Atmel mXT225T-AT
Revision 1.0 (Automotive)

maXTouch 224-node Touchscreen Controller

DATASHEET

Features

- Atmel® maXTouch® Adaptive Sensing Touchscreen Technology
  - Up to 32 X (transmit) lines and 20 Y (receive) lines
  - A maximum of 224 nodes can be allocated to the touchscreen
  - Touchscreen size of 5.0 inches with aspect ratios of 8:3 to 16:10, but larger touchscreens may be possible, subject to configuration
  - Multi-touch support with up to 16 concurrent touches tracked in real time
- Advanced Touch Handling
  - Moisture/Water Compensation
    - No false touch with condensation or water drop up to 22 mm diameter
    - One-finger tracking with condensation or water drop up to 22 mm diameter
  - Hover Support
    - Supports one-finger hover up to 20 mm detection and 15 mm tracking range
  - Glove Support
    - Supports multiple-finger glove touch up to 1.5 mm thickness
    - Supports single-touch gloved operation with various materials up to 5 mm thickness
- Touch Performance
  - Mutual capacitance and self capacitance measurements supported for touch detection
  - Response Times
    - Initial latency <25 ms for first touch from idle, subject to configuration
  - Atmel maXCharger® technology to combat ambient and power-line noise:
    - Up to 240 Vpp between 1 Hz and 1 kHz sinusoidal waveform
    - Up to 20 Vpp between 1 kHz and 1 MHz sinusoidal waveform
  - Scan Speed
    - Typical report rate for 10 touches ≥60 Hz
- Enhanced Algorithms
  - Lens bending algorithms to remove signal distortions
  - Touch suppression algorithms to remove unintentional touches
  - Palm Recovery Algorithm for quick restoration to normal state
- Panel / Cover Glass Support
  - Supports fully-laminated sensors, touch-on-lens stack-ups and on-cell designs
  - Works with PET or glass, including curved profiles
    - Glass from 0.55 mm to 2.5 mm, dependent on screen size, touch size, and stack-up
    - Plastic from 0.2 mm to 3.0 mm, dependent on screen size and touch size, configuration and approval from Atmel
  - Works with all proprietary sensor patterns recommended by Atmel
  - Compatible with True Single-Layer designs

- Keys
  - Up to 32 nodes can be allocated as mutual capacitance sensor keys (subject to other configurations)
  - Adjacent Key Suppression® (AKS®) technology is supported for false touch prevention
- Product Data Store Area
  - Up to 60 bytes of user-defined data can be stored during production
- Power Saving
  - Programmable timeout for automatic transition from active to idle states
  - Pipelined analog sensing detection and digital processing to optimize system power efficiency
- Application Interfaces
  - I^2C-compatible slave mode: Standard/Fast mode 400 kHz, Fast-plus mode 1 MHz, High-speed mode up to 3.4 MHz
  - SPI interface slave speed up to 8 MHz
  - Interrupt to indicate when a message is available
- Power Supply
  - Digital (Vdd) 3.3 V nominal
  - Analog (AVdd) 3.3 V nominal
  - Host interface I/O voltage (VddIO) 1.8 V to 3.3 V nominal
  - High voltage internal X line drive (XVdd) = 2 × Vdd (6.6 V), with internal voltage doubler
- Packages
  - 100-pin TQFP 14 × 14 × 1 mm, 0.5 mm pitch
- Environmental Conditions
  - Operating temperature –40°C to +85°C

Table of Contents

1. Overview of mXT225T-AT

1.1 Introduction

The Atmel maXTouch family of touch controllers brings industry-leading capacitive touch performance to customer automotive applications. The mXT225T-AT features the latest generation of Atmel Adaptive Sensing technology that utilizes a hybrid mutual- and self-capacitive sensing system in order to deliver unparalleled touch features and a robust user experience.

- **Patented capacitive sensing method** – The mXT225T-AT uses a unique charge-transfer acquisition engine to implement the Atmel-patented QMatrix® capacitive sensing method. Coupled with a state-of-the-art CPU, the entire touchscreen sensing solution can measure, classify and track number of individual finger touches with a high degree of accuracy in the shortest response time.
- **Capacitive Touch Engine (CTE)** – The mXT225T-AT features an acquisition engine, which uses an optimal measurement approach to ensure almost complete immunity from parasitic capacitance on the receiver input lines. The engine includes sufficient dynamic range to cope with anticipated touchscreen self and mutual capacitances, which allows great flexibility for use with the Atmel proprietary sensor pattern designs. One- and two-layer ITO sensors are possible using glass or PET substrates.
- **Touch detection** – The mXT225T-AT allows for both mutual- and self-capacitance measurements, with the self-capacitance measurements being used to augment the mutual-capacitance measurements to produce reliable touch information.

  When self-capacitance measurements are enabled, touch classification is achieved using both mutual- and self-capacitance touch data. This has the advantage that both types of measurement systems can work together to detect touches under a wide variety of circumstances.

  During idle mode, the device performs self-capacitance touch scans (interspersed with self-capacitance hover scans, if required). When a touch is detected, the device starts performing mutual-capacitance touch scans as well as self capacitance scans.

  Mutual-capacitance touch data is used wherever possible to classify touches as this has greater granularity than self-capacitance measurements and provides positional information on touches. For this reason, multiple touches can only be determined by mutual-capacitance touch data. If the self-capacitance touch processing detects multiple touches, touchscreen processing is skipped until mutual-capacitance touch data is available.

  Self-capacitance measurements, on the other hand, allow for the detection of single touches in extreme case, such as single thick-glove touches, when touches can only be detected by self-capacitance data and may be missed by mutual-capacitance touch detection.
- **Display Noise Cancellation** – A combination of analog circuitry, hardware noise processing, and firmware that combats display noise without requiring additional listening channels or synchronization to display timing. This enables the use of shieldless touch sensor stacks, including touch-on-lens.
- **Noise filtering** – Hardware noise processing in the capacitive touch engine provides enhanced autonomous filtering and allows a broad range of noise profiles to be handled. The result is good performance in the presence of LCD noise.
- **Processing power** – The main CPU has two powerful microsequencer coprocessors under its control consuming low power. This system allows the signal acquisition, preprocessing, postprocessing and housekeeping to be partitioned in an efficient and flexible way.
- **Interpreting user intention** – The Atmel hybrid mutual- and self-capacitance method provides unambiguous multitouch performance. Algorithms in the mXT225T-AT provide optimized touchscreen position filtering for the smooth tracking of touches, responding to a user's intended touches while preventing false touch triggered by ambient noise or conductive material on the sensor surface, such as water. The suppression of unintentional touches from the user’s gripping fingers, resting palm or touching cheek or ear also help ensure that the user’s intentions are correctly interpreted.

2. Connection and Configuration Information

2.1 Pin Configuration – TQFP100

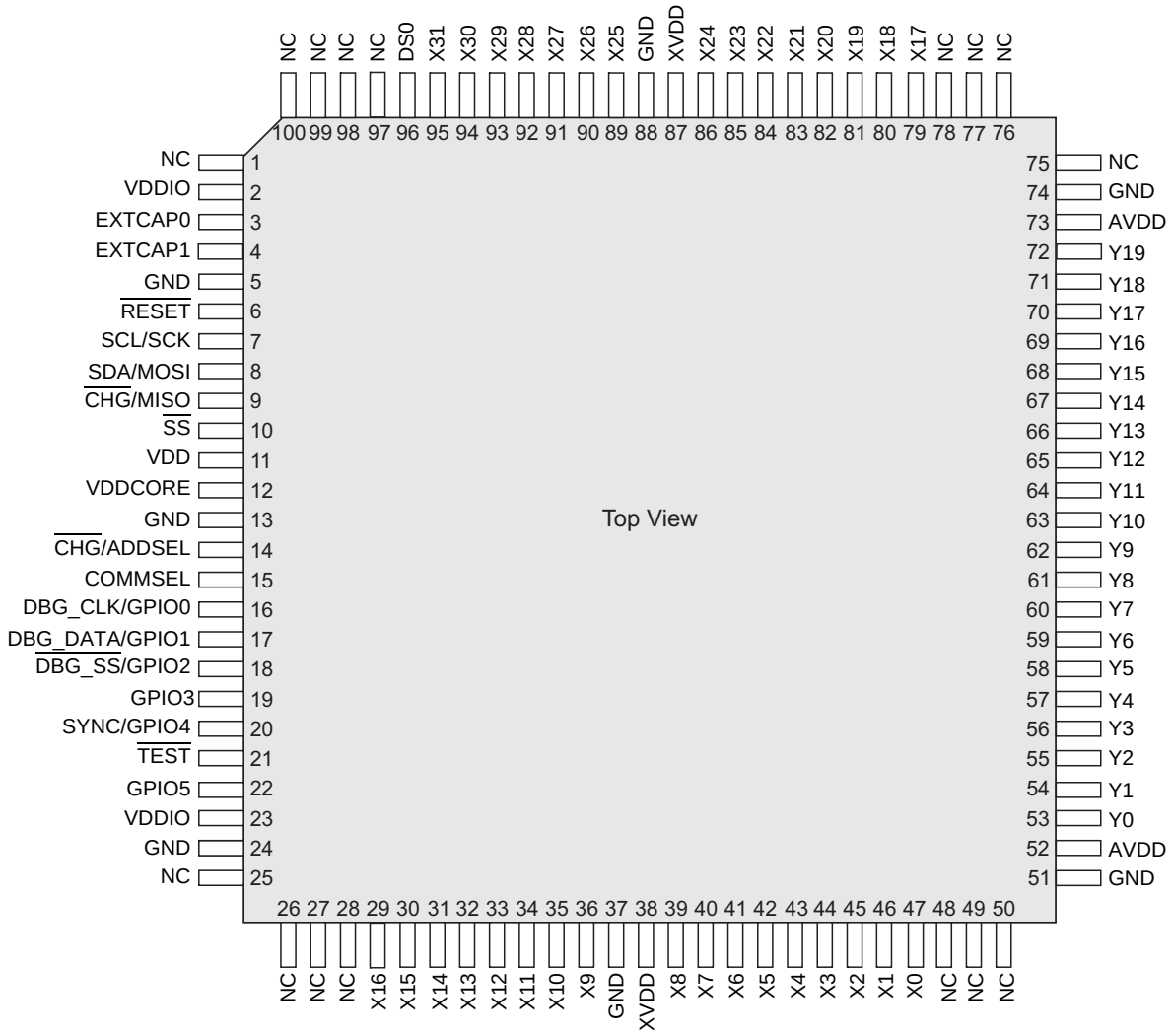


Table 2-1. Pin Listing – TQFP100

Pin	Name	Type	Description	If Unused...
1	NC	–	No connection	Leave open
2	VDDIO	P	Digital IO interface power	–
3	EXTCAP0	P	Connect to EXTCAP1 via capacitor; see schematic notes	Leave open
4	EXTCAP1	P	Connect to EXTCAP0 via capacitor; see schematic notes	Leave open
5	GND	P	Ground	–
6	$\overline{\text{RESET}}$	I	Reset low. Connection to host system is recommended	Pull up to VddIO
7	SCL SCK	OD I	I2C Clock SPI Clock	–
8	SDA MOSI	OD I	I2C Data SPI Data – Master Out Slave In	–
9	$\overline{\text{CHG}}$ MISO	OD O	I2C Mode – State change interrupt SPI Data – Master In Slave Out	–
10	$\overline{\text{SS}}$	I	SPI SS line (active low)	Pull up to VddIO
11	VDD	P	Digital power	–
12	VDDCORE	P	Digital core power	–
13	GND	P	Ground	–
14	$\overline{\text{CHG}}$ ADDSEL	OD I	SPI Mode – State change interrupt I2C address select – For more information see Section 9.2 on page 30	–
15	COMMSEL	I	Selects communications mode – For more information see Section 9.1 on page 30	–
16	DBG_CLK GPIO0	O I/O	Debug clock output General purpose I/O	Input: GND Output: leave open
17	DBG_DATA GPIO1	O I/O	Debug data output General purpose I/O	Input: GND Output: leave open
18	$\overline{\text{DBG\_SS}}$ GPIO2	OD I/O	Debug SS line General purpose I/O	Input: GND Output: leave open
19	GPIO3	I/O	General purpose I/O	Input: GND Output: leave open
20	SYNC GPIO4	I I/O	External synchronization General purpose I/O	Input: GND Output: leave open
21	$\overline{\text{TEST}}$	I	Reserved for factory use; pull up to VddIO	–
22	GPIO5	I/O	General purpose I/O	Input: GND Output: leave open
23	VDDIO	P	Digital IO interface power	–

Table 2-1. Pin Listing – TQFP100 (Continued)

Pin	Name	Type	Description	If Unused...
24	GND	P	Ground	–
25	NC	–	No connection	Leave open
26	NC	–	No connection	Leave open
27	NC	–	No connection	Leave open
28	NC	–	No connection	Leave open
29	X16	S	X line connection	Leave open
30	X15	S	X line connection	Leave open
31	X14	S	X line connection	Leave open
32	X13	S	X line connection	Leave open
33	X12	S	X line connection	Leave open
34	X11	S	X line connection	Leave open
35	X10	S	X line connection	Leave open
36	X9	S	X line connection	Leave open
37	GND	P	Ground	–
38	XVDD	P	X line drive power. For more information see Section 3.2.8 on page 15	–
39	X8	S	X line connection	Leave open
40	X7	S	X line connection	Leave open
41	X6	S	X line connection	Leave open
42	X5	S	X line connection	Leave open
43	X4	S	X line connection	Leave open
44	X3	S	X line connection	Leave open
45	X2	S	X line connection	Leave open
46	X1	S	X line connection	Leave open
47	X0	S	X line connection	Leave open
48	NC	–	No connection	Leave open
49	NC	–	No connection	Leave open
50	NC	–	No connection	Leave open
51	GND	P	Ground	–
52	AVDD	P	Analog power	–
53	Y0	S	Y line connection	Leave open
54	Y1	S	Y line connection	Leave open
55	Y2	S	Y line connection	Leave open

Table 2-1. Pin Listing – TQFP100 (Continued)

Pin	Name	Type	Description	If Unused...
56	Y3	S	Y line connection	Leave open
57	Y4	S	Y line connection	Leave open
58	Y5	S	Y line connection	Leave open
59	Y6	S	Y line connection	Leave open
60	Y7	S	Y line connection	Leave open
61	Y8	S	Y line connection	Leave open
62	Y9	S	Y line connection	Leave open
63	Y10	S	Y line connection	Leave open
64	Y11	S	Y line connection	Leave open
65	Y12	S	Y line connection	Leave open
66	Y13	S	Y line connection	Leave open
67	Y14	S	Y line connection	Leave open
68	Y15	S	Y line connection	Leave open
69	Y16	S	Y line connection	Leave open
70	Y17	S	Y line connection	Leave open
71	Y18	S	Y line connection	Leave open
72	Y19	S	Y line connection	Leave open
73	AVDD	P	Analog power	–
74	GND	P	Ground	–
75	NC	–	No connection	Leave open
76	NC	–	No connection	Leave open
77	NC	–	No connection	Leave open
78	NC	–	No connection	Leave open
79	X17	S	X line connection	Leave open
80	X18	S	X line connection	Leave open
81	X19	S	X line connection	Leave open
82	X20	S	X line connection	Leave open
83	X21	S	X line connection	Leave open
84	X22	S	X line connection	Leave open
85	X23	S	X line connection	Leave open
86	X24	S	X line connection	Leave open
87	XVDD	P	X line drive power. For more information see Section 3.2.8 on page 15	–

Table 2-1. Pin Listing – TQFP100 (Continued)

Pin	Name	Type	Description	If Unused...
88	GND	P	Ground	–
89	X25	S	X line connection	Leave open
90	X26	S	X line connection	Leave open
91	X27	S	X line connection	Leave open
92	X28	S	X line connection	Leave open
93	X29	S	X line connection	Leave open
94	X30	S	X line connection	Leave open
95	X31	S	X line connection	Leave open
96	DS0	S	Driven Shield signal; used as guard track between X/Y signals and ground	Leave open
97	NC	–	No connection	Leave open
98	NC	–	No connection	Leave open
99	NC	–	No connection	Leave open
100	NC	–	No connection	Leave open

Notes: 1. It is recommend that $\overline{\text{RESET}}$ is connected to the host system.

