave open
D7	X1	O	X matrix drive line	Leave open
E1	X9	O	X matrix drive line	Leave open
E2	X10	O	X matrix drive line	Leave open
E3	$\overline{\text{RESET}}$	I	Inter-chip signal	–
E4	SLV_CLK	I	Inter-chip signal	–
E5	SYNC_OUT	O	Inter-chip signal	–
E6	N/C	–	No connection	Leave open

Table 1-2. Pin Listing (Continued)

Ball	Name	Type	Comments	If Unused, Connect To...
E7	N/C	–	No connection	Leave open
F1	VDD	P	Digital power	–
F2	GND	P	Ground	–
F3	BUSY	O	Inter-chip signal	–
F4	MOSI	O	Inter-chip signal	–
F5	GND	P	Ground	–
F6	SCK	O	Inter-chip signal	–
F7	START_SCAN	I	Inter-chip signal	–
G1	N/C	–	No connection	Leave open
G2	N/C	–	No connection	Leave open
G3	REQ	I	Inter-chip signal	–
G4	$\overline{SS}$	O	Inter-chip signal	–
G5	VDD	P	Digital power	–
G6	MISO	I	Inter-chip signal	–
G7	SYNC_IN	I	Inter-chip signal	–

I Input only O Output only, push-pull P Ground or power

1.2.3 Slave mXT154 – 48-pin QFN

Table 1-3. Pin Listing

Pin	Name	Type	Comments	If Unused, Connect To...
1	Y13	I	Y line connection	Leave open
2	GND	P	Ground	–
3	AVDD	P	Analog power	–
4	X3	O	X matrix drive line	Leave open
5	X4	O	X matrix drive line	Leave open
6	X5	O	X matrix drive line	Leave open
7	X6	O	X matrix drive line	Leave open
8	X7	O	X matrix drive line	Leave open
9	X8	O	X matrix drive line	Leave open
10	X9	O	X matrix drive line	Leave open
11	X10	O	X matrix drive line	Leave open
12	GND	P	Ground	–
13	VDD	P	Digital power	–
14	$\overline{\text{RESET}}$	I	Inter-chip signal	–
15	N/C	–	No connection	Leave open
16	N/C	–	No connection	Leave open
17	BUSY	O	Inter-chip signal	–
18	REQ	I	Inter-chip signal	–
19	SLV_CLK	I	Inter-chip signal	–
20	VDD	P	Digital power	–
21	$\overline{\text{SS}}$	O	Inter-chip signal	–
22	MOSI	O	Inter-chip signal	–
23	MISO	I	Inter-chip signal	–
24	SCK	O	Inter-chip signal	–
25	GND	P	Ground	–
26	SYNC_IN	I	Inter-chip signal	–
27	SYNC_OUT	O	Inter-chip signal	–
28	START_SCAN	I	Inter-chip signal	–
29	N/C	–	No connection	Leave open
30	N/C	–	No connection	Leave open
31	X0	O	X matrix drive line	Leave open
32	X1	O	X matrix drive line	Leave open
33	X2	O	X matrix drive line	Leave open
34	AVDD	P	Analog power	–

Table 1-3. Pin Listing (Continued)

Pin	Name	Type	Comments	If Unused, Connect To...
35	GND	P	Ground	–
36	Y0	I	Y line connection	Leave open
37	Y1	I	Y line connection	Leave open
38	Y2	I	Y line connection	Leave open
39	Y3	I	Y line connection	Leave open
40	Y4	I	Y line connection	Leave open
41	Y5	I	Y line connection	Leave open
42	Y6	I	Y line connection	Leave open
43	Y7	I	Y line connection	Leave open
44	Y8	I	Y line connection	Leave open
45	Y9	I	Y line connection	Leave open
46	Y10	I	Y line connection	Leave open
47	Y11	I	Y line connection	Leave open
48	Y12	I	Y line connection	Leave open

I Input only O Output only, push-pull P Ground or power

2. Overview of the mXT1386

2.1 Introduction

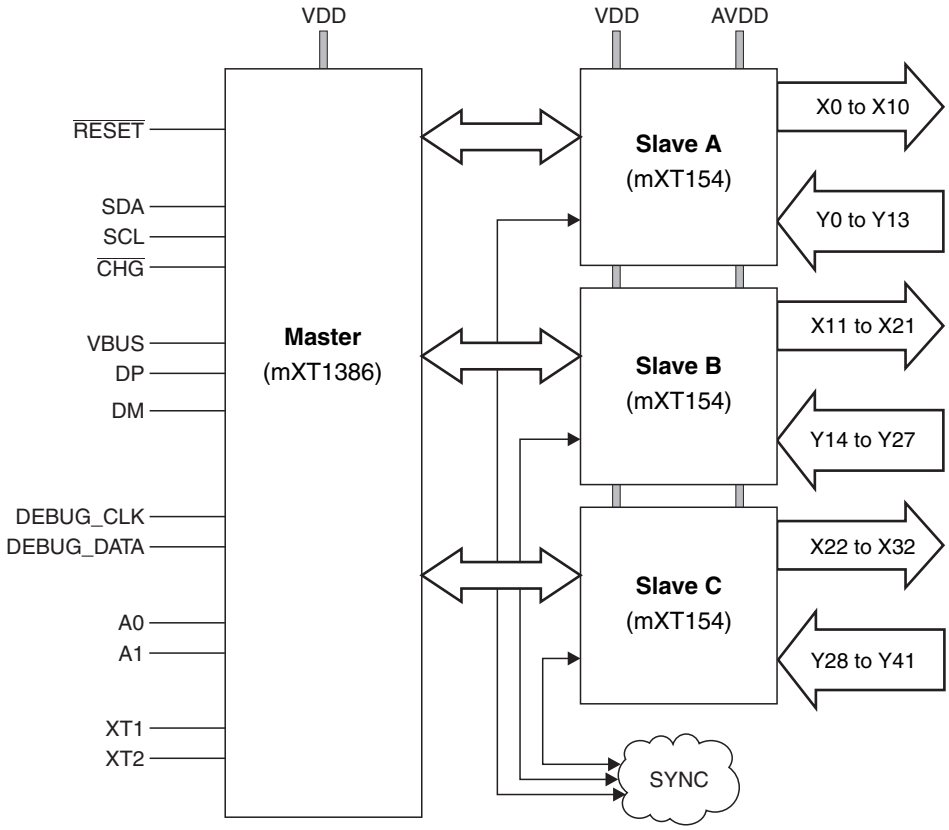
The mXT1386, together with its three associated mXT154 slave devices, is part of the maXTouch™ family of touchscreen controllers. This combined chip set uses a unique charge-transfer acquisition engine to implement the QMatrix™ capacitive-sensing method patented by Atmel®. This allows the measurement of up to 1386mutual-capacitance channelsepending on the master device used). Coupled with a state-of-the-art CPU, the entire touchscreen sensing solution can measure, classify and track finger touches with a high degree of accuracy.

The QMatrix acquisition engine uses an optimal measurement approach to ensure almost complete immunity from parasitic capacitance on the receiver inputs (Y lines). The engine includes sufficient dynamic range to cope with touchscreen mutual capacitances spanning 0.63 pF to 5 pF. This allows great flexibility for use with Atmel's proprietary ITO pattern designs. One- and two-layer ITO sensors are possible using glass or PET substrates.

2.2 Chip Set Architecture

The master mXT1386 device controls three slave mXT154 devices, as shown in Figure 2-1. These combine to allow synchronous signal acquisition, preprocessing, postprocessing and housekeeping to be partitioned in an efficient and flexible way. This gives ample scope for sensing algorithms, touch tracking or advanced shape-based filtering.

Figure 2-1. System Block Diagram



The host interfaces with the single master device only; it never needs to deal with the slave devices. It is the master chip's responsibility to ensure that the configuration and use of the slaves is carried out in a uniform and consistent manner.

Communication with the host is achieved using either the I^2C-compatible interface (see Section 5 on page 24) or the USB interface see Section 6 on page 31). Either interface can be used, depending on the needs of the user's project.

2.3 Understanding Unfamiliar Concepts

If some of the concepts mentioned in this datasheet are unfamiliar, see the following sections for more information:

- Appendix B on page 63 for a glossary of terms
- Appendix C on page 65 for QMatrix technology

2.4 Resources

The following datasheet provides essential information on configuring the chip set:

- *mXT1386 Protocol Guide*

The following documents may also be useful (available by contacting Atmel's Touch Technology division):

- **Configuring the chip set:**
  - Application Note: QTAN0058 – *Rejecting Unintentional Touches with the maXTouch™ Touchscreen Controllers*
- **Miscellaneous:**
  - Application Note QTAN0050 – *Using the maXTouch™ Debug Port*
  - Application Note QTAN0061 – *maXTouch™ Sensitivity Effects for Mobile Devices*
- **Touchscreen design and PCB/FPCB layout guidelines:**
  - Touch Sensors Design Guide (document number: 10620-AT42)
- **Other documents –** The chip set uses the same core technology as the mXT224, so the following documents may also be useful (available by contacting Atmel's Touch Technology division):
  - Application Note QTAN0054 – *Getting Started with mXT224 Touchscreen Designs*
  - Application Note QTAN0048 – *mXT224 PCB/FPCB Layout Guidelines*
  - Application Note QTAN0052 – *mXT224 Passive Stylus Support*

3. Touchscreen Basics

3.1 Sensor Construction

A touchscreen is usually constructed from a number of transparent electrodes. These are typically on a glass or plastic substrate. They can also be made using non-transparent electrodes, such as copper or carbon. Electrodes are normally formed by etching a material called Indium Tin Oxide (ITO). This is a brittle ceramic material, of high optical clarity and varying sheet resistance. Thicker ITO yields lower levels of resistance (perhaps tens to hundreds of Ω/square) at the expense of reduced optical clarity. Lower levels of resistance are generally more compatible with capacitive sensing. Thinner ITO leads to higher levels of resistance (perhaps hundreds to thousands of Ω/square) with some of the best optical characteristics.

Interconnecting tracks formed in ITO can cause problems. The excessive RC time constants formed between the resistance of the track and the capacitance of the electrode to ground can inhibit the capacitive sensing function. In such cases, ITO tracks should be replaced by screen printed conductive inks (non-transparent) outside the touchscreen's viewing area.

A range of trade-offs also exist with regard to the number of layers used for construction. Atmel has pioneered single-layer ITO capacitive touchscreens. For many applications these offer a near optimum cost/performance balance. With a single layer screen, the electrodes are all connected using ITO out to the edges of the sensor. From there the connection is picked up with printed silver tracks. Sometimes two overprinted silver tracking layers are used to reduce the margins between the edge of the substrate and the active area of the sensor.

Two-layer designs can have a strong technical appeal where ultra-narrow edge margins are required. They are also an advantage where the capacitive sensing function needs to have a very precise cut-off as a touch is moved to just off the active sensor area. With a two-layer design the QMatrix transmitter electrodes are normally placed nearest the bottom and the receiver electrodes nearest the top. The separation between layers can range from hundreds of nanometers to hundreds of microns, with the right electrode design and considerations of the sensing environment.

3.2 Electrode Configuration

The specific electrode designs used in Atmel's touchscreens are the subject of various patents and patent applications. Further information is available on request.

The chip set supports various configurations of electrodes as summarized below:

Touchscreens:	1 Touchscreen allowed 3X x 3Y minimum (depends on screen resolution) 33X x 42Y maximum (subject to other configurations)
Keys:	2 Key Arrays allowed Each up to 32 keys (subject to other configurations)

3.3 Scanning Sequence

All channels are scanned in sequence by the chip set. There is full parallelism in the scanning sequence to improve overall response time. The channels are scanned by measuring capacitive changes at the intersections formed between the first X line and all the Y lines. Then the intersections between the next X line and all the Y lines are scanned, and so on, until all X and Y combinations have been measured.

The chip set can be configured in various ways. It is possible to disable some channels so that they are not scanned at all. This can be used to improve overall scanning time.

3.4 Touchscreen Sensitivity

3.4.1 Adjustment

Sensitivity of touchscreens can vary across the extents of the electrode pattern due to natural differences in the parasitics of the interconnections, control chip, and so on. An important factor in the uniformity of sensitivity is the electrode design itself. It is a natural consequence of a touchscreen pattern that the edges form a discontinuity and hence tend to have a different sensitivity. The electrodes at the far edges do not have a neighboring electrode on one side and this affects the electric field distribution in that region.

A sensitivity adjustment is available for the whole touchscreen. This adjustment is a basic algorithmic threshold that defines when a channel is considered to have enough signal change to qualify as being in detect.

The chip set supports mixed configurations of different touch objects, each having independent threshold controls to allow fine tuning with mixed configurations.

3.4.2 Mechanical Stackup

The mechanical stackup refers to the arrangement of material layers that exist above and below a touchscreen. The arrangement of the touchscreen in relation to other parts of the mechanical stackup has an effect on the overall sensitivity of the screen. QMatrix technology has an excellent ability to operate in the presence of ground planes close to the sensor. QMatrix sensitivity is attributed more to the interaction of the electric fields between the transmitting (X) and receiving (Y) electrodes than to the surface area of these electrodes. For this reason, stray capacitance on the X or Y electrodes does not strongly reduce sensitivity.

Front panel dielectric material has a direct bearing on sensitivity. Plastic front panels are usually suitable up to about 1.2 mm, and glass up to about 2.5 mm (dependent upon the screen size and layout). The thicker the front panel, the lower the signal-to-noise ratio of the measured capacitive changes and hence the lower the resolution of the touchscreen. In general, glass front panels are near optimal because they conduct electric fields almost twice as easily as plastic panels.

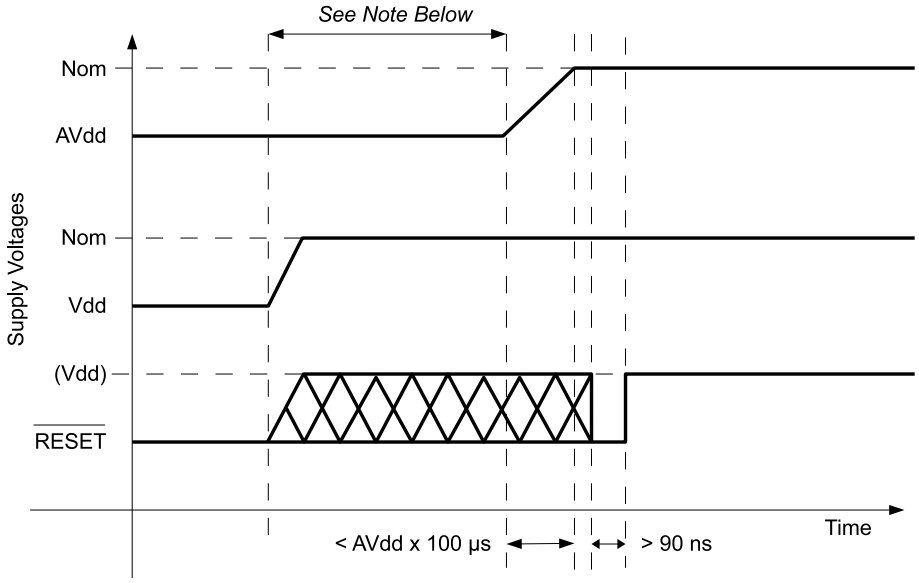
4. Detailed Operation

4.1 Power-up/Reset

The mXT1386 has an internal Power-on Reset (POR) that is executed on power-up.

The device must be held in $\overline{\text{RESET}}$ (active low) while both the digital and analog power supplies (Vdd and AVdd) are powering up. If a slope or slew is applied to the digital or analog supplies, Vdd and AVdd must reach their nominal values before the $\overline{\text{RESET}}$ signal is deasserted (that is, goes high). This is shown in Figure 4-1. See Section 8.2 on page 43 for nominal values for Vdd and AVdd.

Figure 4-1. Power Sequencing on the mXT1386



NOTE: Vdd and AVdd can be powered up in either order.
There is no prerequisite for the length of time between Vdd and AVdd powering up.

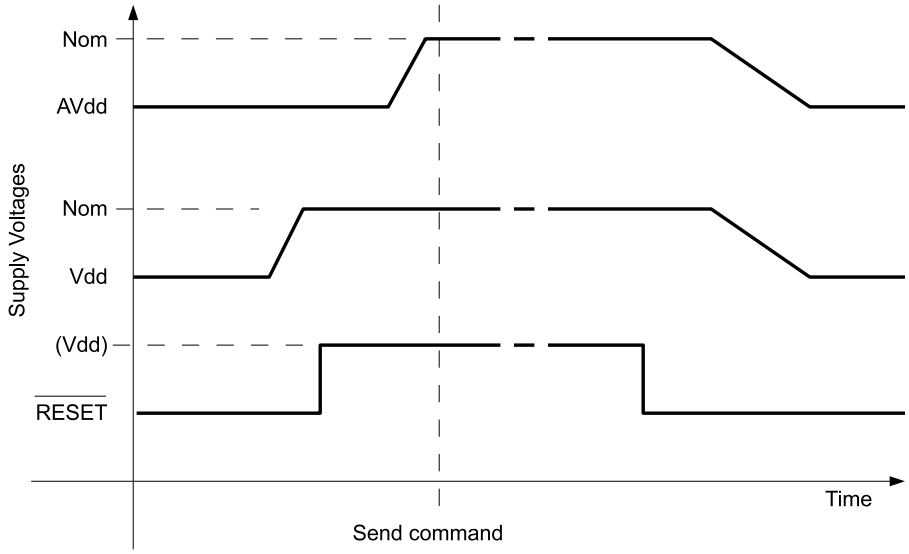
Note that there are no specific power-up, or power-down sequences required for the mXT1386. This means that the digital or analog supplies can be applied independently and in any order during power-up.

After power-up, the mXT1386 takes 90 ms before it is ready to start communications. Vdd must drop to below 1V in order to effect a proper POR. See Section 8 for further specifications.

If the $\overline{\text{RESET}}$ line is released before the AVDD supplies have reached their nominal voltage (see Figure 4-2), then some additional operations need to be carried out by the host. There are two options open to the host controller:

- Start the part in deep sleep mode and then send the command sequence to set the cycle time to wake the part and allow it to run normally. Note that in this case a calibration command is also needed.
- Send a reset command.

Figure 4-2. Power Sequencing on the mXT1386 – Late rise on AVDD



The $\overline{\text{RESET}}$ pin can be used to reset the mXT1386 whenever necessary. The $\overline{\text{RESET}}$ pin must be asserted low for at least 10 ns to cause a reset. After releasing the $\overline{\text{RESET}}$ pin the mXT1386 takes 90 ms before it is ready to start communications. It is recommended to connect the $\overline{\text{RESET}}$ pin to a host controller to allow it to initiate a full hardware reset without requiring a power-down.

A software reset command can also be used to reset the chip set (refer to the Command Processor object in the *mXT1386 Protocol Guide*). A software reset takes ~250 ms. After the chip set has finished initializing it asserts the $\overline{\text{CHG}}$ line to signal to the host that a message is available. The reset flag is set in the Message Processor object to indicate to the host that it has just completed a reset cycle. This bit can be used by the host to detect any unexpected brownout events. This allows the host take any necessary corrective actions, such as reconfiguration.

A checksum check is performed on the configuration settings held in the nonvolatile memory of the master device. If the checksum does not match a stored copy of the last checksum, then this indicates that the settings have become corrupted. This is signaled to the host by setting the configuration error bit in the message data for the Command Processor object (refer to the *mXT1386 Protocol Guide* for more information).

Note that the $\overline{\text{CHG}}$ line is momentarily set (approximately 100 ms) as an input after power-up or reset for diagnostic purposes. It is therefore particularly important that the line should be allowed to float high via the $\overline{\text{CHG}}$ line pull-up resistors during this period. It should not be driven by the host.

4.2 Calibration

Calibration is the process by which a sensor chip assesses the background capacitance on each channel. Channels are only calibrated on power-up and when:

- The channel is enabled (that is, activated).

OR

- The channel is already enabled and one of the following applies:
  - The channel is held in detect for longer than the Touch Automatic Calibration setting (refer to the *mXT1386 Protocol Guide* for more information on TCHAUTOCAL setting in the Acquisition Configuration object).
  - The signal delta on a channel is at least the touch threshold (TCHTHR) in the anti-touch direction, while no other touches are present on the channel matrix (refer to the *mXT1386 Protocol Guide* for more information on the TCHTHR field in the Multiple Touch Touchscreen and Key Array objects).
  - The user issues a recalibrate command.

A status message is generated on the start and completion of a calibration.

Note that the chip set performs a global calibration; that is, all the channels are calibrated together.

