- Panel Thickness
  - Glass up to 2.5 mm, screen size dependent
  - Plastic up to 1.2 mm, screen size dependent
- Interfaces
  - I^2C-compatible slave mode; Standard/Fast Mode: up to 400 kHz, High speed mode: up to 1.7 MHz
  - HID-I^2C interface for Microsoft® Windows® 8
- Power
  - Digital 2.7 V to 3.3 V nominal
  - Analog 2.7 V to 3.3 V nominal
  - High voltage X line drive 2.7 V to 10.0 V nominal
- Packages
  - 144-pin LQFP 20 × 20 × 1.4 mm, 0.5 mm pin pitch

Table of Contents

1. Overview of the mXT1664S-AT

1.1 Introduction

The Atmel maXTouch family of touch controllers has set a new industry benchmark for capacitive touchscreens with their low current consumption, fast response time and high levels of accuracy. The mXT1664S-AT single-chip solution offers the benefits of the maXTouch enhanced architecture on devices with touchscreens up to 12.1 in. diagonal:

- **Patented capacitive sensing method** – The mXT1664S-AT uses a unique charge-transfer acquisition engine to implement the Atmel-patented QMatrix® capacitive sensing method. This allows the measurement of up to 1664 mutual capacitance nodes. Coupled with a state-of-the-art CPU, the entire touchscreen sensing solution can measure, classify and track individual finger touches with a high degree of accuracy.
- **Capacitive Touch Engine (CTE)** – The mXT1664S-AT features an acquisition engine, which uses an optimal measurement approach to ensure almost complete immunity from parasitic capacitance on the receiver inputs (Y lines). The engine includes sufficient dynamic range to cope with anticipated touchscreen mutual capacitances, which allows great flexibility for use with the Atmel proprietary ITO pattern designs. One and two layer ITO sensors are possible using glass or PET substrates.
- **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 charger and LCD noise.
- **Processing power** – The main CPU has two powerful, yet low power, microsequencer coprocessors under its control. These combine to allow the signal acquisition, preprocessing, postprocessing and housekeeping to be partitioned in an efficient and flexible way. This gives ample scope for sensing algorithms, touch tracking or advanced shape-based filtering. An in-circuit reflash can be performed over the chip hardware-driven interface.
- **Interpreting user intention** – The Atmel mutual capacitance method provides unambiguous multitouch performance. Algorithms in the mXT1664S-AT provide optimized touchscreen position filtering for the smooth tracking of touches. Stylus support allows stylus touches to be detected and distinguished from other touches, such as finger touches. 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.

1.2 Understanding Unfamiliar Concepts

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

- Appendix B. on page 58 for a glossary of terms
- Appendix C. on page 60 for QMatrix technology

1.3 Resources

The following datasheet provide essential information on configuring the device:

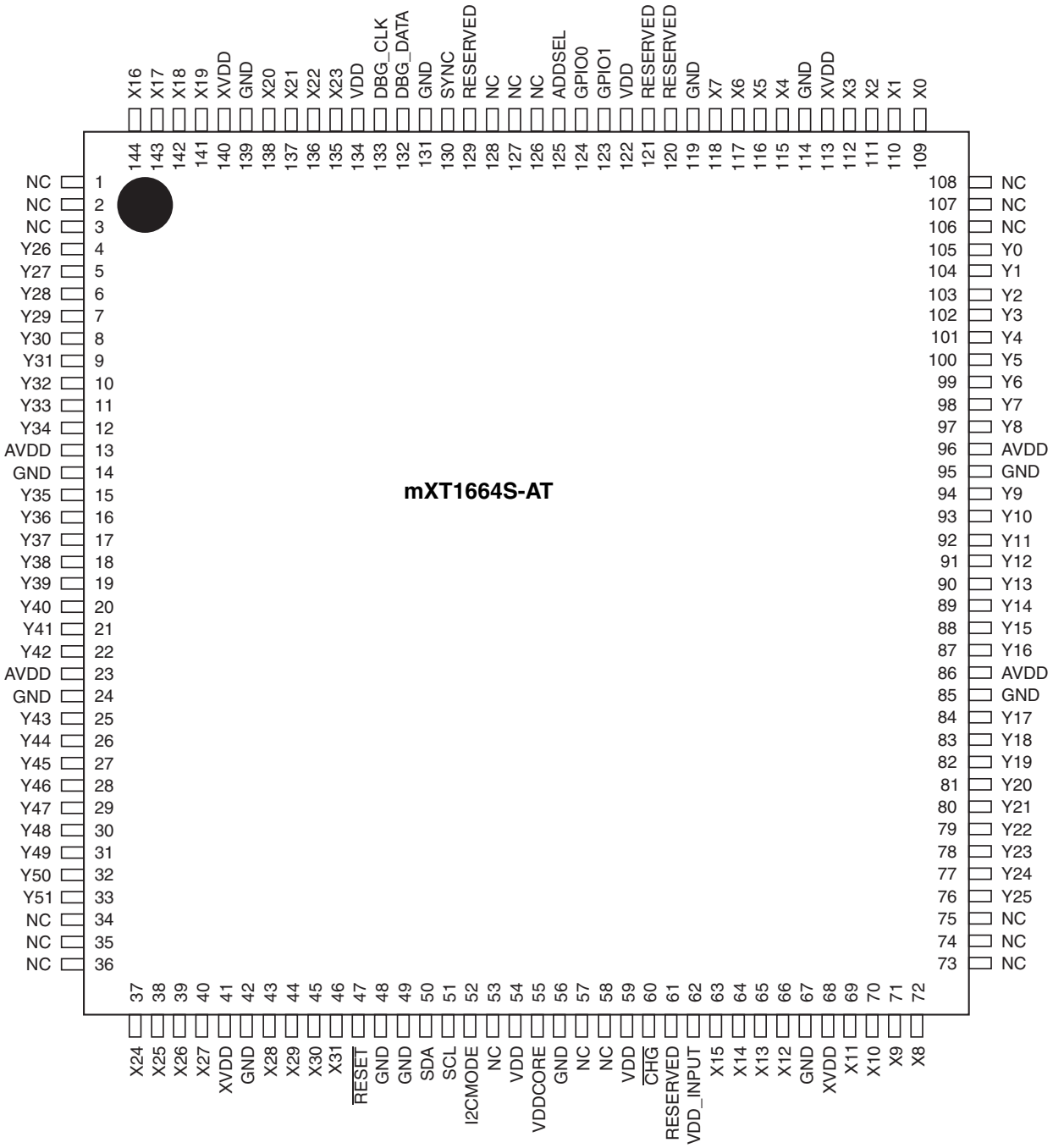
- *mXT1664S-AT 1.0 Protocol Guide*

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

- **Configuring the device:**
  - Application Note: QTAN0058 – *Rejecting Unintentional Touches with the maXTouch Touchscreen Controllers*
  - Application Note: QTAN0078 – *maXTouch Stylus Tuning*
- **Miscellaneous:**
  - Application Note QTAN0050 – *Using the maXTouch Debug Port*
  - Application Note QTAN0061 – *maXTouch Sensitivity Effects for Mobile Devices*
  - Application Note QTAN0086 – *Touchscreen Design for Gloved Operation*
- **Touchscreen design and PCB/FPCB layout guidelines:**
  - Application Note QTAN0054 – *Getting Started with maXTouch Touchscreen Designs*
  - Application Note QTAN0094 – *mXT1664S PCB/FPCB Layout Guidelines*
  - Application Note QTAN0080 – *Touchscreens Sensor Design Guide*
- **Other documents –** The device uses the same core technology as the mXT768E, so the following documents may also be useful (available by contacting the Atmel Touch Technology division):
  - Application Note QTAN0083 – *mXT768E Power and Speed Considerations*
  - Application Note QTAN0052 – *mXT224 Passive Stylus Support*

2. Pinout and Schematic

2.1 Pinout Configuration



2.2 Pinout Descriptions

Table 2-1. Pin Listing

Pin	Name	Type	Comments	If Unused, Connect To...
1	NC	–	No connection	–
2	NC	–	No Connection	–
3	NC	–	No Connection	–
4	Y26	I	Y line connections	Leave open
5	Y27	I	Y line connections	Leave open
6	Y28	I	Y line connections	Leave open
7	Y29	I	Y line connections	Leave open
8	Y30	I	Y line connections	Leave open
9	Y31	I	Y line connections	Leave open
10	Y32	I	Y line connections	Leave open
11	Y33	I	Y line connections	Leave open
12	Y34	I	Y line connections	Leave open
13	AVDD	P	Analog power	–
14	GND	P	Ground	–
15	Y35	I	Y line connections	Leave open
16	Y36	I	Y line connections	Leave open
17	Y37	I	Y line connections	Leave open
18	Y38	I	Y line connections	Leave open
19	Y39	I	Y line connections	Leave open
20	Y40	I	Y line connections	Leave open
21	Y41	I	Y line connections	Leave open
22	Y42	I	Y line connections	Leave open
23	AVDD	P	Analog power	–
24	GND	P	Ground	–
25	Y43	I	Y line connections	Leave open
26	Y44	I	Y line connections	Leave open
27	Y45	I	Y line connections	Leave open
28	Y46	I	Y line connections	Leave open
29	Y47	I	Y line connections	Leave open
30	Y48	I	Y line connections	Leave open
31	Y49	I	Y line connections	Leave open

Table 2-1. Pin Listing (Continued)

Pin	Name	Type	Comments	If Unused, Connect To...
32	Y50	I	Y line connections	Leave open
33	Y51	I	Y line connections	Leave open
34	NC	–	No connection	–
35	NC	–	No connection	–
36	NC	–	No connection	–
37	X24	O	X matrix drive line	Leave open
38	X25	O	X matrix drive line	Leave open
39	X26	O	X matrix drive line	Leave open
40	X27	O	X matrix drive line	Leave open
41	XVDD	P	X line drive voltage	–
42	GND	P	Ground	–
43	X28	O	X matrix drive line	Leave open
44	X29	O	X matrix drive line	Leave open
45	X30	O	X matrix drive line	Leave open
46	X31	O	X matrix drive line	Leave open
47	$\overline{\text{RESET}}$ (1)	I	Reset low, has internal pull-up resistor (20 kΩ to 60 kΩ)	VDD
48	GND	P	Ground	–
49	GND	P	Ground	–
50	SDA	OD	Serial interface data	–
51	SCL	OD	Serial interface clock	–
52	I2CMODE (2)	I	I^2C-compatible protocol select	Leave open
53	NC	–	No connection	–
54	VDD	P	Digital power	–
55	VDDCORE	P	Digital core power	–
56	GND	P	Ground	–
57	NC	–	No connection	–
58	NC	–	No connection	–
59	VDD	P		–
60	$\overline{\text{CHG}}$ (3)	OD	State change interrupt	Leave open
61	Reserved	–	Must be connected to GND	–
62	VDD_INPUT	P	For factory use only – connect to VDD	–
63	X15	O	X matrix drive line	Leave open