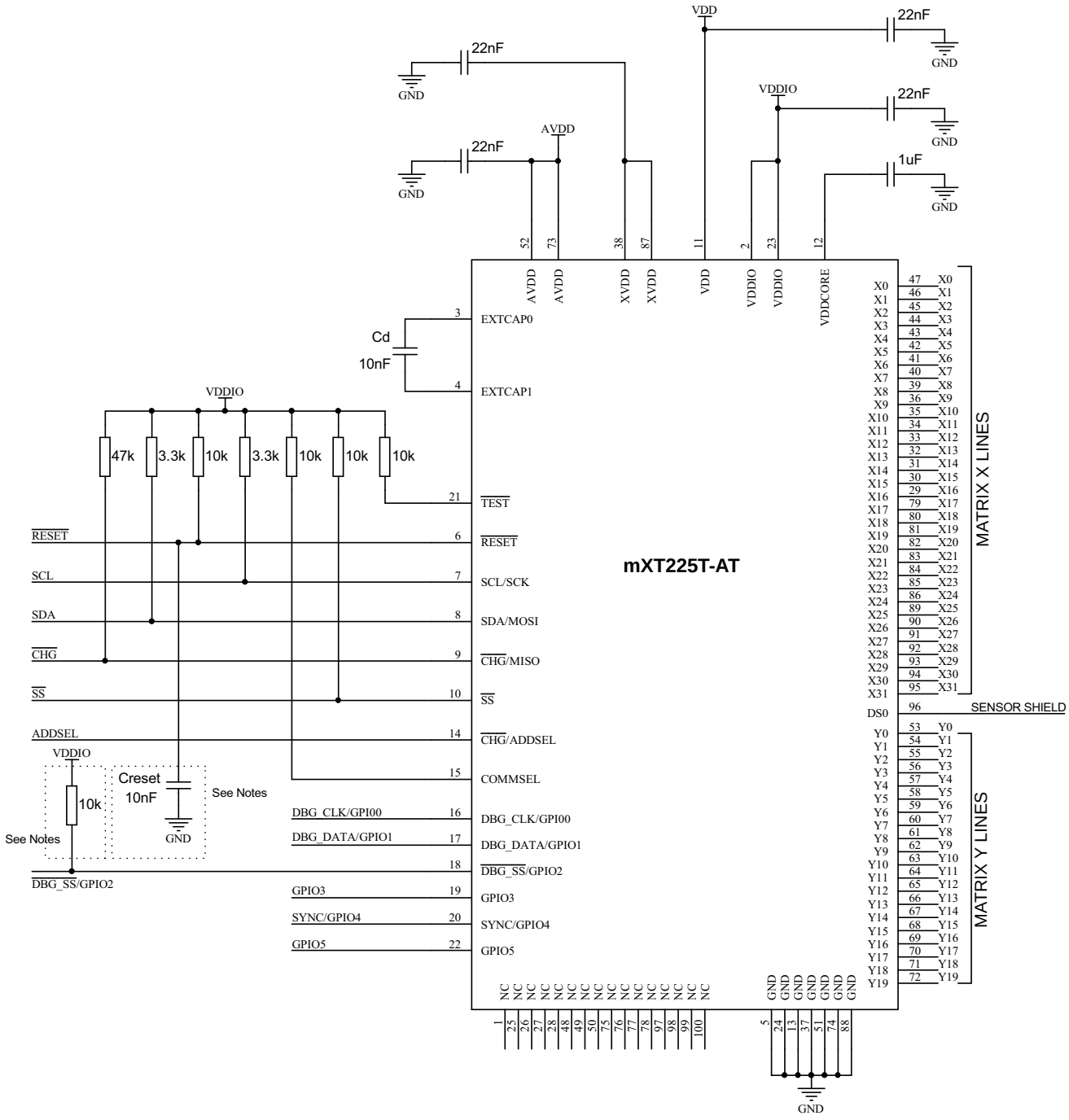
Key:

I	Input only	O	Output only	I/O	Input or output
OD	Open drain output	P	Ground or power	S	Sense pin

3. Schematic

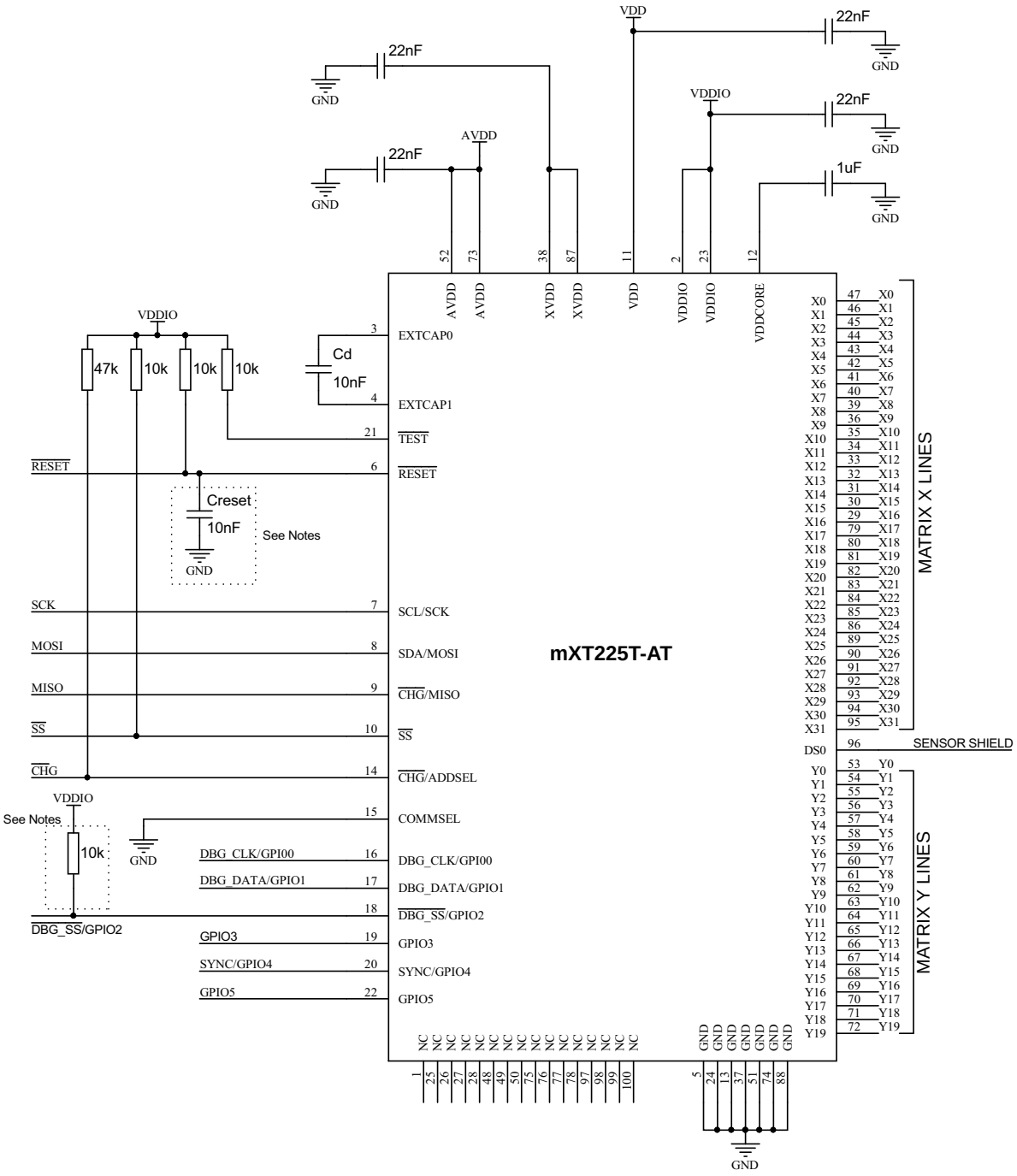
3.1 TQFP 100 Pins

3.1.1 I²C Mode



See Section 3.2 “Schematic Notes” on page 14

3.1.2 SPI Mode



See Section 3.2 “Schematic Notes” on page 14

3.2 Schematic Notes

3.2.1 Number of Available Nodes

Although 32 X lines and 20 Y lines are provided, only a maximum of 224 nodes on the matrix can be used for the touchscreen.

3.2.2 Unconnected Pins

Pins marked as NC (No Connection) can be left open or grounded as they are not internally bonded.

3.2.3 VDDCORE

VddCore is internally generated from the Vdd 3.3 V power supply. To guarantee stability of the internal voltage regulator, a minimum value of 1 µF must be used for decoupling on VDDCORE.

3.2.4 $\overline{\text{DBG\_SS}}$ Line

The $\overline{\text{DBG\_SS}}$ line shares the same pin as GPIO2. Only one of these two functions can be chosen and the circuit should be designed accordingly.

The pull-up resistor in the schematics is optional and should be present only if the pin is used as $\overline{\text{DBG\_SS}}$.

3.2.5 DBG_CLK and DBG_DATA

It is recommended to bring these pins out to accessible test points close to the pins in case they are required during development.

3.2.6 $\overline{\text{RESET}}$ Line

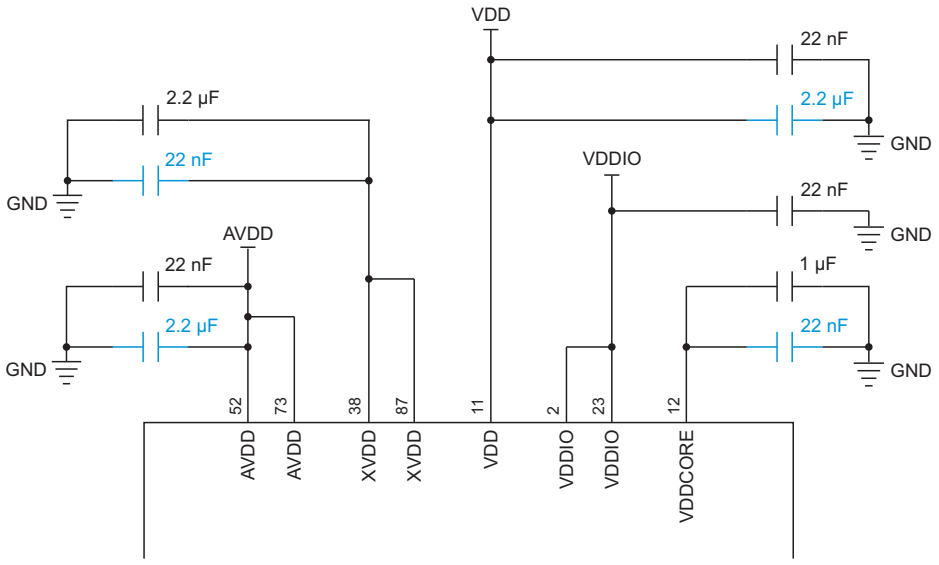
The $\overline{\text{RESET}}$ line is shown on the schematics with a 10 nF capacitor to ground. This capacitor is optional but may help if ESD issues are encountered.

3.2.7 Decoupling Capacitors

1. All decoupling capacitors must be X7R or X5R and placed <5 mm away from the pins for which they act as decoupling capacitors. Pins of the same type can share a capacitor provided no pin is more than 10 mm from the capacitor.
2. The schematics on the previous pages show the minimum capacitors required if the device is placed on the system board. If the pin configuration means that sharing a decoupling capacitor is not possible (distance between pins too great to satisfy condition 1 or routing difficulty), then the number of base capacitors should be increased.

   Note that this requires that the voltage regulator supplies for AVdd, Vdd and VddIO are clean and noise free. It also assumes that the track length between the capacitors and on-board power supplies is < 50 mm.
3. If an active tail design is used, the voltage regulators are likely to be some distance from the device and it may be necessary to implement additional decoupling. In this case, a parallel combination of capacitors is recommended to give high and low frequency filtering as shown in Figure 3-1.

Figure 3-1. Additional Recommended Decoupling Capacitors



NOTE: Recommended additional decoupling capacitors are shown in blue

3.2.8 Voltage Doubler

To use XVdd voltage doubler:

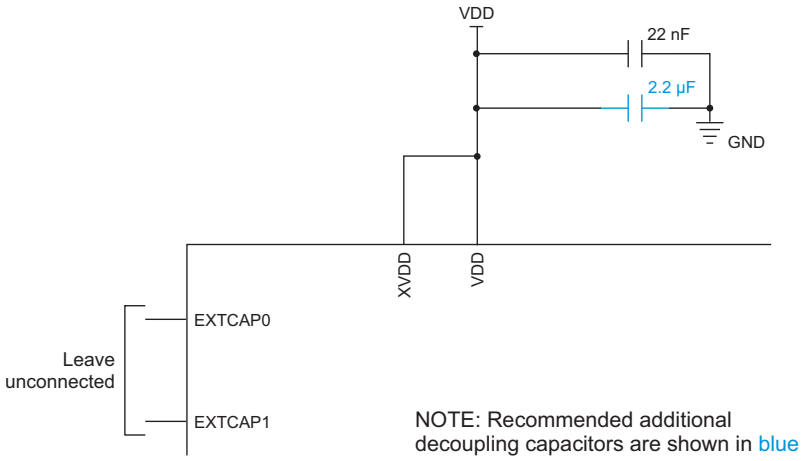
- EXTCAP0 must be connected to EXTCAP1 via a capacitor (Cd) to provide XVdd voltage doubler mode.
- The recommended value of the capacitor is 10 nF. Other values can be used if necessary after consultation with Atmel.
- The capacitor on XVDD should be rated at least 10 V if the voltage doubler is used.

If XVdd voltage doubler is not required:

- Capacitor Cd must be omitted and EXTCAP0 and EXTCAP1 left unconnected.
- XVDD line(s) must be connected to VDD.

These modifications are shown in Figure 3-2.

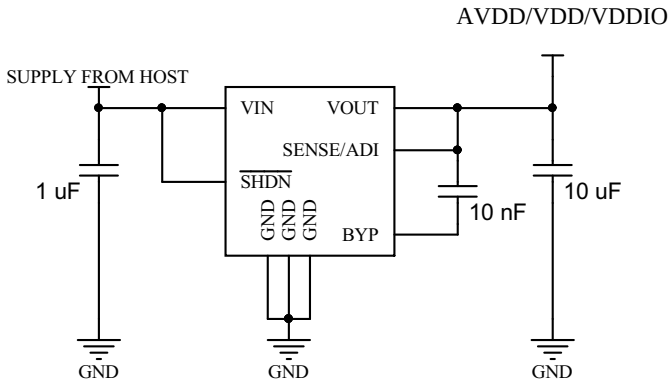
Figure 3-2. No voltage Doubler



3.2.9 Low Drop-Out Voltage Regulators (LDOs)

In applications where the VddIO supply is at the same voltage level as Vdd and AVdd (that is, 3.3 V) it is permissible to use a single LDO for all supply rails (AVDD/VDD/VDDIO). A suitable circuit is shown in Figure 3-3.

Figure 3-3. Low Drop-Out Regulators



Where poor or inadequate tracking or decoupling leads to high noise levels on the supply rails, Atmel recommends that a separate low drop-out voltage regulator supply is used for the AVdd supply.

See Section 4.4 on page 17 for further details. A list of approved regulators is given in Table 4-1 on page 18.

4. Circuit Components

4.1 Decoupling Capacitors

Each power supply pin requires decoupling as described in Section 3.2 on page 14. The capacitors should be ceramic X7R or X5R.

The PCB traces connecting the decoupling capacitors to the pins of the device must not exceed 10 mm in length. This limits any stray inductance that would reduce filtering effectiveness.

4.2 I^2C Line Pull-up Resistors

The values for pull-up resistors on SDA and SCL need to be chosen to ensure rise times are within I^2C specification – if the rise time is too long the overall clock rate will be reduced.

If using a VddIO at the low end of the allowable range it is likely that the pull-up resistor values will need to be reduced from those shown on the schematic.

4.3 Supply Quality

While the device has good Power Supply Rejection Ratio properties, poorly regulated and/or noisy power supplies can significantly reduce performance.

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

4.4 Suggested Voltage Regulators

An LDO regulator should be chosen that provides adequate output capability, low noise, good load regulation and step response.

Suitable fixed output LDO devices are shown in Table 4-1 on page 18.

With a single regulator, PCB layout is more critical than with multiple LDO regulators, and special care with the PCB layout should be taken. See Section 12.5 on page 47 for information concerning PCB design with a single LDO.

4.4.1 Multiple Voltage Regulator Supply

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

4.4.2 Suggested Voltage Regulators

The voltage regulators listed in Table 4-1 have been tested and found to work well with the mXT225T-AT.

Table 4-1. Suitable LDO Regulators

Manufacturer	Device	Current Rating (mA)
Analog Devices	ADP122/ADP123	300
Diodes Inc.	AP2125	300
Diodes Inc.	AP7335	300
Linear Technology	LT1763CS8-3.3	500
NXP	LD6836	300
Texas Instruments	LP2981	100
Texas Instruments	LP3981	300
Texas Instruments	LP5996	150 / 300

1. 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 manufacturer's datasheets should always be referred to when selecting capacitors for these devices and the typical recommended values, types and dielectrics adhered to.

4.4.3 LDO Selection Criteria

The LDO devices in Table 4-1 have been proved to provide satisfactory performance in Atmel maxTouch controllers, however, if it is desired to use an alternative LDO, certain performance criteria should be verified before using the device. These are:

- Stable with low value multi-layer ceramic capacitors on input and output – actual values will be device dependent, but it is good design practice to use values greater than the minimum specified in the LDO regulator data sheet
- Low output noise – less than 100 µV RMS over the range 10 Hz to 1 MHz
- Good load transient response – this should be less than 35 mV peak when a load step change of 100 mA is applied at the device output terminal
- Input supply requirement of between 4.5 V and 5.5 V
- Low quiescent current to improve battery life
- Thermal and current limit overload protection
- Ideally, select an LDO with common footprint, to allow interchanging between regulators

5. Touchscreen Basics

5.1 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 constructed from Indium Tin Oxide (ITO) or metal mesh. Thicker electrodes yield 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 electrodes lead to higher levels of resistance (perhaps hundreds to thousands of Ω/square) with some of the best optical characteristics.

Interconnecting tracks 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, the tracks should be replaced by screen printed conductive inks (non-transparent) outside the touchscreen viewing area.

5.2 Electrode Configuration

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

The device supports various configurations of electrodes as summarized in Section 6. on page 21.

5.3 Scanning Sequence

All nodes are scanned in sequence by the device. There is a full parallelism in the scanning sequence to improve overall response time. The nodes 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 device can be configured in various ways. It is possible to disable some nodes so that they are not scanned at all. This can be used to improve overall scanning time.

5.4 Touchscreen Sensitivity

5.4.1 Adjustment

Sensitivity of touchscreens can vary across the extents of the electrode pattern due to natural differences in the parasitic capacitance 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 node is considered to have enough signal change to qualify as being in detect.

5.4.2 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 3.0 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.

Note: Care should be taken using ultra-thin glass panels as retransmission effects can occur, which can significantly degrade performance.

6. Sensor Layout

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

The physical matrix can be configured to have one or more touch objects. These are configured using the appropriate touch objects (Multiple Touch Touchscreen and Key Array). It is not mandatory to have all the allowable touch objects present. The objects are disabled by default so only those that you wish to use need to be enabled. Refer to the *mXT225T-AT 1.0 Protocol Guide* for more information on configuring the touch objects.