4.3 Operational Modes

The chip set operates in two modes: active (touch detected) and idle (no touches detected). Both modes operate as a series of burst cycles. Each cycle consists of a short burst (during which measurements are taken) followed by an inactive sleep period. The difference between these modes is the length of the cycles. Those in idle mode typically have longer sleep periods. The cycle length is configured using the IDLEACQINT and ACTVACQINT settings in the Power Configuration object. In addition, an Active to Idle timeout (ACTV2IDLETO) setting is provided.

Refer to the *mXT1386 Protocol Guide* for full information on how these modes operate, and how to use the settings provided.

4.4 Sense Lines

Each of the three mXT154 slave devices controls a maximum of 11 X lines (X0 to X10) and 14 Y lines (Y0 to Y13). This makes a total of 33 X lines and 42 Y lines available for use. Each slave controls the sense lines listed in Table 4-1.

Table 4-1. Sense Lines

Slave Device	Controls Sense Lines...	
	X	Y
Slave A	X0 to X10	Y0 to Y13
Slave B	X11 to X21	Y14 to Y27
Slave C	X22 to X32	Y28 to Y41

If fewer lines are required for use in the user's product, unused lines must be dropped from the slave devices in reverse sense line order, starting with the highest line; that is, in the following pin order:

Slave C: Y13 down to Y0 or
X10 down to X0 (as appropriate)

then...

Slave B: Y13 down to Y0 or
X10 down to X0 (as appropriate)

then (unlikely)...

Slave A: Y13 down to Y0 or
X10 down to X0 (as appropriate)

4.5 Touchscreen Layout

4.5.1 Introduction

The physical matrix can be configured to have one or more touch objects. These are configured using the appropriate touch objects (Multiple Touch Touchscreen, Key Array). It is not mandatory to have all the allowable touch objects p0resent. The objects are disabled by default so only those that you wish to use need to be enabled. Refer to the *mXT1386 Protocol Guide* for more information on configuring the touch objects.

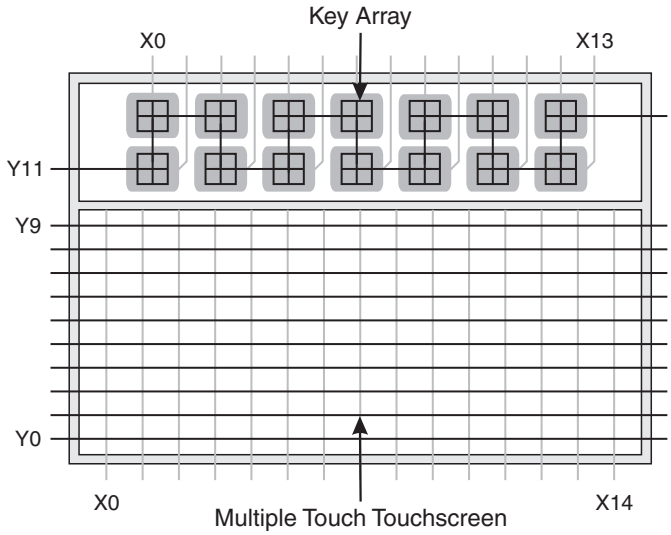
When designing the physical layout of the touch panel, obey the following rules:

- Each touch object should be a regular rectangular shape in terms of the lines it uses.
- The touch objects must not share the Y lines they use. The X lines can, however, be shared.
- The design of the touch objects does not physically need to be on a strict XY grid pattern.

4.5.2 Example Layout – A Multiple Touch Touchscreen and a Key Array

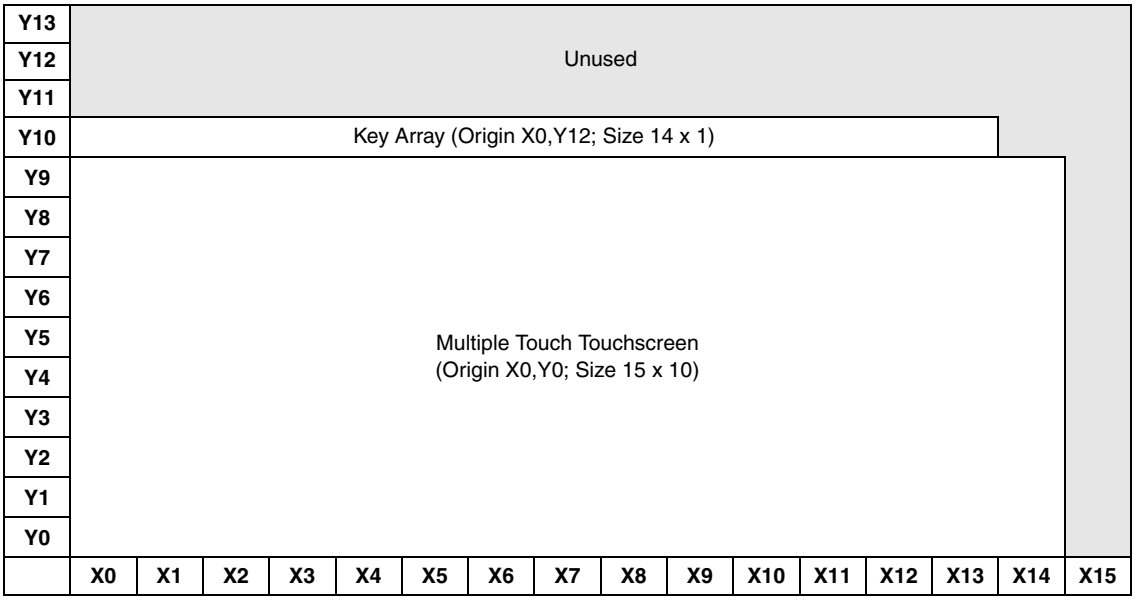
Figure 4-3 shows an example layout of a Multiple Touch Touchscreen with a Key Array. Note how the touch objects do not share the same physical XY grid pattern.

Figure 4-3. Example Screen Layout



The logical XY grid allocation for this layout (as configured by a Multiple Touch Touchscreen and a Key Array object) is shown in Figure 4-4.

Figure 4-4. Example Touchscreen Configuration



4.6 Signal Processing

4.6.1 Adjacent Key Suppression Technology

Adjacent Key Suppression (AKS) technology is a patented method used to detect which touch object is touched when objects are located close together. A touch in a group of AKS objects is only indicated on the object in that group that is touched first. This is assumed to be the intended object. Once an object in an AKS group is in detect, there can be no further detections within that group until the object is released. Objects can be in more than one AKS group.

Note that AKS technology works best when it operates in conjunction with a detect integration setting of several acquisition cycles.

The chip set has two levels of AKS. The first level works between the Touchscreen and Key Array objects. The touch objects are assigned to AKS groups. If a touch occurs within one of the touch objects in a group, then touches within other objects inside that group are suppressed. For example, if a Touchscreen and Key Array are placed in the same AKS group, then a touch in the Touchscreen will suppress touches in the Key Array, and vice versa.

The second level of AKS is internal AKS within an individual Key Array object (note that internal AKS is not present on other types of touch objects, only a Key Array). If internal AKS is enabled, then when one key is touched, touches on all the other keys within the Key Array are suppressed.

AKS is configured using the Multiple Touch Touchscreen or Key Array objects (refer to the *mXT1386 Protocol Guide* for more information).

Note: If a touch is in detect and then AKS is enabled, that touch will not be forced out of detect. It will not go out of detect until the touch is released. AKS will then operate normally. This applies to both levels of AKS.

4.6.2 Detection Integrator

The chip set features a touch detection integration mechanism. This acts to confirm a detection in a robust fashion. A counter is incremented each time a touch has exceeded its threshold and has remained above the threshold for the current acquisition. When this counter reaches a preset limit the sensor is finally declared to be touched. If, on any acquisition, the signal is not seen to exceed the threshold level, the counter is cleared and the process has to start from the beginning.

The detection integrator is configured using the appropriate touch objects (Multiple Touch Touchscreen, Key Array). Refer to the *mXT1386 Protocol Guide* for more information.

4.6.3 Digital Filtering

The chip set supports the use of on-chip Frequency Hopping to filter the acquisition data received from the sensor. The filtering is applied to the delta values, and not to the measured signal. Frequency Hopping is configured using the Noise Suppression object. Refer to the *mXT1386 Protocol Guide* for more information.

4.6.4 Gestures

The chip set supports the on-chip processing of touches so that specific gestures can be detected. These may be a one-touch gesture (such as a tap or a drag) or they may be a two-touch gesture (such as a pinch or a rotate).

Gestures are configured using the One-touch Gesture Processor and the Two-touch Gesture Processor objects. Refer to the *mXT1386 Protocol Guide* for more information on gestures and their configuration.

4.6.5 Grip Suppression

The chip set has a grip suppression mechanism to suppress false detections when the user grips a handheld device.

Grip suppression works by specifying a boundary around a touchscreen, within which touches can be suppressed whilst still allowing touches in the center of the touchscreen. This ensures that a “rolling” hand touch (such as when a user grips a mobile device) is suppressed. A “real” (finger) touch towards the center of the screen is allowed.

Grip suppression is configured using the Grip Suppression object. Refer to the *mXT1386 Protocol Guide* for more information.

4.6.6 Palm Suppression

The chip set has a palm suppression mechanism to suppress false detections when the user places a palm on the touchscreen, while at the same time still allowing finger touches.

Palm suppression allows touches to be suppressed or allowed based on their size: large palm touches are suppressed, whereas small finger touches are allowed. The distance between touches can also taken into consideration. This ensures that a finger touch that is near to a palm touch is assumed to be part of the palm touch and is rejected. Any finger touch that is farther away from the palm touch can still be reported.

Palm suppression is configured using the Palm Suppression object. Refer to the *mXT1386 Protocol Guide* for more information.

4.7 Circuit Components

4.7.1 Bypass Capacitors

The mXT1386 master device requires a 4.7 µF capacitor with 100 nF ceramic X7R or X5R bypass capacitors on each of the Vdd and internal VDD_1V8 supplies.

The mXT154 slave devices require a 100 nF and a 1 µF bypass capacitor on the Vdd supply, and two 100 nF capacitors and a 1 µF capacitor on the AVdd supply. The capacitors should be ceramic X7R or X5R.

See the schematics in Appendix A on page 56 for examples of these.

The PCB traces connecting the capacitors to the pins of the mXT1386 and mXT154 devices must not exceed 5 mm in length. This limits any stray inductance that would reduce filtering effectiveness. See also Section 8.11 on page 47.

4.7.2 PCB Cleanliness

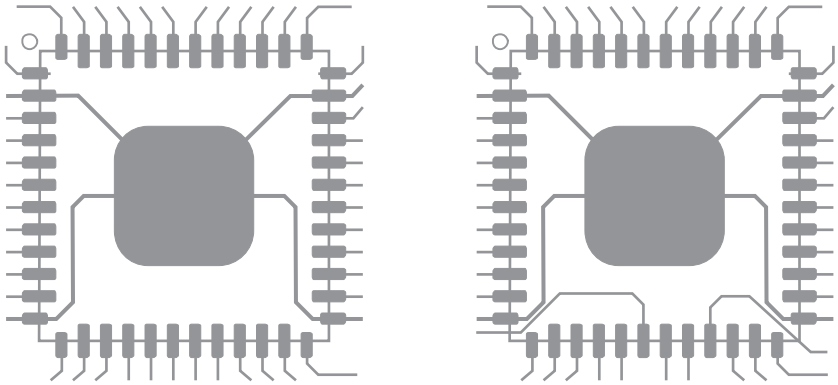
Modern no-clean-flux is generally compatible with capacitive sensing circuits.

CAUTION: If a PCB is reworked to correct soldering faults relating to any of the chip set devices, or to any associated traces or components, be sure that you fully understand the nature of the flux used during the rework process. Leakage currents from hygroscopic ionic residues can stop capacitive sensors from functioning. If you have any doubts, a thorough cleaning after rework may be the only safe option.

4.7.3 QFN Package Restrictions

The central pad on the underside of a QFN chip should be connected to ground. Do not run any tracks underneath the body of the chip, only ground. Figure 4-5 shows an example of good/bad tracking.

Figure 4-5. Examples of Good and Bad Tracking



4.7.4 Supply Quality

While the chip set has good Power Supply Rejection Ratio properties, poorly regulated and/or noisy power can significantly reduce performance. See Section 8.11 on page 47.

Always operate the chip set with a well-regulated and clean AVdd supply. It supplies the sensitive analog stages in the chip set.

There is no separate GND return pin for the analog stages. You are advised to consider return current paths from other current consumers in the system. Try to provide a separate heavy GND return trace or flood for the chip set that connects at a PSU star-point or connector pin. This helps to avoid inductive transient voltages coupling into the capacitive measurements made by the chip.

It is still recommended, however, that a low noise supply is used to prevent cross-talk into the analog sections.

The AVdd level used in the design for the mXT154 slave devices must be written to the VOLTAGE field of the CTE object. Refer to the *mXT1386 Protocol Guide* for more details. See also Section 8.4 on page 44.

4.7.5 Supply Sequencing

Vdd and AVdd can be powered independently of each other without damage to the chip set. Vdd and AVdd should be supplied with the same voltage unless specified by Atmel.

Make sure that any lines connected to the chip set are below or equal to Vdd during power-up. For example, if $\overline{\text{RESET}}$ is supplied from a different power domain to the mXT1386 master device's Vdd pin, make sure that it is held low when Vdd is off. If this is not done, the $\overline{\text{RESET}}$ signal could parasitically couple power via the mXT1386's $\overline{\text{RESET}}$ pin into the Vdd supply.

4.7.6 Oscillator

The chip set requires an 8 MHz crystal oscillator connected to the master device. A crystal oscillator with a minimum accuracy of 100 ppm must be used.

4.7.7 Synchronization Signal

Synchronization between the three slave devices is achieved using the SYNC_IN and SYNC_OUT pins (see Figure 2-1). These should be connected to a common AND gate. This means that when all the SYNC_OUT lines have been asserted, the SYNC_IN line is triggered simultaneously on each of the three slaves. An example of this is shown in the schematic in Section A.2 on page 57.

4.8 Debugging

The chip set provides a mechanism for obtaining raw data for development and testing purposes by reading data from the Diagnostic Debug object. Refer to the *mXT1386 Protocol Guide* for more information on this object.

A second mechanism is provided that allows the host to read the real-time raw data using the low-level debug port. This can be accessed via the SPI interface or the USB interface. Note that if both the I^2C-compatible and USB interfaces are used for normal communcations, the debug data is output on the USB interface. Refer to QTAN0050, *Using the maXTouch Debug Port*, for more information on the debug port.

There is also a Self Test object that runs self-test routines in the mXT1386 to find hardware faults on the sense lines and the electrodes. Refer to the *mXT1386 Protocol Guide* for more information.

4.9 Communications

Communication with the host is achieved using either the I^2C-compatible interface (see Section 5 on page 24) or the USB interface (see Section 6 on page 31). Either interface can be used, depending on the needs of the user's project, but only one interface should be used in any one design.

Note that you only need to connect those pins that are actually required for use with the chosen communications interface. See Section 1.2 on page 4 for details on what should be done with the unconnected pins. This ensures optimal power consumption and correct functioning.

4.10 Configuring the Chip Set

The chip set has an object-based protocol that organizes the features of the chip set into objects that can be controlled individually. This is configured using the Object Protocol common to many of Atmel's touch sensor devices. For more information on the Object Protocol and its implementation on the chip set, refer to the *mXT1386 Protocol Guide*.

5. I²C-compatible Communications

5.1 Communications Protocol

The chip set can use an I²C-compatible interface for communication. See Appendix D on page 67 for details of the I²C-compatible protocol.

The I²C-compatible interface is used in conjunction with the $\overline{\text{CHG}}$ line. The $\overline{\text{CHG}}$ line going active signifies that a new data packet is available. This provides an interrupt-style interface and allows the chip set to present data packets when internal changes have occurred.

5.2 I²C-compatible Addresses

The chip set supports four I²C-compatible device addresses. These are selected at start-up using the A0 and A1 pins on the mXT1386 master device (see Table 5-1). The address pins should be connected to GND to signal a logic "0", and either left open or connected to VDD_3v3 to signal a logic "1" (1).

Table 5-1. I²C-compatible Device Addresses

A1	A0	Address
0	0	0x4C
0	1	0x4D
1	0	0x5A
1	1	0x5B

The addresses are shifted left to form the SLA+W or SLA+R address when transmitted over the I²C-compatible interface (see Table 5-2).

Table 5-2. Format of SLA+W and SLA+R

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Address (see Table 5-1)							Read/write

5.3 Writing To the Chip Set

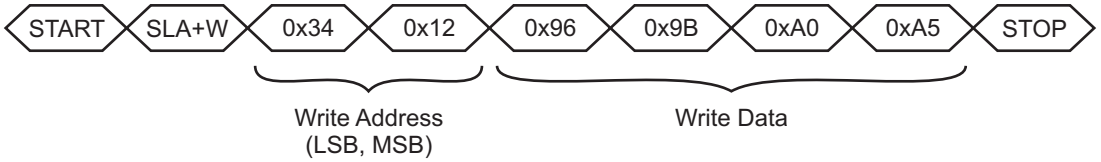
A WRITE cycle to the chip set consists of a START condition followed by the I²C-compatible address of the device (SLA+W). The next two bytes are the address of the location into which the writing starts. The first byte is the Least Significant Byte (LSByte) of the address, and the second byte is the Most Significant Byte (MSByte). This address is then stored as the address pointer.

Subsequent bytes in a multibyte transfer form the actual data. These are written to the location of the address pointer, location of the address pointer +1, location of the address pointer + 2, and so on. The address pointer returns to its starting value when the WRITE cycle's STOP condition is detected.

Figure 5-1 shows an example of writing four bytes of data to contiguous addresses starting at 0x1234.

1. No external pull-down resistors are required on the A0 and A1 pins.

Figure 5-1. Example of a Four-byte Write Starting at Address 0x1234



5.4 I²C-compatible Writes in Checksum Mode

In I²C-compatible checksum mode an 8-bit CRC is added to all I²C-compatible writes. The CRC is sent at the end of the data write as the last byte before the STOP condition. All the bytes sent are included in the CRC, including the two address bytes. Any command or data sent to the chip set is processed even if the CRC fails.

To indicate that a checksum is to be sent in the write, the most significant bit of the MSByte of the address is set to 1. For example, the I²C-compatible command shown in Figure 5-2 writes a value of 150 (0x96) to address 0x1234 with a checksum. The address is changed to 0x9234 to indicate checksum mode.

Figure 5-2. Example of a Write To Address 0x1234 With a Checksum

5.5 Reading From the Chip Set

Two I²C-compatible bus activities must take place to read from the chip set. The first activity is an I²C-compatible write to set the address pointer (LSByte then MSByte). The second activity is the actual I²C-compatible read to receive the data. The address pointer returns to its starting value on detection of the NACK condition immediately before the STOP condition.

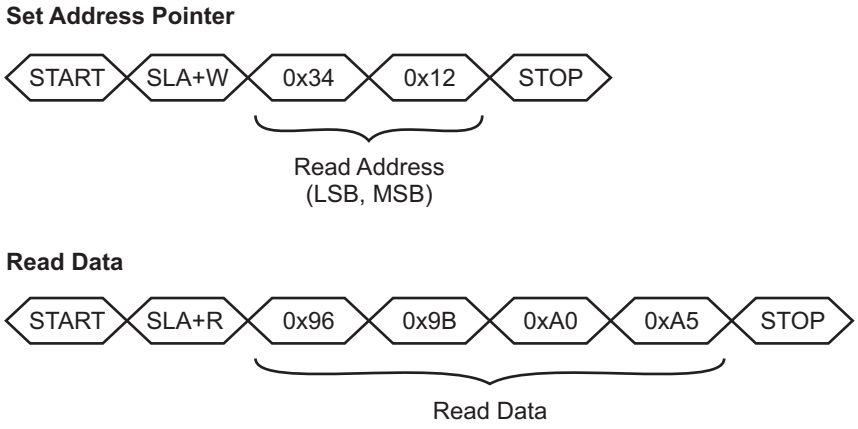
It is not necessary to set the address pointer before every read. The address pointer is updated automatically after every read operation. The address pointer will be correct if the reads occur in order. In particular, when reading multiple messages from the Message Processor object, the address pointer is automatically reset to allow continuous reads (see Section 5.6).

The WRITE and READ cycles consist of a START condition followed by the I²C-compatible address of the device (SLA+W or SLA+R respectively).

Figure 6-8 shows the I²C-compatible commands to read four bytes starting at address 0x1234.