Table 2-1. Pin Listing (Continued)

Pin	Name	Type	Comments	If Unused, Connect To...
64	X14	O	X matrix drive line	Leave open
65	X13	O	X matrix drive line	Leave open
66	X12	O	X matrix drive line	Leave open
67	GND	P	Ground	–
68	XVDD	P	X line drive voltage	–
69	X11	O	X matrix drive line	Leave open
70	X10	O	X matrix drive line	Leave open
71	X9	O	X matrix drive line	Leave open
72	X8	O	X matrix drive line	Leave open
73	NC	–	No connection	–
74	NC	–	No connection	–
75	NC	–	No connection	–
76	Y25	I	Y line connections	Leave open
77	Y24	I	Y line connections	Leave open
78	Y23	I	Y line connections	Leave open
79	Y22	I	Y line connections	Leave open
80	Y21	I	Y line connections	Leave open
81	Y20	I	Y line connections	Leave open
82	Y19	I	Y line connections	Leave open
83	Y18	I	Y line connections	Leave open
84	Y17	I	Y line connections	Leave open
85	GND	P	Ground	–
86	AVDD	P	Analog power	–
87	Y16	I	Y line connections	Leave open
88	Y15	I	Y line connections	Leave open
89	Y14	I	Y line connections	Leave open
90	Y13	I	Y line connections	Leave open
91	Y12	I	Y line connections	Leave open
92	Y11	I	Y line connections	Leave open
93	Y10	I	Y line connections	Leave open
94	Y9	I	Y line connections	Leave open
95	GND	P	Ground	–
96	AVDD	P	Analog power	–

Table 2-1. Pin Listing (Continued)

Pin	Name	Type	Comments	If Unused, Connect To...
97	Y8	I	Y line connections	Leave open
98	Y7	I	Y line connections	Leave open
99	Y6	I	Y line connections	Leave open
100	Y5	I	Y line connections	Leave open
101	Y4	I	Y line connections	Leave open
102	Y3	I	Y line connections	Leave open
103	Y2	I	Y line connections	Leave open
104	Y1	I	Y line connections	Leave open
105	Y0	I	Y line connections	Leave open
106	NC	–	No connection	–
107	NC	–	No connection	–
108	NC	–	No connection	–
109	X0	O	X matrix drive line	Leave open
110	X1	O	X matrix drive line	Leave open
111	X2	O	X matrix drive line	Leave open
112	X3	O	X matrix drive line	Leave open
113	XVDD	P	X line drive voltage	–
114	GND	P	Ground	–
115	X4	O	X matrix drive line	Leave open
116	X5	O	X matrix drive line	Leave open
117	X6	O	X matrix drive line	Leave open
118	X7	O	X matrix drive line	Leave open
119	GND	P	Ground	–
120	Reserved	–	Connect to GND via 10 kΩ	–
121	Reserved	–	Reserved for future use	–
122	VDD	P	Digital power	–
123	GPIO1	I/O	General purpose I/O	Leave open
124	GPIO0	I/O	General purpose I/O	Leave open
125	ADDSEL	I	I^2C-compatible address select	Leave open
126	NC	–	No connection	–
127	NC	–	No connection	–
128	NC	–	No connection	–

Table 2-1. Pin Listing (Continued)

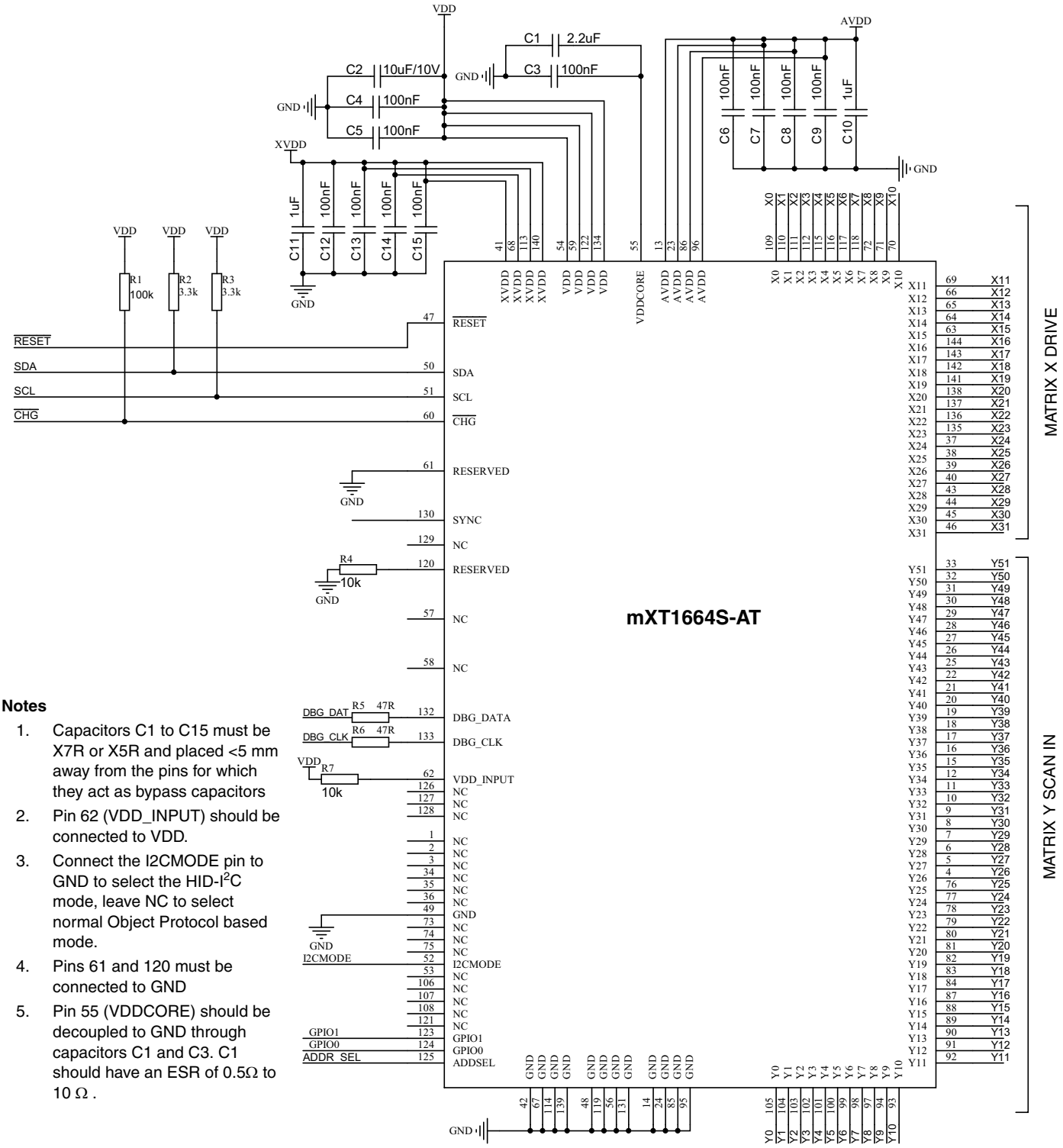
Pin	Name	Type	Comments	If Unused, Connect To...
129	Reserved	–	Reserved for future use	–
130	SYNC	I	External synchronization	Leave open
131	GND	P	Ground	–
132	DBG_DATA	O	Debug data	Leave open
133	DBG_CLK	I	Debug clock	Leave open
134	VDD	P	Digital power	–
135	X23	O	X matrix drive line	Leave open
136	X22	O	X matrix drive line	Leave open
137	X21	O	X matrix drive line	Leave open
138	X20	O	X matrix drive line	Leave open
139	GND	P	Ground	–
140	XVDD	P	X line drive voltage	–
141	X19	O	X matrix drive line	Leave open
142	X18	O	X matrix drive line	Leave open
143	X17	O	X matrix drive line	Leave open
144	X16	O	X matrix drive line	Leave open

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

1. It is recommend that $\overline{\text{RESET}}$ is connected to the host system.
2. Leave open for standard Atmel object protocol, or connect to GND to select the HID-I^2C mode.
3. $\overline{\text{CHG}}$ is momentarily set (approximately 100 ms) as an input after power-up or reset for diagnostic purposes.

2.3 Schematics

2.3.1 I²C-compatible Mode – Digital Supply 2.7 V – 3.3 V



Notes

1. Capacitors C1 to C15 must be X7R or X5R and placed <5 mm away from the pins for which they act as bypass capacitors
2. Pin 62 (VDD_INPUT) should be connected to VDD.
3. Connect the I2CMODE pin to GND to select the HID-I²C mode, leave NC to select normal Object Protocol based mode.
4. Pins 61 and 120 must be connected to GND
5. Pin 55 (VDDCORE) should be decoupled to GND through capacitors C1 and C3. C1 should have an ESR of 0.5Ω to 10 Ω .

3. Touchscreen Basics

3.1 Sensor Construction

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

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

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

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

3.2 Electrode Configuration

The specific electrode designs used in Atmel 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 below:

Touchscreens:	2 Touchscreens allowed
	3 X × 3 Y minimum (depends on screen resolution)
	32 X × 52 Y maximum (subject to other configurations)
Keys:	1 Key Array allowed
	Up to 32 keys (subject to other configurations)

3.3 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.

3.4 Touchscreen Sensitivity

3.4.1 Adjustment

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

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

3.4.2 Mechanical Stackup

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

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

Note: Care should be taken using ultra-thin glass panels as retransmission effects can occur.