The device supports various configurations of electrodes as summarized below:

- Touchscreen: 32 X × 20 Y maximum (subject to other configurations)
- Keys: Up to 32 keys in an X/Y grid

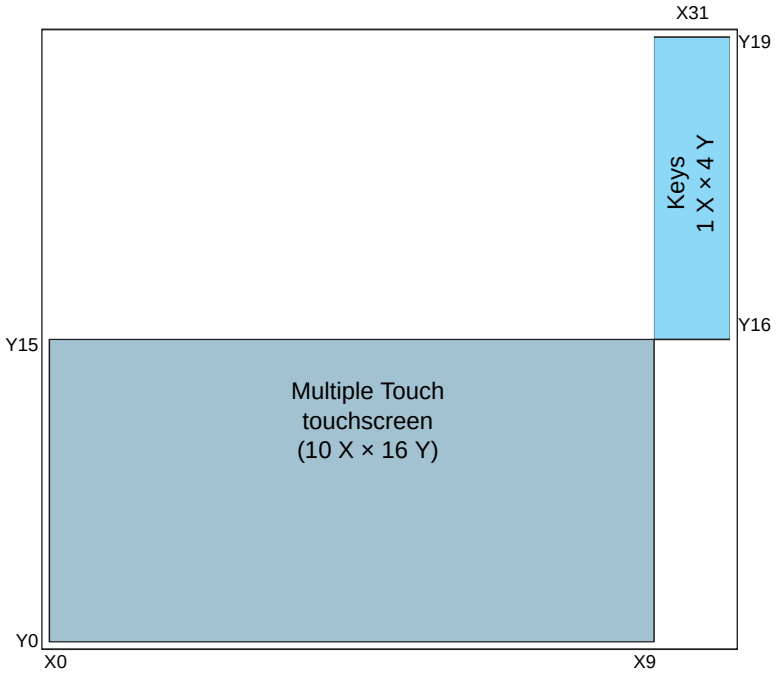
Although there are a total of 52 lines, arranged as a matrix of 32 X by 20 Y, only a maximum of 224 nodes can be used for all the touch objects on this device. The matrix can be made up of any combination of X and Y lines in the design, provided the X and Y lines are contiguous and subject to the maximum of 224 nodes. For example the matrix could be constructed as a matrix of 32 X by 7 Y lines (giving 224 nodes) or as a matrix of 11 X by 20 Y (giving 220 nodes). The arrangement chosen depends on the application.

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 cannot share either X or Y lines if self-capacitance measurements are enabled.
- It is recommended that the touchscreen should start at X0, Y0; if self-capacitance measurements are enabled, the touchscreen **must** start at X0, Y0.
- It is recommended that the keys should occupy the highest X and Y lines.

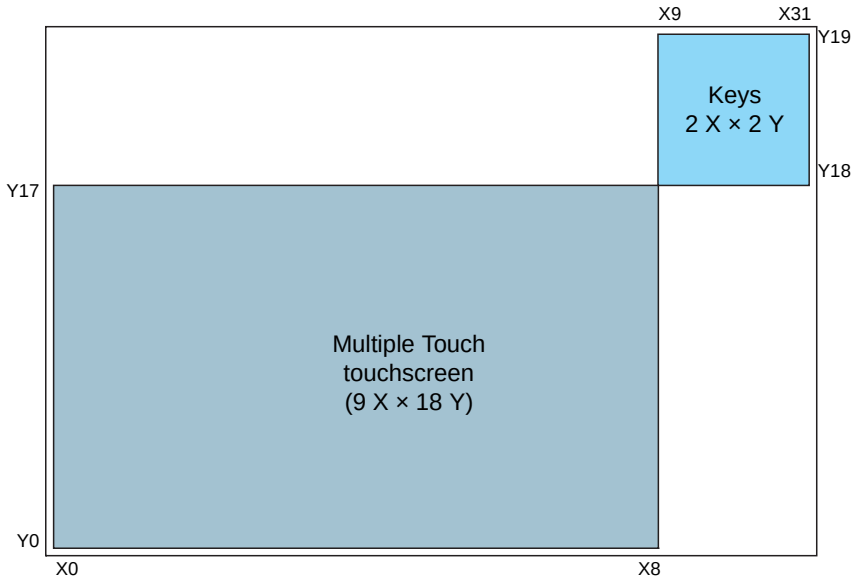
For optimal performance in terms of cycle time overhead, it is recommended that the number of X (drive) lines used for the key array is kept to the minimum and designs should favor using Y lines where possible. Figure 6-1 shows an example layout for a key array of 1 X × 4 Y lines. Note that in this case using 1 X × 4 Y lines would give better performance than using 4 X × 1 Y lines.

Figure 6-1. Example Layout – Optimal Cycle Time



If, however, the intention is to preserve a larger touchscreen size and maintain an optimal aspect ratio, then using equal X and Y lines for the key array can be considered, as in Figure 6-2.

Figure 6-2. Example Layout – Optimal Aspect Ratio

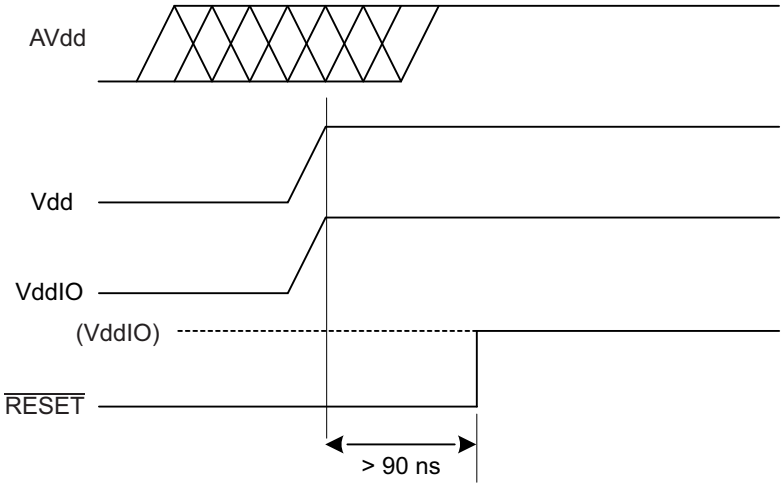


7. Power-up / Reset Requirements

There is an internal Power-on Reset (POR) in the device.

If an external reset is to be used the device must be held in RESET (active low) while the digital (Vdd) analog (AVdd) and I/O (VddIO) power supplies are powering up. The supplies must have reached their nominal values before the RESET signal is deasserted (that is, goes high). This is shown in Figure 7-1. See Section 15.2 on page 54 for nominal values for Vdd, VddIO, and AVdd.

Figure 7-1. Power Sequencing on the mXT225T-AT



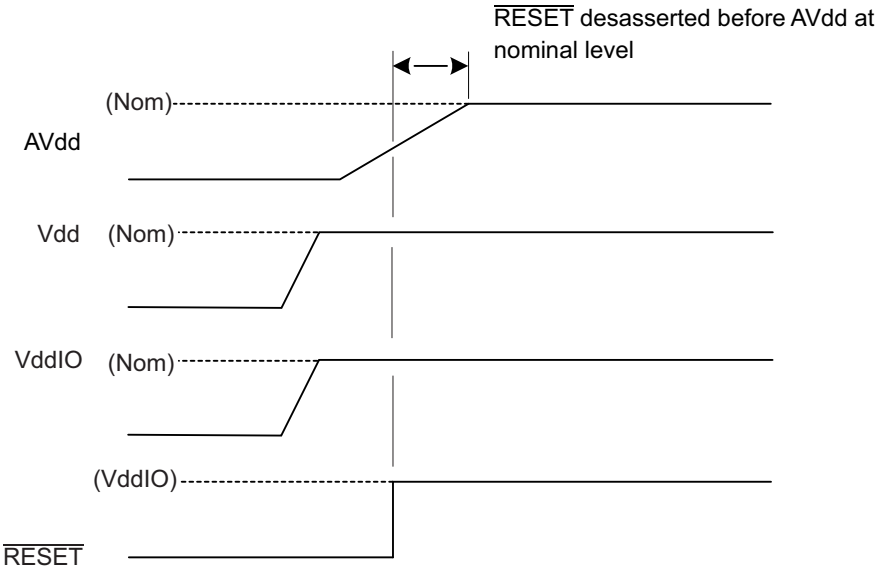
Note: When using external RESET at power-up, VddIO must not be enabled after Vdd

After power-up, the device takes 93.8 ms before it is ready to start communications.

If the RESET line is released before the AVdd supply has reached its nominal voltage (see Figure 7-2 on page 24), 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 7-2. Power Sequencing on the mXT225T-AT – Late rise on AVdd or XVdd



The RESET pin can be used to reset the device whenever necessary. The RESET pin must be asserted low for at least 90 ns to cause a reset. After releasing the RESET pin the device takes 93.3 ms before it is ready to start communications. It is recommended to connect the RESET pin to a host controller to allow it to initiate a full hardware reset without requiring a power-down.

Make sure that any lines connected to the device are below or equal to Vdd during power-up. For example, if RESET is supplied from a different power domain to the VDDIO pin, make sure that it is held low when Vdd is off. If this is not done, the RESET signal could parasitically couple power via the RESET pin into the Vdd supply.

Note that the voltage level on the RESET pin of the device must never exceed VddIO (digital supply voltage).

A software reset command can be used to reset the chip (refer to the Command Processor T6 object in the *mXT225T-AT 1.0 Protocol Guide*. A software reset takes typically 115 ms. After the chip has finished it asserts the 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 to take any necessary corrective actions, such as reconfiguration.

A checksum check is performed on the configuration settings held in the nonvolatile memory. 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 T6 object (refer to the *mXT225T-AT 1.0 Protocol Guide* for more information).

Note that the CHG line is briefly set as an input during power-up or reset. It is therefore particularly important that the line should be allowed to float high via the CHG line pull-up resistor during this period. It should not be driven by the host (see Table 15.5.3 on page 60).

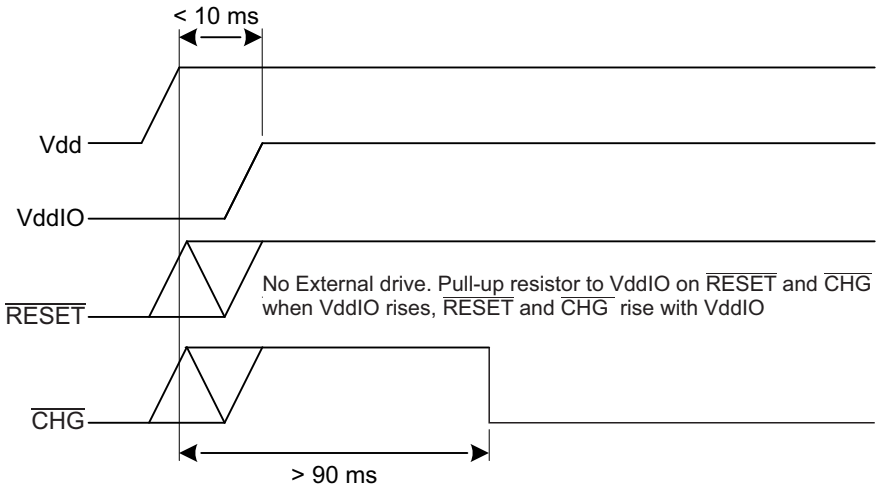
At power-on, the device performs a self-test routine to check for shorts that might cause damage to the device. Refer to the Self Test T25 object in the *mXT225T-AT 1.0 Protocol Guide* for more details about this process.

7.1 Power-up and Reset Sequence – VddIO Enabled after Vdd

The Power-up sequence that can be used in applications where VddIO must be powered up after Vdd, is shown in Figure 7-3.

In this case the communication interface to the maXTouch device is not driven by the host system. The RESET and CHG pins are connected to VddIO using suitable pull-up resistors. Vdd is powered up, followed by VddIO, no more than 10 ms after Vdd. Due to the pull-up resistors, RESET and CHG will rise with VddIO. The internal POR system ensures reliable boot up of the device and the CHG line will go low approximately 93.8 ms after Vdd to notify the host that the device is ready to start communication.

Figure 7-3. Power-up Sequence



7.1.1 Summary

The Power-up and reset requirements for the maXTouch devices are summarized in Table 7-1.

Table 7-1. Power-up and Reset Requirements

Condition	External $\overline{\text{RESET}}$	VddIO Delay (After Vdd)	AVdd Power-Up	Comments
1	Low at Power-up	0 ms	Before $\overline{\text{RESET}}$ is released	If AVdd bring-up is delayed then additional actions will be required by the host. See notes in Figure 7-1 on page 23
2	Not driven	<10 ms	Before VddIO	

8. Detailed Operation

8.1 Touch Detection

The mXT225T-AT allows for both mutual and self capacitance measurements, with the self capacitance measurements being used to augment the mutual capacitance measurements to produce reliable touch information.

When self capacitance measurements are enabled, touch classification is achieved using both mutual and self capacitance touch data. This has the advantage that both types of measurement systems can work together to detect touches under a wide variety of circumstances.

Mutual capacitance touch data is used wherever possible to classify touches as this has greater granularity than self capacitance measurements and provides positional information on touches. Refer to the *mXT225T-AT 1.0 Protocol Guide* for more information on measurements.

Self capacitance measurements, on the other hand, allow for the detection of single touches in extreme case, such as single thick glove touches, when touches can only be detected by self capacitance data and may be missed by mutual capacitance touch detection.

8.2 Operational Modes

The device 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 T7. In addition, an *Active to Idle Timeout* setting is provided.

Refer to the *mXT225T-AT 1.0 Protocol Guide* for full information on how these modes operate, and how to use the settings provided.

8.3 Detection Integrator

The device 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 T100, Key Array T15). Refer to the *mXT225T-AT 1.0 Protocol Guide* for more information.

8.4 Sensor Acquisition

The maximum acquisition time for one X line on the mXT225T-AT is 5 µs. Care should be taken to ensure that the total time for one X line configured by the Acquisition Configuration T8 and CTE Configuration T46 objects do not exceed this (refer to the *mXT225T-AT 1.0 Protocol Guide* for details on these objects).

8.5 Calibration

Calibration is the process by which a sensor chip assesses the background capacitance on each node. Nodes are only calibrated on reset and when:

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

or

- The node is already enabled and one of the following applies:
  - The node is held in detect for longer than the Touch Automatic Calibration setting (refer to the *mXT225T-AT 1.0 Protocol Guide* for more information on TCHAUTOCAL setting in the Acquisition Configuration object).

- The signal delta on a node is at least the touch threshold (TCHTHR) in the anti-touch direction, while it meets the criteria in the Touch Recovery Processes that results in a recalibration. (Refer to the *mXT225T-AT 1.0 Protocol Guide* for objects Acquisition Configuration T8 and Self Capacitance Configuration T111).
- The host issues a recalibrate command.
- Certain configuration settings are changed.

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

Note that the device performs a global calibration; that is, all the nodes are calibrated together.

8.6 Digital Filtering and Noise Suppression

The mXT225T-AT supports on-chip filtering of the acquisition data received from the sensor. Specifically, the maXCharger T72 object provides an algorithm to suppress the effects of noise (for example, from a noisy charger plugged into the user's product). This algorithm can automatically adjust some of the acquisition parameters on-the-fly to filter the analog-to-digital conversions (ADCs) received from the sensor.

Additional noise suppression is provided by the Self Capacitance maXCharger T108 object. Similar in both design and configuration to the maXCharger T72 object, the Self Capacitance maXCharger T108 object is the noise suppression interface for self capacitance touch measurements.

Noise suppression is triggered when a noise source is detected.

- The host driver code can indicate when a noise source is present.
- The noise suppression is also triggered based on the noise levels detected using internal line measurements.The maXCharger T72 and Self Capacitance maXCharger T108 object selects the appropriate controls to suppress the noise present in the system.

Refer to the *mXT225T-AT 1.0 Protocol Guide* for more information on the maXCharger T72 and Self Capacitance maXCharger T108 objects.

8.7 Shieldless Support and Display Noise Suppression

The mXT225T-AT can support shieldless sensor design even with a noisy LCD by using the following features.

- **Optimal Integration:** This feature is not filtering as such, but enables the user to use a shorter integration window. The integration window optimizes the amount of charge collected against the amount of noise collected, to ensure an optimal SNR. This feature also benefits the system in the presence of an external noise source. This feature is configured using the Shieldless T56 object. Refer to the *mXT225T-AT 1.0 Protocol Guide* for more information
- **Display noise suppression:** This feature is based on filtering provided by the Lens Bending T65 object (See Section 8.10 on page 28). This feature allows the device to overcome display noise simultaneously with external noise. Refer to the *mXT225T-AT 1.0 Protocol Guide* for more information

8.8 Retransmission Compensation

The device can limit the undesirable effects on the mutual capacitance touch signals caused by poor device coupling to ground, such as poor sensitivity and touch break-up. This is achieved using the Retransmission Compensation T80 object. This object can be configured to allow the touchscreen to compensate for signal degradation due to these undesirable effects. If self capacitance measurements are also scheduled, the Retransmission Compensation T80 object will use the resultant data to enhance the compensation process.

The Retransmission Compensation T80 object is also capable of compensating for water presence on the sensor if self capacitance measurements are scheduled. In this case, both mutual capacitance and self capacitance measurements are used to detect moisture and then, once moisture is detected, self capacitance measurements are used to detect single touches in the presence of moisture.

Refer to the *mXT225T-AT 1.0 Protocol Guide* for more information on the Retransmission Compensation T80 object.

8.9 Grip Suppression

The device has two grip suppression mechanisms to suppress false detections from a user's grip.

Mutual 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.

Mutual grip suppression is configured using the Grip Suppression T40 object. There is one instance of the Grip Suppression T40 object for each Multiple Touch Touchscreen T100 object present on the device.