Note: Although some chips may tolerate an illegal ACK before a STOP condition, the mXT1386 will not tolerate this. The correct I²C-specified sequence to terminate a read transfer is a NACK followed by a STOP condition.

Figure 5-3. Example of a Four-byte Read Starting at Address 0x1234



5.6 Reading Status Messages with DMA

The device facilitates the easy reading of multiple messages using a single continuous read operation. This allows the host hardware to use a direct memory access (DMA) controller for the fast reading of messages, as follows:

1. The host uses a write operation to set the address pointer to the start of the Message Count object, if necessary. (1) If a checksum is required on each message, the most significant bit of the MSByte of the read address must be set to 1.
2. The host starts the read operation of the message by sending a START condition.
3. The host reads the Message Count object (one byte) to retrieve a count of the pending messages (refer to the *mXT1386 Protocol Guide* for details).
4. The host calculates the number of bytes to read by multipling the message count by the size of the Message Processor object. (2)

   Note that the size of the Message Processor object as recorded in the Object Table includes a checksum byte. If a checksum has not been requested, one byte should be deducted from the size of the object. That is: number of bytes = count x (size-1).
5. The host reads the calculated number of message bytes. It is important that the host does *not* send a STOP condition during the message reads, as this will terminate the continuous read operation and reset the address pointer. No START and STOP conditions must be sent between the messages.
6. The host sends a STOP condition at the end of the read operation after the last message has been read. The NACK condition immediately before the STOP condition resets the address pointer to the start of Message Count object.

Figure 5-4 shows an example of using a continuous read operation to read three messages from the device without a checksum. Figure 5-5 on page 28 shows the same example with a checksum.

1. The STOP condition at the end of the read resets the address pointer to its intial location, so it may already be pointing at the Message Count object following a previous message read.
2. The host should have already read the size of the Message Processor object in its initialization code.

Figure 5-4. Continuous Message Read Example – No Checksum

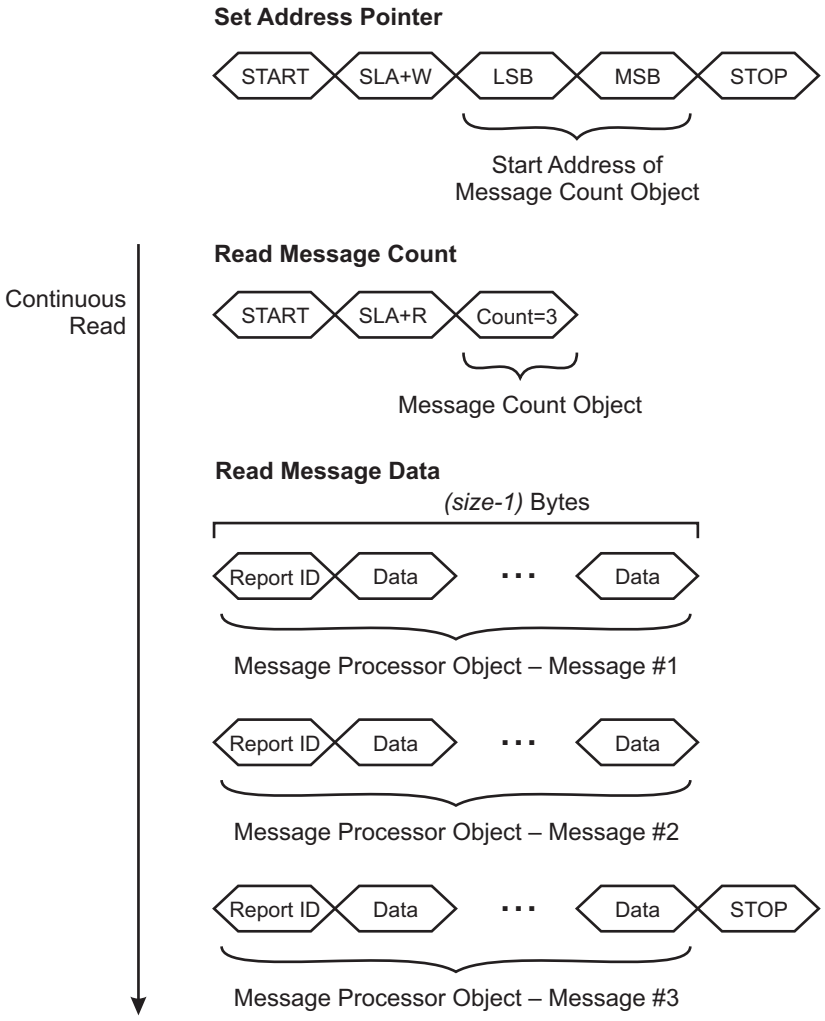
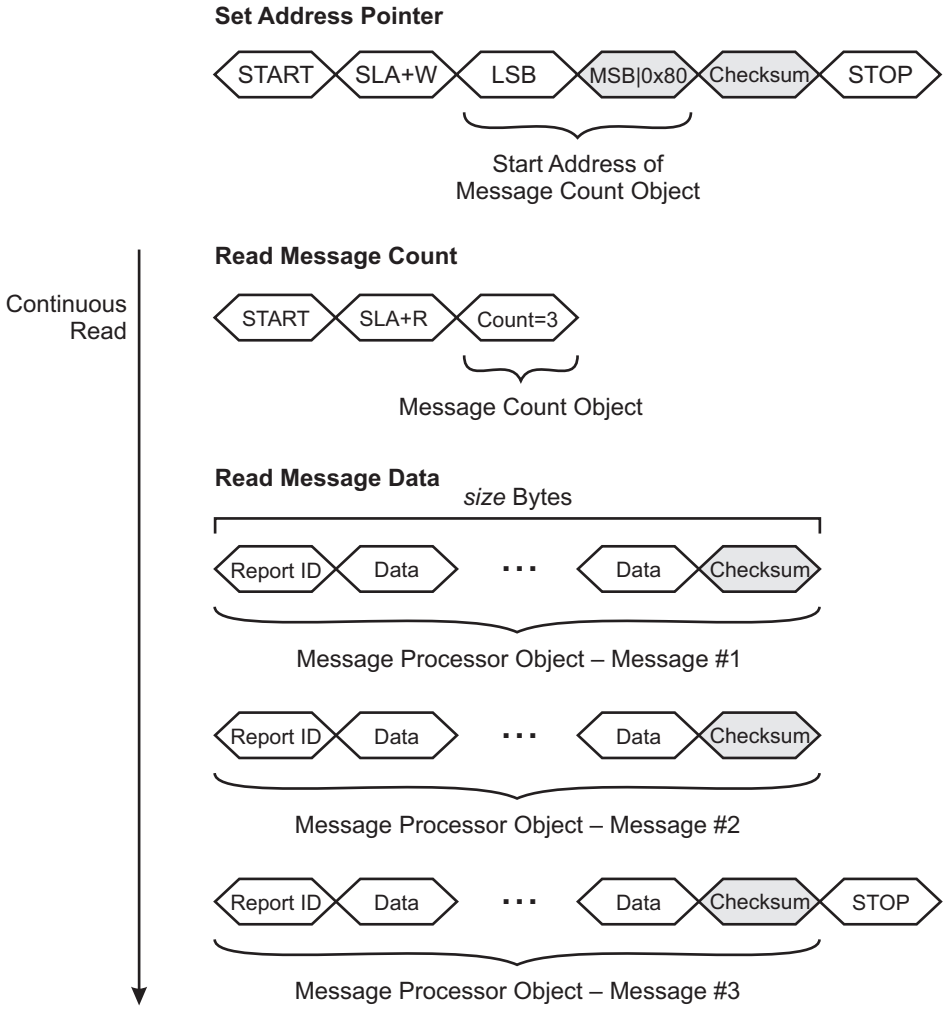


Figure 5-5. Continuous Message Read Example – I²C-compatible Checksum Mode



There are no checksums added on any other I²C-compatible reads. An 8-bit CRC can be added, however, to all I²C-compatible writes, as described in Section 5.4 on page 25.

An alternative method of reading messages using the $\overline{\text{CHG}}$ line is given in Section 5.7.

5.7 $\overline{\text{CHG}}$ Line

The $\overline{\text{CHG}}$ line is an active-low, open-drain output that is used to alert the host that a new message is available in the Message Processor object. This provides the host with an interrupt-style interface with the potential for fast response times. It reduces the need for wasteful I²C-compatible communications.

The $\overline{\text{CHG}}$ line operates in two modes, as defined by the Communications Configuration object (refer to the *mXT1386 Protocol Guide*).

In Mode 0:

1. The $\overline{\text{CHG}}$ line goes low to indicate that a message is present.
2. The $\overline{\text{CHG}}$ line goes high when the first byte of the first message (that is, its report ID) has been sent and acknowledged (ACK sent) and the next byte has been prepared in the buffer.
3. The STOP condition at the end of an I²C-compatible transfer causes the $\overline{\text{CHG}}$ line to stay high if there are no more messages. Otherwise the $\overline{\text{CHG}}$ line goes low to indicate a further message.

Mode 0 allows the host to continually read messages. Messaging reading ends when a report ID of 255 (“invalid message”) is received. Alternatively the host ends the transfer by sending a NACK after receiving the last byte of a message, followed by a STOP condition. If and when there is another present, the $\overline{\text{CHG}}$ line goes low, as in step 1. In this mode the state of the $\overline{\text{CHG}}$ line does not need to be checked during the I²C-compatible read.

In Mode 1:

1. The $\overline{\text{CHG}}$ line goes low to indicate that a message is present.
2. The $\overline{\text{CHG}}$ line remains low while there are further messages to be sent after the current message.
3. The $\overline{\text{CHG}}$ line goes high again only once the first byte of the last message (that is, its report ID) has been sent and acknowledged (ACK sent) and the next byte has been prepared in the output buffer.

Mode 1 allows the host to continually read the messages until the $\overline{\text{CHG}}$ line goes high, and the state of the $\overline{\text{CHG}}$ line determines whether or not the host should continue receiving messages from the chip set. Note that the $\overline{\text{CHG}}$ line state should be checked only between messages and not between the bytes of a message.

The $\overline{\text{CHG}}$ line should be allowed to float during normal usage. It should also be pulled up with a pull-up resistor, typically 10 kΩ to Vdd. This is particularly important after power-up or reset (see Section 4.1 on page 14).

5.8 $\overline{\text{WAKE}}$ Line

The $\overline{\text{WAKE}}$ line is an active-low input that is used to wake the mXT1386 up from deep sleep mode before communicating with it via the I²C-compatible interface. It can be used to minimize current consumption when the mXT1386 is in deep sleep mode. Refer to the *mXT1386 Protocol Guide* for information on deep sleep mode.

Note that the $\overline{\text{WAKE}}$ line is not used when the mXT1386 is not in deep sleep mode.

This pin can be connected in one of the following ways:

- It can be left permantly low (at the expense of increased power consumption in deep sleep mode)
- It can be connected to the I²C-compatible SCL pin
- It can be connected to a GPIO pin on the host

The mXT1386 is ready to accept I²C-compatible communications 25 ms after the $\overline{\text{WAKE}}$ line is asserted. This means that if the $\overline{\text{WAKE}}$ line is connected to a GPIO line, the line must be asserted 25 ms before the host attempts to communicate with the mXT1386.

If the $\overline{\text{WAKE}}$ line is connected to the SCL pin, the mXT1386 will send a NACK on the first attempt to address it; the host must then retry 25 ms later.

The mXT1386 remains ready to accept I²C-compatible communications for 2 seconds after the WAKE line is asserted, after which time the chip will timeout and return to deep sleep mode. This timeout period is reset every time there is an I²C-compatible communication with the mXT1386, or if the WAKE line is held asserted.

Note that when the mXT1386 is sent into deep sleep mode, it goes to sleep immediately. In this case the two-second timeout does not apply until the WAKE pin is asserted.

5.9 SDA, SCL

The I²C-compatible bus transmits data and clock with SDA and SCL, respectively. These are open-drain. The I²C-compatible master and slave devices can only drive these lines low or leave them open. The termination resistors (Rp) pull the line up to Vdd if no I²C-compatible device is pulling it down.

The termination resistors commonly range from 1 kΩ to 10 kΩ. They should be chosen so that the rise times on SDA and SCL meet the I²C-compatible specifications (see Section 8.6 on page 45).

5.10 Clock Stretching

The chip set supports clock stretching in accordance with the I²C specification. It may also instigate a clock stretch if a communications event happens during a period when the chip set is busy internally. The maximum clock stretch is approximately TBD ms.

The chip set has an internal bus monitor that can reset the internal I²C-compatible hardware if SDA or SCL is stuck low. This means that if a prolonged clock stretch is seen by the chip set, then any ongoing transfers with the chip set may be corrupted. The bus monitor is enabled or disabled using the Communications Configuration object. Refer to the *mXT1386 Protocol Guide* for more information.

6. USB Communications

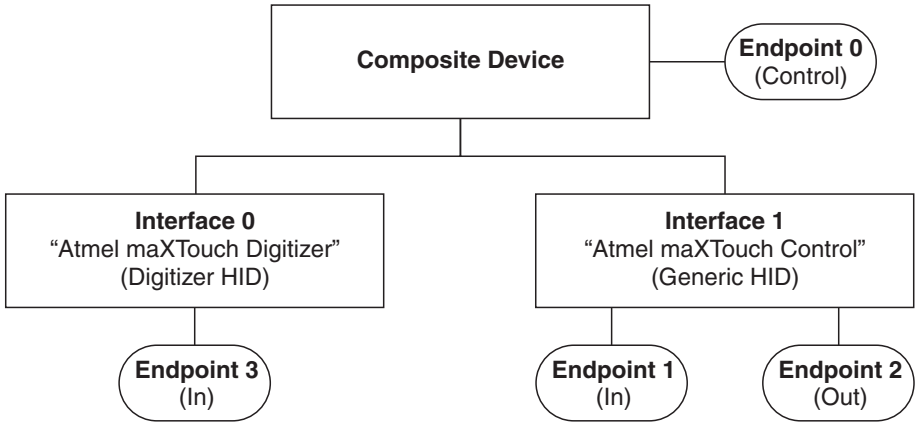
6.1 Communications Protocol

The chip set is a composite USB device with two Human Interface Device (HID) interfaces:

- **Interface 0** – This interface provides a Digitizer HID that supplies touch information to the Host for passing on to a PC's operating system. This interface is supported by Microsoft® Windows® 7 without the need for additional software. The HID identifier string is "Atmel maXTouch Digitizer".
- **Interface 1** – This interface provides a Generic HID that allows the host to communicate with the chip set using the Object Protocol. The HID identifier string is "Atmel maXTouch Control".

The topography of the USB device is shown in Figure 6-1.

Figure 6-1. USB Topography



Communication takes place using Full-speed USB at 12 Mbps.

For more information on the USB HID specifications visit www.usb.org.

6.2 Endpoint Addresses

The endpoint addresses are listed in Table 6-1.

Table 6-1. Endpoint Addresses

Endpoint	Direction	Address
Endpoint 0	Bidirectional (control)	–
Endpoint 1	In	0x81
Endpoint 2	Out	0x02
Endpoint 3	In	0x83

6.3 Composite Device

The composite device is a USB 2.0-compliant USB composite device running at full speed (12 Mbps). It has the following specification:

Vendor ID: 0x03EB (Atmel)

Product ID: 0x211C (mXT1386)

Version: 16-bit Version & Build Identifier in the form 0xVVBB, where:
VV = Version Major (Upper 4 bits) / Minor (Lower 4 bits)
BB = Build number

The composite device has one bidirectional enpoint: the Control Endpoint (Endpoint 0). It is used by the USB Host to interrogate the USB device for details on its configurations, interfaces and report structures. It is also used to apply general device settings relating to USB Implementation.

6.4 Interface 0 (Digitizer HID)

Interface 0 is a Digitizer-class HID, compliant with HID specification 1.11 with amendments. [1]

This interface consists of a single interrupt-In endpoint (Endpoint 3).

Each Input report consists of a USB Report ID [2] (value 0x01), followed by 5 bytes that describe the status of one active touch (see Figure 6-2).

Figure 6-2. Input Report Packet

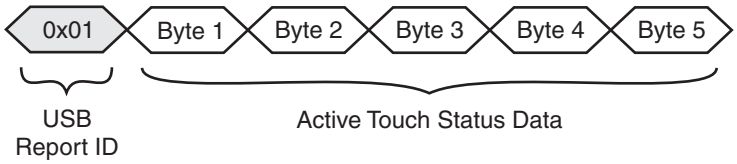


Table 6-2 gives the detailed format of an input report packet.

Table 6-2. Input Report Format

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	USB Report ID							
1	Touch ID					1	1	Status
2	X Position LSByte							
3	0	0	0	0	X Position MSBits			
4	Y Position LSByte							
5	0	0	0	0	Y Position MSBits			

1. This is an implementation of Microsoft's USB HID specification for Multitouch digitizers.
2. The term USB Report ID should not be confused with the term Report Id as used in the Object Protocol; the two are entirely different concepts.

In Table 6-2:

- **Byte 1:**

  Touch ID: Identifies the touch for which this is a status report (starting from 1).

  Bit 3: Always reads “0” (byte alignment padding).

  Bit 2 (Data Valid): Always set to 1.

  Bit 1 (In Range): Always set to 1.

  Status: 1 = In detect, 0 = Not in detect.

- Byte 2 to 5:

  X and Y positions: These are scaled to 12-bit resolution. This means that the upper four bits of the MSByte will always be zero.

There are two update conditions:

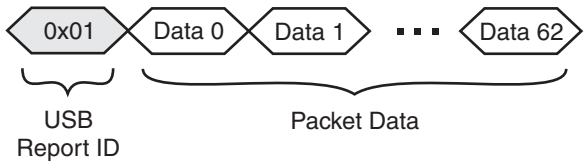
- **Change:** A change in status of any contact (touch) triggers a touch update message to be sent to the host.
- **Idle:** The idle delay of the Digitizer Interface may be controlled via the Control Endpoint as per the HID 1.11 specification (Set Idle command). By default this is set to a delay of 2 (8 ms).

6.5 Interface 1 (Generic HID)

Interface 1 is a Generic Human Interface Device, compliant with HID specification 1.11 with amendments. [1]

It consists of two endpoints: an interrupt-In endpoint (Endpoint 1) and an interrupt-out endpoint (Endpoint 2). The data packet in each case contains a 1-byte USB Report ID followed by 63 bytes of data, totalling 64 bytes (see Figure 6-3).

Figure 6-3. Data Packet for Interface 1



Commands are sent by the application software over the Interrupt-out endpoint, Endpoint 2. The command is sent as the first data byte of the packet data (data byte 0), followed by conditions and/or data.

The supported commands are as follows:

- Read/write Memory Map
- Send Auto-return messages
- Start debug monitoring
- End debug monitoring

Responses from the device are sent via the interrupt-In endpoint, Endpoint 1.

1. This is an implementation of Microsoft’s USB HID specification for Multitouch digitizers.

6.5.1 Read/Write Memory Map

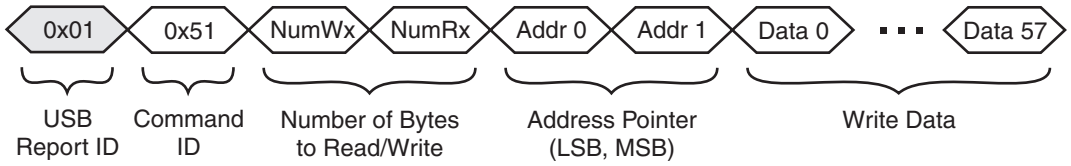
6.5.1.1 Introduction

This command is used to carry out a write/read operation on the memory map of the chip set.

The USB Report ID is 0x01.

The command packet has the generic format given in Figure 6-4. The following sections give examples on using the command to write to the memory map and to read from the memory map.

Figure 6-4. Generic Command Packet Format

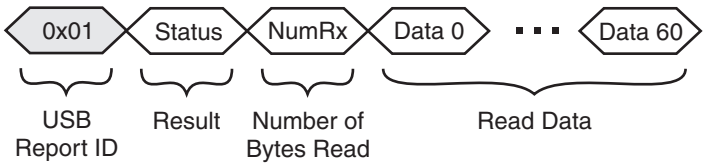


In Figure 6-4:

- **NumWx** is the number of data bytes to write to the memory map (may be zero). If the address pointer is being sent, this must include the size of the address pointer.
- **NumRx** is the number of data bytes to read from the memory map (may be zero).
- **Addr 0** and **Addr 1** form the address pointer to the memory map (where necessary; may be zero if not needed).
- **Data 0** to **Data 57** are the bytes of data to be written (in the case of a write). Note that data locations beyond the number specified by NumWx will be ignored.