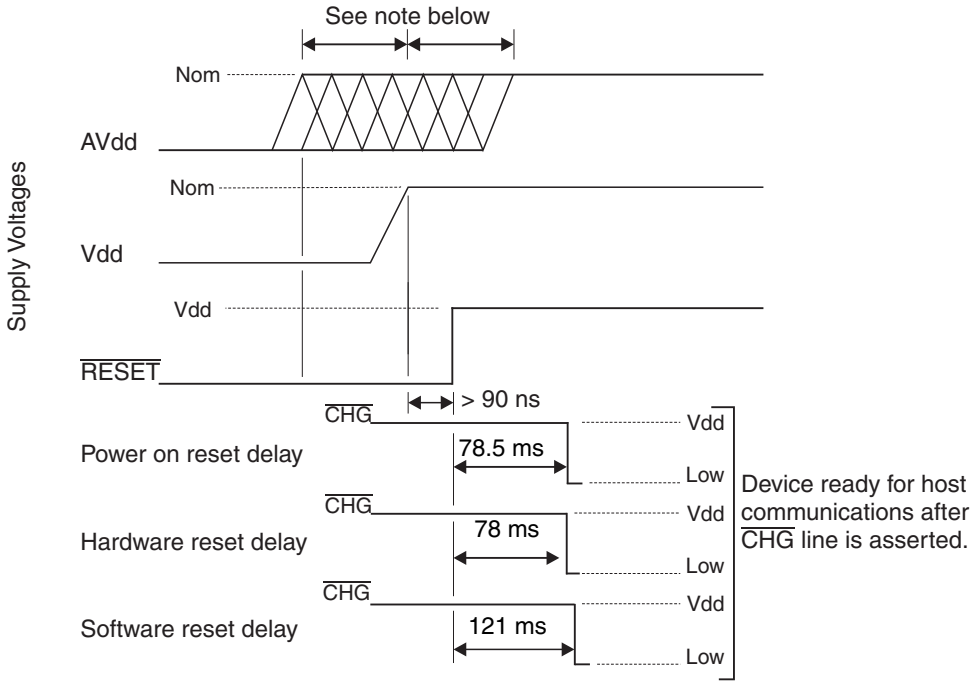
4. Detailed Operation

4.1 Power-up/Reset

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

The device must be held in RESET (active low) while the power supplies (Vdd and AVdd) are powering up. If a slope or slew is applied to the digital or analog supplies (Vdd, AVdd and XVdd) must reach their nominal values before the RESET signal is de-asserted (that is, goes high). This is shown in Figure 4-1. See Section 8.2 on page 44 for nominal values for Vdd, AVdd and XVdd. **Please note that the XVdd rail has a maximum rate of rise specification** (see Section 8.3.3 on page 45), **that is, a soft-start XVdd supply must be used**.

Figure 4-1. Power Sequencing on the mXT1664S-AT



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

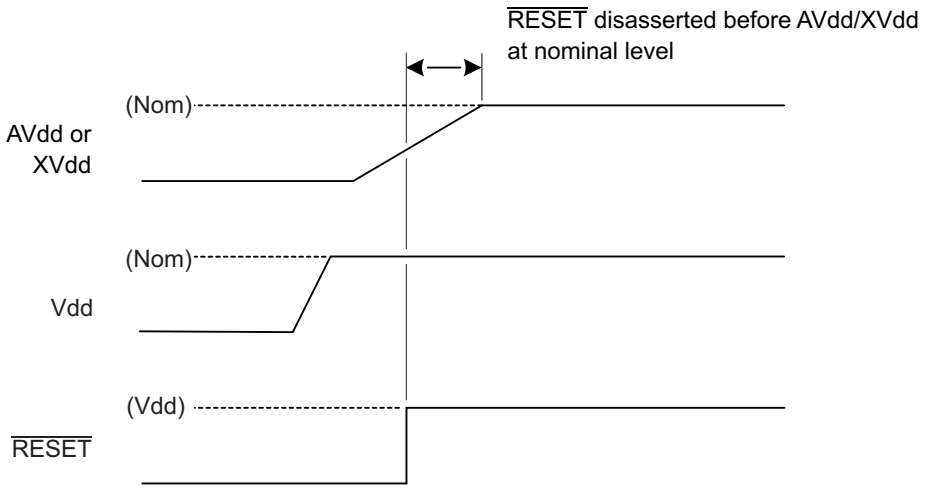
The digital or analog (AVdd) supplies can be applied independently and in any order on the mXT1664S-AT during power-up. Vdd must be applied to the device before XVdd to ensure that the different power domains in the device are initialized correctly. Typically this can be done by connecting the enable pin of the Switched Mode Power Supply (SMPS) supplying XVdd to a 10 kΩ pull-up resistor connected to the Vdd, but the XVdd can be controlled separately by the host, if required.

After power-up, the device takes 78.5 ms before it is ready to start communications. Vdd must drop to below 1.45 V in order to effect a proper POR. See Section 8. on page 44 for further specifications.

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

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

Figure 4-2. Power Sequencing on the mXT1664S-AT – Late rise on AVDD



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 ~78 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.

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

A software reset command can be used to reset the chip (refer to the Command Processor object in the *mXT1664S-AT 1.0 Protocol Guide*). A software reset takes a maximum of 121 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 object (refer to the *mXT1664S-AT 1.0 Protocol Guide* for more information).

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

At power-on, the device performs a self-test routine to check for shorts which might cause damage to the device. Refer to the Self Test T25 section of the *mXT1664S-AT 1.0 Protocol Guide* for more details about this process.

4.2 Watchdog Timer

A Watchdog timer is included and enabled within the mXT1664S-AT firmware, the period of this watchdog timer is fixed at 500 ms. This value is selected since it is twice the maximum system cycle time of 254 ms.

4.3 Calibration

Calibration is the process by which a sensor chip assesses the background capacitance on each node. Nodes are only calibrated on power-up 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 *mXT1664S-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 no other touches are present on the node matrix (refer to the *mXT1664S-AT 1.0 Protocol Guide* for more information on the TCHTHR field in the Multiple Touch Touchscreen and Key Array objects).
- 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.

4.4 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 object. In addition, an Active to Idle timeout (ACTV2IDLETO) setting is provided.

Refer to the *mXT1664S-AT 1.0 Protocol Guide* for full information on how these modes operate, and how to use the settings provided.

4.5 Touchscreen Layout

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 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 *mXT1664S-AT 1.0 Protocol Guide* for more information on configuring the touch objects.

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

- Each touch object should be a regular rectangular shape in terms of the lines it uses.
- Touch objects can share X and Y lines, as necessary. Note, however, that the first instance (instance 0) of the Multiple Touch Touchscreen T9 object cannot share Y lines if the Shieldless T56 object is enabled.
- The design of the touch objects does not physically need to be on a strict XY grid pattern.

4.6 Signal Processing

4.6.1 Adjacent Key Suppression Technology

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

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

The device has two levels of AKS. The first level works between the touch objects (Multiple Touch Touchscreen T9 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). 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 T9 or Key Array T15). Refer to the *mXT1664S-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.

4.6.2 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 T9, Key Array T15). Refer to the *mXT1664S-AT 1.0 Protocol Guide* for more information.

4.6.3 Digital Filtering and Noise Suppression

The mXT1664S-AT supports the on-chip filtering of the acquisition data received from the sensor. Specifically, the maXCharger T62 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. The algorithm can make use of a Grass Cutter (which rejects any samples outside a predetermined limit).

Noise suppression is triggered when a noise source is detected (typically when a charger is turned on). A hardware trigger can be implemented using the CHRG_IN pin. Alternatively, the host driver code can indicate when a noise source is present.

An alternative burst mode on the X lines, known as Dual X Drive, is provided. This improves the signal-to-noise ratio (SNR) on a closely spaced X sensor matrix (when finger touches are likely to cover more than one X line).

Refer to the *mXT1664S-AT 1.0 Protocol Guide* for more information on the maXCharger T62 object.

4.6.4 GPIO Pins

The mXT1664S-AT has two GPIO pins. The pins can be set to be either an input or an output, as required. The GPIO pins are configured using the GPIO/PWM Configuration T19 object.

4.6.5 Grip Suppression

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

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

Grip suppression is configured using the Grip Suppression T40 object. There is one instance of the Grip Suppression T40 object for each Multiple Touch Touchscreen T9 object present on the device. Refer to the *mXT1664S-AT 1.0 Protocol Guide* for more information.

4.6.6 Baseline Reference

The device supports using a set of known-good factory references to eliminate problems associated with calibrating in the presence of touches, moisture or foreign objects.

The maXStartup T66 object enables baseline references to be generated and stored in a controlled environment on the production line. These values can then be restored on calibration instead of relying on current state of the screen which may be in contact with a touch, moisture or foreign objects such as keys or coins. Supporting algorithms are used to compensate the references for variations in different environments. Refer to the *mXT1664S-AT 1.0 Protocol Guide* for more information.

4.6.7 Lens Bending

The device supports algorithms to eliminate disturbances from the measured signal and also to measure the bend component.

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 *mXT1664S-AT 1.0 Protocol Guide* for more information.

4.6.8 Shieldless Support

The device can support shieldless sensor design even with a noisy LCD. The Shieldless T56 object provides a number of algorithms to suppress the effect of noise emitted by the display.

The Shieldless T56 display noise suppression operates on a completely different mechanism to the maXCharger T62 object. This allows the device to overcome display noise simultaneously with charger noise.

The device can make use of the following mechanisms to overcome display noise:

- Optimal Integration is not filtering as such, instead it is a feature that 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 such as charger.

Refer to the *mXT1664S-AT 1.0 Protocol Guide* for more information on the Shieldless T56 object.

4.6.9 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 T9 object present on the device. Refer to the *mXT1664S-AT 1.0 Protocol Guide* for more information.

4.6.10 Zone Indication

The mXT1664S-AT allows the host to define a Zone on the touchscreen and touches detected in the defined zone. The zone can be configured using the Zone Indication T73 object. Note that only finger touches can be detected in the defined zone, refer to the *mXT1664S-AT 1.0 Protocol Guide* for more information on the Zone Indication T73 object.

4.7 Circuit Components

4.7.1 XVdd Power Supply

The X line driver power supply XVdd can be used in two different modes:

- XVdd connected to AVdd. This mode limits the range of XVdd to 2.7 V to 3.3 V.
- XVdd connected to an external supply. In this configuration the external supply should be in the range 2.7 V to 10.0 V. The higher voltages improve the SNR of the system.
- If XVdd < 4.75 V, please note restriction on minimum Cx in Section 8.2 on page 44.