Refer to the *mXT225T-AT 1.0 Protocol Guide* for more information on the Grip Suppression T40 and the Self Capacitance Grip Suppression T112 objects.

Self Capacitance grip suppression works by looking for characteristic shapes in the self capacitance measurement along the touchscreen boundary, and thereby distinguishing between a grip and a touch further into the sensor.

8.10 Lens Bending

The device supports algorithms to eliminate disturbances from the measured signal.

When the sensor suffers from the screen deformation (lens bending) the signal values acquired by normal procedure are corrupted by the disturbance component (bend). The amount of bend depends on:

- The mechanical and electrical characteristics of the sensor
- The amount and location of the force applied by the user touch to the sensor

The Lens Bending T65 object measures the bend component and compensates for any distortion caused by the bend. As the bend component is primarily influenced by the user touch force, it can be used as a secondary source to identify the presence of a touch. The additional benefit of the Lens Bending T65 object is that it will eliminate LCD noise as well.

Refer to the *mXT225T-AT 1.0 Protocol Guide* for more information on the Lens Bending T65 object.

8.11 Glove Detection

The device has glove detection algorithms that process the measurement data received from the touchscreen classifying touches as potential gloved touches.

The Glove Detection T78 object is used to detect glove touches. In Normal Mode the Glove Detection T78 object applies vigorous glove classification to small signal touches to minimize the effect of unintentional hovering finger reporting. Once a gloved touch is found, the Glove Detection T78 object enters Glove Confidence Mode. In this mode the device expects the user to be wearing gloves so the classification process is much less stringent.

Refer to the *mXT225T-AT 1.0 Protocol Guide* for more information on the Glove Detection T78 object.

8.12 Stylus Support

The mXT225T-AT allows for the particular characteristics of passive stylus touches, whilst still allowing conventional finger touches to be detected. The touch sensitivity and threshold controls for stylus touches are configured separately from those for conventional finger touches so that both types of touches can be accommodated.

Stylus support ensures that the small touch area of a stylus registers as a touch, as this would otherwise be considered too small for the touchscreen. Additionally, there are controls to distinguish a stylus touch from an unwanted approaching finger (such as on the hand holding the stylus).

Passive stylus touches are configured by the Passive Stylus T47 object. There is one instance of the Passive Stylus T47 object for each Multiple Touch Touchscreen T100 object present on the device.

Refer to the *mXT225T-AT 1.0 Protocol Guide* for more information on configuring a stylus.

8.13 Hover Support

The mXT225T-AT supports hover and is configured using the Touchscreen Hover Configuration T101 and the Self Capacitance Configuration T111 (instance 1) objects. The mXT225T-AT allow for the configuration of both the hover measurements and also the data that defines the hover touch detection and post processing. Hover status messages are reported through the reporting mechanisms of the linked Multiple Touch Touchscreen T100 object. Refer to the *mXT225T-AT 1.0 Protocol Guide* for more information on the Touchscreen Hover Configuration T101 and Self Capacitance Configuration T111 objects.

8.14 Unintentional Touch Suppression

The Touch Suppression T42 object provides a mechanism to suppress false detections from unintentional touches from a large body area, such as from a face, ear or palm. The Touch Suppression T42 object also provides Maximum Touch Suppression to suppress all touches if more than a specified number of touches has been detected. There is one instance of the Touch Suppression T42 object for each Multiple Touch Touchscreen T100 object present on the device.

Refer to the *mXT225T-AT 1.0 Protocol Guide* for more information on the Touch Suppression T42 object.

8.15 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 device has two levels of AKS. The first level works between the touch objects (Multiple Touch Touchscreen T100 and Key Array T15). 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 a 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 T15). 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 touch objects (Multiple Touch Touchscreen T100 or Key Array T15).

Refer to the *mXT225T-AT 1.0 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.

8.16 GPIO Pins

The mXT225T-AT has 6 GPIO pins. The pins can be set to be either an input or an output, as required. Note that unused GPIO pins can be left externally unconnected as long as they are given a defined state by using the GPIO/PWM Configuration T19 object. With the GPIO/PWM Configuration T19 object, an unused GPIO pin can be either set to Input mode, with internal pull-up, or Output mode. If the GPIO/PWM Configuration T19 is not enabled for use, all the GPIO pins are unused.

By default GPIO pins are set to be inputs. If not used they should be connected to GND. Alternatively, they can be set as outputs using the GPIO/PWM Configuration T19 object and left open.

9. Host Communications

9.1 Communication Mode Selection (COMMSEL Pin)

The selection of the I²C or SPI interface is determined by the COMMSEL pin:

Table 9-1. Interface Selection

COMMSEL	Interface Selected
Connected to GND	SPI
Pulled up to VddIO	I²C

9.2 I²C Address Selection (ADDSEL Pin)

The I²C address is selected by connecting the ADDSEL pin according to Table 9-2.

Table 9-2. I²C Address Selection

ADDSEL	I²C Address
Connected to GND	0x4A
Pulled up to VddIO	0x4B

10. I²C Communications

The device can use an I²C interface for communication.

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

10.1 I²C Addresses

The device supports two I²C device addresses that are selected using the ADDSEL line at start up. The two internal I²C device addresses are 0x4A and 0x4B. The selection of the address (and the communication mode) is described in Section 9.2 on page 30. These are shifted left to form the SLA+W or SLA+R address when transmitted over the I²C interface, as shown in Figure 10-1.

Table 10-1. Format of an I²C Address

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Address: 0x4A or 0x4B							Read/write

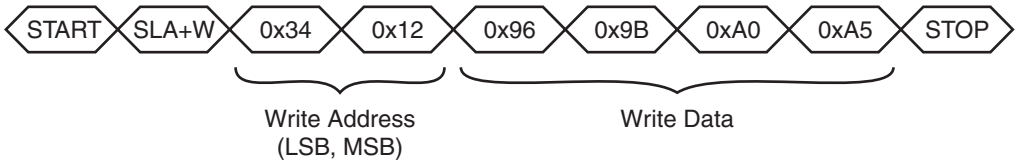
10.2 Writing To the Device

A WRITE cycle to the device consists of a START condition followed by the I²C 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 multi-byte 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 STOP condition is detected.

Figure 10-1 shows an example of writing four bytes of data to contiguous addresses starting at 0x1234.

Figure 10-1. Example of a Four-byte Write Starting at Address 0x1234

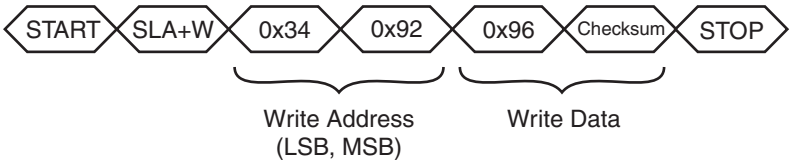


10.3 I²C Writes in Checksum Mode

In I²C checksum mode an 8-bit CRC is added to all I²C 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 device 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 command shown in Figure 10-2 writes a value of 150 (0x96) to address 0x1234 with a checksum. The address is changed to 0x9234 to indicate checksum mode.

Figure 10-2. Example of a Write To Address 0x1234 With a Checksum



10.4 Reading From the Device

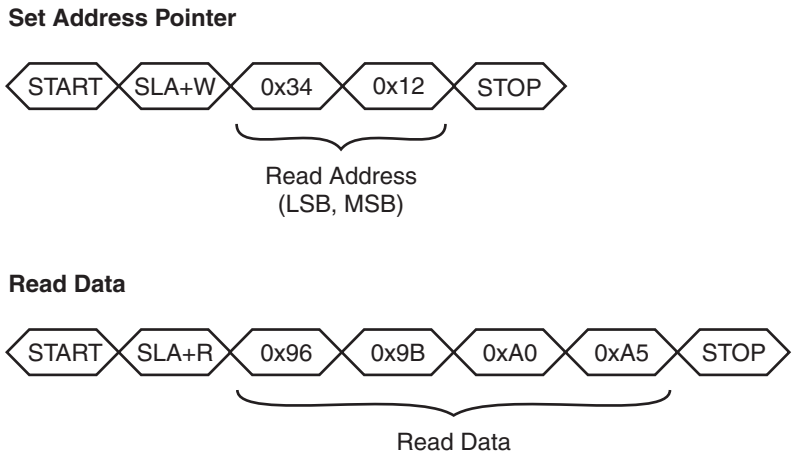
Two I²C bus activities must take place to read from the device. The first activity is an I²C write to set the address pointer (LSByte then MSByte). The second activity is the actual I²C read to receive the data. The address pointer returns to its starting value when the read cycle NACK is detected.

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 T5 object, the address pointer is automatically reset to allow continuous reads (see Section 10.5).

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

Figure 10-3 shows the I²C commands to read four bytes starting at address 0x1234.

Figure 10-3. Example of a Four-byte Read Starting at Address 0x1234



10.5 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 T44 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 T44 object (one byte) to retrieve a count of the pending messages (refer to the *mXT225T-AT 1.0 Protocol Guide* for details).
4. The host calculates the number of bytes to read by multiplying the message count by the size of the Message Processor T5 object (2).
5. Note that the size of the Message Processor T5 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 × (size – 1).
6. 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.

1. The STOP condition at the end of the read resets the address pointer to its initial location, so it may already be pointing at the Message Count T44 object following a previous message read.
2. The host should have already read the size of the Message Processor T5 object in its initialization code.

7. 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 T44 object.

Figure 10-4 shows an example of using a continuous read operation to read three messages from the device without a checksum. Figure 10-5 on page 34 shows the same example with a checksum.

Figure 10-4. Continuous Message Read Example – No Checksum

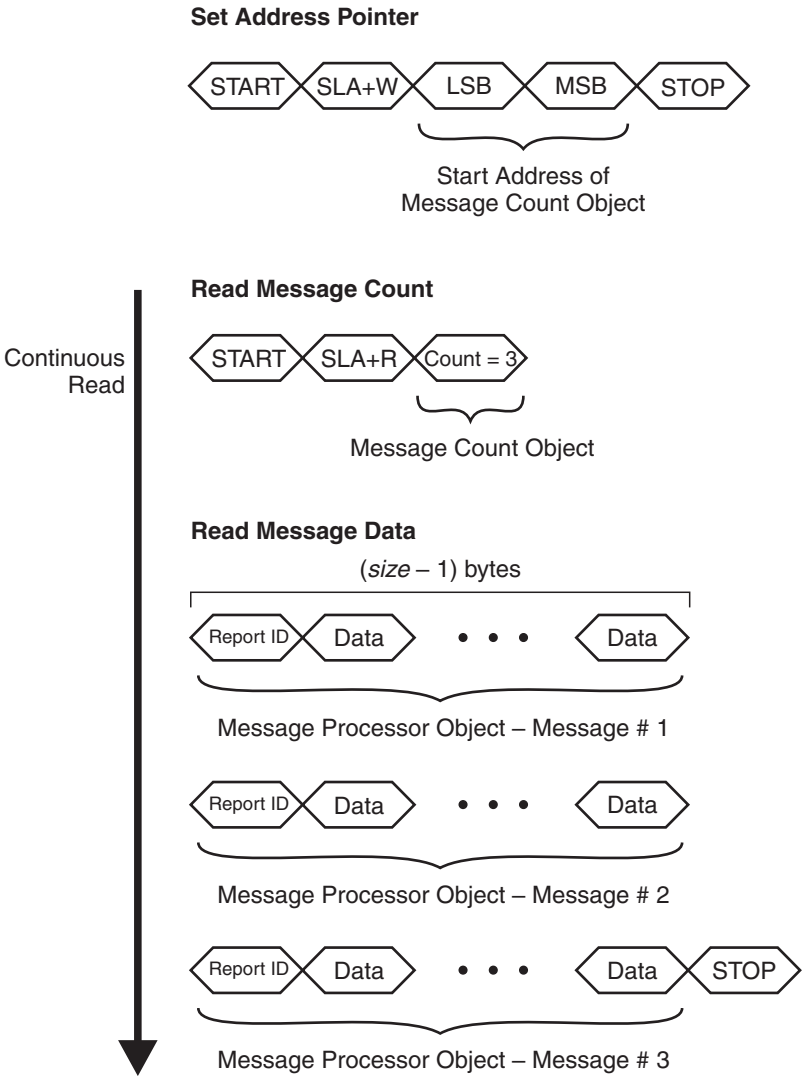
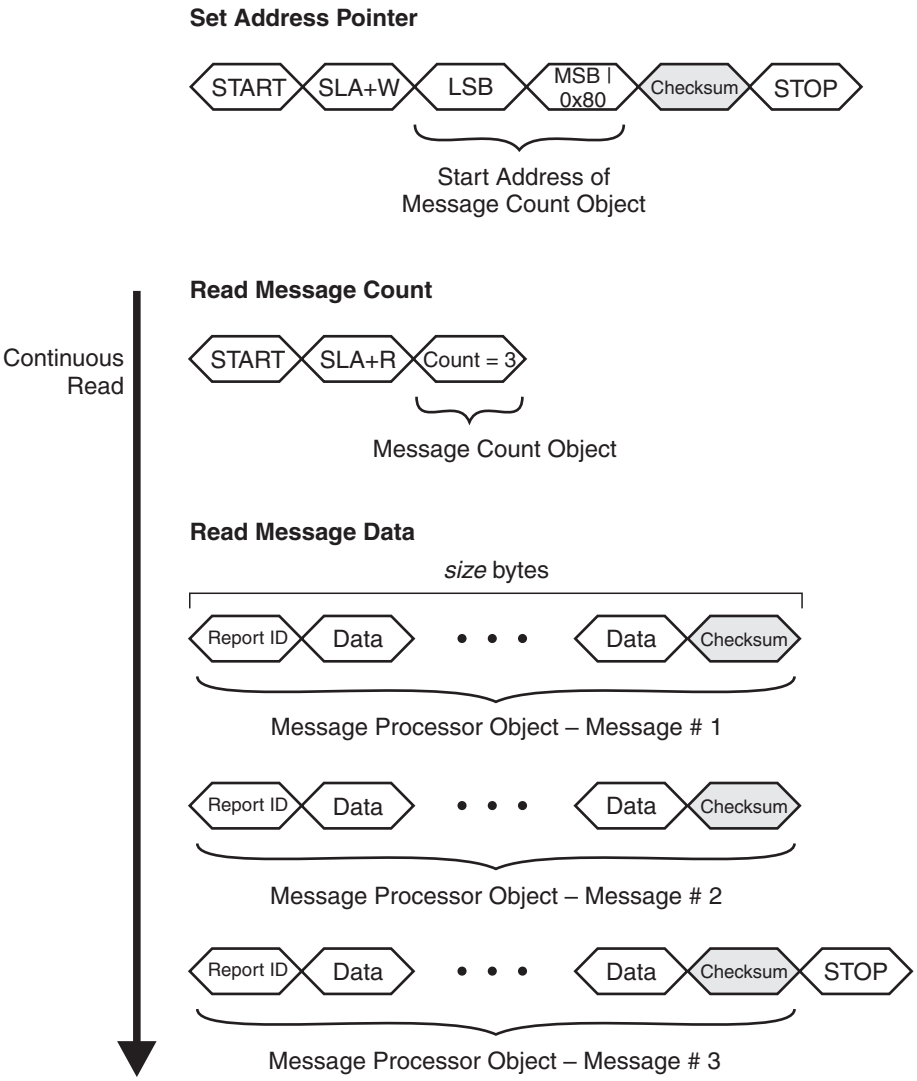


Figure 10-5. Continuous Message Read Example – I²C Checksum Mode



There are no checksums added on any other I²C reads. An 8-bit CRC can be added, however, to all I²C writes, as described in Section 10.3 on page 31.

An alternative method of reading messages using the $\overline{\text{CHG}}$ line is given in Section 10.6.

10.6 $\overline{\text{CHG}}$ 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 T5 object. This provides the host with an interrupt-style interface with the potential for fast response times. It reduces the need for wasteful I²C communications.

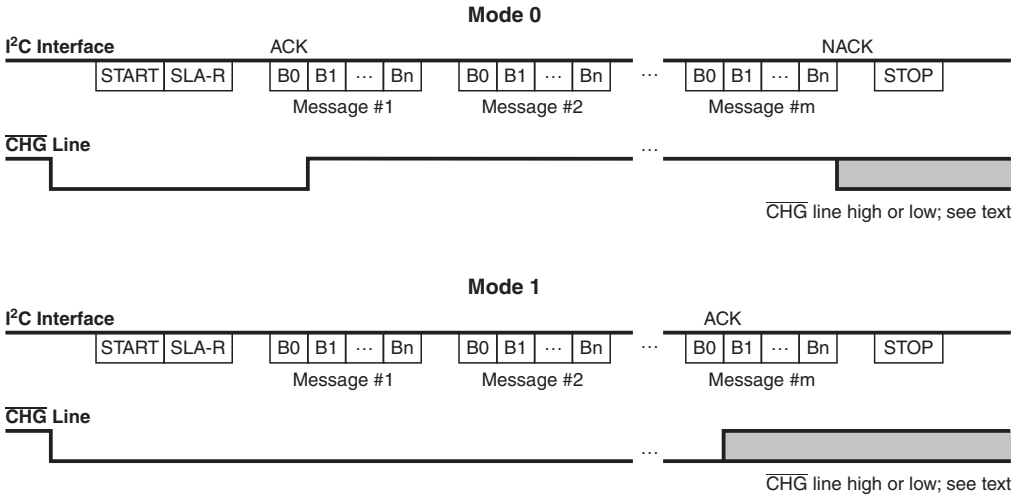
The $\overline{\text{CHG}}$ line remains low as long as there are messages to be read. The host should be configured so that the $\overline{\text{CHG}}$ line is connected to an interrupt line that is level-triggered. The host should not use an edge-triggered interrupt as this means adding extra software precautions.