The response packet has the generic format given in Figure 6-5.

Figure 6-5. Response Packet Format



In Figure 6-5:

- **Status** indicates the result of the command:

  0x00 = read and write completed; read data returned

  0x04 = write completed; no read data requested
- **NumRx** is the number of bytes following that have been read from the memory map (in the case of a read). This will be the same value as NumRx in the command packet.
- **Data 0** to **Data 60** are the data bytes read from the memory map.

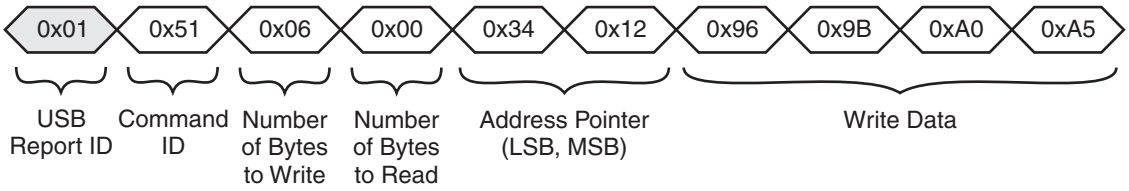
6.5.1.2 *Writing To the Chip Set*

A write operation cycle to the chip set consists of sending a packet that contains six header bytes. These specify the USB report ID, the Command ID, the number of bytes to read, the number of bytes to write, and the 16-bit address pointer.

Subsequent bytes in a multibyte transfer form the actual data. These are written to the location of the address pointer, location of the address pointer +1, location of the address pointer + 2, and so on.

Figure 6-6 shows an example command packet to write four bytes of data to contiguous addresses starting at 0x1234.

Figure 6-6. Example of a Four-byte Write Starting at Address 0x1234

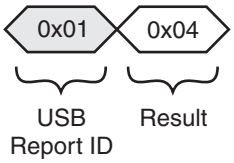


In Figure 6-6:

- The number of bytes to read is set to zero as this is a write-only operation.
- The number of bytes to write is six: that is, four data bytes plus the two address pointer bytes.

Figure 6-7 shows the response to this command. Note that the result status returned is 0x04 (that is, the write operation was completed but no read data was requested).

Figure 6-7. Response to Example Four-byte Write

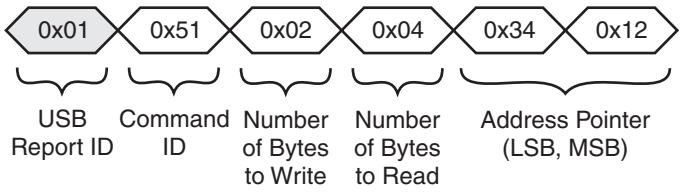


6.5.1.3 *Reading From the Chip Set*

A read operation consists of sending a packet that contains the six header bytes only and no write data.

Figure 6-8 shows an example command packet to read four bytes starting at address 0x1234. Note that the address pointer is included in the number of bytes to write, so the number of bytes to write is set to 2 as there are no other data bytes to be written.

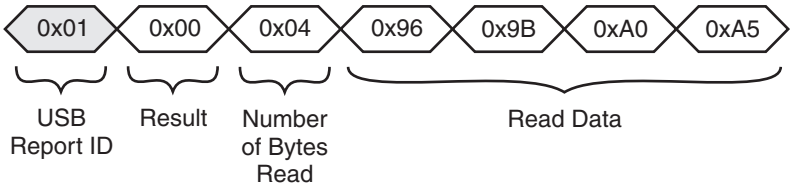
Figure 6-8. Example of a Four-byte Read Starting at Address 0x1234



It is not necessary to set the address pointer before every read. The address pointer is updated automatically after every read operation, so the address pointer will be correct if the reads occur in order.

Figure 6-9 shows the response to this command. The result status returned is 0x00 (that is the write operation was completed and the data was returned). The number of bytes returned will be the same as the number requested (4 in this case).

Figure 6-9. Response to Example Four-byte Read

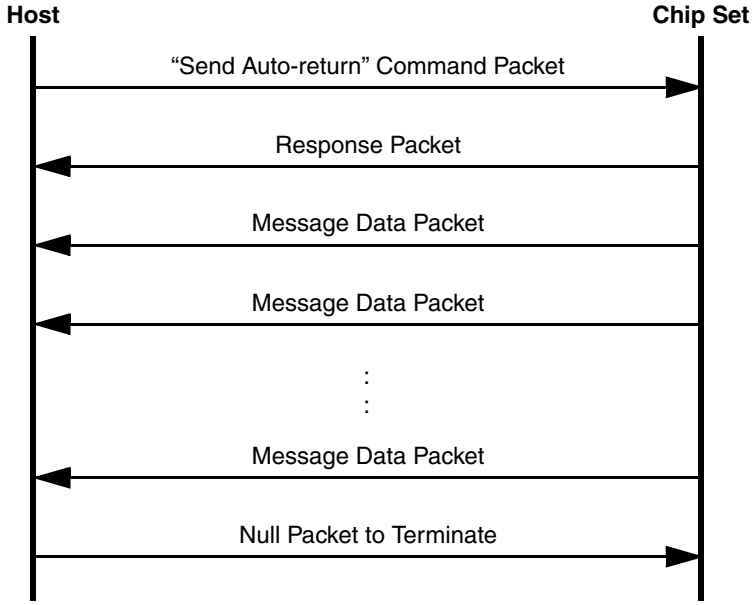


6.5.2 Send Auto-return Messages

6.5.2.1 Introduction

With this command the chip set can be configured to return new messages from the Message Processor object autonomously. The packet sequence to do this is shown in Figure 6-10.

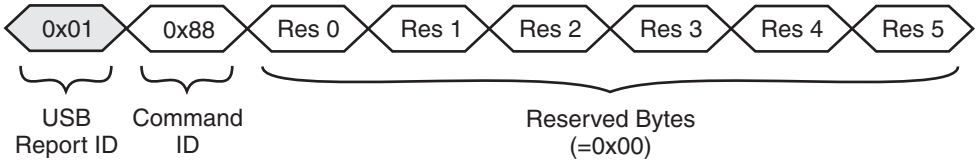
Figure 6-10. Packet Sequence for “Send Auto-return” Command.



The USB Report ID is 0x01.

The command packet has the format given in Figure 6-11.

Figure 6-11. Command Packet Format

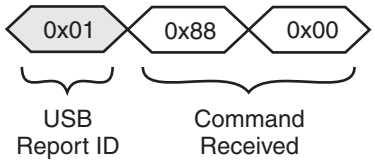


In Figure 6-11:

- **Res 0** to **Res 5** are reserved bytes with a value of 0x00.

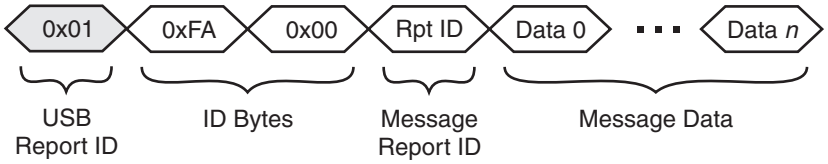
The response packet has the format given in Figure 6-12. Note that with this command, the command packet does not include an address pointer as the chip set already knows the address of the Message Processor object.

Figure 6-12. Response Packet Format



Once the chip set has responded to the command, it starts sending message data. Each time a message is generated in the Message Processor object, the chip set automatically sends a message packet to the host with the data. The message packets have the format given in Figure 6-13.

Figure 6-13. Message Packet Format



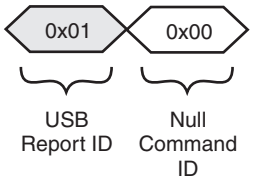
In Figure 6-13:

- **ID Bytes** identify the packet as an auto-return message packet.
- **Rpt ID** is the Report ID returned by the Message Processor object. (1)
- **Message Data** bytes are the bytes of data returned by the Message Processor. The size of the data depends on the source object for which this is the message data. Refer to the *mXT1386 Protocol Guide* for more information.

1. This is the Report ID used in the Object Protocol and should not be confused with the USB Report ID. Refer to the *mXT1386 Protocol Guide* for more information on the use of Report IDs in the the Object Protocol.

To stop the sending of the messages, the host can send a null command packet. This consists of two bytes: a report ID of 0x01 and a command byte of 0x00 (see Figure 6-14).

Figure 6-14. Null Command Packet Format

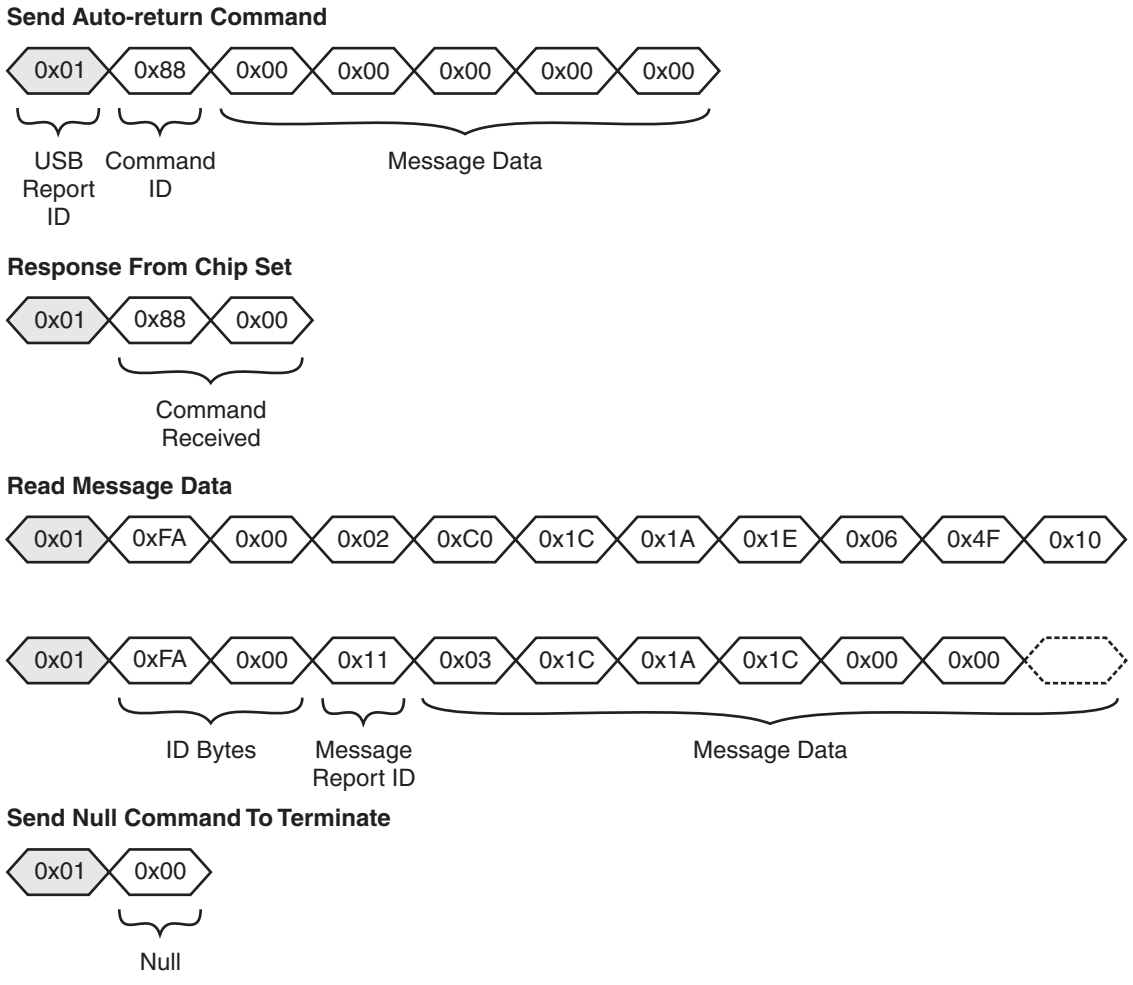


Note that the “Start Debug Monitoring” command may also terminate any currently enabled auto-return mode (see Section 6.5.3).

6.5.2.2 *Reading Status Messages*

Figure 6-8 shows an example sequence of packets to receive messages from the Message Processor object using the “Send Auto-return” command.

Figure 6-15. Example Auto-return Command Packet



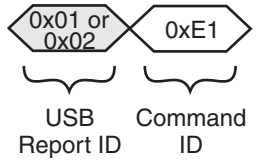
6.5.3 Start Debug Monitoring

This command instructs the device to return debug-monitoring data packets using the debug port, if this feature has been enabled in the Command Processor object.

The USB Report ID can be either 0x01 or 0x02. This allows the source of the request to be identified. The main difference is that a USB Report ID of 0x01 will terminate any currently enabled auto-return mode (see Section 6.5.2 on page 36).

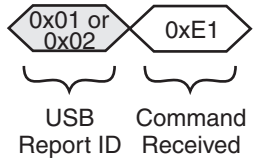
The command packet has the format given in Figure 6-16.

Figure 6-16. Command Packet Format



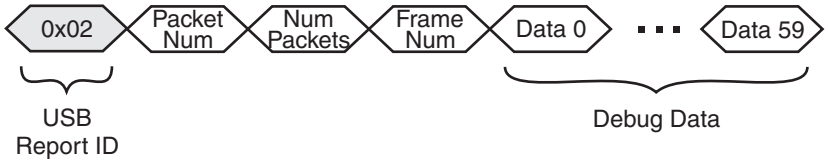
The response packet has the format given in Figure 6-17. Note that the USB Report ID will be the same as that used in the command packet.

Figure 6-17. Response Packet Format



The debug data packet has the format given in Figure 6-18.

Figure 6-18. Debug Data Packet Format



In Figure 6-18:

- **PacketNum** is the number of this USB packet in the debug data frame (full set of debug data). Refer to QTAN0050, *Using the maXTouch Debug Port*, for more information on the format of the debug data.
- **NumPackets** is the total number of USB packets that make up a debug data frame.
- **FrameNum** is the ID number of this frame.
- **Data 0** to **Data 59** are 60 bytes of debug data.

6.5.4 Stop Debug monitoring

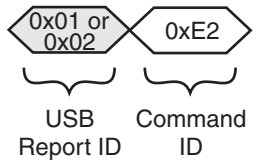
This command instructs the device to cease returning debug-monitoring data packets.

The command packet has the following format:

The USB Report ID is either 0x01 or 0x02.

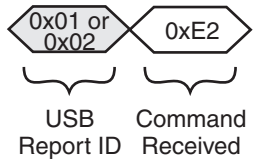
The command packet has the format given in Figure 6-19.

Figure 6-19. Command Packet Format



The response packet has the format given in Figure 6-20.

Figure 6-20. Response Packet Format



6.6 USB Suspend Mode

When the mXT1386 is used in USB configuration, the USB “System Suspend” event can be used to minimize current consumption. Note that it is possible to put the mXT1386 into deep sleep mode without also sending a “System Suspend” event on the USB bus, but the current consumption is not as low. The USB controller must send a USB “System Wake Up” event on the bus to bring the mXT1386 out of suspend mode.

6.6.1 USB “Remote Wakeup” – Firmware Version 1.1 Only

The mXT1386 can also be configured to respond to USB “Remote Wakeup” requests. In this case, if the operating system enables remote wakeup and the mXT1386 is suspended, the chip set will continue to scan at a preset sensor refresh rate. Use of the remote wake up feature and the sensor refresh rate are configured using the Digitizer HID Configuration object (refer to the *mXT1386 Protocol Guide* for more information).

7. Getting Started With the mXT1386

7.1 Establishing Contact

7.1.1 Communication with the Host

The host can use either the I²C-compatible bus (see Section 5.1 on page 24) or the USB interface (see Section 6.1 on page 31) to communicate with the chip set.

7.1.2 I²C-compatible Interface

On power-up, the $\overline{\text{CHG}}$ line goes low to indicate that there is new data to be read from the Message Processor object. If the $\overline{\text{CHG}}$ line does not go low, there is a problem with the chip set.

The host should attempt to read any available messages to establish that the chip set is present and running following power-up or a reset. Examples of messages include reset or calibration messages. The host should also check that there are no configuration errors reported.

7.1.3 USB Interface

The host can establish contact with the chip set as specified in the USB 2.0 specification and the USB HID specification (both available from www.usb.org).

7.2 Using the Object Protocol

The chip set has an object-based protocol that is used to communicate with the chip set. Typical communication includes configuring the chip set, sending commands to the chip set, and receiving messages from the chip set. Refer to the *mXT1386 Protocol Guide* for more information.

The host must perform the following initialization so that it can communicate with the chip set:

1. Read the start positions of all the objects in the chip set from the Object Table and build up a list of these addresses.
2. Use the Object Table to calculate the report IDs so that messages from the chip set can be correctly interpreted.

7.3 Writing to the Chip Set

There are two mechanisms for writing to the chip set:

- Using an I²C-compatible write operation (see Section 6.5.1.2 on page 35).
- Using the USB Generic HID's "Read/Write Memory Map" command (see Section 6.5.1 on page 34).

To communicate with the chip set, you write to the appropriate object:

- To send a command to the chip set, you write the appropriate command to the Command Processor object (refer to the *mXT1386 Protocol Guide*).
- To configure the chip set, you write to an object. For example, to configure the chip set's power consumption you write to the global Power Configuration object, and to set up a touchscreen you write to a Multiple Touch Touchscreen object. Some objects are optional and need to be enabled before use. Refer to the *mXT1386 Protocol Guide* for more information on the objects.

7.4 Reading from the Chip Set

Status information is stored in the Message Processor object. This object can be read to receive any status information from the chip set. There are two mechanisms that provide an interrupt-style interface for reading messages in the Message Processor object:

- When using the I²C-compatible interface, the $\overline{\text{CHG}}$ line is asserted whenever a new message is available in the Message Processor object (see Section 5.7 on page 28). See Section 6.5.1.3 on page 35 for information on the format of the I²C-compatible read operation.
- When using the USB interface, the Generic HID interface provides an interrupt-driven interface that sends the messages automatically (See Section 6.5.2 on page 36).

Note that in both cases the host should always wait to be notified of messages. The host should not poll the chip set for messages.

The USB Digitizer HID provides a third alternative interrrupt-style mechanism for reading a subset of the touch data. See Section 6.4 on page 32 for more information.

7.5 Configuring the Chip Set

The objects are designed such that a default value of zero in their fields is a “safe” value that typically disables functionality. The objects must be configured before use and the settings written to the nonvolatile memory using the Command Processor object. Refer to the *mXT1386 Protocol Guide* for more information.

The following objects must be configured before use:

- Power Configuration
  - Set up the Idle Acquisition Interval, Active Acquisition Interval and Active to Idle Timeout.
- Acquisition Configuration

The following objects should also be configured and enabled, as required:

- Touch objects: Multiple Touch Touchscreen, Key Array
  - Enable the object.
  - Configure the origin and the number of channels it occupies. Configure the other fields in the object, as required. For example, set up the AKS group(s), specify the burst length and threshold.
  - Enable reporting to receive touch messages from the object.
- Signal processing objects: One-touch Gesture Processor, Two-touch Gesture Processor, Grip Suppression, Palm Suppression, Noise Suppression
  - Enable the object.
  - Configure the fields in the object, as required.
  - Enable reporting to receive signal processing messages from the object.
- Support objects: Communications Configuration, CTE Configuration, Self Test, User Data
  - Enable the object, if the object requires it.
  - Configure the fields in the object, as required.
  - Enable reporting, if the object supports messages, to receive messages from the object.

Refer to the *mXT1386 Protocol Guide* for more information on configuring the objects.

8. Specifications

8.1 Absolute Maximum Specifications

Vdd	3.6V
AVdd	3.6V
DP, DM and VBUS pins	5.5V
Max continuous pin current, any control or drive pin	20 mA
Voltage forced onto any pin	-0.5V to (Vdd or AVdd)+0.5V
Configuration parameters maximum writes	10,000

CAUTION: Stresses beyond those listed under *Absolute Maximum Specifications* may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum specification conditions for extended periods may affect device reliability.