4.7.2 Bypass Capacitors

Each power supply (Vdd, XVdd and AVdd) requires a 1 µF bypass capacitor. If the internal 1.8 V VDDCORE regulator is used, the Vdd 1 µF should be replaced with a 10 µF capacitor and a 2.2 µF capacitor should be added on the VDDCORE pin. In addition, there should be a 100 nF bypass capacitor on each power trace. The capacitors should be ceramic X7R or X5R. See the schematics in Section 2.3 on page 14 for more details.

The PCB traces connecting the bypass capacitors to the pins of the device must not exceed 5 mm in length. This limits any stray inductance that would reduce filtering effectiveness. See also Section 8.11 on page 50.

4.7.3 Supply Quality

While the device has good Power Supply Rejection Ratio properties, poorly regulated and/or noisy power can significantly reduce performance. See Section 8.11 on page 50.

Always operate the device with a well-regulated and clean AVdd (and XVdd, if used) supply. It supplies the sensitive analog stages in the device. See Figure A-1 on page 56 for an example XVdd supply.

4.7.4 Supply Sequencing

Vdd and AVdd can be powered independently of each other without damage to the device. Vdd must be applied to the device before XVdd to ensure proper initialization of the device. All voltage ranges should be used with in the limits specified in Section 8.2 on page 44.

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

4.7.5 Decoupling Requirements

Certain pins have specific decoupling requirements:

- Pin L7 (VDD_INPUT) should be connected to VDD.
- Pin N8 (VDDCORE) is a decoupling connection for an internal LDO (Low Drop-Out regulator) circuit.

See also the schematics in Section 2.3 on page 14.

4.7.6 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.

4.8 PCB Layout

See Appendix A. on page 54 for general advice on PCB layout.

4.9 Debugging

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 *mXT1664S-AT 1.0 Protocol Guide* for more information on this object.

A second mechanism is provided that allows the host to read the real-time raw data using the low-level debug port. This can be accessed via the SPI interface. Refer to QTAN0050, *Using the maXTouch Debug Port*, for more information on the debug port.

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 *mXT1664S-AT 1.0 Protocol Guide* and QTAN0059, *Using the maXTouch Self Test Feature*, for more information on the Self Test T25 object.

4.10 Communications

Communication with the host is achieved using either the I^2C-compatible interface (see Section 5. on page 24), the HID-I^2C-compatible interface (see Section 6. on page 31). Any interface can be used, depending on the needs of the user's project, but only one interface should be used in any one design. The selection of the I^2C or the HID-I^2C interface is determined by the I2CMODE pin.

Note that you only need to connect those pins that are actually required for use with the chosen communications interface. This ensures optimal power consumption and correct functioning. See Section 2.2 on page 9 for details on what should be done with the unconnected pins.

4.11 Configuring the Device

The device has an object-based protocol that organizes the features of the device into objects that can be controlled individually. This is configured using the Object Protocol common to many of Atmel touch sensor devices. For more information on the Object Protocol and its implementation on the device, refer to the *mXT1664S-AT 1.0 Protocol Guide*.

5. I²C-compatible Communications

5.1 Communications Protocol

The device can use an I²C-compatible interface for communication. See Appendix C.4 on page 61 for details of the I²C-compatible protocol.

The I²C-compatible 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.

5.2 I²C-compatible Addresses

5.2.1 I²C-compatible Addresses

The device supports two I²C-compatible device addresses that are selected using the ADDSEL line at start-up. The two internal I²C-compatible device addresses are 0x4A (ADDSEL low) and 0x4B (ADDSEL floating or high). These are shifted left to form the SLA+W or SLA+R address when transmitted over the I²C-compatible interface, as shown in Figure 5-1.

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

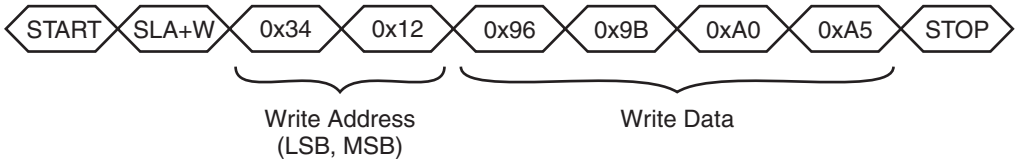
5.3 Writing To the Device

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

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

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

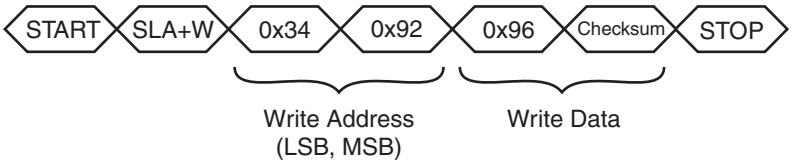


5.4 I²C-compatible Writes in Checksum Mode

In I²C-compatible checksum mode an 8-bit CRC is added to all I²C-compatible writes. The CRC is sent at the end of the data write as the last byte before the STOP condition. All the bytes sent are included in the CRC, including the two address bytes. Any command or data sent to the 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-compatible command shown in Figure 5-2 writes a value of 150 (0x96) to address 0x1234 with a checksum. The address is changed to 0x9234 to indicate checksum mode.

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



5.5 Reading From the Device

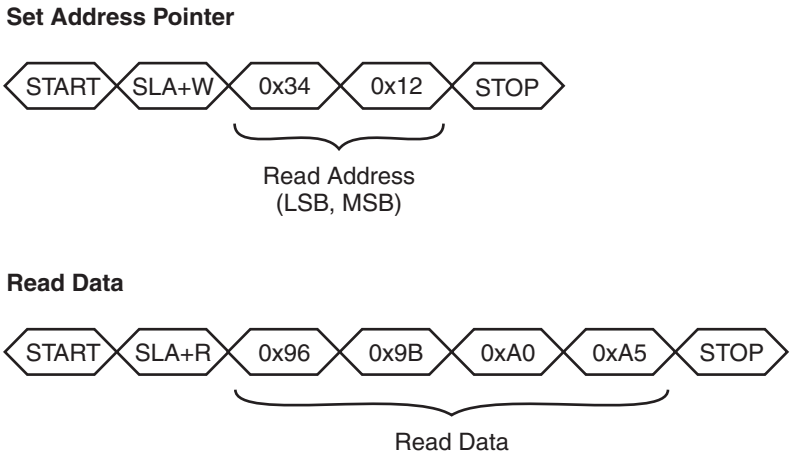
Two I²C-compatible bus activities must take place to read from the device. The first activity is an I²C-compatible write to set the address pointer (LSByte then MSByte). The second activity is the actual I²C-compatible read to receive the data. The address pointer returns to its starting value when the 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 5.6).

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

Figure 5-3 shows the I²C-compatible commands to read four bytes starting at address 0x1234.

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



5.6 Reading Status Messages with DMA

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

1. The host uses a write operation to set the address pointer to the start of the Message Count 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 *mXT1664S-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)

   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 x (size – 1).
5. The host reads the calculated number of message bytes. It is important that the host does *not* send a STOP condition during the message reads, as this will terminate the continuous read operation and reset the address pointer. No START and STOP conditions must be sent between the messages.
6. The host sends a STOP condition at the end of the read operation after the last message has been read. The NACK condition immediately before the STOP condition resets the address pointer to the start of Message Count T44 object.

Figure 5-4 on page 27 shows an example of using a continuous read operation to read three messages from the device without a checksum. Figure 5-5 on page 28 shows the same example with a checksum.

1. The STOP condition at the end of the read resets the address pointer to its 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.