The $\overline{\text{CHG}}$ line should be allowed to float during normal usage. This is particularly important after power-up or reset (see Section 7. on page 23).

A pull-up resistor is required, typically 47 kΩ to VddIO.

The $\overline{\text{CHG}}$ line operates in two modes, as defined by the Communications Configuration T18 object (refer to the *mXT225T-AT 1.0 Protocol Guide*).

Figure 10-6. $\overline{\text{CHG}}$ Line Modes for I²C-compatible Transfers



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 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 message 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 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 device.

Note: The state of the $\overline{\text{CHG}}$ line should be checked only between messages and not between the bytes of a message. The precise point at which the $\overline{\text{CHG}}$ line changes state cannot be predicted and so the state of the $\overline{\text{CHG}}$ line cannot be guaranteed between bytes.

The Communications Configuration T18 object can be used to configure the behavior of the $\overline{\text{CHG}}$ line. In addition to the $\overline{\text{CHG}}$ line operation modes described above, this object allows the use of edge-based interrupts, as well as direct control over the state of the $\overline{\text{CHG}}$ line. Refer to the *mXT225T-AT 1.0 Protocol Guide* for more information.

10.7 SDA, SCL

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

The termination resistors should be chosen so that the rise times on SDA and SCL meet the I²C specifications for the interface speed being used, bearing in mind other loads on the bus (see Section 15.7 on page 61).

10.8 Clock Stretching

The device supports clock stretching in accordance with the I^2C specification. It may also instigate a clock stretch if a communications event happens during a period when the device is busy internally. The maximum clock stretch is approximately 10 – 15 ms.

The device has an internal bus monitor that can reset the internal I^2C hardware if SDA or SCL is stuck low for more than 200 ms. This means that if a prolonged clock stretch of more than 200 ms is seen by the device, then any ongoing transfers with the device may be corrupted. The bus monitor is enabled or disabled using the Communications Configuration T18 object. Refer to the *mXT225T-AT 1.0 Protocol Guide* for more information.

11. SPI Communications

11.1 Communications Protocol

All communication with the device is carried out over the Serial Peripheral Interface (SPI). The host communicates with the mXT225T-AT over the SPI using a master-slave relationship, with the mXT225T-AT acting in slave mode.

11.2 SPI Operation

The SPI uses four logic signals:

- **Serial Clock (SCK)** – output from the host.
- **Master Output, Slave Input (MOSI)** – output from the host, input to the mXT225T-AT. Used by the host to send data to the mXT225T-AT.
- **Master Input, Slave Output (MISO)** – input to the host, output from the mXT225T-AT. Used by the mXT225T-AT to send data to the host.
- **Slave Select ($\overline{SS}$)** – active low output from the host.

In addition the following pin is used:

- **Change Line ($\overline{CHG}$)** – active low input to the host, output from the mXT225T-AT. Used by the mXT225T-AT to indicate that a response is ready for transmission (see Section 11.2.1) or that an OBP message is pending.

The master pulls $\overline{SS}$ low at the start of the SPI transaction and it remains low until the end of it.

At each byte, the master generates 8 clock pulses on SCK. With these 8 clock pulses, a byte of data is transmitted from the master to the slave over MOSI, most significant bit first.

Simultaneously a byte of data is transmitted from the slave to the master over MISO, also most significant bit first.

The mXT225T-AT requires that the clock idles “high” (CPOL=1). The data on MOSI and MISO pins are set at the falling edges and sampled at the rising edges (CPHA=1). This is known as SPI Mode 3.

The mXT225T-AT SPI interface can operate at a SCK frequency of up to 8 MHz.

The SPI interface is used in half duplex mode, even though it is a full duplex communication bus by its nature. This simplifies the protocol, minimizes the CPU processing required and avoids possible timing critical scenarios. This means that only one of the two in/out data lines (MOSI/MISO) will be meaningful at a time. During a read operation, therefore, the host must transmit 0xFF bytes on the MOSI line while it is reading data from the slave device. Similarly, during a write operation, the host must ignore the data on the MISO line.

An SPI transaction is considered as initiated when the $\overline{SS}$ line is asserted (active low) by the host and terminated when it is deasserted. The host can abort a transfer at any time by deasserting the $\overline{SS}$ line.

11.2.1 Change Line ($\overline{CHG}$)

The $\overline{CHG}$ line is an active-low, open-drain output that is used as an interrupt to alert the host that the slave is ready to send a response or that an OBP message is pending and ready to be read from the Host.

The change line must be handled by the host as a falling edge triggered line. It must not be used a level triggered line. This avoids the situation in which the host initiates a new read/write operation (because the interrupt line is still asserted following a previous SPI transaction) but the target is not yet ready to handle it.

To prevent the host missing an interrupt, the target device can use a retriggering mechanism for the interrupt line. This guarantees that any pending message is always delivered. This mechanism must be enabled in the Communications Configuration T18 object (refer to the *mXT225T-AT 1.0 Protocol Guide* for more information).

11.2.2 SPI Protocol Opcodes

The allowed operations and responses codes used by the SPI protocol are shown in Table 11-1.

Table 11-1. SPI Opcodes

Name	Value	Operation
Write Operation and Responses (see Section 11.3)		
SPI_WRITE_REQ	0x01	Write operation request
SPI_WRITE_OK	0x81	Write operation succeeded (response)
SPI_WRITE_FAIL	0x41	Write operation failed (response)
Read Operation and Responses (see Section 11.4 on page 41)		
SPI_READ_REQ	0x02	Read operation request
SPI_READ_OK	0x82	Read operation succeeded (response)
SPI_READ_FAIL	0x42	Read operation failed (response)
General Responses (see Section 11.5 on page 44)		
SPI_INVALID_REQ	0x04	Invalid operation (response)
SPI_INVALID_CRC	0x08	Invalid CRC (response)

All the responses reported in Table 11-1 require the Interrupt line to go from inactive (deasserted) to active (asserted) before the host can read a response following an SPI_READ_REQ or SPI_WRITE_REQ operation.

11.2.3 SPI Transaction Header

Every SPI transaction includes a 6-byte HEADER that has the format shown in Table 11-2.

Table 11-2. Header Format

Byte	Field	Description
0	Opcode	Op code for the transaction
1	Address LSByte	The memory address of the slave device where the Host wants to write to or read from.
2	Address MSByte	
3	Length LSByte	The number of bytes that the host wants to write to or read from the slave device.
4	Length MSByte	
5	CRC	8-bit CRC

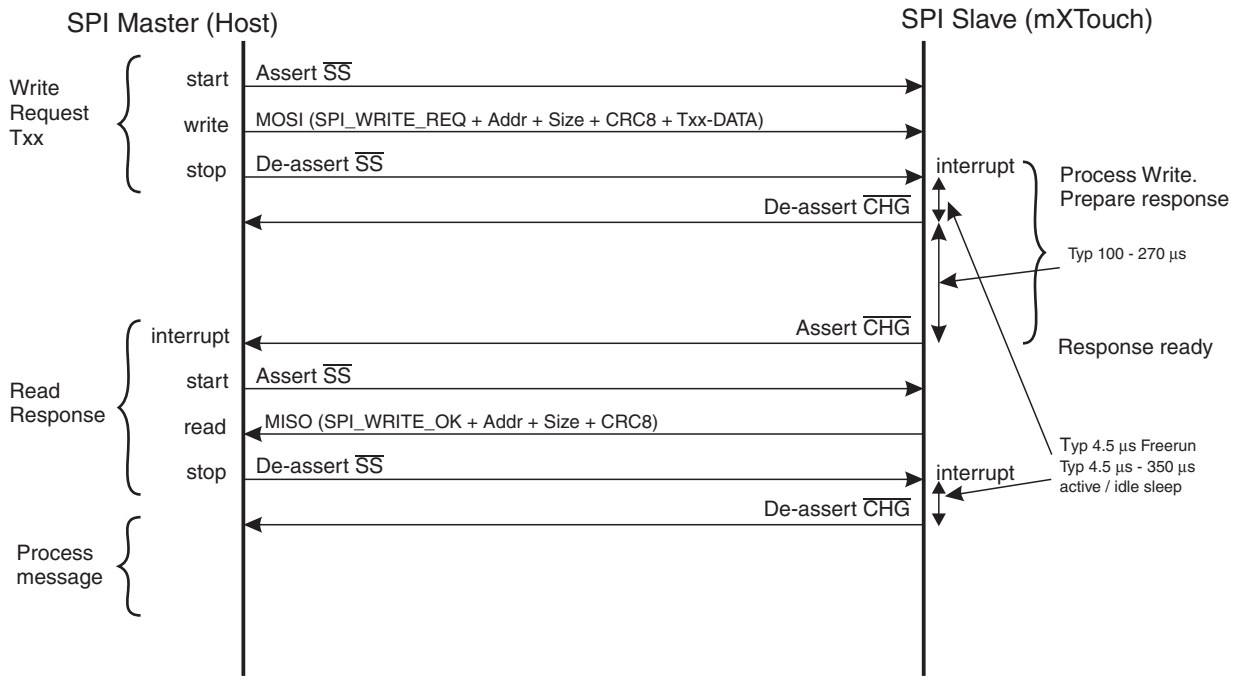
An 8-bit CRC is used to detect errors on the 5 bytes of the header (that is: Opcode, Address LSB, Address MSB, Length LSByte, Length MSByte) in order to prevent the writing to or reading from unwanted objects if the header gets corrupted during the SPI transfer. The 8-bit CRC algorithm is the same as that used to calculate the CRC for Message Processor T5 messages (refer to the *mXT225T-AT 1.0 Protocol Guide* for details).

11.3 Write Operation and Responses

The write operation and its responses allows the host to write to an object configuration area.

The flow and timing are shown in Figure 11-1

Figure 11-1. SPI Write Configuration Message Flow and Timing

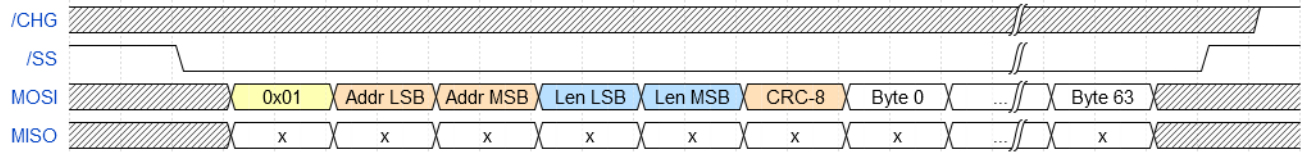


Note: No detection mechanism is provided at the SPI network layer level on the data written, but further to the possibility of reading back the data written to check its correctness the Host can also use the Communication Checksum Mode (as reported in the appendix of the Protocol Guide) that provides to the Host the capability to detect whether the payload of the write operation was corrupted or not during the SPI transaction. (see Figure 11-5 on page 41).

11.3.1 SPI_WRITE_REQ

Figure 11-2 shows the message format used for the write request operation.

Figure 11-2. SPI_WRITE_REQ



In Figure 11-2:

- **0x01** is the opcode
- **Addr LSB and Addr MSB** together specify the address to which the host wishes to write
- **Len LSB and Len MSB** together specify the length of the data in bytes. This is the total number of bytes that the Host wishes to write to the slave device (excluding the header bytes)
- **CRC-8** is the 8-bit CRC
- **Byte 0 .. Byte 63** contain the data that is to be written (64 bytes maximum).

If the host needs to write more than 64 bytes of data then multiple SPI_WRITE_REQ operations are required.

Following an SPI_WRITE_REQ operation, the host must wait for a response from the device before accessing the SPI bus again. If the slave system does not assert the interrupt line within 10 ms, a HW reset or a retry from the Host is necessary. When the response is ready to be sent, the target device asserts the interrupt line to notify the host that a message is ready to be read. Only at this point is the host allowed to initiate a new SPI transaction to read back the response related to the previous write operation.

This means that an object message will be blocked during the time that a response related to a previous read or write request is pending and has not yet been read back by the Host.

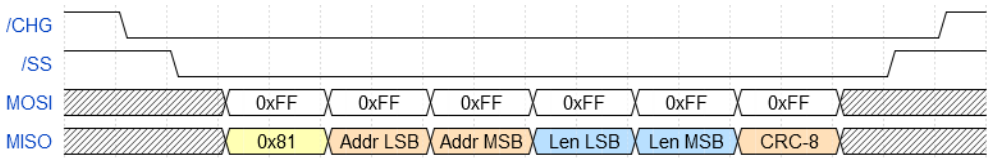
The following responses are possible following an SPI_WRITE_REQ operation:

- SPI_WRITE_OK – Generated if the write operation was successfully completed (the memory address and length specified by the host were within the allowed accessible memory map regions). See Section 11.3.2
- SPI_WRITE_FAIL – Generated if the write operation failed, for example if the host tries to write to an address outside the available memory map. See Section 11.3.3
- SPI_INVALID_REQ – See Section 11.5.1 on page 44
- SPI_INVALID_CRC – See Section 11.5.2 on page 44

11.3.2 SPI_WRITE_OK

Figure 11-3 shows the message format used for the write OK response.

Figure 11-3. SPI_WRITE_OK



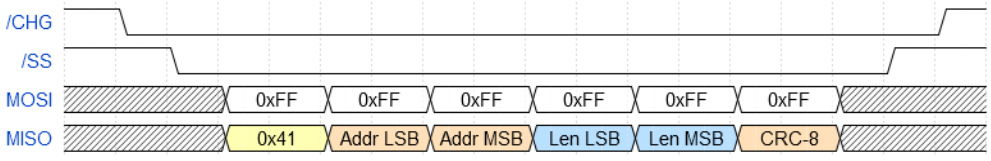
In Figure 11-3:

- **0x81** is the opcode
- **Addr LSB and Addr MSB** together specify the address to which the data was written
- **Len LSB and Len MSB** together specify the length of the data in bytes. This is the total number of bytes that was written to the slave device (excluding the header bytes)
- **CRC-8** is the 8-bit CRC

11.3.3 SPI_WRITE_FAIL

Figure 11-4 shows the message format used for the write fail response.

Figure 11-4. SPI_WRITE_FAIL



In Figure 11-4:

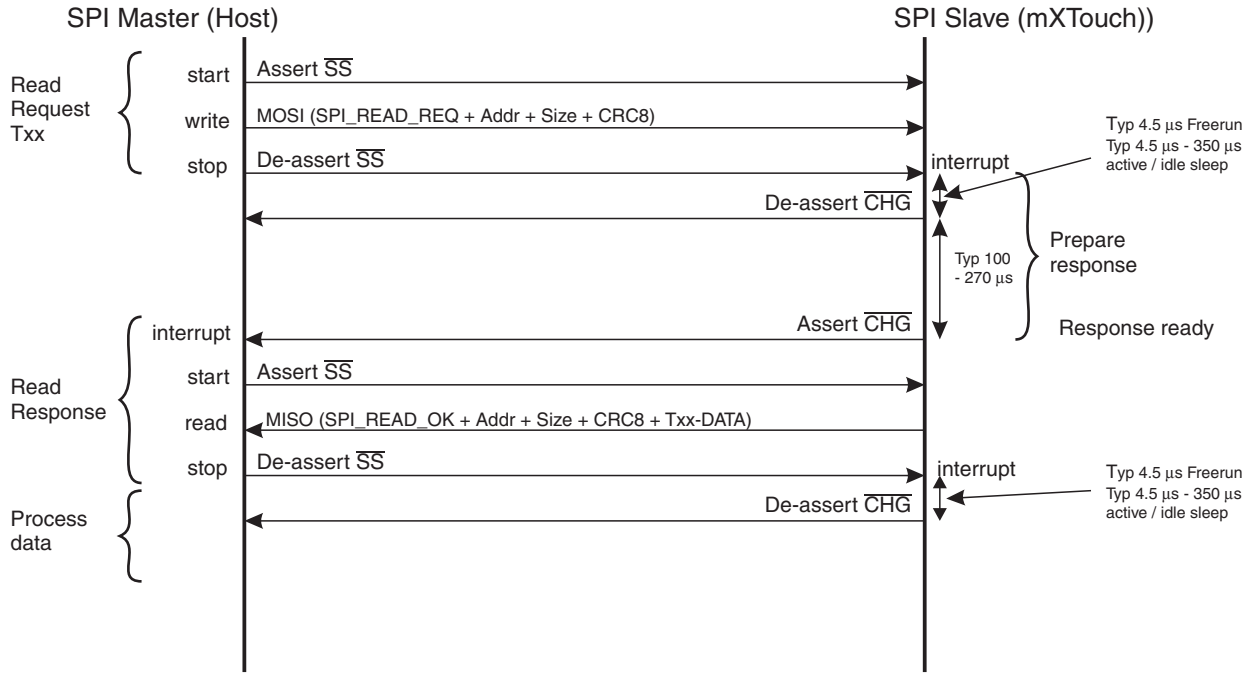
- **0x41** is the opcode
- **Addr LSB and Addr MSB** together specify the address to which the host requested the write
- **Len LSB and Len MSB** together specify the length of the data in bytes. This is the total number of bytes that the Host attempted to write to the slave device (excluding the header bytes)
- **CRC-8** is the 8-bit CRC

11.4 Read Operation and Responses

The read request operation allows the host to read from the object memory map for the device. This allows the host to read a message from the Message Processor T5 object or read from an object configuration area.