8.2 Recommended Operating Conditions

Operating temp	-20°C to +85°C
Storage temp	-65°C to +150°C
Vdd	3.3V ±5%
AVdd	3.3V ±5%
Vdd vs AVdd power sequencing	No sequencing required
Supply ripple + noise	See Section 8.11 on page 47
Cx transverse load capacitance per channel	0.63 pF to 5 pF

8.3 DC Specifications

8.3.1 Digital Power (DVdd_3V3)

Parameter	Description	Min	Typ	Max	Units	Notes
Vdd	Operating limits	3.14	3.3	3.47	V	Common to master and slaves
Vil	Low input logic level	-0.3		+0.8	V	
Vih	High input logic level	2		3.6	V	
Vol	Low output voltage			0.4	V	
Voh	High output voltage	Vdd-0.4			V	
Iil	Input leakage current			1	µA	

8.3.2 Analog Power (AVdd_3V3_A/B/C)

Parameter	Description	Min	Typ	Max	Units	Notes
AVdd	Operating limits	3.14	3.3	3.47	V	
Slew rate	Minimum slew rate	1			V/100 µs	

Note: AVdd must be stable and have a nominal tolerance in the host system of ±5% or better. The VOLTAGE field of the CTE Configuration object must be configured with the Avdd voltage level (60 = 3.3V). The VOLTAGE setting must be within ±5% of the voltage applied to the slave devices.

8.4 Supply Current

Note: The tables below list the total current consumed from both Vdd and AVdd power supplies for the touch conditions listed (see the schematics in Appendix A on page 56).

8.4.1 Using the I²C-compatible Interface

XSIZE = 27, YSIZE = 42, CHRGTIME = 2.5 µs, IDLE/ACTVGCAFDEPTH = 8

Parameter	Description	Min	Typ	Max	Units	Notes
Isupply	Active average supply current		26.2		mA	100Hz (IDLE/ACTVACQINT = 50), 1 moving touch,
	Idle average supply current		5.1		mA	20Hz (IDLE/ACTVACQINT = 50)
	Sleep average supply current		95		µA	IDLE/ACTVACQINT = 0

8.4.2 Using the USB Bus

XSIZE = 27, YSIZE = 42, CHRGTIME = 2.5 µs, IDLE/ACTVGCAFDEPTH = 8

Parameter	Description	Min	Typ	Max	Units	Notes
Isupply	Active average supply current		29.3		mA	100Hz (IDLE/ACTVACQINT = 50), 1 moving touch,
	Idle average supply current		8.1		mA	20Hz (IDLE/ACTVACQINT = 50)
	Sleep average supply current		4.9		mA	IDLE/ACTVACQINT = 0
	Suspend (remote wakeup); Firmware version 1.1 only		5		mA	10Hz (IDLE/ACTVACQINT = 100), RWKEN = 1, RWKRATE = 100, With USB hub

8.5 Reset Timings

Parameter	Min	Typ	Max	Units	Notes
Power on to $\overline{\text{CHG}}$ line low		90		ms	
Hardware reset to $\overline{\text{CHG}}$ line low		90		ms	
Software reset to $\overline{\text{CHG}}$ line low		250		ms	
USB soft reset to device ready		90		ms	

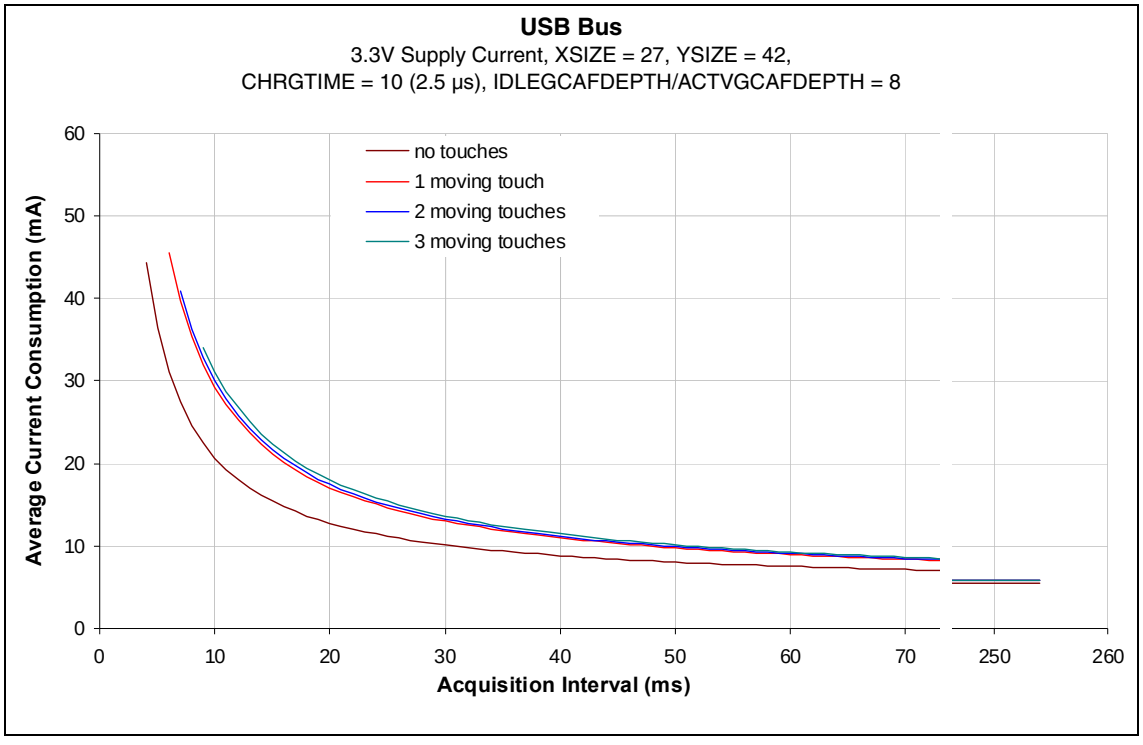
8.6 I²C-compatible Bus Specifications

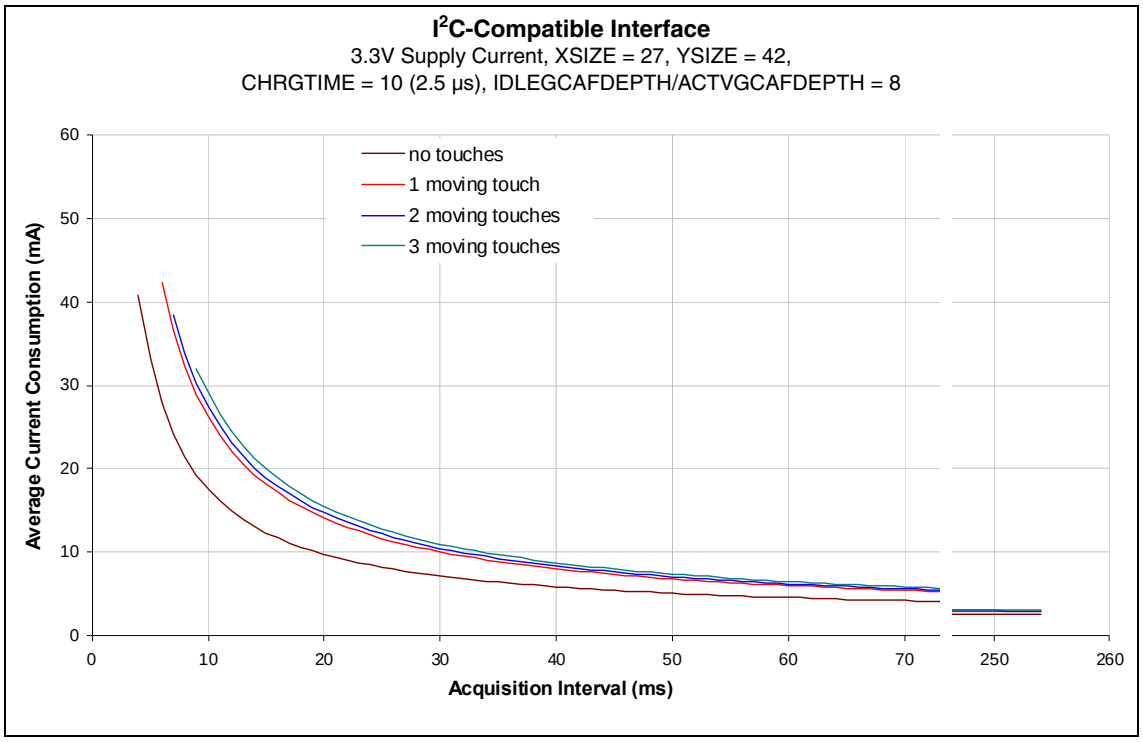
Parameter	Operation
Addresses	0x4C, 0x4D, 0x5A or 0x5B
Maximum bus speed (SCL)	400 kHz
I²C specification	Version 2.1

8.7 USB Bus Specification

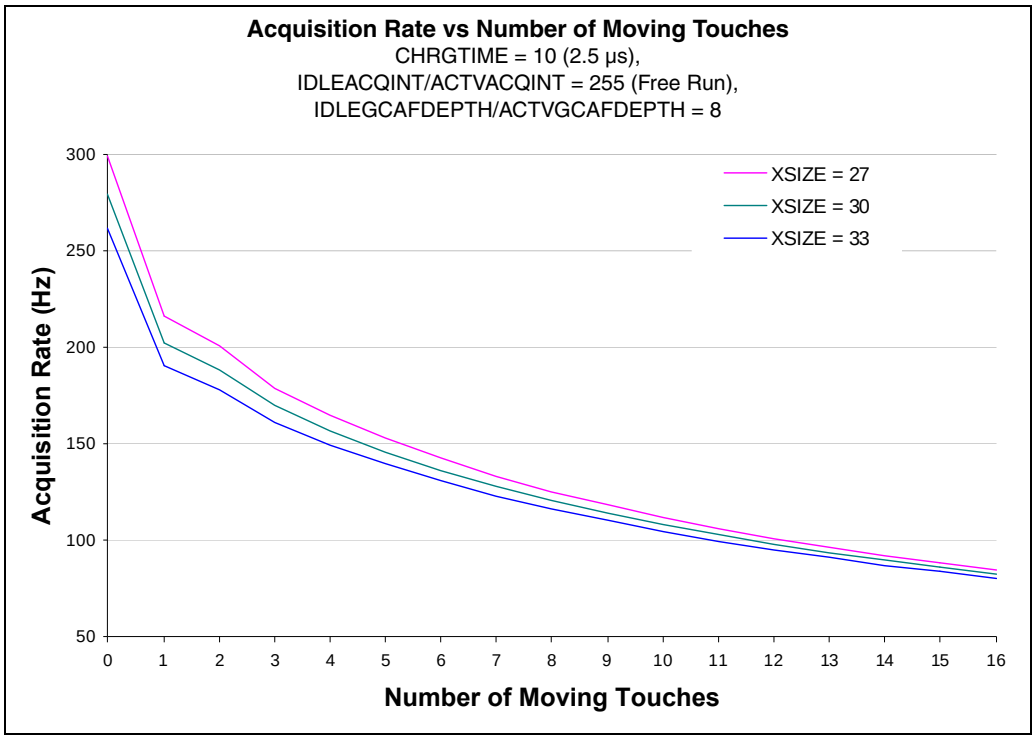
Parameter	Operation
Endpoint Addresses	0x81 (Endpoint 1) 0x02 (Endpoint 2) 0x83 (Endpoint 3)
Maximum bus speed	12 Mbps
Vendor ID	0x03EB (Atmel)
Product ID	0x211C (mXT1386)
USB specification	USB 2.0 HID specification 1.11 with amendments for multitouch digitizers

8.8 Power Consumption




I²C-Compatible Interface
3.3V Supply Current, XSIZE = 27, YSIZE = 42,
CHRGTIME = 10 (2.5 µs), IDLEGCAFDEPTH/ACTVGCAFDEPTH = 8
Average Current Consumption (mA)
Acquisition Interval (ms)
no touches
1 moving touch
2 moving touches
3 moving touches
0
10
20
30
40
50
60
70
250
260

8.9 Speed


Acquisition Rate vs Number of Moving Touches
CHRGTIME = 10 (2.5 µs),
IDLEACQINT/ACTVACQINT = 255 (Free Run),
IDLEGCAFDEPTH/ACTVGCAFDEPTH = 8
Acquisition Rate (Hz)
Number of Moving Touches
XSIZE = 27
XSIZE = 30
XSIZE = 33
300
250
200
150
100
50
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16

8.10 Touch Accuracy and Repeatability

Touchscreen pitch= 4.7 mm, front panel = 1 mm, touch size = 8 mm

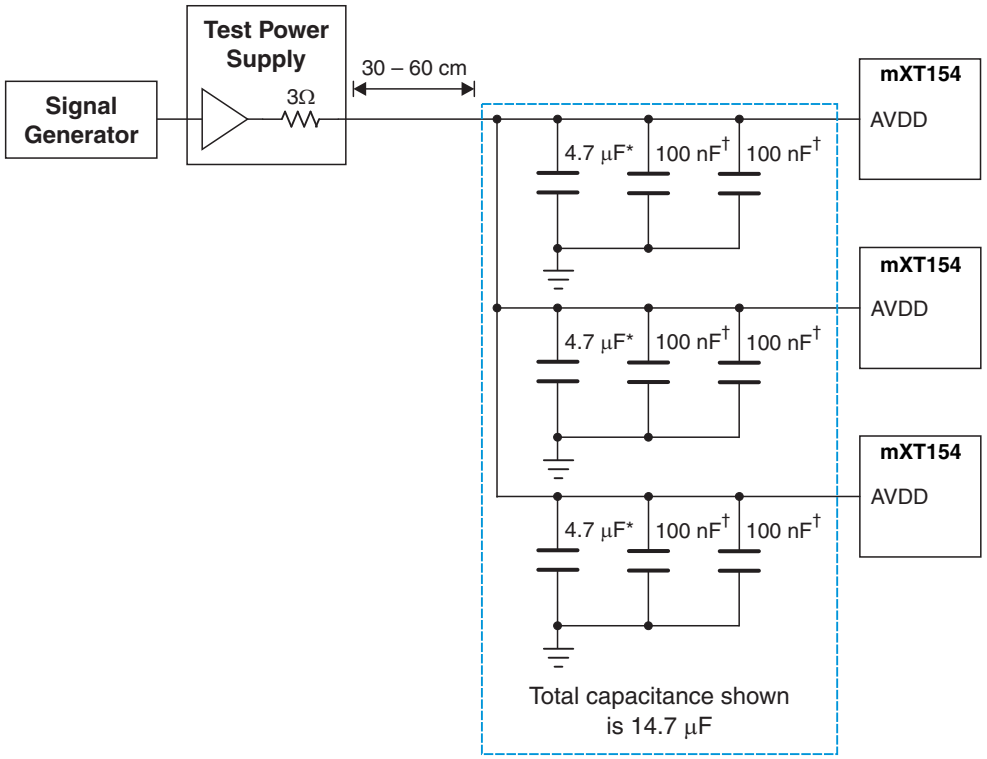
Parameter	Min	Typ	Max	Units	Notes
Linearity		±0.5		mm	
Accuracy		±1		mm	
Accuracy edge		±2		mm	
Repeatability		±0.25		%	X axis with 12-bit resolution

8.11 Power Supply Ripple and Noise

See Section 8.11.1 on page 48 and Section 8.11.2 on page 49 for the power supply ripple and noise characterization charts.

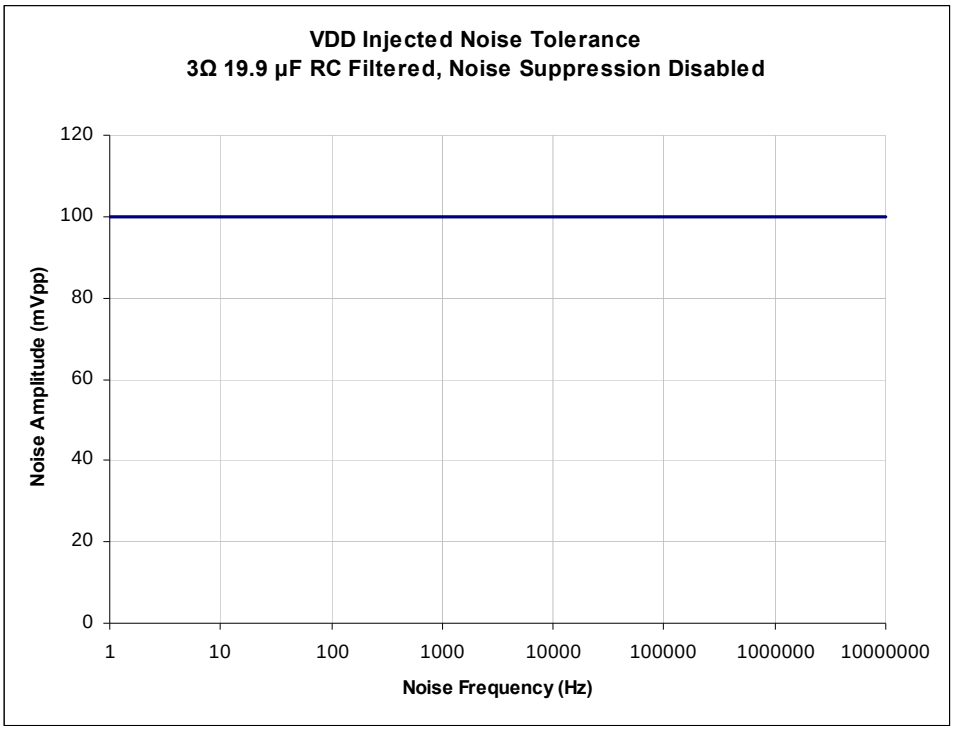
The test circuit used for the charts is shown in Figure 8-1.

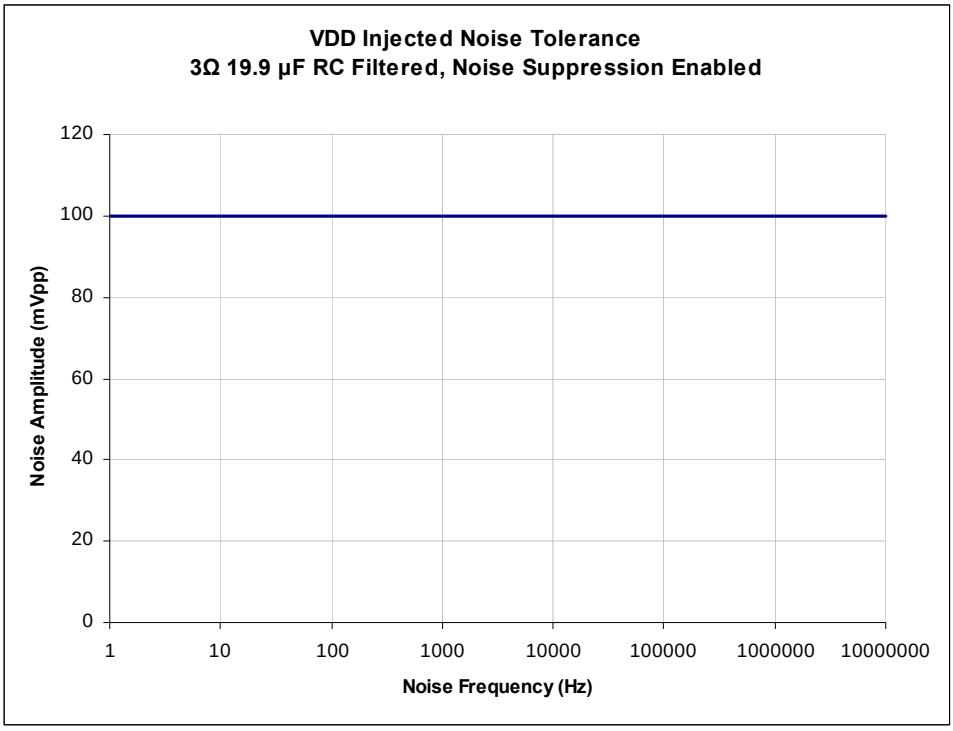
Figure 8-1. Circuit Used for Power Supply Ripple Characterization Charts



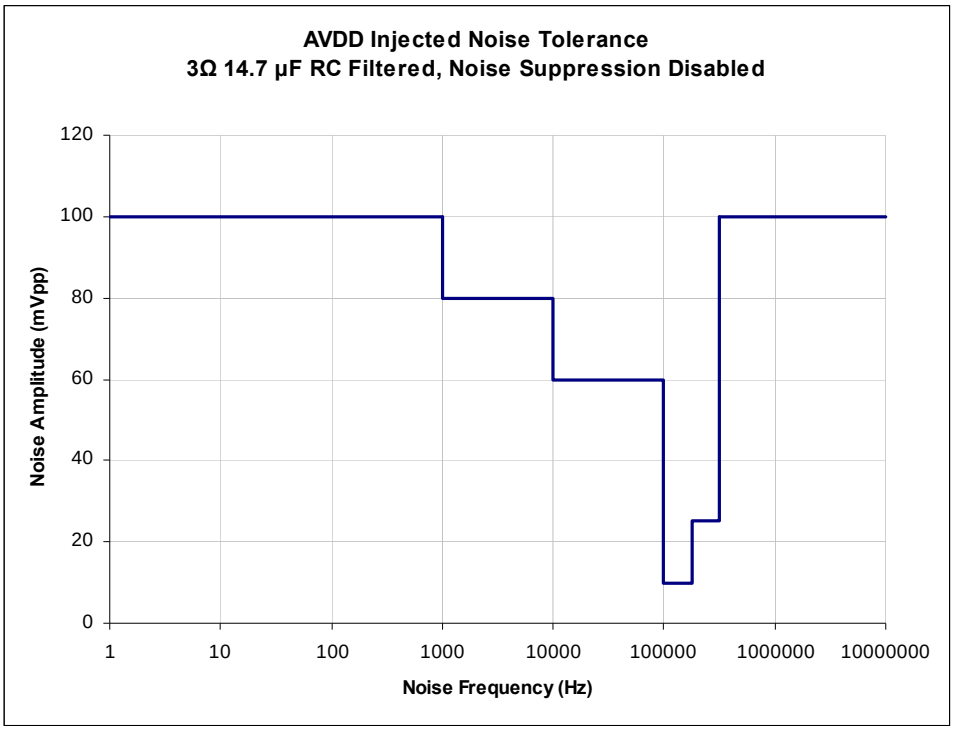
NOTES: *Bypass capacitors are <5 mm away from the chip.
† Bypass capacitors are <2 mm away from the chip.