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

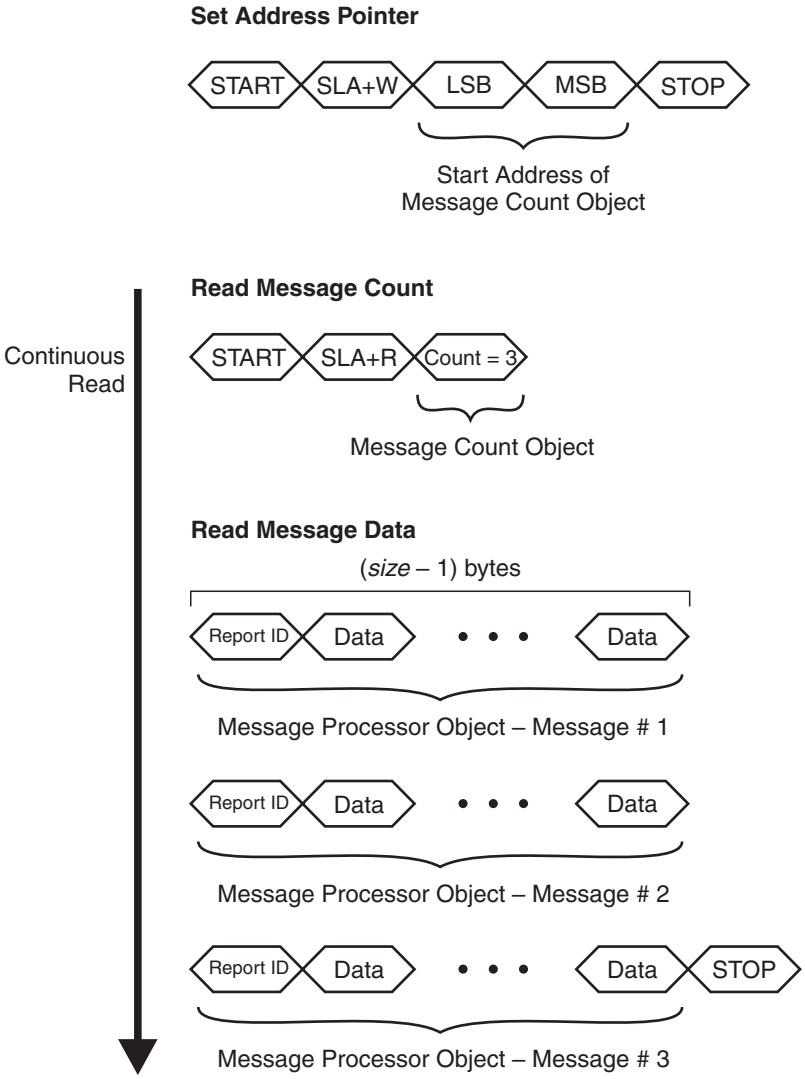
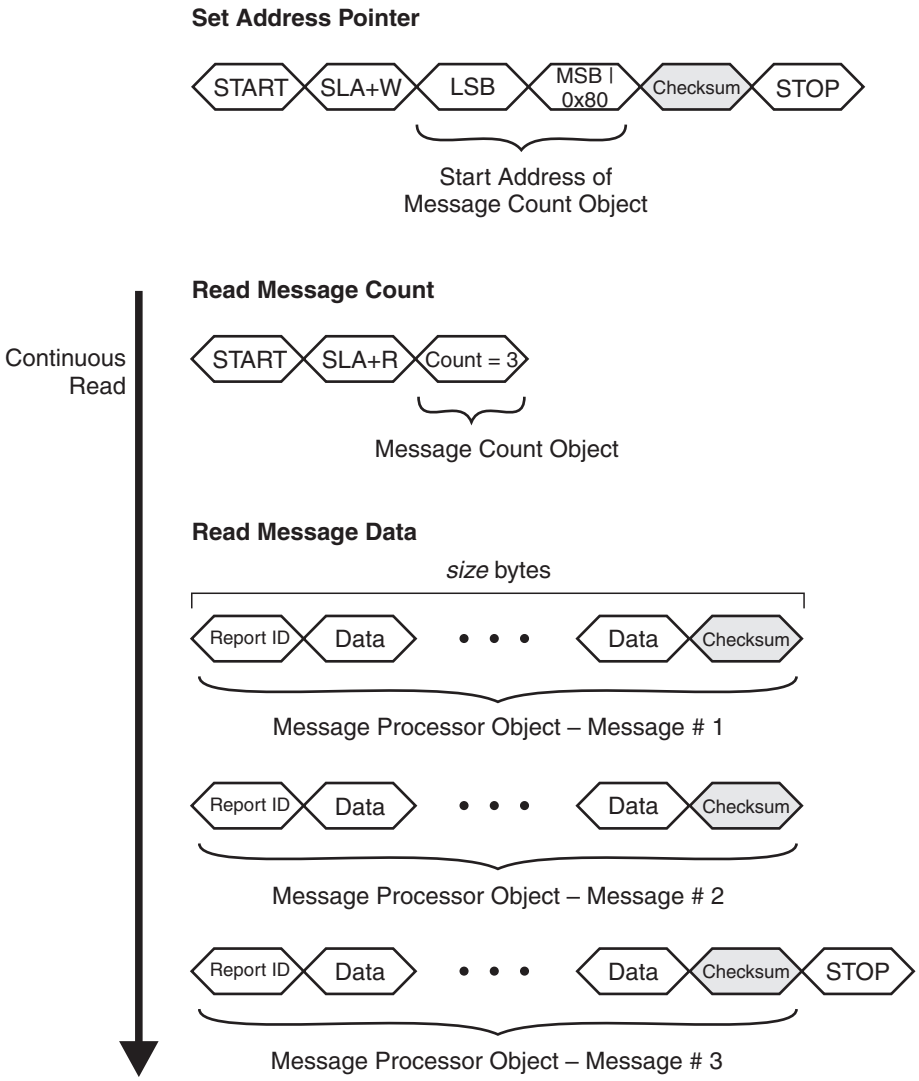


Figure 5-5. Continuous Message Read Example – I²C-compatible Checksum Mode



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

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

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

The $\overline{\text{CHG}}$ line is an active-low, open-drain output that is used to alert the host that a new message is available in the Message Processor 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-compatible communications.

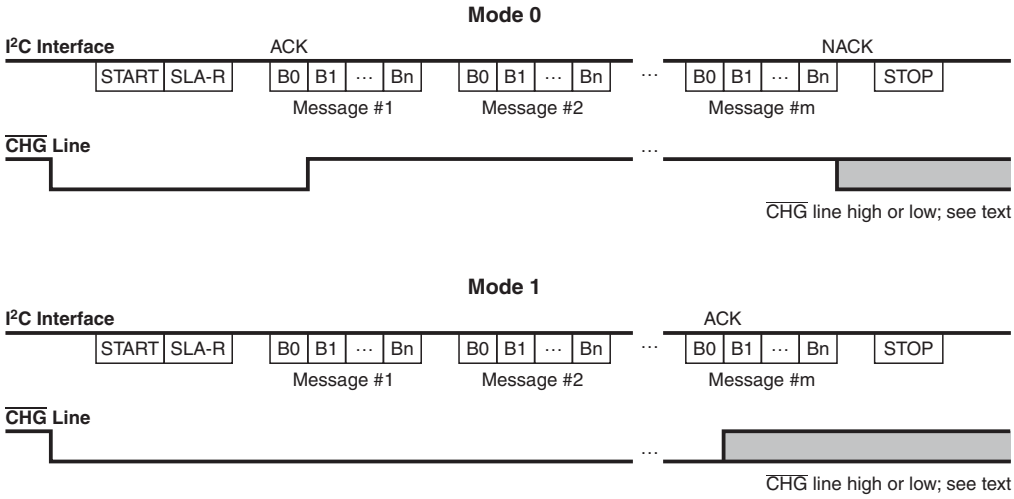
The $\overline{\text{CHG}}$ line remains low as long as there are messages to be read. The host should be configured so that the $\overline{\text{CHG}}$ line is connected to an interrupt line that is level-triggered. The host should not use an edge-triggered interrupt as this means adding extra software 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 4.1 on page 17).

A pull-up resistor is required, typically 10 kΩ to Vdd.

The $\overline{\text{CHG}}$ line operates in two modes, as defined by the Communications Configuration T18 object (refer to the *mXT1664S-AT 1.0 Protocol Guide*).

Figure 5-6. $\overline{\text{CHG}}$ 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-compatible transfer causes the $\overline{\text{CHG}}$ line to stay high if there are no more messages. Otherwise the $\overline{\text{CHG}}$ line goes low to indicate a further message.

Mode 0 allows the host to continually read messages. Messaging reading ends when a report ID of 255 (“invalid message”) is received. Alternatively the host ends the transfer by sending a NACK after receiving the last byte of a message, followed by a STOP condition. If and when there is another 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-compatible read.

In Mode 1:

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

Mode 1 allows the host to continually read the messages until the $\overline{\text{CHG}}$ line goes high, and the state of the $\overline{\text{CHG}}$ line determines whether or not the host should continue receiving messages from the 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 *mXT1664S-AT 1.0 Protocol Guide* for more information.

5.8 SDA, SCL

The I²C-compatible 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-compatible device is pulling it down.

The termination resistors should be chosen so that the rise times on SDA and SCL meet the I²C-compatible specifications for the interface speed being used, bearing in mind other loads on the bus, (see Section 8.8 on page 50).

5.9 Clock Stretching

The device supports clock stretching in accordance with the I²C specification. It may also instigate a clock stretch if a communications event happens during a period when the 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 I²C-compatible 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 *mXT1664S-AT 1.0 Protocol Guide* for more information.

6. HID-I²C-compatible Communications

6.1 Communications Protocol

The device is an HID-I²C DEVICE presenting two top-level collections (TLCs):

Interface 0 (Digitizer HID-I²C) – supplies touch information to the host. This interface is supported by Microsoft Windows 8 without the need for additional software.

Interface 1 (Generic HID-I²C) – This interface provides a generic HID-I²C interface that allows the host to communicate with the chipset using the object protocol.

To use the device in HID-I²C mode, the I2CMODE pin should be pulled low. Other features are identical to standard I²C communication described in Section 5.2 on page 24.

6.2 I²C-compatible Addresses

See Section 5.2 on page 24.

6.3 Device

The device is compliant with HID-I²C specification V0.9. It has the following specification:

Vendor ID:	0x03EB (Atmel)
Product ID:	0x212C (mXT1664S-AT)
Version:	16-bit Version & Build Identifier in the form 0xVVBB, where: VV = Version Major (Upper 4 bits) / Minor (Lower 4 bits) BB = Build number in BCD format
HID descriptor address:	0x0000.

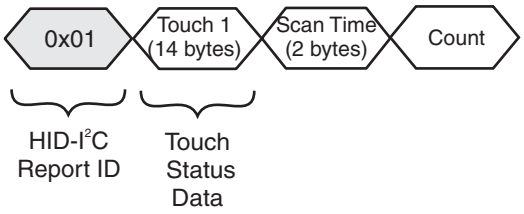
6.4 Interface 0 (Digitizer HID-I²C)

Interface 0 is a digitizer class HID.

6.4.1 Normal Touch Report

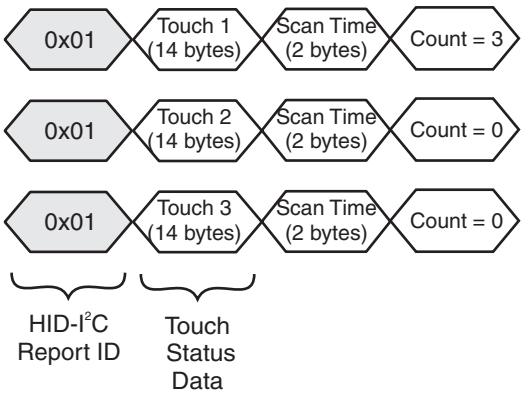
The format of an input report is shown in Figure 6-1. Each input report starts with a report ID and each input report message report contains data of one touch.

Figure 6-1. Input Report Packet



An example of the input report packets for 3 active touches is shown in Figure 6-2 on page 32.

Figure 6-2. Example Input Report Packets for 3 Active Touches



Each input report consists of a HID-I^{2}C report ID followed by 17 bytes of that describe the status of one active touch. The input report format depends on the geometry calculation field (TCHGEOMEN) of the Digitizer HID Configuration T43 object. Table 6-2 and Table 6-2 explains the detailed format of an input report packet.