The flow and timing are shown in Figure 11-5

Figure 11-5. SPI Read Configuration Message Flow and Timing



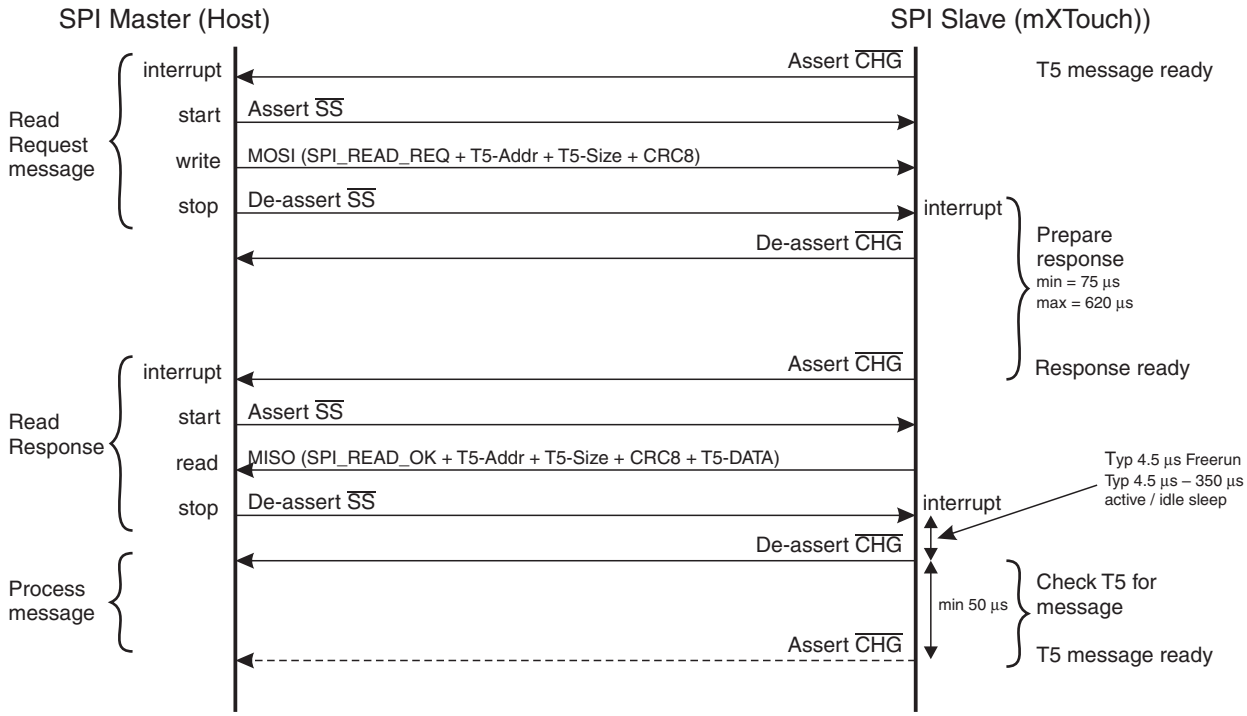
Normally a limit of 64 bytes is allowed for data reads. If the host tries to read more than 64 bytes, the slave returns SPI_READ_FAIL (see Section 11.4.3 on page 43). A mechanism is provided, however, that supports the DMA transfer of a large block of data that exceeds this limit. This is achieved by the provision of multiple instances of the Data Container T117 object within the device that allow up to 1290 bytes of data to be read in a contiguous manner. Refer to the *mXT225T-AT 1.0 Protocol Guide* for more information on the Data Container T117 object and the Data Container Controller T118 used to configure the data.

Under certain circumstances, a CRC can be used as an error detection mechanism when reading an object:

- Message Processor T5 – When reading a message from the Message Processor T5 object, an optional CRC as an error detection mechanism is provided. This is enabled in the Message Processor T5 object (refer to the *mXT225T-AT 1.0 Protocol Guide* for details).
- Data Container T117 – When performing a block data transfer from Data Container T117 instances, however, the header bytes within the data can be configured to provide a CRC on the data. Refer to the *mXT225T-AT 1.0 Protocol Guide* for details on the Data Container Controller T118 used to configure the CRC.
- All other objects – When reading from any other object configuration area, no error detection mechanism is provided, as this operation is typically performed only at system startup. It is possible, however, to verify a read operation by performing it twice and comparing the results.

The complete flow of messages to read a T5 message and the associated timing is shown in Figure 11-6.

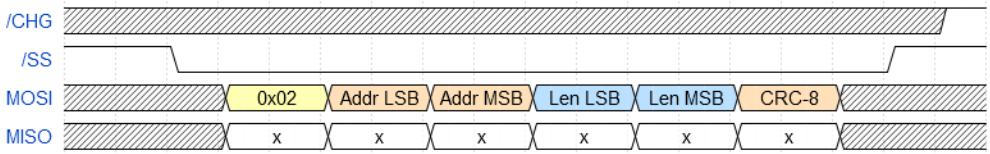
Figure 11-6. SPI Read T5 Message Flow and Timing



11.4.1 SPI_READ_REQ

Figure 11-7 shows the message format used for the read request operation.

Figure 11-7. SPI_READ_REQ



The SPI_READ_REQ operation can be initiated by the host at any time, regardless of the state of the interrupt line. The slave device will assert the interrupt line when there are object messages pending. When the master asserts SS (whether to respond to the slave asserting the interrupt line or because the master wants to initiate a transaction), the interrupt line is deasserted until the message from the master has been received and processed.

In Figure 11-7:

- **0x02** is the opcode
- **Addr LSB and Addr MSB** together specify the address from which the host wishes to read
- **Len LSB and Len MSB** together specify the length of the data in bytes. This is the total number of bytes (excluding the header bytes) that the Host wishes to read from the slave device. The limit is 64 bytes for normal reads, and 1290 maximum for a block data transfer from Data Container T117 instances
- **CRC-8** is the 8-bit CRC

The actual data is sent in the subsequent SPI_READ_OK operation.

Following an SPI_READ_REQ operation, the host must wait for a response to be ready from the device before accessing the SPI bus again. If the slave system does not assert the interrupt line within 10 ms, a HW reset or a retry from the Host is necessary. When the response is ready to be sent, the target device asserts the interrupt line to notify the host that a message is ready to be read. Only at this point is the host allowed to initiate a new SPI transaction to read back the response related to the previous write operation.

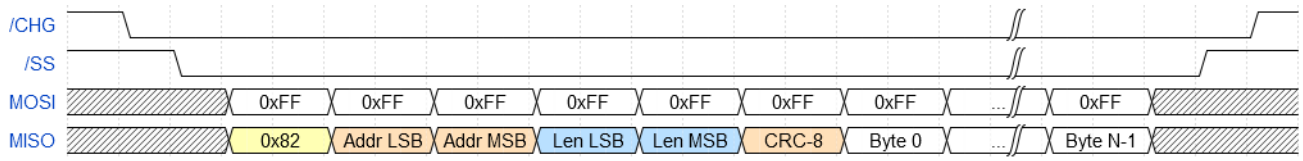
The following responses are possible following an SPI_READ_REQ operation:

- SPI_READ_OK – Generated if the read operation was successfully completed (the memory address and length specified by the host were within the allowed accessible memory map regions). See Section 11.4.2
- SPI_READ_FAIL – Generated if the read operation failed, for example if the host tries to read from an address outside the available memory map. See Section 11.4.3
- SPI_INVALID_REQ – See Section 11.5.1 on page 44
- SPI_INVALID_CRC – See Section 11.5.2 on page 44

11.4.2 SPI_READ_OK

Figure 11-8 shows the message format used for the read OK response.

Figure 11-8. SPI_READ_OK



In Figure 11-8:

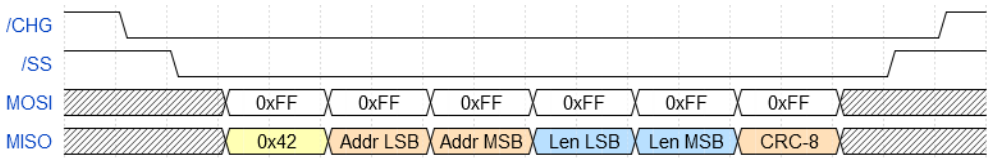
- **0x82** is the opcode
- **Addr LSB and Addr MSB** together specify the address from which the host requested the data should be read
- **Len LSB and Len MSB** together specify the length of the data in bytes. This is the total number of bytes that the Host requested to read from the slave device (excluding the header bytes)
- **CRC-8** is the 8-bit CRC
- **Byte 0 .. Byte N-1** contain the data that is to be written, where *N* the number of bytes (maximum 64 bytes for normal reads, and 1290 for block data transfers from Data Container T117 instances)

Note that, although the slave device flushes the transmit buffer when the host performs a read operation, any attempt by the Host to read more data than expected (that is, greater than Len bytes) could cause the slave device to transmit junk data on the MISO line.

11.4.3 SPI_READ_FAIL

Figure 11-9 shows the message format used for the read fail response.

Figure 11-9. SPI_READ_FAIL



In Figure 11-9:

- **0x42** is the opcode
- **Addr LSB and Addr MSB** together specify the address from which the host requested the data should be read

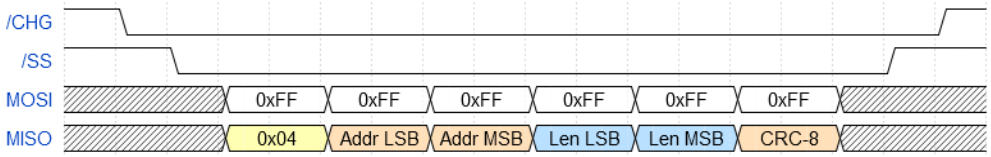
- **Len LSB and Len MSB** together specify the length of the data in bytes. This is the total number of bytes that the Host attempted to read from the slave device (excluding the header bytes)
- **CRC-8** is the 8-bit CRC

11.5 General Operations

11.5.1 SPI_INVALID_REQ

Figure 11-10 shows the message format used for the Invalid Request response. The purpose of this opcode is to report to the host that the opcode of the last request was not recognized or that the Host has tried to perform another read or write operation without waiting for the response from the previous request.

Figure 11-10. SPI_INVALID_REQ



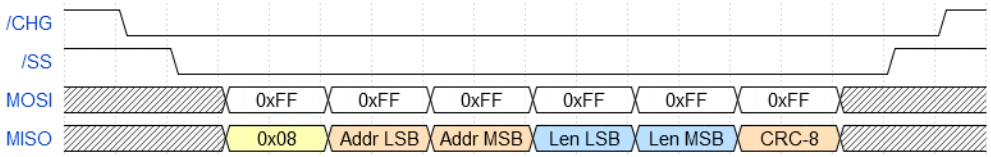
In Figure 11-10:

- **0x04** is the opcode
- **Addr LSB and Addr MSB** together specify the address received in the invalid request
- **Len LSB and Len MSB** together specify the length of the data in bytes. This is the total number of bytes that the Host attempted to read from or write to from the slave device (excluding the header bytes)
- **CRC-8** is the 8-bit CRC

11.5.2 SPI_INVALID_CRC

Figure 11-11 shows the message format used for the Invalid CRC response. The purpose of this opcode is to report an error in the CRC check performed on the received data.

Figure 11-11. SPI_INVALID_CRC



In Figure 11-10:

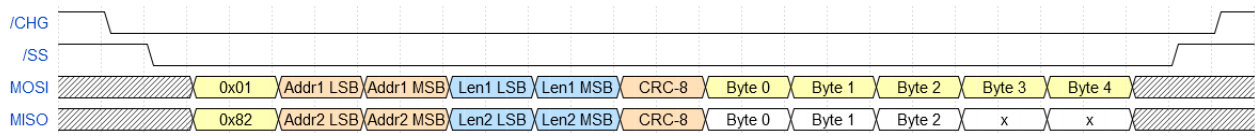
- **0x08** is the opcode
- **Addr LSB and Addr MSB** together specify the address received in the last request
- **Len LSB and Len MSB** together specify the length of the data in bytes. This is the total number of bytes that the Host attempted to read from or write to from the slave device in the last request (excluding the header bytes)
- **CRC-8** is the 8-bit CRC

11.6 Example of Failed Protocol

In order to prevent unpredictable system behavior, the host *must* always wait for the response of the last request issued to be ready before initiating a new SPI request transaction. If the host does not comply with the protocol specification, clashes can occur.

For example, Figure 11-12 shows the situation in which an SPI_READ_OK (0x82) response with a payload of 3 bytes is expected, but the host performs an SPI_WRITE_ REQ (0x01) operation instead to write 5 bytes to address *Addr1*. In this case, the slave device outputs the SPI_READ_OK data on the MISO line (this will have been prepared in advance before the interrupt line was asserted) and ignores the new Host request received on the MOSI line. In this case the slave device will return to the Host, in response to the following read or write request, an SPI_INVALID_REQ response to indicate a violation of the SPI protocol.

Figure 11-12. Clash of SPI_WRITE_REQ while SPI_READ_OK is expected



12. PCB Design Considerations

12.1 Introduction

The following sections give the design considerations that should be adhered to when designing a PCB layout for use with the mXT225T-AT. Of these, power supply and ground tracking considerations are the most critical.

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.

12.2 Printed Circuit Board

Atmel recommends the use of a four-layer printed circuit board for mXT225T-AT 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.

12.2.1 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 device 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.

12.3 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 0 V plane. The flood filling should be done on the outside layers of the board.

12.4 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 supply pins as possible (less than 10 mm away). The ground connection of these capacitors should be tracked to 0 V by the shortest, heaviest traces possible.

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

In addition, at least one ‘bulk’ 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 PLDs* (doc0484.pdf; available on the Atmel website) for further general information on decoupling capacitors.

12.5 Single Supply Operation

When designing a PCB for an application using a single LDO, extra care should be taken to ensure short, low inductance traces between the supply and the touch controller supply input pins. Ideally, tracking for the individual supplies should be arranged in a star configuration, with the LDO at the junction of the star. This will ensure that supply current variations or noise in one supply rail will have minimum effect on the other supplies. In applications where a ground plane is not practical, this same star layout should also apply to the power supply ground returns.

12.6 Analog I/O

In general, tracking for the analog I/O signals from the device 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.

12.7 Component Placement and Tracking

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.

12.7.1 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.

12.8 EMC and Other Observations

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

- 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 0 V 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 high capacity ceramic type, have the requisite low ESR, ESL and good stability/temperature properties. Refer to the regulator manufacturer’s datasheet for more information.

13. Getting Started with mXT225T-AT

13.1 Establishing Contact

13.1.1 Communication with the Host

The host can use the following interface to communicate with the device:

- I^2C interface (see Section 10. on page 31)
- SPI interface (see Section 11. on page 37)

13.1.2 Power-up Sequence

On power-up, the $\overline{\text{CHG}}$ line goes low to indicate that there is new data to be read from the Message Processor T5 object. If the $\overline{\text{CHG}}$ line does not go low, there is a problem with the device.

The host should attempt to read any available messages to establish that the device 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.

13.2 Using the Object Protocol

The device has an object-based protocol that is used to communicate with the device. Typical communication includes configuring the device, sending commands to the device, and receiving messages from the device. Refer to the *mXT225T-AT 1.0 Protocol Guide* for more information.

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

1. Read the start positions of all the objects in the device 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 device can be correctly interpreted.

13.2.1 Classes of Objects

The mXT225T-AT contains the following classes of objects:

- **Debug objects** – provide a raw data output method for development and testing.
- **General objects** – required for global configuration, transmitting messages and receiving commands.
- **Touch objects** – operate on measured signals from the touch sensor and report touch data.
- **Signal processing objects** – process data from other objects (typically signal filtering operations).
- **Support objects** – provide additional functionality on the device.

13.2.2 Object Instances

Table 13-1. Objects on the mXT225T-AT

Object	Description	Number of Instances	Usage
Debug Objects			
Diagnostic Debug T37	Allows access to diagnostic debug data to aid development.	1	Debug commands only. No configuration/tuning necessary. Not for use in production.
General Objects			
Message Processor T5	Handles the transmission of messages. This object holds a message in its memory space for the host to read.	1	No configuration necessary.

Table 13-1. Objects on the mXT225T-AT (Continued)

Object	Description	Number of Instances	Usage
Command Processor T6	Performs a command when written to. Commands include reset, calibrate and backup settings	1	No configuration necessary.
Power Configuration T7	Controls the sleep mode of the device. Power consumption can be lowered by controlling the acquisition frequency and the sleep time between acquisitions.	1	Must be configured before use.
Acquisition Configuration T8	Controls how the device takes each capacitive measurement.	1	Must be configured before use.
Touch Objects			
Key Array T15	Creates a rectangular array of keys. A Key Array T15 object reports simple on/off touch information.	2	Enable and configure as required.
Multiple Touch Touchscreen T100	Creates a Touchscreen that supports the tracking of more than one touch.	1	Enable and configure as required.
Signal Processing Objects			
One-touch Gesture Processor T24	Operates on the data from a Touchscreen object. A One-touch Gesture Processor T24 converts touches into one-touch finger gestures (for example, taps, double taps and drags).	1	Enable and configure as required.
Two-touch Gesture Processor T27	Operates on the data from a One-touch Gesture Processor T24 object. A Two-touch Gesture Processor T27 converts touches into two-touch finger gestures (for example, pinches, stretches and rotates).	1	Enable and configure as required.
Grip Suppression T40	Suppresses false detections caused, for example, by the user gripping the edge of the touchscreen.	1	Enable and configure as required.
Touch Suppression T42	Suppresses false detections caused, for example, by the user placing their face too near the touchscreen on a mobile phone.	1	Enable and configure as required.
Passive Stylus T47	Processes passive stylus input.	1	Enable and configure as required.
Shieldless T56	Allows a sensor to use true single-layer co-planar construction.	1	Enable and configure as required.
Lens Bending T65	Compensates for lens deformation (lens bending) by attempting to eliminate the disturbance signal from the reported deltas.	3	Enable and configure as required.
maXCharger T72	Performs various noise reduction techniques during touchscreen signal acquisition.	1	Enable and configure as required.
Glove Detection T78	Allows for the reporting of glove touches.	1	Enable and configure as required.
Retransmission Compensation T80	Limits the negative effects on touch signals caused by poor device coupling to ground.	1	Enable and configure as required.
Gesture Processor T84	Performs gesture processing.	1	Enable and configure as required.