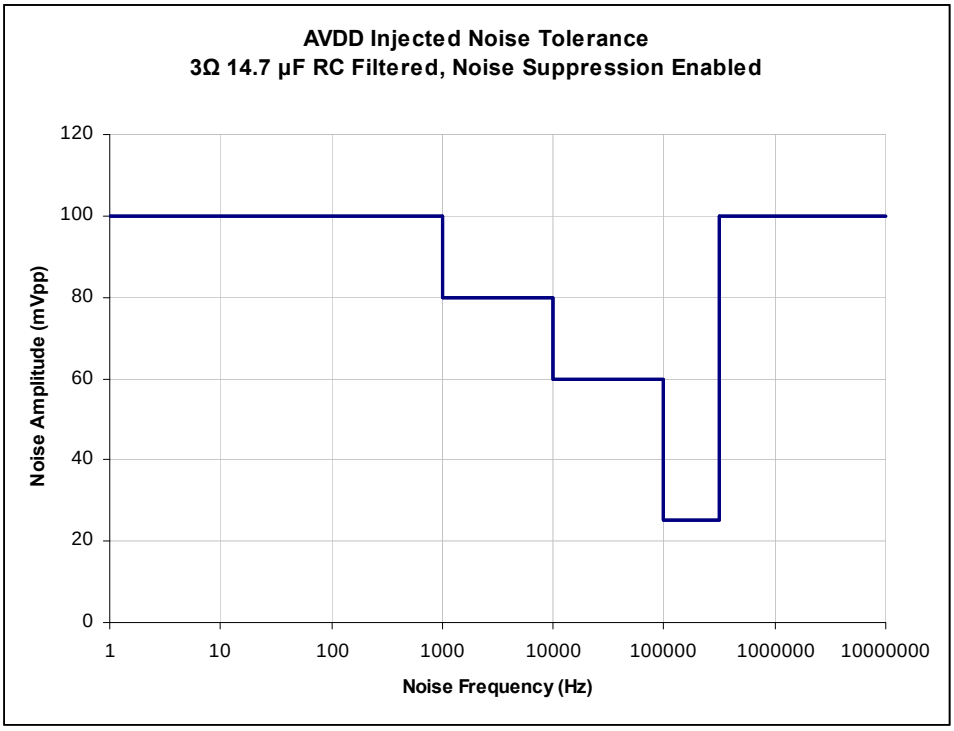
8.11.1 Digital Power (Vdd) Characterization





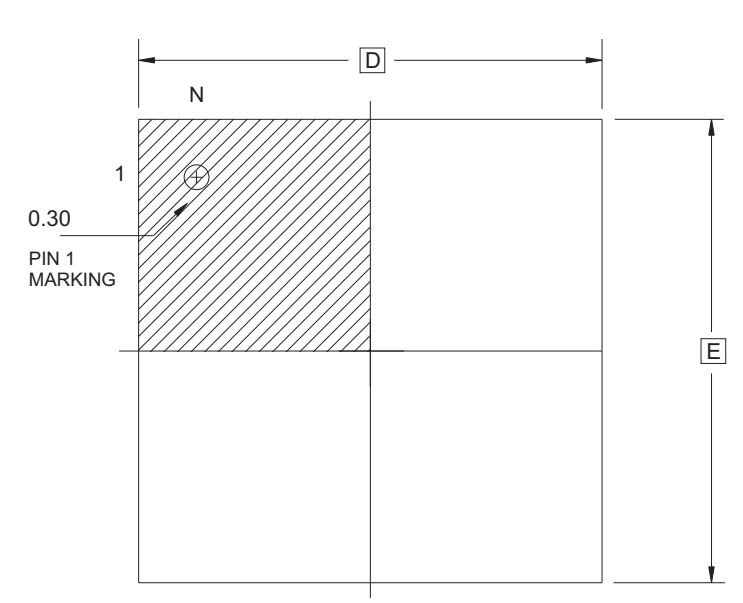
8.11.2 Analog Power (AVdd) Characterization



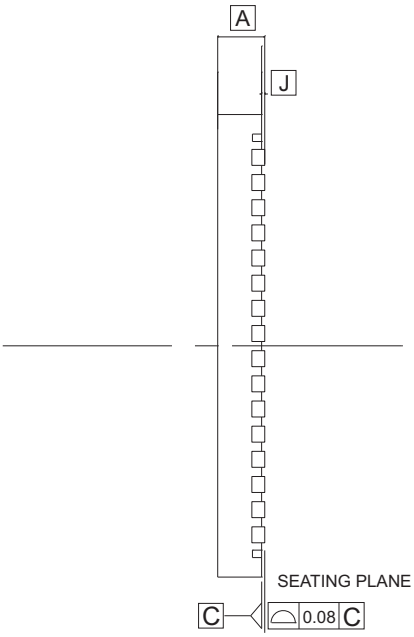


8.12 Mechanical Dimensions

8.12.1 ATMXT1386-Z2U – 64-pin QFN

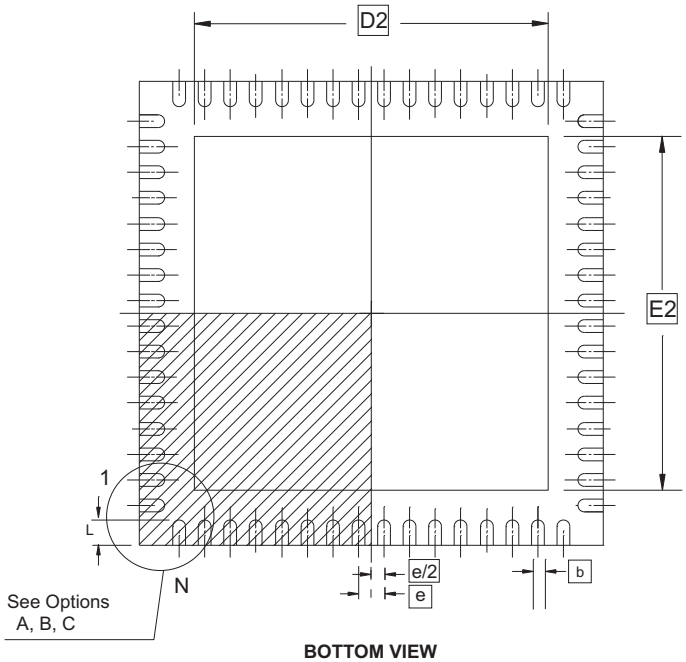


TOP VIEW



SIDE VIEW

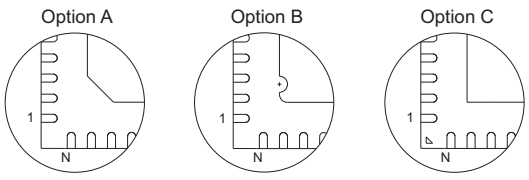
DRAWINGS NOT SCALED



BOTTOM VIEW

COMMON DIMENSIONS
(Unit of Measure = mm)

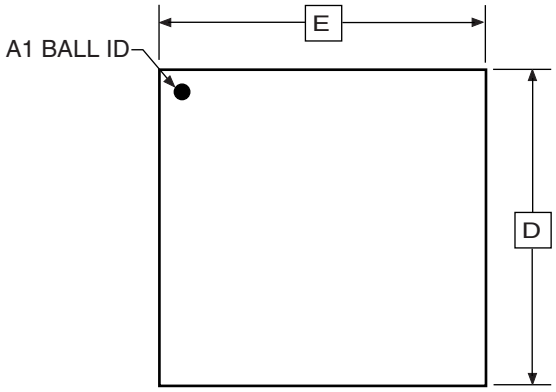
SYMBOL	MIN	NOM	MAX	NOTE
A	0.08	–	1.00	
J	0.00		1.05	
D/E		9.00 BSC		
D2/E2	3.25	–	7.50	
N		64		
e		0.50 BSC		
L	0.30	0.40	0.55	
b	0.18	0.25	0.30	



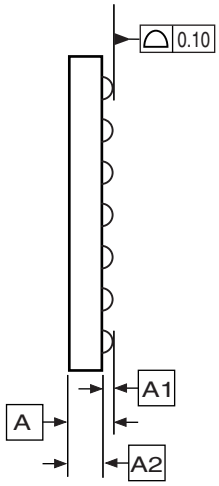
6/24/10

Package Drawing Contact: touch@atmel.com	TITLE	DRAWING NO.	REV.
	64Z2, 64 Leads - body 9.0 x 9.0 mm - pitch 0.5 mm Quad Flat No Lead Package (QFN)	64Z2	A

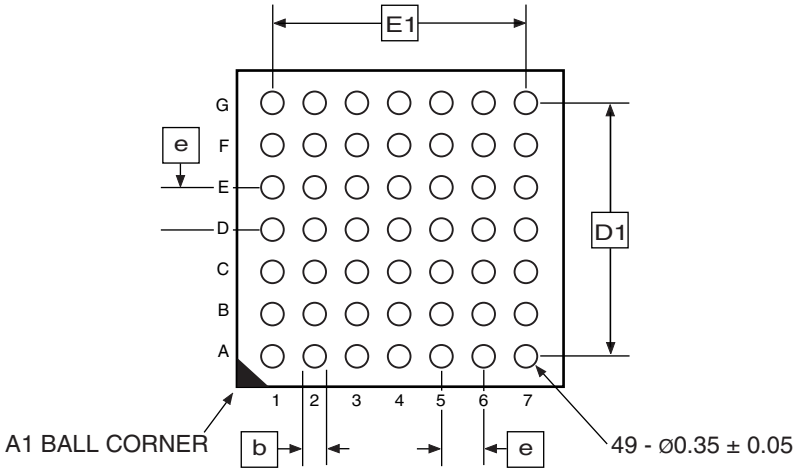
8.12.2 ATMXT154-CU – 49-ball VFBGA



TOP VIEW



SIDE VIEW



BOTTOM VIEW

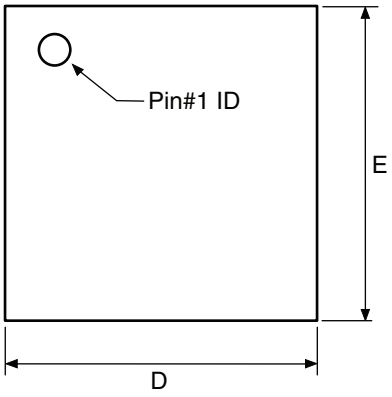
COMMON DIMENSIONS
(Unit of Measure = mm)

SYMBOL	MIN	NOM	MAX	NOTE
A	–	–	1.00	
A1	0.20	–	–	
A2	0.65	–	–	
D	4.90	5.00	5.10	
D1	3.90 BSC			
E	4.90	5.00	5.10	
E1	3.90 BSC			
b	0.30	0.35	0.40	
e	0.65 BSC			

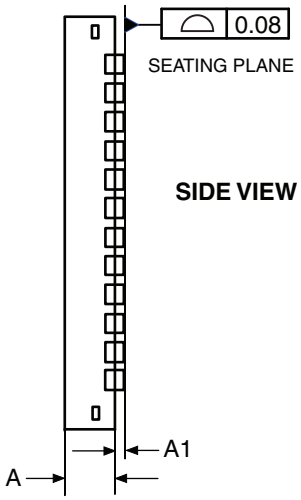
3/14/08

	TITLE	GPC	DRAWING NO.	REV.
Package Drawing Contact: packagedrawings@atmel.com	**49C2,** 49-ball (7 x 7 Array), 0.65 mm Pitch, 5.0 x 5.0 x 1.0 mm, Very Thin, Fine-Pitch Ball Grid Array Package (VFBGA)	CBD	49C2	A

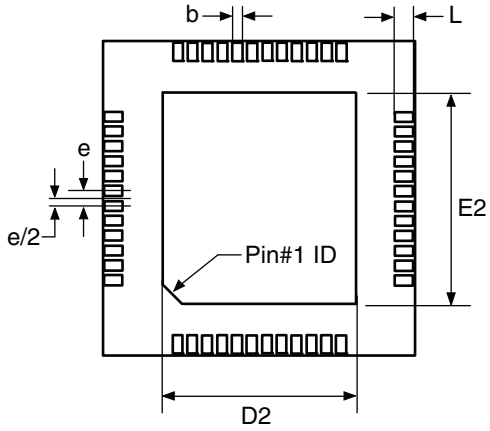
8.12.3 ATMXT154-MAH – 48-pin QFN



TOP VIEW



SIDE VIEW



BOTTOM VIEW

COMMON DIMENSIONS
(Unit of Measure = mm)

SYMBOL	MIN	NOM	MAX	NOTE
A	–	–	0.60	
A1	0.00	–	0.05	
b	0.15	0.20	0.25	2
D		6.00 BSC		
D2	4.40	4.50	4.60	
E		6.00 BSC		
E2	4.40	4.50	4.60	
e	–	0.40	–	
L	0.35	0.40	0.45	

Note1: Refer to JEDEC Drawing MO-248, variation UHHE-1 (saw singulation).

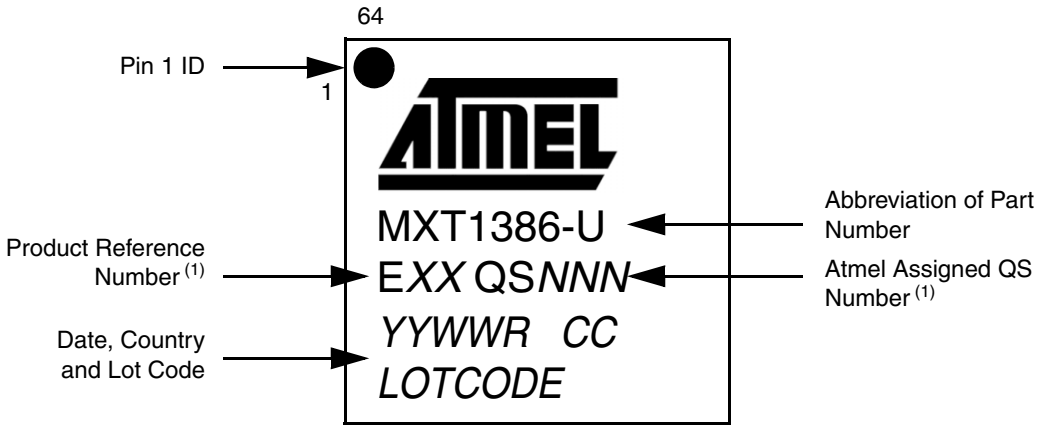
Note2: Dimension “b” refers to metalized terminal and is measured between 0.15 and 0.30mm from the terminal tip. If the terminal has the optional radius on the other end of the terminal, the dimension should not be measured in that radius area.

30/06/09

	TITLE	GPC	DRAWING NO.	REV.
Package Drawing Contact: packagedrawings@atmel.com	**48MA1,** 48 pad, 6 x 6 x 0.6mm body, 0.40mm pitch,4.5 x 4.5mm exposed pad, Saw singulated Thermally enhanced plastic Ultra thin quad flat no lead package (UQFN).	ZAL	48MA1	A

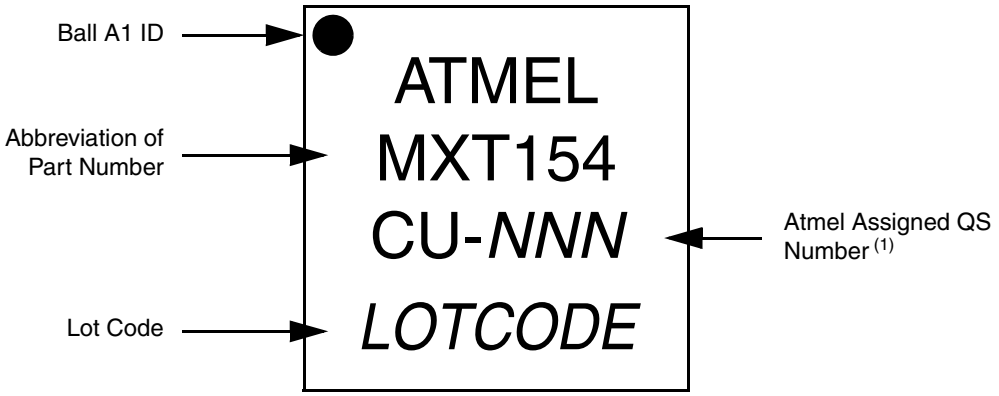
8.13 Part Markings

8.13.1 ATMXT1386-Z2U – 64-pin QFN



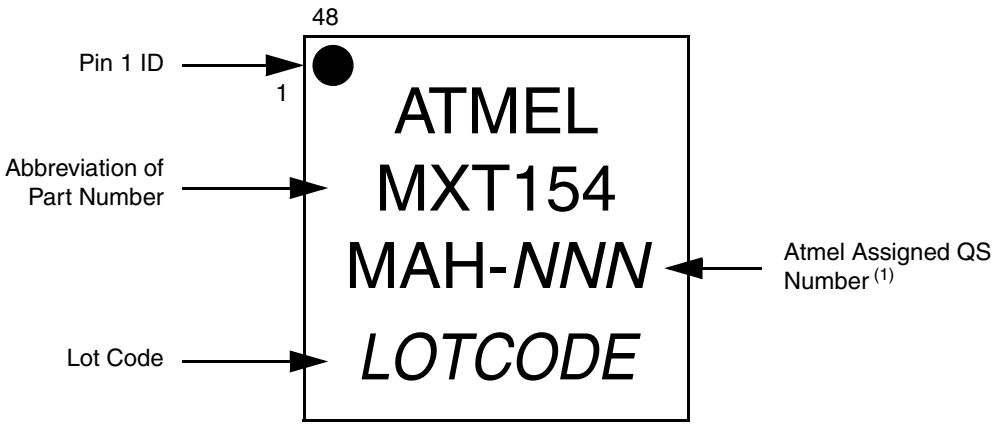
1. See Section 8.14.2 on page 55 for the relevant Product Reference Number and QS Number.