Table 6-1. Input Report Format when TCHGEOMEN = 1

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	HID-I2C Report ID							
1	Reserved							Status
2	Touch ID							
3 – 4	X Position							
5 – 6	X’ Position							
7 – 8	Y Position							
9 – 10	Y’ Position							
11	Touch Width							
12	Reserved							
13	Touch Height							
14	Reserved							
15 – 16	Scan Time							
17	Number of active touches to be sent in one package							

Table 6-2. Input Report Format when TCHGEOMEN = 0

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	HID-I2C Report ID							
1	Reserved							Status
2	Touch ID							
3 – 4	X Position							
5 – 6	Reserved							
7 – 8	Y Position							
9 – 10	Reserved							
11	Reserved							
12	Reserved							
13	Reserved							
14	Reserved							
15 – 16	Scan Time							
17	Number of active touches to be sent in one package							

- **Byte 2:**

  Touch ID: Identifies the touch for which this is a status report (starting from 0).
- **Bytes 3 to 10:**

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

  Bytes 5,6, 9, 10 are Reserved when TCHGEOMEN is set to 0.
- **Byte 11:**

  Touch Width: Reports the width of the detected touch when TCHGEOMEN is set to 1.
- **Byte 13:**

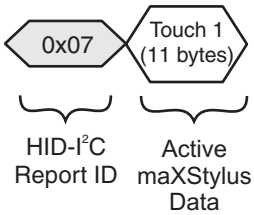
  Touch Height: Reports the height of the detected touch when TCHGEOMEN is set to 1.
- **Byte 15 to 16:**

  Scan Time: Timestamp associated with the current report packet with a 10 kHz resolution.

6.4.2 Active maXStylus Report

The format of an active maXStylus report packet is shown in Figure 6-3.

Figure 6-3. Example Active maXStylus Report



Each active maXStylus report start with a HID-I^{2}C Report ID (value 0x06). Table 6-3 gives the detailed format of an active stylus report packet.

Table 6-3. Active maXStylus Report

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	HID-I^{2}C Report ID = 0x07							
1	Reserved			In Range	Reserved	Eraser	Barrel	Tip
2	X Position							
3								
4	X Center Position							
5								
6	Y Position							
7								
8	Y Center Position							
9								
10	Tip Pressure							

- **Byte 1:**

  Tip: 1 = Contact of the stylus with the touch screen surface, 0 = No contact.

  Barrel: 1 = Barrel button on, 0 = Barrel button off

  Eraser: 1 = Eraser function on, 0 = Eraser function off.

  In Range: 1 = Stylus approaching the touchscreen detected, 0 = No stylus approaching the touchscreen detected.

- **Byte 2 to 3:**

  X position

- **Byte 4 to 5:**

  X center position

- **Byte 6 t o7:**

  Y position

- **Byte 8 to 9:**

  Y center position

- **Byte 10:**

  Tip Pressure: Force exerted against the touch screen surface by the stylus.

6.5 Interface 1 (Generic HID-I²C)

Interface 1 is a generic human interface device. It supports an input report for receiving data from the device and an output report for sending data to the device.

Commands are sent by the host using the output reports. Responses from the device are sent using input reports.

Supported commands are:

- Read/Write Memory Map
- Send Auto-return Messages

6.5.1 Read/Write Memory Map

6.5.1.1 Introduction

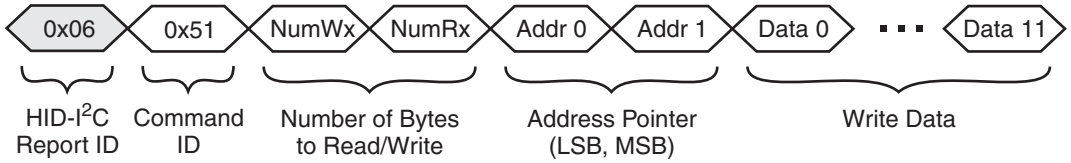
This command is used to carry out a write/read operation on the memory map of the device.

The HID-I²C Report ID is 0x06.

Note: This value may change.

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

Figure 6-4. Generic Command Packet Format

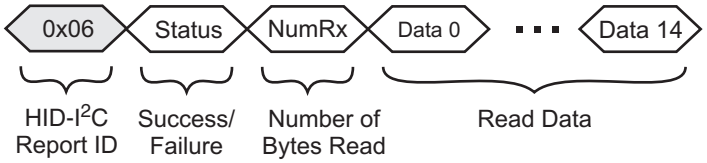


In Figure 6-4:

- **NumWx** is the number of data bytes to write to the memory map (may be zero). If the address pointer is being sent, this must include the size of the address pointer.
- **NumRx** is the number of data bytes to read from the memory map (may be zero).
- **Addr 0** and **Addr 1** form the address pointer to the memory map (where necessary; may be zero if not needed).
- **Data 0** to **Data 11** are the bytes of data to be written (in the case of a write). Note that data locations beyond the number specified by NumWx will be ignored.

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

Figure 6-5. Response Packet Format



In Figure 6-5 on page 35:

- **Status** indicates the result of the command:
  - 0x00 = read and write completed; read data returned
  - 0x04 = write completed; no read data requested
- **NumRx** is the number of bytes following that have been read from the memory map (in the case of a read). This will be the same value as NumRx in the command packet.
- **Data 0** to **Data 14** are the data bytes read from the memory map.

6.5.1.2 Writing To the Device

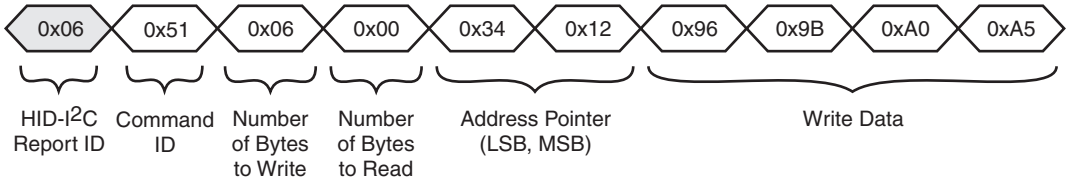
A write operation cycle to the device consists of sending a packet that contains six header bytes. These specify the HID-I^{2}C report ID, the Command ID, the number of bytes to read, the number of bytes to write, and the 16-bit address pointer.

Subsequent bytes in a multibyte transfer form the actual data. These are written to the location of the address pointer, location o

If the address pointer +1, location of the address pointer + 2, and so on.

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

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

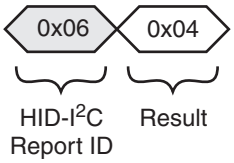


In Figure 6-6:

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

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

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

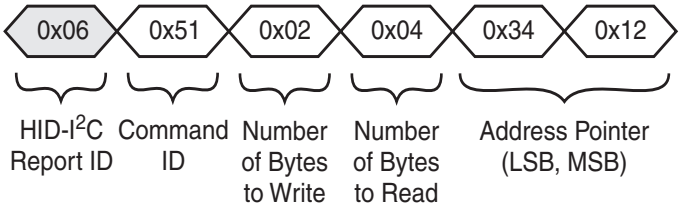


6.5.1.3 Reading From the Device

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

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

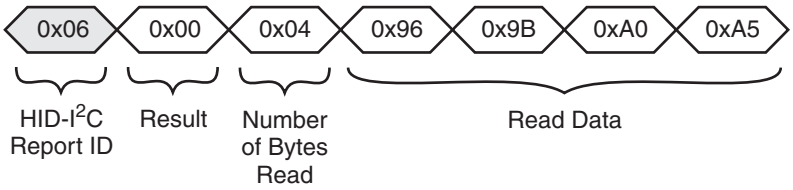
Figure 6-8. Example of a Four-byte Read Starting at Address 0x1234



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

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

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

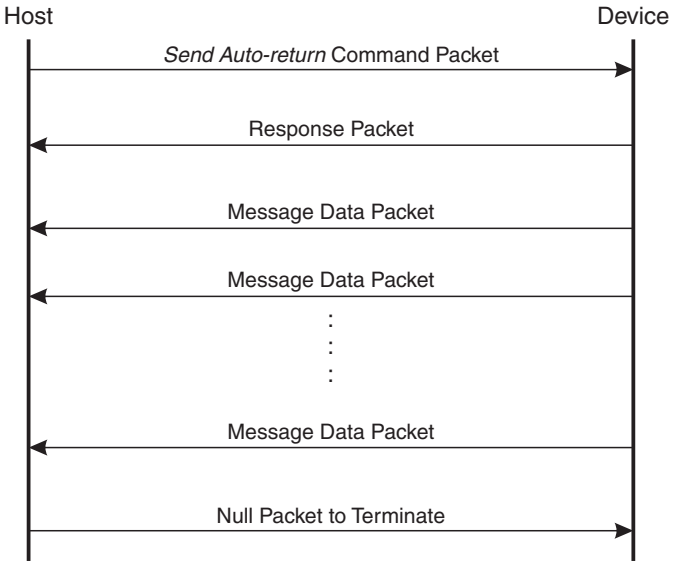


6.5.2 Send Auto-return Messages

6.5.2.1 Introduction

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

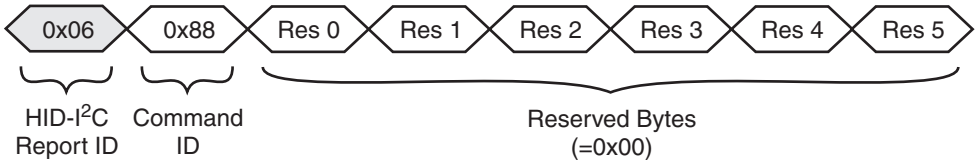
Figure 6-10. Packet Sequence for "Send Auto-return" Command



The HID-I2C Report ID is 0x06.

The command packet has the format given in Figure 6-11 on page 37.

Figure 6-11. Command Packet Format

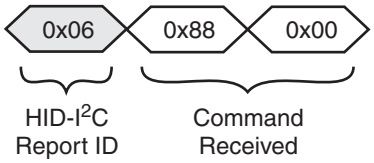


In Figure 6-11:

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

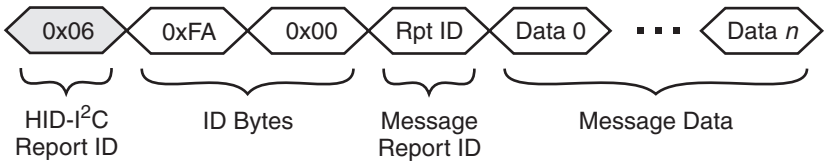
The response packet has the format given in Figure 6-12. Note that with this command, the command packet does not include an address pointer as the device already knows the address of the Message Processor object.