Table 13-1. Objects on the mXT225T-AT (Continued)

Object	Description	Number of Instances	Usage
Self Capacitance maXCharger T108	Suppresses the effects of external noise within the context of self capacitance touch measurements.	1	Enable and configure as required.
Self Capacitance Grip Suppression T112	Allows contacting and hovering touches to be reported from the self capacitance measurements while the device is being gripped.	2	Enable and configure as required.
Support Objects			
Communications Configuration T18	Configures additional communications behavior for the device.	1	Check and configure as necessary.
GPIO/PWM Configuration T19	Creates a bank of digital I/O pins. These pins can then be controlled from the host by writing to the object's configuration memory space.	1	Enable and configure as required.
Self Test T25	Configures and performs self-test routines to find faults on a touch sensor.	1	Pin test commands only. No configuration/tuning necessary. Not for use in production.
User Data T38	Provides a data storage area for user data.	1	Read-only object. Configure as required.
Message Count T44	Provides a count of pending messages.	1	Read only object.
CTE Configuration T46	Controls the capacitive touch engine for the device.	1	Must be configured.
Timer T61	Provides control of a timer.	6	Enable and configure as required.
Serial Data Command T68	Provides an interface for the host driver to deliver various data sets to the device.	1	Enable and configure as required.
Dynamic Configuration Controller T70	Allows rules to be defined that respond to system events.	20	Enable and configure as required.
Dynamic Configuration Container T71	Allows the storage of user configuration on the device that can be selected in run-time based on rules defined in the T70 object.	1	Configure if Dynamic Configuration Controller T70 is in use.
CTE Scan Configuration T77	Configures enhanced X line scanning features.	1	Enable and configure as required.
Touch Event Trigger T79	Configures touch triggers for use with the event handler.	3	Enable and configure as required.
Touchscreen Hover Configuration T101	Provides controls specific to self-capacitance hover measurements and hovering touch support.	1	Enable and configure as required.
Auxiliary Touch Configuration T104	Allows the setting of self capacitance gain and thresholds for a particular measurement to generate auxiliary touch data for use by other objects.	1	Enable and configure if using self capacitance measurements
Self Capacitance Global Configuration T109	Provides configuration for a self capacitance measurements employed on the device.	1	Check and configure as required (if using self capacitance measurements).
Self Capacitance Tuning Parameters T110	Provides configuration space for a generic set of settings for self capacitance measurements.	9	Use under the guidance of Atmel field engineers only.

Table 13-1. Objects on the mXT225T-AT (Continued)

Object	Description	Number of Instances	Usage
Self Capacitance Configuration T111	Provides configuration for a self capacitance measurements employed on the device.	3	Check and configure as required (if using self capacitance measurements).
Self Capacitance Measurement Configuration T113	Configures self capacitance measurements to generate data for use by other objects.	1	Enable and configure as required.
Data Container T117	Provides a mechanism for storing contiguous data in a set of read-only objects.	6	Read-only object. No configuration necessary.
Data Container Controller T118	Provides direct access to internal data in memory for use with the Data Container T117 objects.	1	Enable and configure as required.

13.2.3 Configuring and Tuning the Device

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 T6 object.

Perform the following actions for each object:

1. Enable the object, if the object requires it.
2. Configure the fields in the object, as required.
3. Enable reporting, if the object supports messages, to receive messages from the object.

For information on configuring and tuning the mXT225T-AT, refer to the following document:

- *maXTouch T Series Tuning Guide*

Refer also to the *mXT225T-AT 1.0 Protocol Guide* for detailed information on the configuration parameters for the objects.

13.3 Writing to the Device

There are a number of mechanisms for writing to the device:

- Using an I^2C write operation (see Section 10.2 on page 31).
- Using the SPI write operation (see Section 11.3 on page 38).

To communicate with the device, you write to the appropriate object:

- To send a command to the device, you write the appropriate command to the Command Processor T6 object (refer to the *mXT225T-AT 1.0 Protocol Guide*).
- To configure the device, you write to an object. For example, to configure the device power consumption you write to the global Power Configuration T7 object, and to set up a touchscreen you write to a Multiple Touch Touchscreen T100 object. Some objects are optional and need to be enabled before use. Refer to the *mXT225T-AT 1.0 Protocol Guide* for more information on the objects.

13.4 Reading from the Device

Status information is stored in the Message Processor T5 object. This object can be read to receive any status information from the device. The following mechanisms provide an interrupt-style interface for reading messages in the Message Processor T5 object:

- The $\overline{\text{CHG}}$ line is asserted whenever a new message is available in the Message Processor T5 object (see Section 10.6 on page 34). See Section 10.4 on page 32 for information on the format of the I^2C read operation.
- When using the SPI interface, two SPI transactions must take place: the first is an SPI Read request which is used to set the address pointer (Address LSByte and MSByte) and to indicate to the slave device how many bytes (Length LSByte and MSByte) the Host wants to read; the second is a response which comes with a payload that actually contains the data that was requested (see Section 11.4 on page 41.

Note that the host should always wait to be notified of messages. The host should not poll the device for messages. When the SPI interface is used, it is mandatory to not poll the interrupt line (because in such case the Host handling of the $\overline{\text{CHG}}$ line will be level based instead of falling edge based as it is strictly required).

14. Debugging and Tuning

14.1 Hardware SPI Debug Interface

This interface is used for tuning and debugging when running the system and allows the development engineer's PC running Atmel maXTouch Studio to read the real-time raw data using the low-level debug port. This can be accessed via the SPI interface.

The SPI debug interface consists of the DBG_CLK, DBG_DAT and $\overline{\text{DBG\_SS}}$ lines. These pins should be routed to test points on all designs such that they can be connected to external hardware during system development. These lines should not be connected to power or GND.

The touch controller will take care of the pin configuration. When these lines are in use, any alternative function for the pins cannot be used.

The Hardware interface is enabled by the Command Processor T6 object and by default will be off.

Refer to the following documents for more information:

- *mXT225T-AT 1.0 Protocol Guide* for information on the Command Processor T6 object
- QTAN0050, *Using the maXTouch Debug Port*, for information on using the debug port
- *maXTouch T Series Tuning Guide* (distributed with Atmel approval only) for guidance on using the debug interface for tuning purposes

14.2 Object-based Protocol

The device provides a mechanism for obtaining raw data for development and testing purposes by reading data from the Diagnostic Debug T37 object. Refer to the *mXT225T-AT 1.0 Protocol Guide* for more information on this object.

Note that the Diagnostic Debug T37 is of most use for simple tuning purposes. When debugging a design, it is preferable to use a hardware debug interface, as this will have a much higher bandwidth and can provide real time data.

14.3 Self Test

There is also a Self Test T25 object that runs self-test routines in the device to find hardware faults on the sense lines and the electrodes. This object also performs an initial pin fault test on power-up to ensure that there is no X-to-Y short before the high-voltage supply is enabled inside the chip. A high-voltage short into the analog circuitry would break the device.

Refer to the *mXT225T-AT 1.0 Protocol Guide* for more information on the Self Test T25 object.

15. Specifications

15.1 Absolute Maximum Specifications

Vdd	3.6 V
VddIO	3.6 V
AVdd	3.6 V
XVdd	9.0 V
Voltage forced onto any pin	–0.3 V to (Vdd, VddIO or AVdd) + 0.3 V
Configuration parameters maximum writes (flash memory write cycles)	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.

15.2 Recommended Operating Conditions

Operating temp	–40°C to +85°C
Storage temp	–60°C to +150°C
Vdd	3.3 V
VddIO	1.8 V to 3.3 V
AVdd	3.3 V
XVdd with internal voltage doubler	Vdd to 2 × Vdd
Cx transverse load capacitance per node	0.6 pF to 3 pF
Temperature slew rate	10°C/min

15.2.1 DC Characteristics

15.2.1.1 Analog Voltage Supply – AVdd

Parameter	Min	Typ	Max	Units	Notes
AVdd					
Operating limits	3.14	3.3	3.47	V	
Supply Rise Rates	–	–	0.25	V/µs	

15.2.1.2 Digital Voltage Supply

Parameter	Min	Typ	Max	Units	Notes
VddIO					
Operating limits	1.71	–	3.47	V	I^2C-compatible
Supply Rise Rates	–	–	0.25	V/µs	
Vdd					
Operating limits	3.14	3.3	3.47	V	
Supply Rise Rates	–	–	0.25	V/µs	
Supply Fall Rates	–	–	50	V/ms	

15.2.2 Power Supply Ripple and Noise

Parameter	Min	Typ	Max	Units	Notes
Vdd	–	–	±50	mV	Across frequency range 1 Hz to 1 MHz
AVdd	–	–	±50	mV	Across frequency range 1 Hz to 1 MHz
AVdd (with noise suppression enabled)	–	–	±40	mV	Across frequency range 1 Hz to 1 MHz, with Noise Suppression enabled

15.3 Test Configuration

The values listed below were used in the reference unit to validate the interfaces and derive the characterization data provided in the following sections.

See *mXT225T-AT 1.0 Protocol Guide* for information about the individual objects and their fields.

The values for the user application will depend on the circumstances of that particular project and will vary from those listed here. Further tuning will be required to achieve an optimal performance.

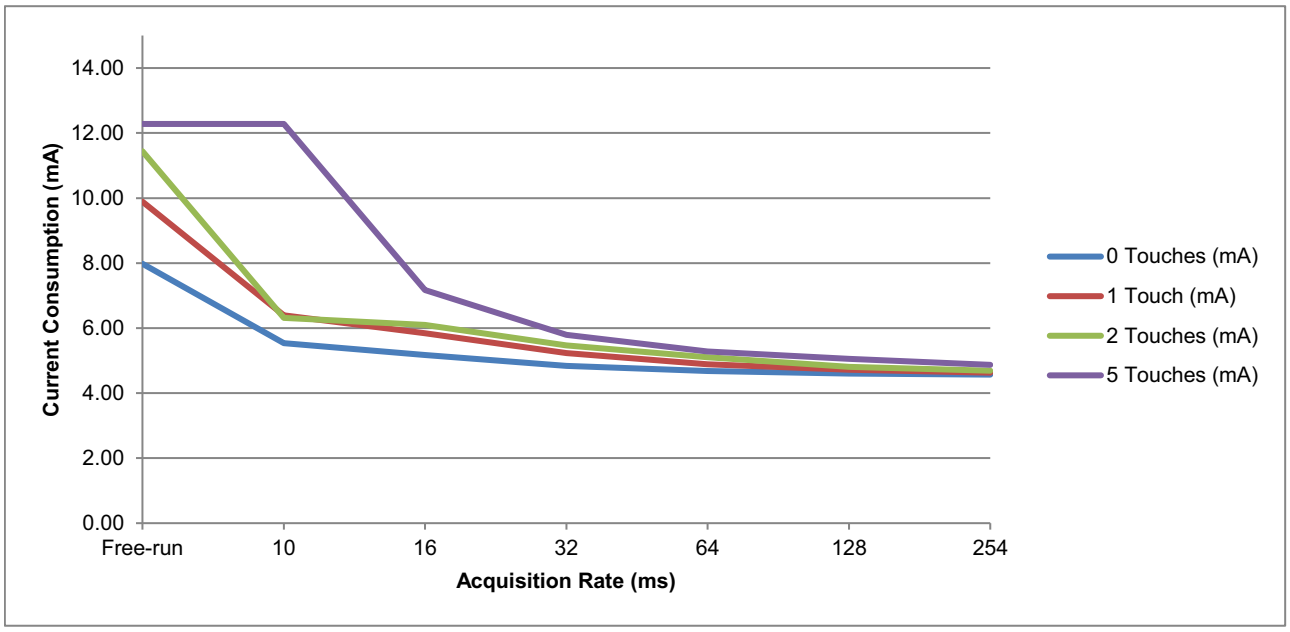
Table 15-1. Test Configuration

Object/Parameter	Description/Setting (Numbers in Decimal)
Acquisition Configuration T8	
CHRGTIME	40
MEASALLOW	1
MEASIDLEDEF	1
MEASACTVDEF	1
Self Test T25	Object Enabled
Touch Suppression T42	Object Enabled
CTE Configuration T46	
IDLESYNCSPERX	16
ACTVSYNCSPERX	16
Glove Detection T78	Object Enabled
Multiple Touch Touchscreen T100	Object Enabled
XSIZE	19
YSIZE	11
CHRGTIME	40
MEASALLOW	1
MEASIDLEDEF	1
MEASACTVDEF	1
Self Test T25	Object Enabled
Touch Suppression T42	Object Enabled
CTE Configuration T46	
IDLESYNCSPERX	16
ACTVSYNCSPERX	16
Glove Detection T78	Object Enabled
Multiple Touch Touchscreen T100	Object Enabled
XSIZE	19

15.4 Supply Current

15.4.1 Analog Supply

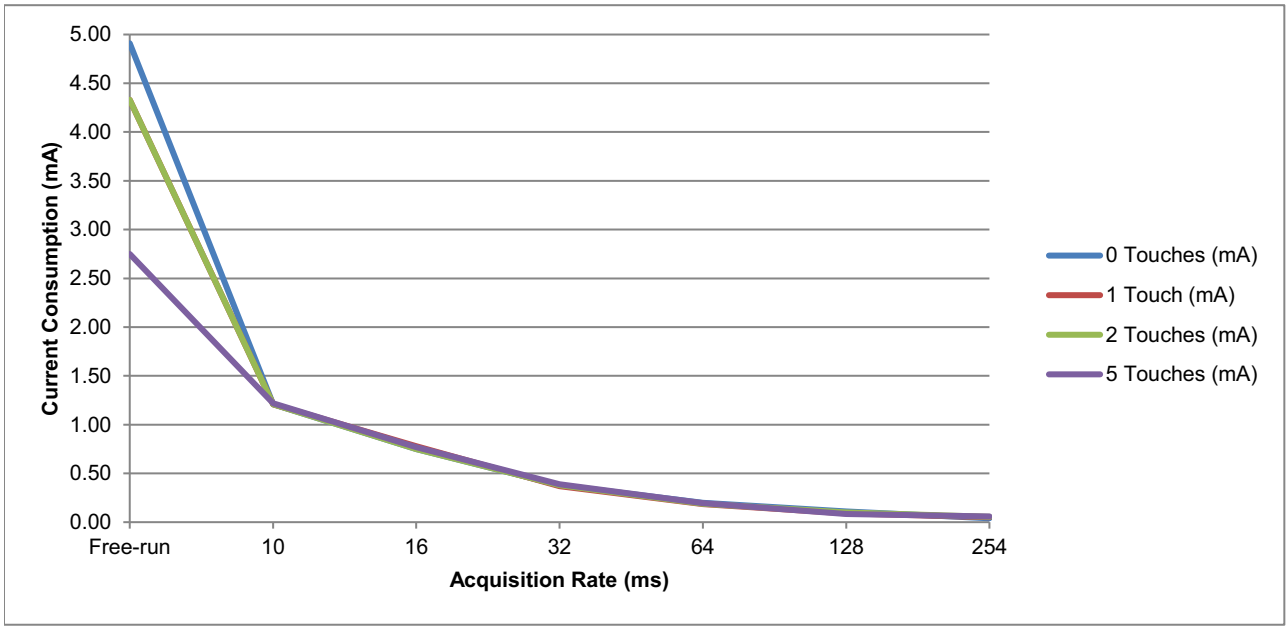
Acquisition Rate (ms)	0 Touches (mA)	1 Touch (mA)	2 Touches (mA)	5 Touches (mA)
Free-run	7.98	9.89	11.44	12.28
10	5.54	6.39	6.31	12.28
16	5.17	5.84	6.10	7.17
32	4.84	5.23	5.47	5.79
64	4.68	4.89	5.10	5.28
128	4.60	4.72	4.81	5.05
254	4.57	4.63	4.69	4.87



15.4.2 Digital Supply

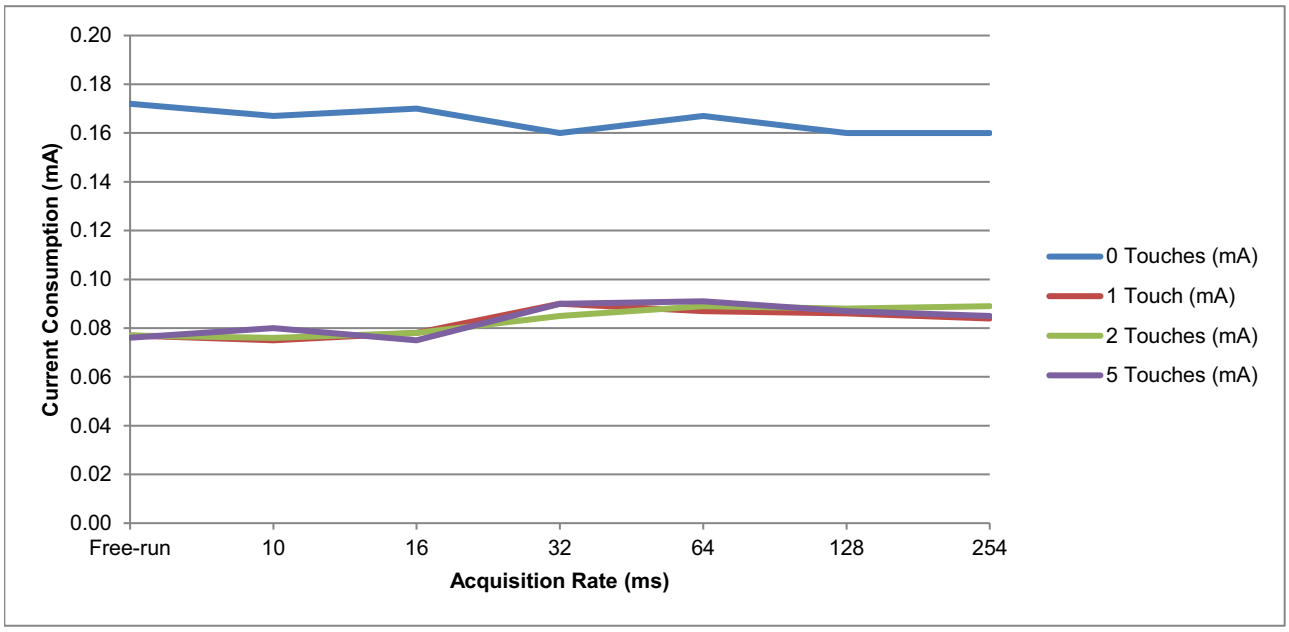
15.4.2.1 Vdd

Acquisition Rate (ms)	0 Touches (mA)	1 Touch (mA)	2 Touches (mA)	5 Touches (mA)
Free-run	4.91	4.33	4.33	2.75
10	1.21	1.21	1.21	1.22
16	0.75	0.78	0.75	0.77
32	0.38	0.37	0.38	0.39
64	0.20	0.19	0.19	0.20
128	0.11	0.09	0.10	0.08
254	0.04	0.05	0.06	0.06