8.13.2 ATMXT154-CU – 49-ball VFBGA



1. See Section 8.14.2 on page 55 for the relevant QS Number.

8.13.3 ATMXT154-MAH – 48-pin QFN



1. See Section 8.14.2 on page 55 for the relevant QS Number.

8.14 Part Numbers

8.14.1 Orderable Chip Set Bundles

8.14.1.1 Firmware 1.0

Orderable Part Number	QS Number	Description
ATMXT1386-CHPSET1	–	Consists of: • 1 x ATMXT1386-Z2UR (supplied in tape and reels) • 3 x ATMXT154-CUR (supplied in tape and reels)
ATMXT1386-CHPSET1U	–	
ATMXT1386-CHPSET2	–	Consists of: • 1 x ATMXT1386-Z2UR (supplied in tape and reels) • 3 x ATMXT154-MAHR (supplied in tape and reels)
ATMXT1386-CHPSET2U	–	
ATMXT1386-CHPSET3	–	Consists of: • 1 x ATMXT1386-Z2U (supplied in trays) 3 x ATMXT154-CU (supplied in trays)
ATMXT1386-CHPSET3U	–	
ATMXT1386-CHPSET4	–	Consists of: • 1 x ATMXT1386-Z2U (supplied in trays) • 3 x ATMXT154-MAH (supplied in trays)
ATMXT1386-CHPSET4U	–	

8.14.1.2 Firmware 1.1

Orderable Part Number	QS Number	Description
ATMXT1386-CHPSET1	QS550	Consists of: • 1 x ATMXT1386-Z2UR (supplied in tape and reels) • 3 x ATMXT154-CUR (supplied in tape and reels)
ATMXT1386-CHPSET1U	QS548	
ATMXT1386-CHPSET2	QS550	Consists of: • 1 x ATMXT1386-Z2UR (supplied in tape and reels) • 3 x ATMXT154-MAHR (supplied in tape and reels)
ATMXT1386-CHPSET2U	QS548	
ATMXT1386-CHPSET3	QS550	Consists of: • 1 x ATMXT1386-Z2U (supplied in trays) 3 x ATMXT154-CU (supplied in trays)
ATMXT1386-CHPSET3U	QS548	
ATMXT1386-CHPSET4	QS550	Consists of: • 1 x ATMXT1386-Z2U (supplied in trays) • 3 x ATMXT154-MAH (supplied in trays)
ATMXT1386-CHPSET4U	QS548	

8.14.2 Orderable Individual Parts

8.14.2.1 Firmware 1.0

Orderable Part Number	Product Reference	QS Number [1]	Description
ATMXT1386-Z2UIR (tape and reels)	EEI	QS524	64-pin 9 x 9 mm QFN RoHS compliant
	EEV	QS537	
ATMXT1386-Z2UI (trays)	EEI	QS524	64-pin 9 x 9 mm QFN RoHS compliant
	EEV	QS537	
ATMXT154-CUIR (tape and reels)	–	QS520	49-ball 5 x 5 mm VFBGA RoHS compliant
ATMXT154-CUI (trays)	–	QS520	
ATMXT154-MAHIR (tape and reels)	–	QS521	48-pin 6 x 6 mm QFN RoHS compliant
ATMXT154-MAHI (trays)	–	QS521	

1. Alternative QS Numbers represent different die foundaries.

8.14.2.2 Firmware 1.1

Orderable Part Number	Product Reference	QS Number [1]	Description
ATMXT1386-Z2UIR (tape and reels)	EFH	QS540	64-pin 9 x 9 mm QFN RoHS compliant
	EFI	QS542	
ATMXT1386-Z2UI (trays)	EFH	QS540	64-pin 9 x 9 mm QFN RoHS compliant
	EFI	QS542	
ATMXT154-CUIR (tape and reels)	–	QS545	49-ball 5 x 5 mm VFBGA RoHS compliant
ATMXT154-CUI (trays)	–	QS545	
ATMXT154-MAHIR (tape and reels)	–	QS545	48-pin 6 x 6 mm QFN RoHS compliant
ATMXT154-MAHI (trays)	–	QS545	

1. Alternative QS Numbers represent different die foundaries.

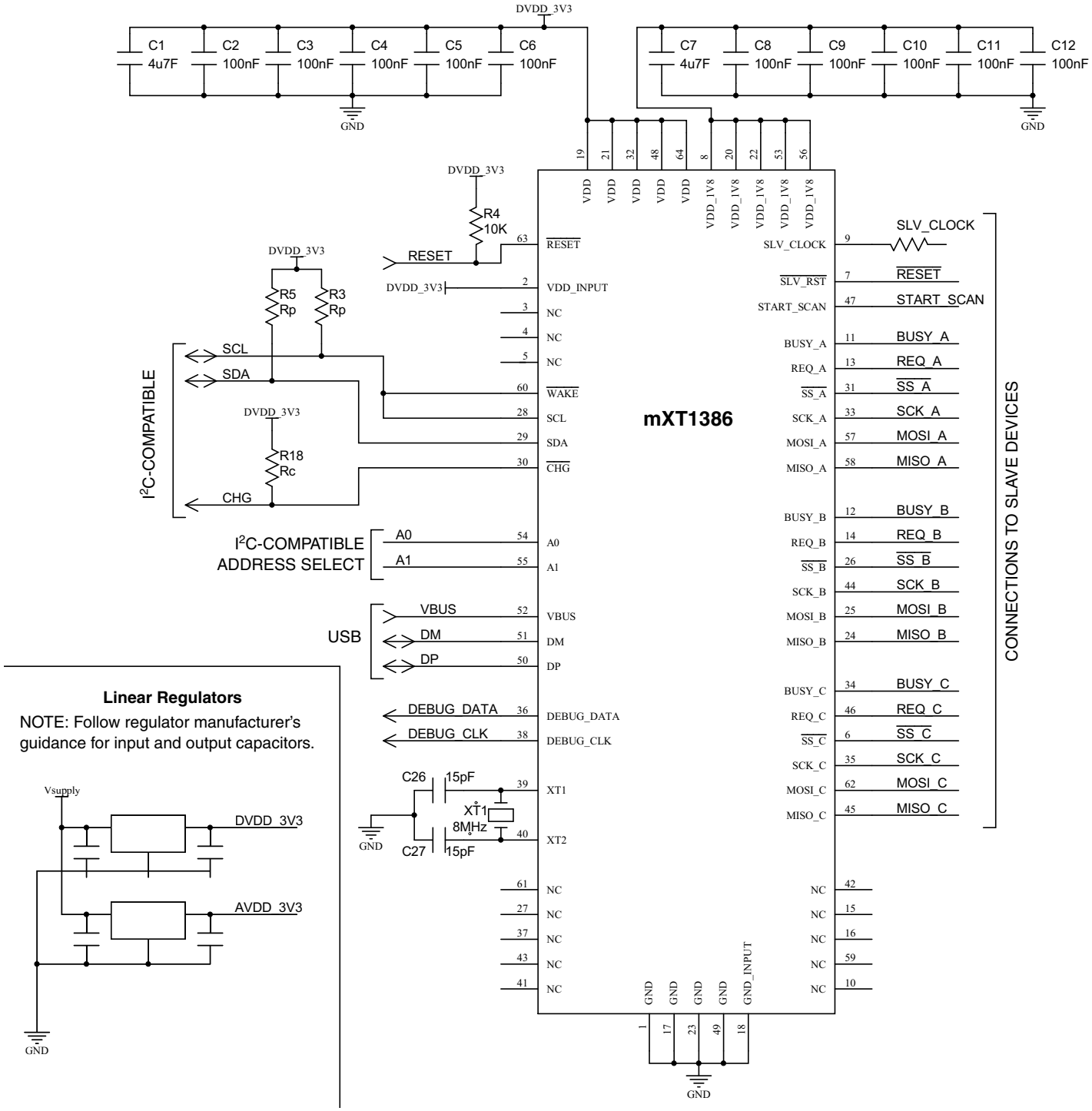
8.15 Moisture Sensitivity Level (MSL)

MSL Rating	Peak Body Temperature	Specifications
MSL3	260°C	IPC/JEDEC J-STD-020

Appendix A. Schematics and PCB Design Considerations

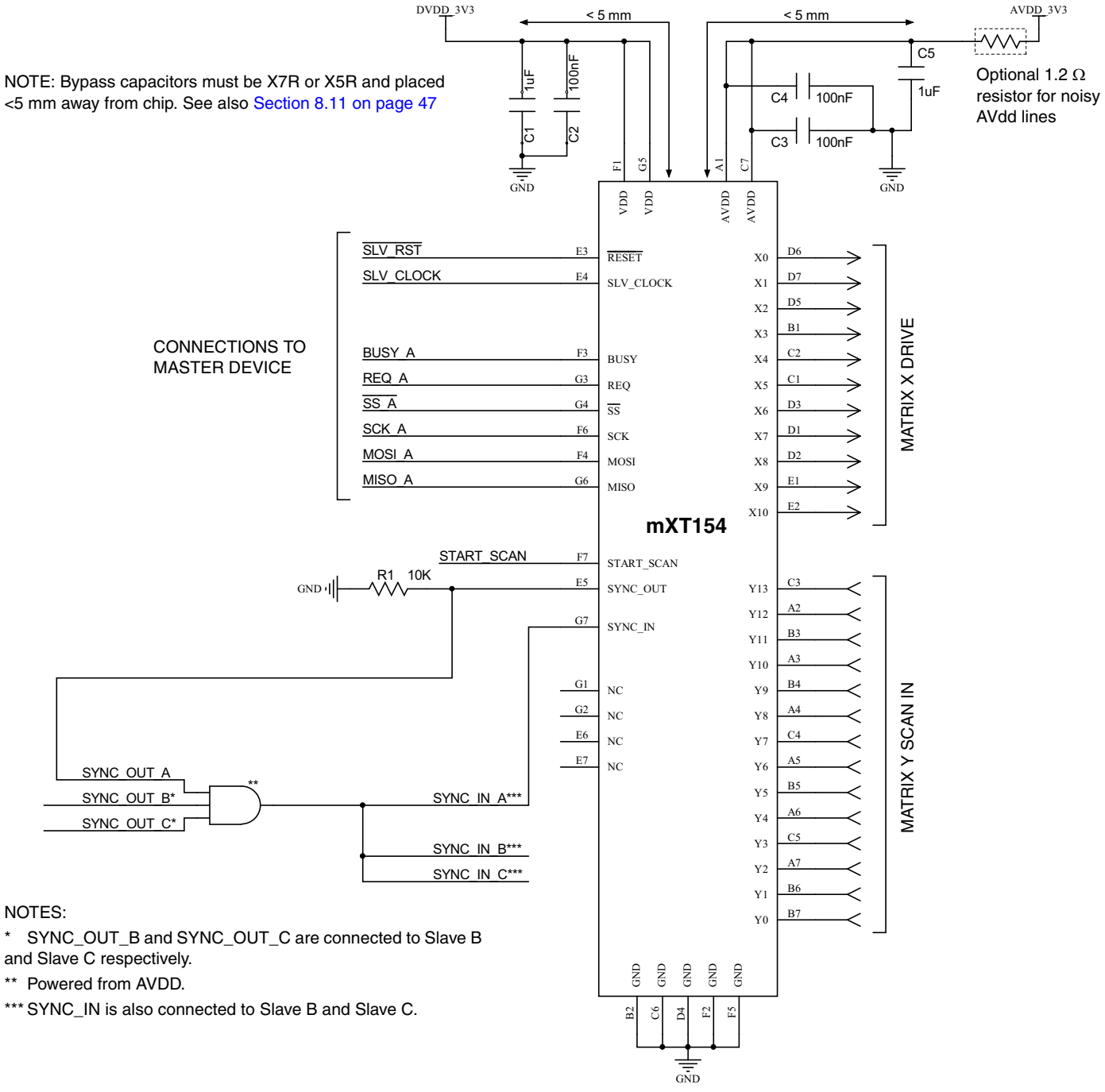
A.1 Master Device (mXT1386) – 64-pin QFN

Notes:
1. Capacitors C2 – C6 and C8 – C12 must be X7R or X5R and placed <5 mm away from the pins for which they act as bypass capacitors. See also Section 8.11 on page 47
2. Either I²C-compatible or USB interface can be used, but only one interface should be used in any one design.



A.2 Slave Devices (3 x mXT154) – 49-ball VFBGA

Note: Instance Slave A only is shown; Slave B and Slave C are omitted for simplicity.



NOTES:

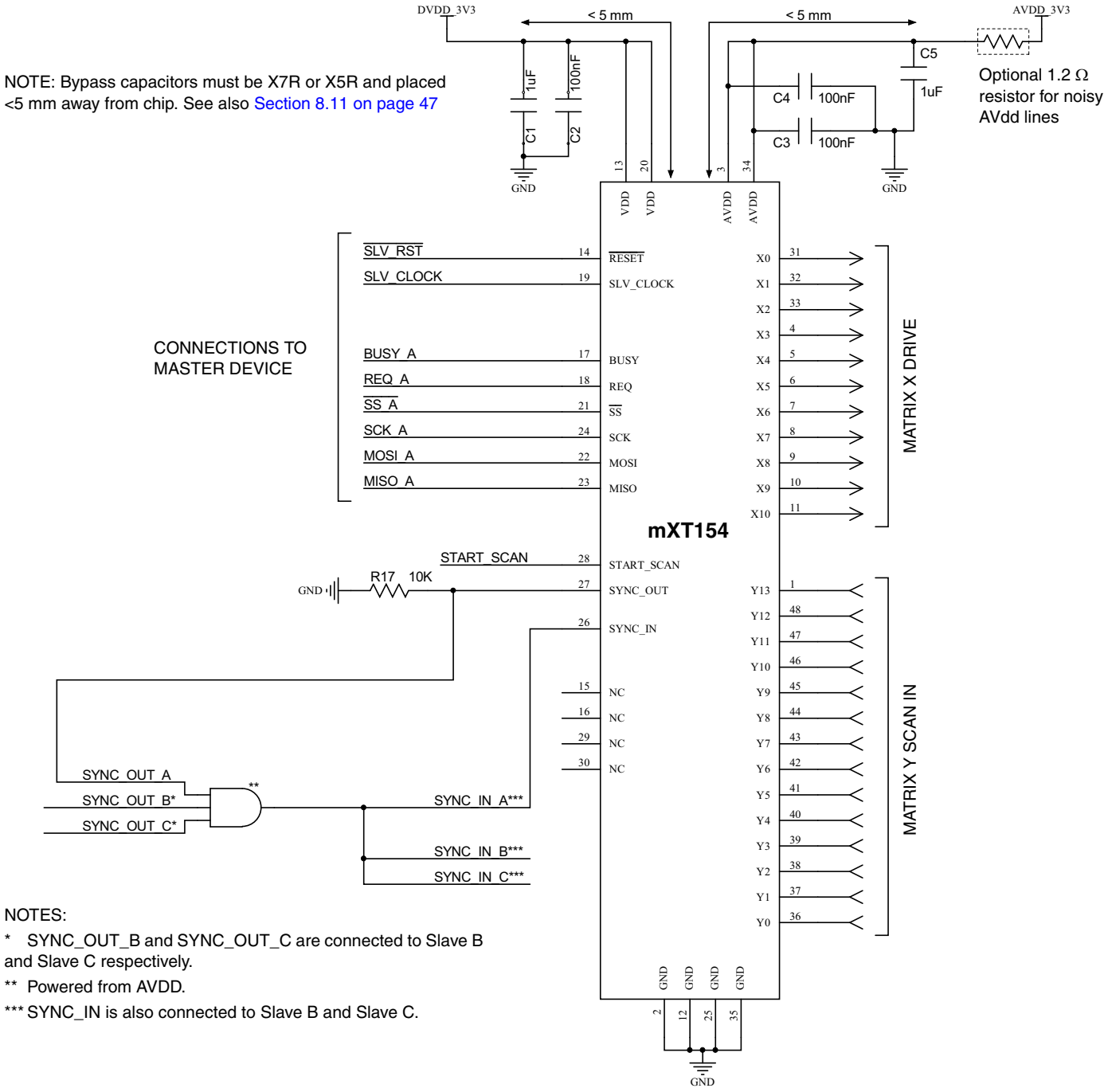
* SYNC_OUT_B and SYNC_OUT_C are connected to Slave B and Slave C respectively.

** Powered from AVDD.

*** SYNC_IN is also connected to Slave B and Slave C.

A.3 Slave Devices (3 x mXT154) – 48-pin QFN

Note: Instance Slave A only is shown; Slave B and Slave C are omitted for simplicity.



NOTES:

* SYNC_OUT_B and SYNC_OUT_C are connected to Slave B and Slave C respectively.

** Powered from AVDD.

*** SYNC_IN is also connected to Slave B and Slave C.

A.4 PCB Design Considerations

The following sections give the design considerations that should be adhered to when designing a PCB layout for use with the mXT1386. Of these, the supply issues and ground tracking considerations are the most critical, followed by careful tracking of the SLV_CLOCK line.

By observing the following design rules, and with careful preparation for the PCB layout exercise, designers will be assured of a far better chance of success and a correctly functioning product.

A.4.1 Printed Circuit Board

Atmel recommends the use of a four layer printed circuit board for mXT1386 applications. This, together with careful layout, will ensure that the board meets relevant EMC requirements for both noise radiation and susceptibility, as laid down by the various national and international standards agencies.

A.4.2 Supply Rails and Ground tracking

Power supply and clock distribution are the most critical parts of any board layout. Because of this, it is advisable that these be completed before any other tracking is undertaken. After these, supply decoupling, and analog and high speed digital signals should be addressed. Track widths for all signals, especially power rails should be kept as wide as possible in order to reduce inductance.

The Power and Ground planes themselves can form a useful capacitor. Flood filling for either or both of these supply rails, therefore, should be used where possible. It is important to ensure that there are no floating copper areas remaining on the board: all such areas should be connected to the 0V plane. The flood filling should be done on the outside layers of the board.

In applications where the USB bus supplies power to the board, care should be taken to ensure that suitable capacitive decoupling is provided close to the USB connector. The tracking to the on-board regulators should also be kept as short as possible.

It should also be remembered that the screen of the USB cable is not intended to be connected to the ground or 0V supply of a remote device. It should either be left open circuit (being connected only at the host computer end) or decoupled with a suitable high voltage capacitor (typically 4.7 nF – 250V) and a parallel resistor (typically 1 MΩ). Note that these components may not be required when the USB cabling is internal and permanently wired, and is routed away from the noisier parts of the system.

A.4.3 Power Supply Decoupling

As a rule, a suitable decoupling capacitor should be placed on each and every supply pin on all digital devices. It is important that these capacitors are placed as close to the chip's supply pins as possible (less than 5mm away). The ground connection of these capacitors should be tracked to 0V by the shortest, heaviest traces possible.

Capacitors with a Type II dielectric, such as X5R or X7R and with a value of at least 100nF, should be used for this purpose.

In addition, at least one 'bulk' tantalum decoupling capacitor, with a minimum value of 4.7 µF should be placed on each power rail, close to where the supply enters the board.

Surface mounting capacitors are preferred to wire leaded types due to their lower ESR and ESL. It is often possible to fit these decoupling capacitors underneath and on the opposite side of the PCB to the digital ICs. This will provide the shortest tracking, and most effective decoupling possible.

Refer to the application note *Selecting Decoupling Capacitors for Atmel's PLDs* (doc0484.pdf; available on Atmel's website) for further general information on decoupling capacitors.

A.4.4 Suggested Voltage Regulator Manufacturers

The AVdd supply stability is critical for the mXT1386 because this supply interacts directly with the analog front end. Atmel therefore recommends that the supply for the analog section of the board be supplied by a regulator that is separate from the logic supply regulator. This reduces the amount of noise injected into the sensitive, low signal level parts of the design.

When frequency hopping is enabled for noise suppression, a poorly controlled AVdd supply can cause an apparent shift in the measured signal values. This can result in false detection or poor burst frequency selection.

A single regulator can be used to provide the analog power to the three on-board mXT154 devices, or three separate regulators can be used, one for each device. If a single regulator is used for the three devices, it is important to ensure that the tracking to each of the mXT154 devices is kept as direct as possible.

A single low value series resistor (around 1Ω) is required from the regulator output to the analog supply input on each mXT154 device. This, together with the regulator output capacitor, and the capacitors at the DC input to each device, forms a simple filter on the supply rail. This applies whether a single regulator or three separate regulators are used in the design.

Low noise devices should be chosen for the local regulators. If possible these should have provision for adding a capacitor across the internal reference for further noise reduction. Reference should be made to the manufacturer's datasheet.

The AVdd supply regulator must be tolerant of the transient load currents drawn by the three mXT154 devices and not show voltage droop on load. Any voltage regulator intended to be used for the 3.3V AVdd supply in mXT1386 designs should be checked to ensure it does not cause signal shifts on frequency change.

The voltage regulators listed in Table 8-1 have been tested and found to work well with the mXT1386. They have compatible footprints and pin-out specifications, and are available in the SOT-23 package.

Table 8-1. Recommended Voltage Regulators

Manufacturer	Part Number
Linear Technology	LT1761
National Semiconductor	LP2981
Micrel	MIC5255
Torex	XC6204

Note some manufacturers claim that minimal or no capacitance is required for correct regulator operation. However, in all cases, a minimum of a 1.0 µF ceramic, low ESR capacitor at the input and output of these devices should be used. The manufacturers' datasheets should always be referred to when selecting capacitors for these devices and the typical recommended values, types and dielectrics adhered to.

A.4.5 Clock Distribution

A distributed clock signal clock signal is derived from pin 9 of the master mXT1386 device. This clock is routed via a single resistor to each of the three mXT154 slaves devices. This distributed clock runs at 12 MHz.

The clock track should be implemented using the shortest, most direct tracking possible. The series terminator should be a low value resistor (approximately 33Ω). It is important that this clock track be located on one of the central layers of the PCB, with grounded copper on each side, and above and below the clock track. It is important to ensure that there is an unbroken ground return path along the whole length of this signal. Failure to observe this can lead to changes in characteristic impedance along the track, resulting in radiated noise.