Figure 6-12. Response Packet Format



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

Figure 6-13. Message Packet Format

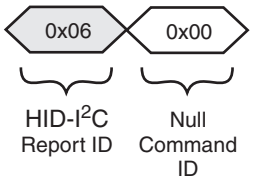


In Figure 6-13:

- **ID Bytes** identify the packet as an auto-return message packet.
- **Rpt ID** is the Report ID returned by the Message Processor object. (1)
- **Message Data** bytes are the bytes of data returned by the Message Processor. The size of the data depends on the source object for which this is the message data. Refer to the *mXT1664S-AT 1.0 Protocol Guide* for more information.

To stop the sending of the messages, the host can send a null command packet. This consists of two bytes: a report ID of 0x01 and a command byte of 0x00 (see Figure 6-13).

Figure 6-14. Null Command Packet Format



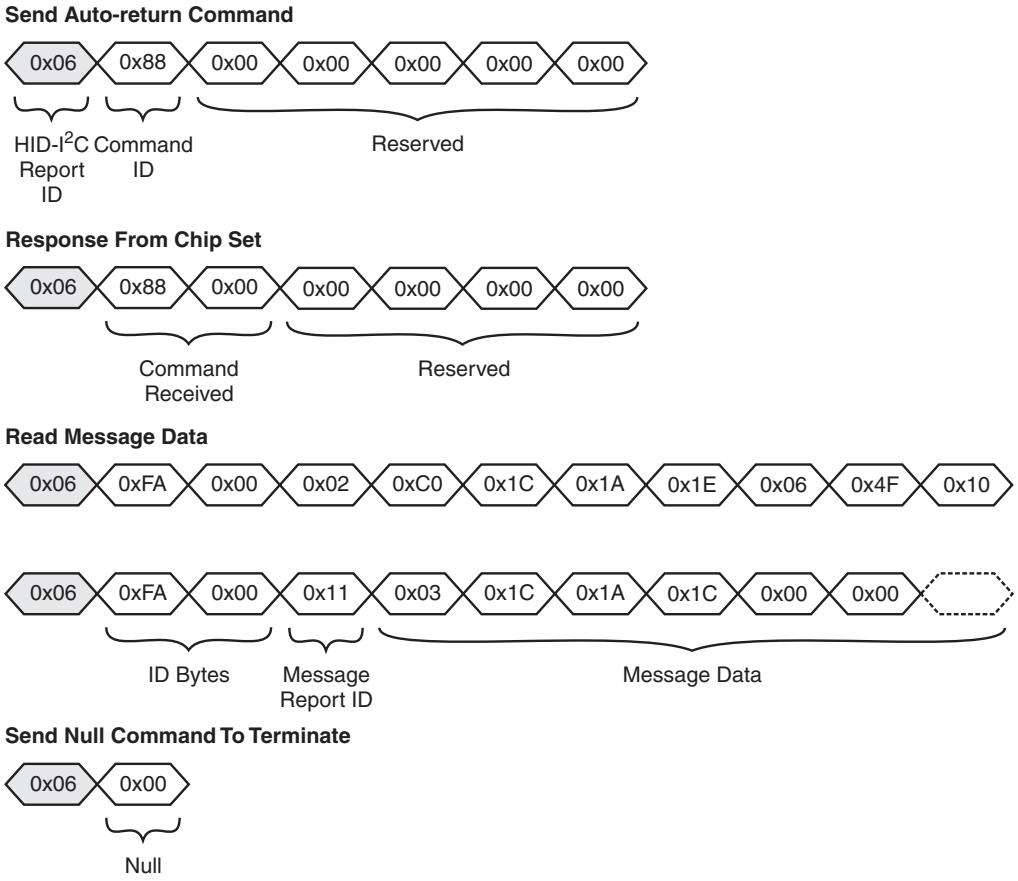
Note that any read or write will also terminate any currently enabled auto-return mode.

1. This is the Report ID used in the Object Protocol and should not be confused with the USB Report ID. Refer to the *mXT1664S-AT 1.0 Protocol Guide* for more information on the use of Report IDs in the Object Protocol.

6.5.2.2 Reading Status Messages

Figure 6-15 shows an example sequence of packets to receive messages from the Message Processor object using the "Send Auto-return" command.

Figure 6-15. Example Auto-return Command Packet



6.6 $\overline{\text{CHG}}$ 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 Input Buffer. This provides the host with an interrupt-style interface with the potential for fast response times. It reduces the need for wasteful I²C-compatible communications.

Further information on the $\overline{\text{CHG}}$ line is given in Section 5.7 on page 29.

6.7 SDA, SCL

Identical to standard I²C operation. Refer to Section 5.8 on page 30.

6.8 Clock Stretching

Identical to standard I²C operation. Refer to Section 5.9 on page 30.

6.9 Power Control

The mXT1664S-AT supports the use of the HID-I²C "SET POWER" commands to put the device into a low power state analogous to the USB "SUSPEND" command.

6.10 Microsoft Windows 8 Compliance

The mXT1664S-AT has algorithms within the Digitizer HID Configuration T43 and Multiple Touch Touchscreen T9 specifically to ensure Microsoft Windows 8 compliance.

The device also supports Microsoft Touch Hardware Quality Assurance (THQA) in the Serial Data Command T68 object. Refer to the Microsoft whitepaper *How to Design and Test Multitouch Hardware Solutions for Windows 8*.

These, and other device features, may need specific tuning.

7. Getting Started with mXT1664S-AT

7.1 Establishing Contact

7.1.1 Communication with the Host

The host can use either the I²C-compatible interface (see Section 5.1 on page 24) or the HID-I²C interface (see Section 6. on page 31) to communicate with the device.

7.1.2 I²C-compatible Interface

On power-up, the $\overline{\text{CHG}}$ line goes low to indicate that there is new data to be read from the Message Processor 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.

7.1.3 HID-I²C Interface

The host can use the HID-I²C interface by connecting the I2CMODE pin to GND.

7.2 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 *mXT1664S-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.

7.3 Writing to the Device

There are three mechanisms for writing to the device:

- Using an I²C-compatible write operation (see Section 5.3 on page 24).
- Using the Generic HID-I²C write operation (see Section 6.5.1.2 on page 36).

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 *mXT1664S-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 T9 object. Some objects are optional and need to be enabled before use. Refer to the *mXT1664S-AT 1.0 Protocol Guide* for more information on the objects.

7.4 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. There are two mechanisms that provide an interrupt-style interface for reading messages in the Message Processor T5 object:

- When using the I²C-compatible interface, the $\overline{\text{CHG}}$ line is asserted whenever a new message is available in the Message Processor T5 object (see Section 5.7 on page 29). See Section 6.5.1.3 on page 36 for information on the format of the I²C-compatible read operation.
- When using the HID-I²C interface, the Generic HID-I²C interface provides an interrupt-driven interface that sends the messages automatically (see Section 6.5.1.3 on page 36)

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

7.5 Configuring 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.

Refer to the *mXT1664S-AT 1.0 Protocol Guide* for more information on configuring the objects.

The following objects are read-only and require no configuration:

- Debug Objects
  - Diagnostic Debug T37
- General objects:
  - Message Processor T5
- Support objects:
  - User Data T38
  - Message Count T44

The following objects must be configured before use:

- General objects
  - Power Configuration T7
  - Acquisition Configuration T8

The following objects should be checked and configured as necessary:

- General objects:
  - Command Processor T6
- Support objects:
  - Communications Configuration T18
  - GPIO/PWM Configuration T19
  - CTE Configuration T46

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

- Touch objects:
  - Multiple Touch Touchscreen T9
  - Key Array T15
- Signal processing objects:
  - Grip Suppression T40
  - Stylus T47
  - One-touch Gesture Processor T24
  - Two-touch Gesture Processor T27
  - Touch Suppression T42
  - Adaptive Threshold T55
  - Shieldless T56
  - Extra Touchscreen Data T57
  - maXCharger T62
  - Active Stylus T63

- Lens Bending T65
- Zone Indication T73

- Support objects:
  - Digitizer HID Configuration T43
  - Self Test T25
  - Timer T61
  - maXStartup T66
  - Serial Data Command T68
  - Dynamic Configuration Controller T70
  - Dynamic Configuration Container T71
  - CTE_SCAN Configuration T77
  - Touch Event Trigger T79

8. Specifications

8.1 Absolute Maximum Specifications

Vdd	3.6 V
XVdd	12 V
AVdd	3.6 V
Max continuous pin current, any control or drive pin	20 mA
Voltage forced onto any pin	–0.3 V to (Vdd or AVdd) + 0.3 V
Configuration parameters maximum writes	10,000

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

8.2 Recommended Operating Conditions

Operating temp	–40°C to +85°C
Storage temp	–60°C to +150°C
Vdd External	2.7 V to 3.3 V ± 5%
AVdd	2.7 V to 3.3 V ± 5%
XVdd	2.7 V to 10.0 V ± 5%
Vdd vs AVdd power sequencing	No sequencing required
Vdd vs XVdd power sequencing	XVdd must not be powered before Vdd
Vdd supply ripple	±50 mV 1 Hz to 1 MHz
XVdd supply ripple	±25 mV 1 Hz to 1MHz
AVdd supply ripple	±25 mV 1 Hz to 1 MHz
Cx transverse load capacitance per node	0.95 pF to 4.8 pF, when XVdd = 2.7 V to 4.75 V 0.5 pF to 4.8 pF, when XVdd = 4.75 V to 10.0 V
Temperature slew rate	10°C/min

8.3 DC Characteristics

8.3.1 Digital Voltage Supply

Parameter	Description	Min	Typ	Max	Units	Notes
Vdd	Operating limits	2.57	–	3.47	V	I^2C-compatible mode
Rate of rise	Rate of rise	0.002	–	2.5	V/µs	

8.3.2 Analog Voltage Supply

Parameter	Description	Min	Typ	Max	Units	Notes
AVdd	Operating limits	2.57	–	3.47	V	Section 4.7.3 on page 22
Rate of rise	Rate of rise	0.002	–	2.5	V/µs	

8.3.3 XVdd Supply

Parameter	Description	Min	Typ	Max	Units	Notes
XVdd	Operating limits	2.57	10.0	11.0	V	
Rate of rise	Rate of rise	2	–	30	V/ms	

Note: The rate of rise values must be followed to avoid permanent damage to the device.