15.4.2.2 VddIO

Acquisition Rate (ms)	0 Touches (mA)	1 Touch (mA)	2 Touches (mA)	5 Touches (mA)
Free-run	0.17	0.08	0.08	0.08
10	0.17	0.08	0.08	0.08
16	0.17	0.08	0.08	0.08
32	0.16	0.09	0.09	0.09
64	0.17	0.09	0.09	0.09
128	0.16	0.09	0.09	0.09
254	0.16	0.08	0.09	0.09



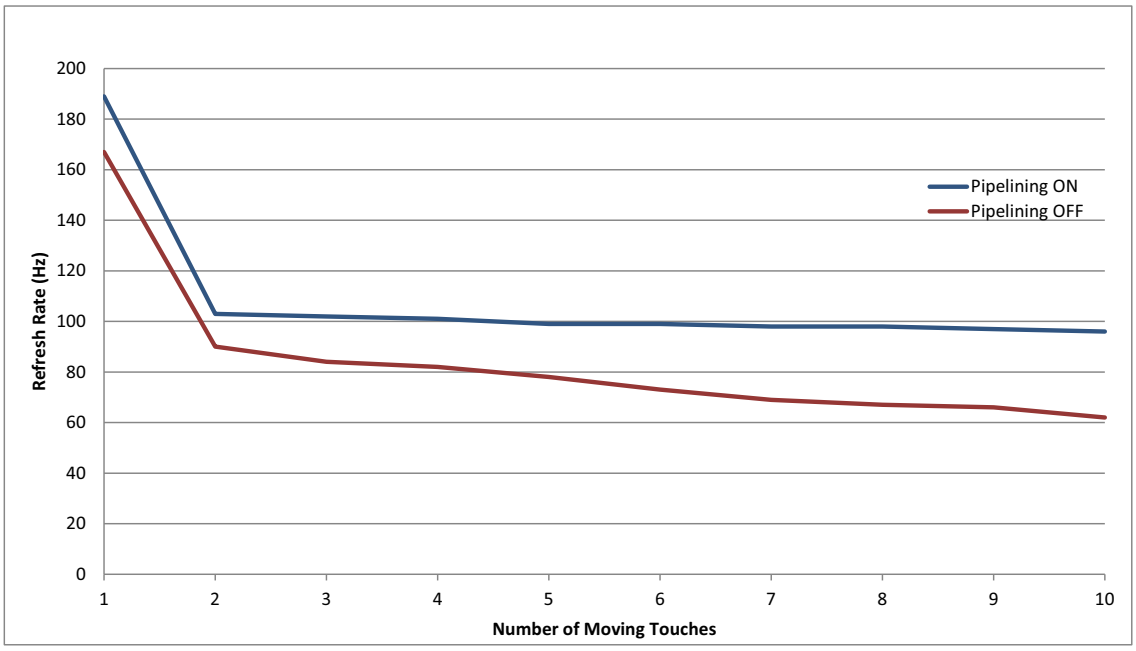
15.5 Timing Specifications

15.5.1 Touch Latency

Conditions: XSIZE = 16; YSIZE = 14 CHRGTIME = 60; IDLE/ACTSYNCPERX = 8/10/12; TCHDI = 0, NEXTTCHDI = 0; AMPLHYST = 0; T56 disabled. T= –40°C, 25°C, 85°C. The values were derived using CPK calculations, CPK = 1.66.

Parameter	Min	Typ	Max	Units	Notes
Touch Latency	18.3	26	56.6	ms	Idle/active acquisition interval = 8 ms
	18.5	25.5	56.9	ms	Idle/active acquisition interval = 10 ms
	14.2	27.1	51.5	ms	Idle/active acquisition interval = 12 ms

15.5.2 Speed



15.5.3 Reset Timing

Parameter	Min	Typ	Max	Units	Notes
Power on to $\overline{\text{CHG}}$ line low	82.5	93.8	118	ms	Vdd supply for POR VddIO supply for external reset

Parameter	Min	Typ	Max	Units	Notes
Hardware reset to $\overline{\text{CHG}}$ line low	82.0	93.3	117	ms	
Software reset to $\overline{\text{CHG}}$ line low	94.8	115	148	ms	

Notes: 1. Any $\overline{\text{CHG}}$ line activity before the power-on or reset period has expired should be ignored by the host. Operation of this signal cannot be guaranteed before the power-on/reset periods have expired.
2. T= –40°C, 25°C, 85°C. The values were derived using CPK calculations, CPK = 1.66

15.6 Input/Output Characteristics

Parameter	Description	Min	Typ	Max	Units	Notes
Input ($\overline{\text{RESET}}$, GPIO, SDA, SCL)						
Vil	Low input logic level	–0.3	–	0.3 × VddIO	V	VddIO = 1.8 V to Vdd
Vih	High input logic level	0.6 × VddIO	–	VddIO	V	VddIO = 1.8 V to Vdd
Iil	Input leakage current	–	–	0.5	µA	Pull-up resistors disabled
$\overline{\text{RESET}}$ pin	Internal pull-up resistor	20	40	60	kΩ	
Output ($\overline{\text{CHG}}$, GPIO, SDA, SCL)						
Vol	Low output voltage	0	–	0.2 × VddIO	V	VDDIO = 1.8 V to VDD. Iol = -2 mA
Voh	High output voltage	0.8 × VddIO	–	VddIO	V	VDDIO = 1.8 V to VDD. Ioh = 2 mA

15.7 I²C Specifications

Parameter	Value
Addresses	0x4A or 0x4B
Maximum bus speed (SCL)	3.4 MHz
I²C specification	Version 6.0
Required pull-up resistance for standard mode (100 kHz)	1 kΩ to 10 kΩ (1)
Required pull-up resistance for fast mode (400 kHz)	1 kΩ to 3 kΩ (1)
Required pull-up resistance for fast+ mode (1 MHz)	0.7 kΩ (max) (1)
Required pull-up resistance for high-speed mode (3.4 MHz)	0.5 kΩ to 0.75 kΩ (1)

Notes: 1. The values of pull-up resistors should be chosen to ensure SCL and SDA rise and fall times meet the I²C specification. The value required will depend on the amount of stray capacitance on the line.
2. In systems with heavily laden I²C lines, bus speed may limited by loading and minimum pull-up resistor values.

More detailed information on I²C operation is available from www.nxp.com/documents/user_manual/UM10204.pdf.

15.8 SPI Bus Specifications

Parameter	Specification
Mode	Mode 3 (CPOL = 1 and CPHA = 1)
Clock idle state	High
Setup on	Leading (falling) edge
Sample on	Trailing (rising) edge
Word size	8-bit
Maximum clock rate	8 MHz

15.9 Touch Accuracy and Repeatability

Parameter	Min	Typ	Max	Units	Notes
Linearity (touch only; 5.4 mm electrode pitch)	–	±1	–	mm	8 mm or greater finger
Linearity (touch only; 4.2 mm electrode pitch)	–	±0.5	–	mm	4 mm or greater finger
Accuracy	–	±1	–	mm	
Accuracy at edge	–	±2	–	mm	
Repeatability	–	±0.25	–	%	X axis with 12-bit resolution

15.10 Thermal Packaging

15.10.1 Thermal Data

Parameter	Typ	Unit	Condition	Package
Junction to ambient thermal resistance	56.4	°C/W	Still air	TQFP 100, 14 x 14 mm
Junction to case thermal resistance	8.5	°C/W	–	TQFP 100, 14 x 14 mm

15.10.2 Junction Temperature

The average chip junction temperature, T_J in °C can be obtained from the following:

$$T_J = T_A + (P_D \times \theta_{JA})$$

If a cooling device is required, use this equation:

$$T_J = T_A + (P_D \times (\theta_{HEATSINK} + \theta_{JC}))$$

where:

- θ_{JA}= package thermal resistance, Junction to ambient (°C/W).
- θ_{JC} = package thermal resistance, Junction to case thermal resistance (°C/W).
- $\theta_{HEATSINK}$ = cooling device thermal resistance (°C/W), provided in the cooling device datasheet.
- P_D = device power consumption (W).
- T_A is the ambient temperature (°C).

15.11 ESD Information

Parameter	Value	Reference standard
Human Body Model (HBM)	±2000 V	JEDEC JS–001
Charge Device Model (CDM)	±500 V	

15.12 Soldering Profile

Profile Feature	Green Package
Average Ramp-up Rate (217°C to Peak)	3°C/s max
Preheat Temperature 175°C ±25°C	150 – 200°C
Time Maintained Above 217°C	60 – 150 s
Time within 5°C of Actual Peak Temperature	30 s
Peak Temperature Range	260°C
Ramp down Rate	6°C/s max
Time 25°C to Peak Temperature	8 minutes max

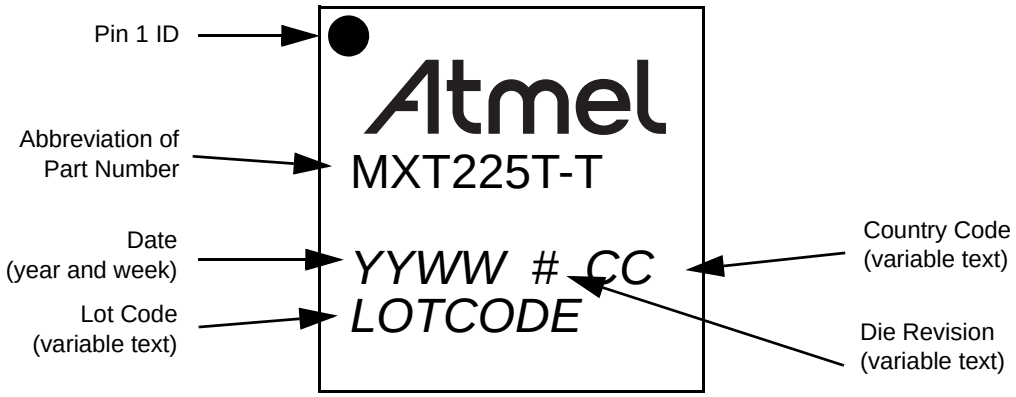
15.13 Moisture Sensitivity Level (MSL)

MSL Rating	Package Type(s)	Peak Body Temperature	Specifications
MSL3	QFP	260°C	IPC/JEDEC J-STD-020

16. Package Information

16.1 Part Marking

16.1.1 ATMXT225T-AT



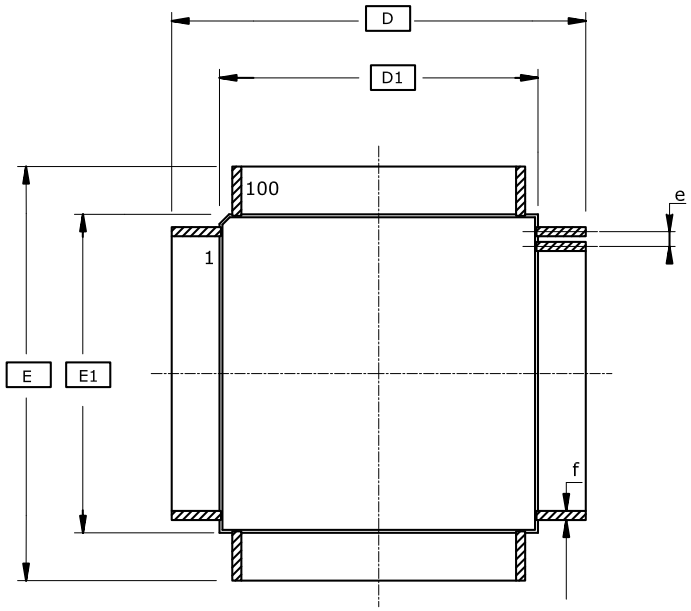
16.2 Orderable Part Numbers

Orderable Part Number	Firmware Version	Description
ATMXT225T-AT (Supplied in trays)	1.0	100-pin 14 × 14 × 1 mm TQFP RoHS compliant
ATMXT225T-ATR (Supplied in tape and reels)	1.0	100-pin 14 × 14 × 1 mm TQFP RoHS compliant

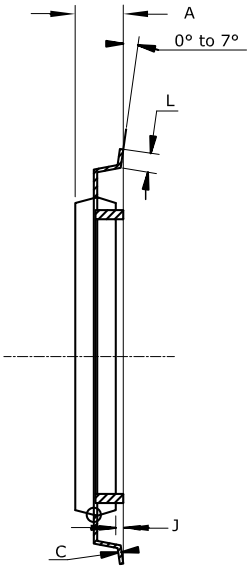
16.3 Mechanical Drawings

16.3.1 TQFP 100 Pins

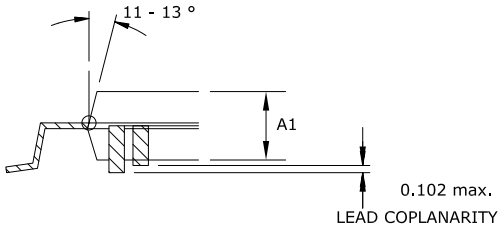
DRAWINGS NOT SCALED



TOP VIEW



SIDE VIEW



DETAIL VIEW

COMMON DIMENSIONS

(Unit of Measure = mm)

SYMBOL	MIN	NOM	MAX	NOTE
A	-----	-----	1.20	
A1	0.95	-----	1.05	
C	0.09	-----	0.20	
D/E		16.00		
D1/E1		14.00		2
J	0.05	-----	0.15	
L	0.45	-----	0.75	
e		0.50 BSC		
f	0.17	-----	0.27	
n		100		

Notes : 1. This drawing is for general information only. Refer to JEDEC Drawing MS-026, Variation AED.
2. Dimensions D1 and E1 do not include mold protrusion. Allowable protrusion is 0.25mm per side. Dimensions D1 and E1 are maximum plastic body size dimensions including mold mismatch.
3. Lead coplanarity is 0.10mm maximum.

05/31/2013

Atmel	TITLE	GPC	DRAWING NO.	REV.
Package Drawing Contact: packagedrawings@atmel.com	MT, 100 Lds - 0.50mm Pitch, 14x14x1.0mm Body size Thin Profile Plastic Quad Flat Package (TQFP)	AAD	MT	F

Associated Documents

The following documents may be useful (available by contacting Atmel Touch Technology Division):

Note: The documents listed below are available under NDA only. In addition, some documents may have further restrictions placed upon them.

For information on using and configuring the device, see the following:

- *maXTouch T Series Tuning Guide* (distributed with Atmel approval only)
- *mXT225T-AT 1.0 Protocol Guide* (distributed with Atmel approval only)

The following documents may also be useful:

- **Touchscreen design and PCB/FPCB layout guidelines:**
  - Application Note: QTAN0054 – *Getting Started with maXTouch Touchscreen Designs*
  - Application Note: MXTAN0208 – *Design Guide for PCB Layouts for Atmel Touch Controllers*
  - Application Note: QTAN0080 – *Touchscreens Sensor Design Guide*
- **Configuring the device:**
  - Application Note: QTAN0059 – *Using the maXTouch Self Test Feature*
- **Miscellaneous:**
  - Application Note: QTAN0050 – *Using the maXTouch Debug Port*
  - Application Note: QTAN0058 – *Rejecting Unintentional Touches with the maXTouch Touchscreen Controllers*
  - Application Note: QTAN0061 – *maXTouch Sensitivity Effects for Mobile Devices*
  - Application Note: QTAN0051 – *Bootloading Procedure for Atmel Touch Sensors Based on the Object Protocol*
  - Application note: QTAN0071 – *Bootloading Procedure for Atmel® Touch Sensors Based on the Object Protocol – SPI Interface*
- **Tools:**
  - *maXTouch Studio User Guide* (distributed as on-line help with maXTouch Studio)

Revision History

Revision Number	History
Revision ASX – March 2015	Initial edition for firmware revision 1.0 – Summary
Revision BX – April 2015	Updated for general distribution – Preliminary • Revised Orderable Part Reference Number
Revision CX – June 2015	Updated for general distribution – Release • Added characterization data
Revision DX – September 2015	• Corrected pinout information for XVDD and SCK

Atmel | Enabling Unlimited Possibilities®

Atmel Corporation
1600 Technology Drive
San Jose, CA 95110
USA
Tel: (+1) (408) 441-0311
Fax: (+1) (408) 487-2600
www.atmel.com

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

Atmel München GmbH
Business Campus
Parkring 4
D-85748 Garching bei München
GERMANY
Tel: (+49) 89-31970-0
Fax: (+49) 89-3194621

Atmel Japan G.K.
16F Shin-Osaki Kangyo Bldg
1-6-4 Osaki, Shinagawa-ku
Tokyo 141-0032
JAPAN
Tel: (+81) (3) 6417-0300
Fax: (+81) (3) 6417-0370