Where possible, microstrip techniques should be used for the clock, and other high-speed tracks such as the MISO, MOSI, and SCK signals in all three channels.

A.4.6 Crystal Oscillator

If a crystal oscillator is used, its placement is critical to the performance of the design. The connecting leads between the mXT1386 device and the crystal should be as short as possible. These tracks, together with the crystal itself, should be placed above a suitable ground plane. It is also important that no other signal tracks are placed close to, or under, these tracks. The crystal input pins are at a relatively high impedance and cross-talk from other signals will seriously affect oscillator stability and accuracy. The crystal's case should also be connected to ground if possible.

If an oscillator module is used, care still needs to be taken when tracking to the mXT1386 device. The clock signal should be kept as short as possible, with a solid ground return underneath the clock output.

A.4.7 Analog I/O

In general, tracking for the analog I/O signals from the mXT154 devices should be kept as short as possible. These normally go to a connector which interfaces directly to the touchscreen.

Ensure that adequate ground-planes are used. An analog ground plane should be used in addition to a digital one. Care should be taken to ensure that both ground planes are kept separate and are connected together only at the point of entry for the power to the PCB. This is usually at the input connector.

A.4.8 Component Placement

It is important to orient all devices so that the tracking for important signals (such as power and clocks) are kept as short as possible. This simple point is often overlooked when initially planning a PCB layout and can save hours of work at a later stage.

A.4.9 Digital Signals

In general, when tracking digital signals, it is advisable to avoid sharp directional changes, sensitive signal tracks (such as analog I/O) and any clock or crystal tracking.

A good ground return path for all signals should be provided, where possible, to ensure that there are no discontinuities in the ground return path.

A.4.10 EMC and Other Observations

The following recommendations are not mandatory, but may help in situations where particularly difficult EMC or other problems are present:

- A small common mode choke is recommended on the differential USB data pair. This should be placed directly at the USB connector, between the connector and the relevant mXT1386 pins. Tracking lengths for the USB data pair should be kept as short as possible.
- Try to keep as many signals as possible on the inside layers of the board. If suitable ground flood fills are used on the top and bottom layers, these will provide a good level of screening for noisy signals, both into and out of the PCB.
- Ensure that the on-board regulators have sufficient tracking around and underneath the devices to act as a heatsink. This heatsink will normally be connected to the 0V or ground supply pin. Increasing the width of the copper tracking to any of the device pins will aid in removing heat. There should be no solder mask over the copper track underneath the body of the regulators.
- Ensure that the decoupling capacitors, especially tantalum, or high capacity ceramic types, have the requisite low ESR, ESL and good stability/temperature properties. Refer to the regulator manufacturer's datasheet for more information.

Appendix B. Glossary of Terms

Channel

One of the capacitive measurement points at which the sensor controller can detect capacitive change.

Jitter

The peak-to-peak variance in the reported location for an axis when a fixed touch is applied. Typically jitter is random in nature and has a Gaussian [1] distribution, therefore measurement of peak-to-peak jitter must be conducted over some period of time, typically a few seconds. Jitter is typically measured as a percentage of the axis in question.

For example a 100 x 100 mm touchscreen that shows ±0.5 percent jitter in X and ±1 percent jitter in Y would show a peak deviation from the average reported coordinate of ±0.5 mm in X and ±1 mm in Y. Note that by defining the jitter relative to the average reported coordinate, the effects of linearity are ignored.

Linearity

The measurement of the peak-to-peak deviation of the reported touch coordinate in one axis relative to the absolute position of touch on that axis. This is often referred to as the nonlinearity. Nonlinearities in either X or Y axes manifest themselves as regions where the perceived touch motion along that axis (alone) is not reflected correctly in the reported coordinate giving the sense of moving too fast or too slow. Linearity is measured as a percentage of the axis in question.

For each axis, a plot of the true coordinate versus the reported coordinate should be a perfect straight line at 45°. A non linearity makes this plot deviate from this ideal line. It is possible to correct modest nonlinearities using on-chip linearization tables, but this correction trades linearity for resolution in regions where stronger corrections are needed (because there is a stretching or compressing effect to correct the nonlinearity, so altering the resolution in these regions). Linearity is typically measured using data that has been sufficiently filtered to remove the effects of jitter. For example, a 100 mm slider with a nonlinearity of ±1 percent reports a position that is, at most, 1 mm away in either direction from the true position.

One-touch Gesture

A touch gesture that consists of a single touch. The combination of the duration of the touch and any change in position (that is, movement) of the touch characterizes a specific gesture. For example, a tap gesture is characterized by a short-duration touch followed by a release, and no significant movement.

1. Sometimes called Bell-shaped or Normal distribution.

Resolution

The measure of the smallest movement on a slider or touchscreen in an axis that causes a change in the reported coordinate for that axis. Resolution is normally expressed in bits and tends to refer to resolution across the whole axis in question. For example, a resolution of 10 bits can resolve a movement of 0.0977 mm on a slider 100 mm long. Jitter in the reported position degrades usable resolution.

Touchscreen

A two-dimensional arrangement of electrodes whose capacitance changes when touched, allowing the location of touch to be computed in both X and Y axes. The output from the XY computation is a pair of numbers, typically 12-bits each, ranging from 0 to 4095, representing the extents of the touchscreen active region.

Two-touch Gesture

A touch gesture that consists of two simultaneous touches. The change in position of the two touches in relation to each other characterizes a specific gesture. For example, a pinch gesture is characterized by two long-duration touches that have a decreasing distance between them (that is, they are moving closer together).

Appendix C. QMatrix Primer

C.1 Acquisition Technique

QMatrix capacitive acquisition uses a series of pulses to deposit charge into a sampling capacitor, Cs. The pulses are driven on X lines from the controller. The rising edge of the pulse causes current to flow in the mutual capacitance, Cx, formed between the X line and a neighboring receiver electrode or Y line. While one X line is being pulsed, all others are grounded. This leads to excellent isolation of the particular mutual capacitances being measured [1], a feature that makes for good inherent touchscreen performance.

After a fixed number of pulses (known as the burst length) the sampling capacitor's voltage is measured to determine how much charge has accumulated. This charge is directly proportional to Cx and therefore changes if Cx [2] changes. The transmit-receive charge transfer process between the X lines and Y lines causes an electric field to form that loops from X to Y. The field itself emanates from X and terminates on Y. If the X and Y electrodes are fixed directly [3] to a dielectric material like plastic or glass, then this field tends to channel through the dielectric with very little leakage of the field out into free-space (that is, above the panel). Some proportion of the field does escape the surface of the dielectric, however, and so can be influenced during a touch.

When a finger is placed in close proximity (a few millimeters) or directly onto the dielectric's surface, some of this stray field and some of the field that would otherwise have propagated via the dielectric and terminated onto the Y electrode, is diverted into the finger and is conducted back to the controller chip via the human body rather than via the Y line.

This means that less charge is accumulated in Cs, and hence the terminal voltage present on Cs, after all the charge transfer pulses are complete, becomes less. In this way, the controller can measure changes in Cx during touch. This means that the measured capacitance Cx goes down during touch, because the coupled field is partly diverted by the touching object.

The spatial separation between the X and Y electrodes is significant to make the electric field to propagate well in relation to the thickness of the dielectric panel.

C.2 Moisture Resistance

A useful side effect of the QMatrix acquisition method is that placing a floating conductive element between the X and Y lines tends to increase the field coupling and so increases the capacitance Cx. This is the opposite change direction to normal touch, and so can be quite easily be ignored or compensated for by the controller. An example of such floating conductive elements is the water droplets caused by condensation.

As a result, QMatrix-based touchscreens tend not to go into false detect when they are covered in small non-coalesced water droplets. Once the droplets start to merge, however, they can become large enough to bridge the field across to nearby ground return paths (for example, other X lines not currently driven, or ground paths in mechanical chassis components). When this happens, the screen's behavior can become erratic.

1. A common problem with other types of capacitive acquisition technique when used for touchscreens, is that this isolation is not so pronounced. This means that when touching one region of the screen, the capacitive signals also tend to change slightly in nearby channels too, causing small but often significant errors in the reported touch position.
2. To a first approximation.
3. Air gaps in front of QMatrix sensors massively reduce this field propagation and kill sensitivity. Normal optically clear adhesives work well to attach QMatrix touchscreens to their dielectric front panel.

There are some measures used in these controllers to help with this situation, but in general there comes a point where the screen is so contaminated by moisture that false detections become inevitable. It should also be noted that uniform condensation soon becomes non-uniform once a finger has spread it around. Finger grease renders the water highly conductive, making the situation worse overall.

In general, QMatrix has industry-leading moisture tolerance but there comes a point when even the best capacitive touchscreen suffers due to moisture on the dielectric surface.

C.3 Interference Sources

C.3.1 Power Supply

See Section 8.2 on page 43 for the power supply range. The chip set can tolerate short-term power supply fluctuations. If the power supply fluctuates slowly with temperature, the chip set tracks and compensate for these changes automatically with only minor changes in sensitivity. If the supply voltage drifts or shifts quickly, the drift compensation mechanism is not able to keep up, causing sensitivity anomalies or false detections.

The chip set itself uses the AVdd power supply as an analog reference, so the power should be very clean and come from a separate regulator. A standard inexpensive Low Dropout (LDO) type regulator should be used that is not also used to power other loads, such as LEDs, relays, or other high current devices. Load shifts on the output of the LDO can cause AVdd to fluctuate enough to cause false detection or sensitivity shifts. The digital Vdd supply is far more tolerant to noise.

CAUTION: A regulator IC shared with other logic can result in erratic operation and is not advised.

See Section B on page 63 for suggested regulator manufacturers.

Noise on AVdd can appear directly in the measurement results. Vdd should be checked to ensure that it stays within specification in terms of noise, across a whole range of product operating conditions.

Ceramic bypass capacitors on AVdd and Vdd, placed very close (<5 mm) to the chip are recommended. A bulk capacitor of at least 1 µF and a higher frequency capacitor of around 10 nF to 100 nF in parallel are recommended; both must be X7R or X5R dielectric capacitors.

C.3.2 Other Noise Sources

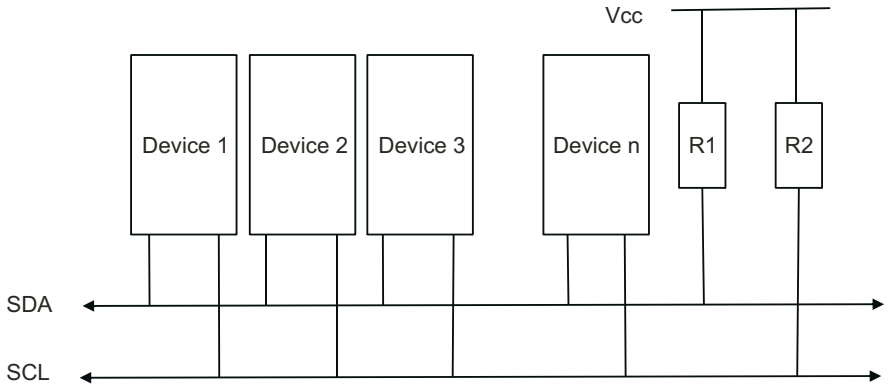
Refer to the *Touch Sensors Design Guide* (downloadable from the Touch Technology area of Atmel's website) for information.

Appendix D. I²C Basics (I²C-compatible Operation)

D.1 Interface Bus

The device communicates with the host over an I²C-compatible bus, in accordance with version 2.1 of the I²C specification. The following sections give an overview of the bus; more detailed information is available from www.i2C-bus.org. Devices are connected to the I²C-compatible bus as shown in Figure D-1. Both bus lines are connected to Vdd via pull-up resistors. The bus drivers of all I²C-compatible devices must be open-drain type. This implements a wired “AND” function that allows any and all devices to drive the bus, one at a time. A low level on the bus is generated when a device outputs a zero.

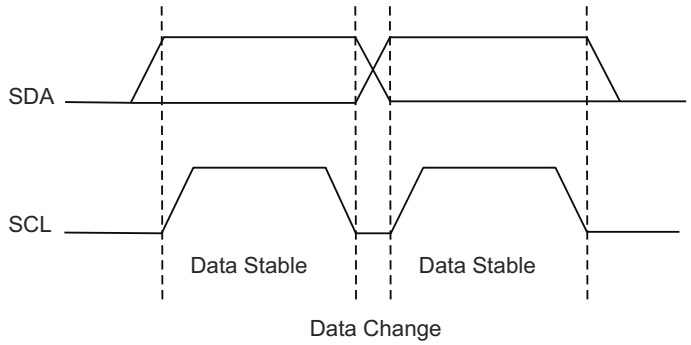
Figure D-1. I²C-compatible Interface Bus



D.2 Transferring Data Bits

Each data bit transferred on the bus is accompanied by a pulse on the clock line. The level of the data line must be stable when the clock line is high; the only exception to this rule is for generating START and STOP conditions.

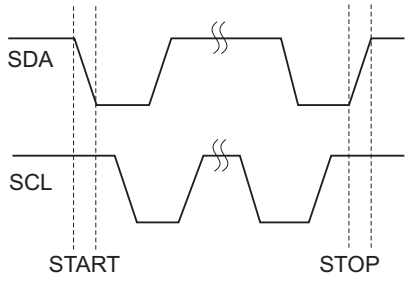
Figure D-2. Data Transfer



D.3 START and STOP Conditions

The host initiates and terminates a data transmission. The transmission is initiated when the host issues a START condition on the bus, and is terminated when the host issues a STOP condition. Between the START and STOP conditions, the bus is considered busy. As shown in Figure D-3 on page 68, START and STOP conditions are signaled by changing the level of the SDA line when the SCL line is high.

Figure D-3. START and STOP Conditions

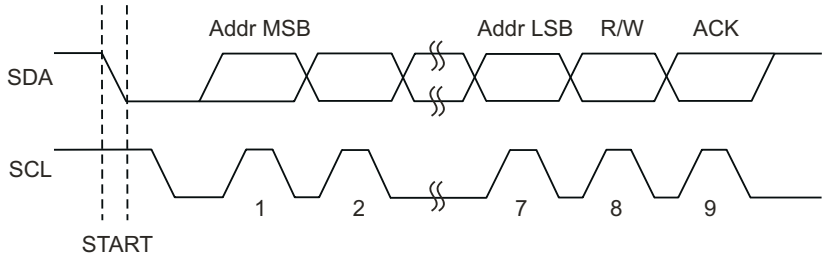


D.4 Address Byte Format

All address bytes are 9 bits long. They consist of 7 address bits, one READ/WRITE control bit and an acknowledge bit. If the READ/WRITE bit is set, a read operation is performed. Otherwise a write operation is performed. An address byte consisting of a slave address and a READ or a WRITE bit is called SLA+R or SLA+W, respectively. When the device recognizes that it is being addressed, it acknowledges by pulling SDA low in the ninth SCL (ACK) cycle.

The most significant bit of the address byte is transmitted first.

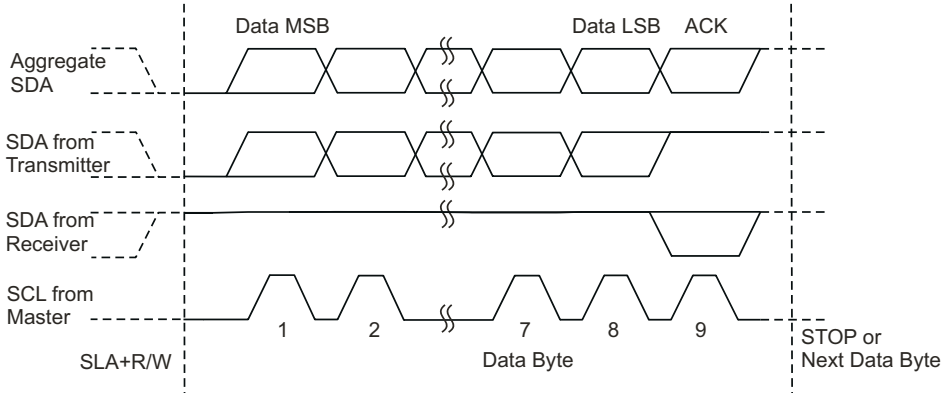
Figure D-4. Address Byte Format



D.5 Data Byte Format

All data bytes are 9 bits long, consisting of 8 data bits and an acknowledge bit. During a data transfer, the host generates the clock and the START and STOP conditions. The slave device is responsible for acknowledging the reception. An acknowledge (ACK) is signaled by the slave device pulling the SDA line low during the ninth SCL cycle. If the slave device leaves the SDA line high, a NACK is signaled.

Figure D-5. Data Byte Format



D.6 Combining Address and Data Bytes into a Transmission

A transmission consists of a START condition, an SLA+R or SLA+W, one or more data bytes and a STOP condition. The wired “ANDing” of the SCL line is used to implement handshaking between the host and the device. The device extends the SCL low period by pulling the SCL line low whenever it needs extra time for processing between the data transmissions.

Figure D-6 shows a typical data transmission. Note that several data bytes can be transmitted between the SLA+R or SLA+W and the STOP.

Figure D-6. Byte Transmission

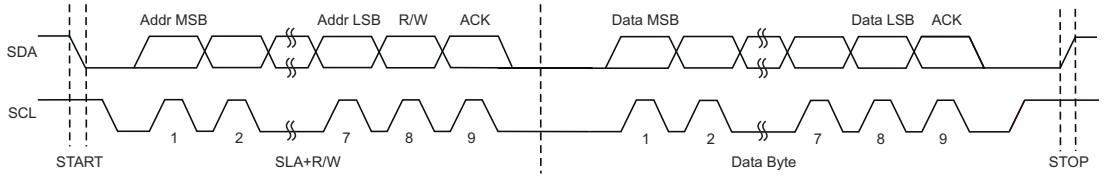


Table of Contents

Revision History

Revision Number	History
Revision AX – April 2010	Initial release
Revision BX – April 2010	Minor updates
Revision CX – May 2010	Updated for chip revision 0.8
Revision DX – June 2010	Updated for chip revision 0.9
Revision EX – July 2010	Minor updates
Revision FX – September 2010	Updated for chip revision 1.0 (released) • Section 8 (Specifications) updated. Reset timings added. Part markings added. Charts added for power consumption, speed and power supply ripple and noise. • Section 4.7.6: Additional information added on crystal oscillator. • Section 5: Description of I^2C-compatible communications updated. • Other minor changes as necessary.
Revision GX – October 2010	• Section 4.1 (Power-up and reset sequence) updated. • Section 8: Orderable chip set bundles updated.
Revision HX – December 2010	• Appendix A updated. Minor corrections to schematics. Information on PCB design added. • Table of Contents added.
Revision IX – January 2011	• Section 8 updated. Alternative part markings and orderable chip set numbers added. • Appendix A updated. Minor correction to mXT1386 schematic to show series terminator resistor on SLV_CLOCK line.
Revision JX – February 2011	Updated for chip revision 1.1
Revision KX – March 2011	• Section 5.5: Note added concerning I^2C-compatable reads. • Section 6.6: USB remote wakeup feature added. • Section 8: Specifications updated. Current consumption figure for USB remote wakeup feature added. Part markings and orderable part numbers updated.
Revision LX – July 2011	Updated for chip revisions 1.0 and 1.1 • Section 8: Specifications updated. Part markings and orderable part numbers updated. • Appendix A (Schematics and PCB Design Considerations): List of recommended voltage regulators updated.

Headquarters

Atmel Corporation
2325 Orchard Parkway
San Jose, CA 95131
USA
Tel: (+1) (408) 441-0311
Fax: (+1) (408) 487-2600

International

Atmel Asia
Unit 01-05 & 16, 19F
BEA Tower, Millennium City 5
418 Kwun Tong Road
Kwun Tong
Kowloon
HONG KONG
Tel: (+852) 2245-6100
Fax: (+852) 2722-1369

Touch Technology Division
1560 Parkway
Solent Business Park
Whiteley
Fareham
Hampshire
PO15 7AG
UNITED KINGDOM
Tel: (+44) 844 894 1920
Fax: (+44) 1489 557 066

Atmel Munich GmbH
Business Campus
Parkring 4
D- 85748 Garching b.
MUNICH
Tel: (+49) 89-31970-111
Fax: (+49) 89-3194621

Atmel Japan
9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
JAPAN
Tel: (+81) 3-3523-3551
Fax: (+81) 3-3523-7581

Product Contact

Web Site
www.atmel.com

Technical Support
touch@atmel.com

Sales Contact
www.atmel.com/contacts

Literature Requests
www.atmel.com/literature