8.3.4 Input/Output Characteristics

Parameter	Description	Min	Typ	Max	Units	Notes
Input ($\overline{\text{RESET}}$, SDA, SCL)						
Vil	Low input logic level	–0.3	–	0.3 × Vdd	V	Vdd = 1.8 V to 3.3 V
Vih	High input logic level	0.7 × Vdd	–	Vdd + 0.3	V	Vdd = 1.8 V to 3.3 V
Iil	Input leakage current	–	–	1	µA	Pull-up resistors disabled
Output ($\overline{\text{CHG}}$)						
Vol	Low output voltage	–	–	0.2 × Vdd	V	Vdd = 3 V, I_{OL} = 2.7 mA, High Drive Enabled Vdd = 3 V, I_{OL} = 1.35 mA
Voh	High output voltage	0.8 × Vdd	–	–	V	Vdd = 3 V, I_{OH} = 2.7 mA, High Drive Enabled Vdd = 3 V, I_{OH} = 1.35 mA

8.4 ESD Information

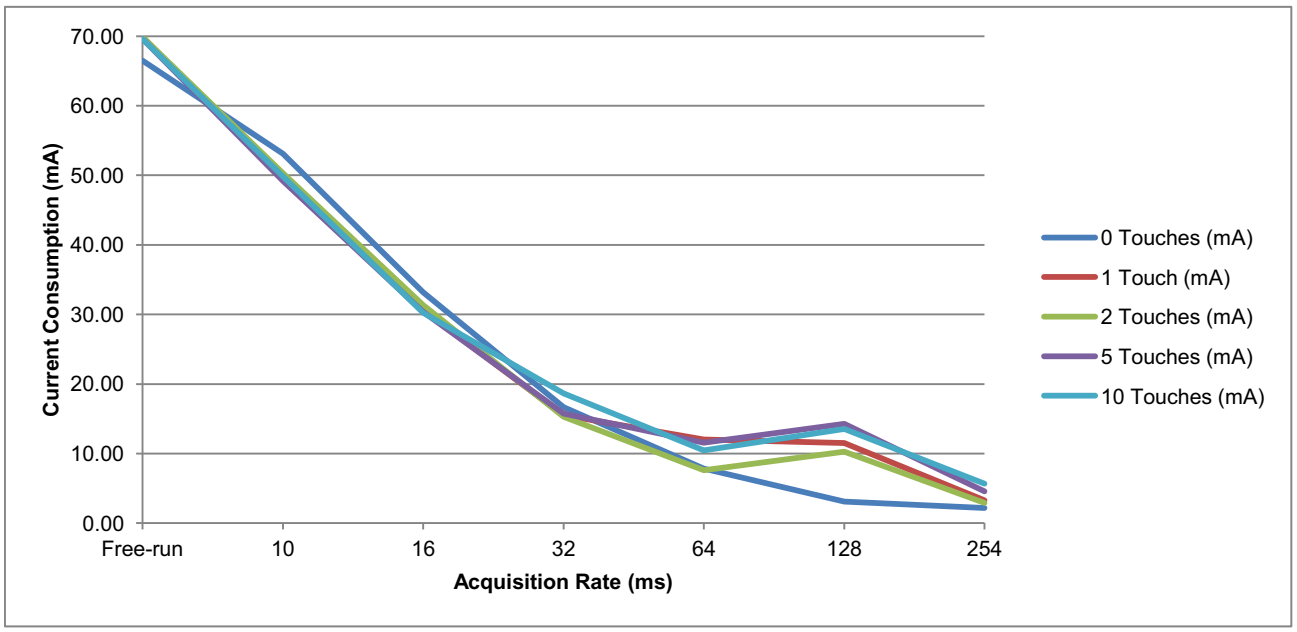
Parameter	Value	Reference standard
Electrostatic Discharge HBM	±2000 V	MIL– STD883 Method 3015.7

8.5 Current Consumption - I²C-compatible Interface

8.5.1 Analog Current

ACTVSYNCSPERX_T46	16

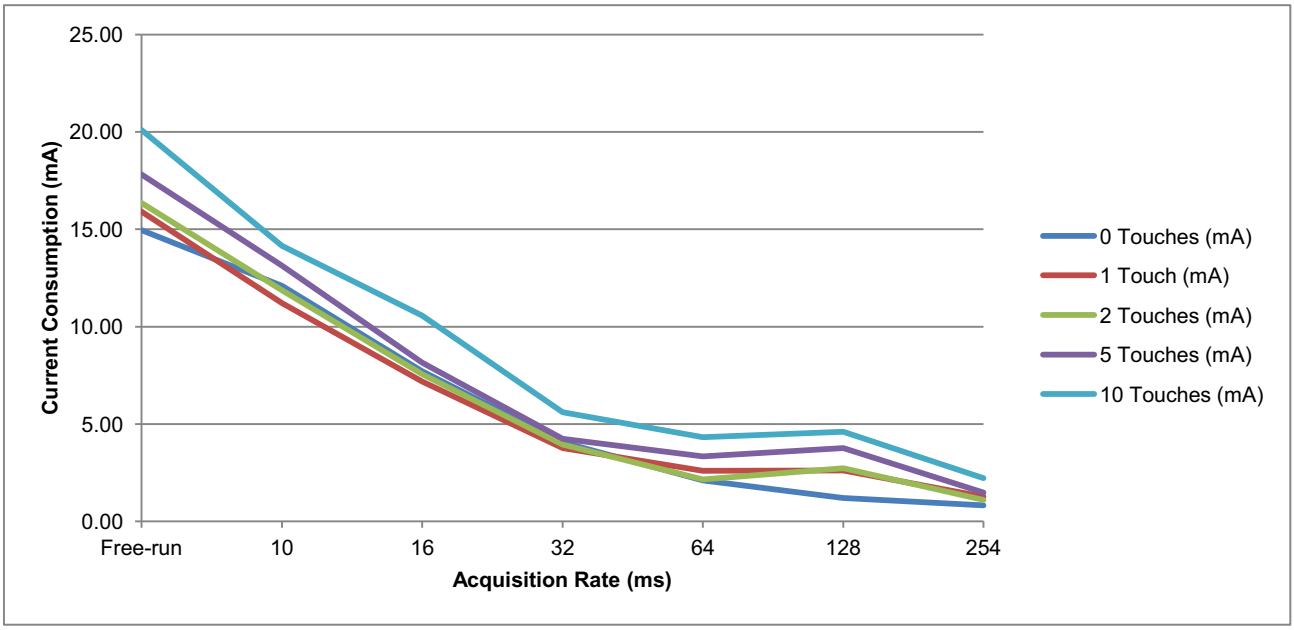
Acquisition Rate (ms)	0 Touches (mA)	1 Touch (mA)	2 Touches (mA)	5 Touches (mA)	10 Touches (mA)
Free-run	66.49	69.89	69.89	69.60	69.61
10	53.11	49.27	50.32	49.20	49.81
16	33.17	30.56	31.35	30.36	30.26
32	16.66	15.53	15.31	15.73	18.65
64	7.82	12.03	7.60	11.56	10.47
128	3.08	11.51	10.27	14.27	13.53
254	2.17	3.27	2.93	4.58	5.66



8.5.2 Digital Current

ACTVSYNCSPERX_T46	16

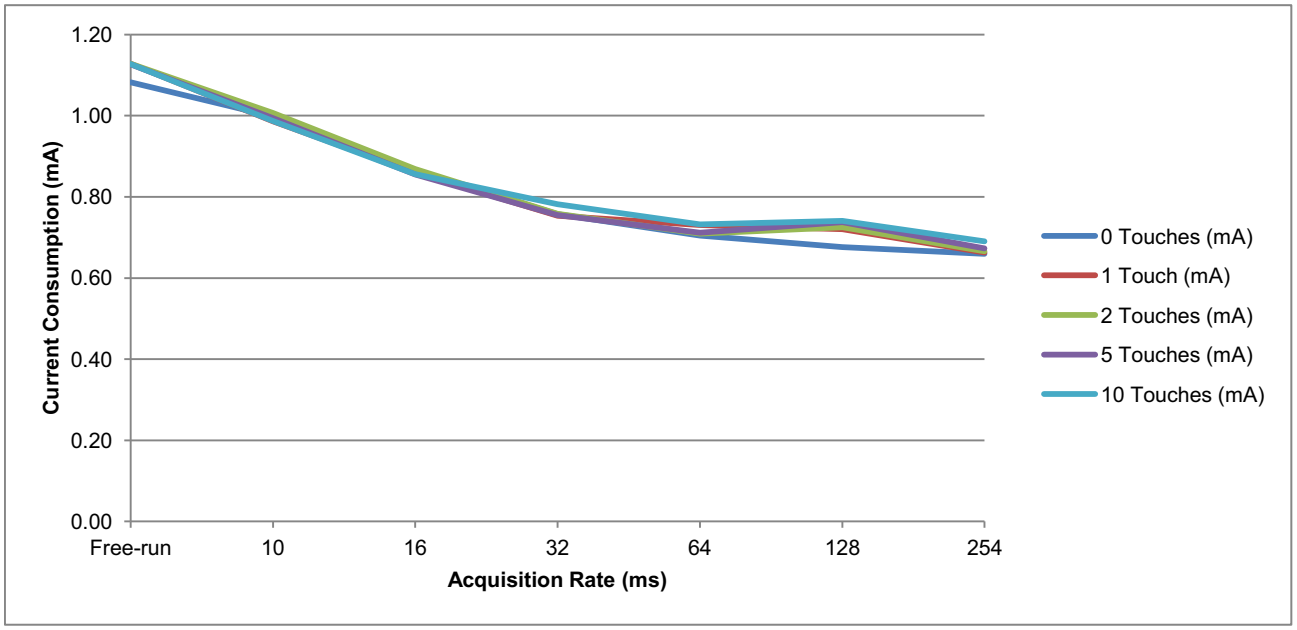
Acquisition Rate (ms)	0 Touches (mA)	1 Touch (mA)	2 Touches (mA)	5 Touches (mA)	10 Touches (mA)
Free-run	14.96	15.90	16.35	17.81	20.10
10	12.10	11.22	11.88	13.13	14.15
16	7.71	7.18	7.56	8.15	10.58
32	4.06	3.77	3.94	4.24	5.61
64	2.11	2.60	2.15	3.34	4.32
128	1.20	2.62	2.74	3.77	4.60
254	0.84	1.27	1.10	1.48	2.22



8.5.3 XVdd Current

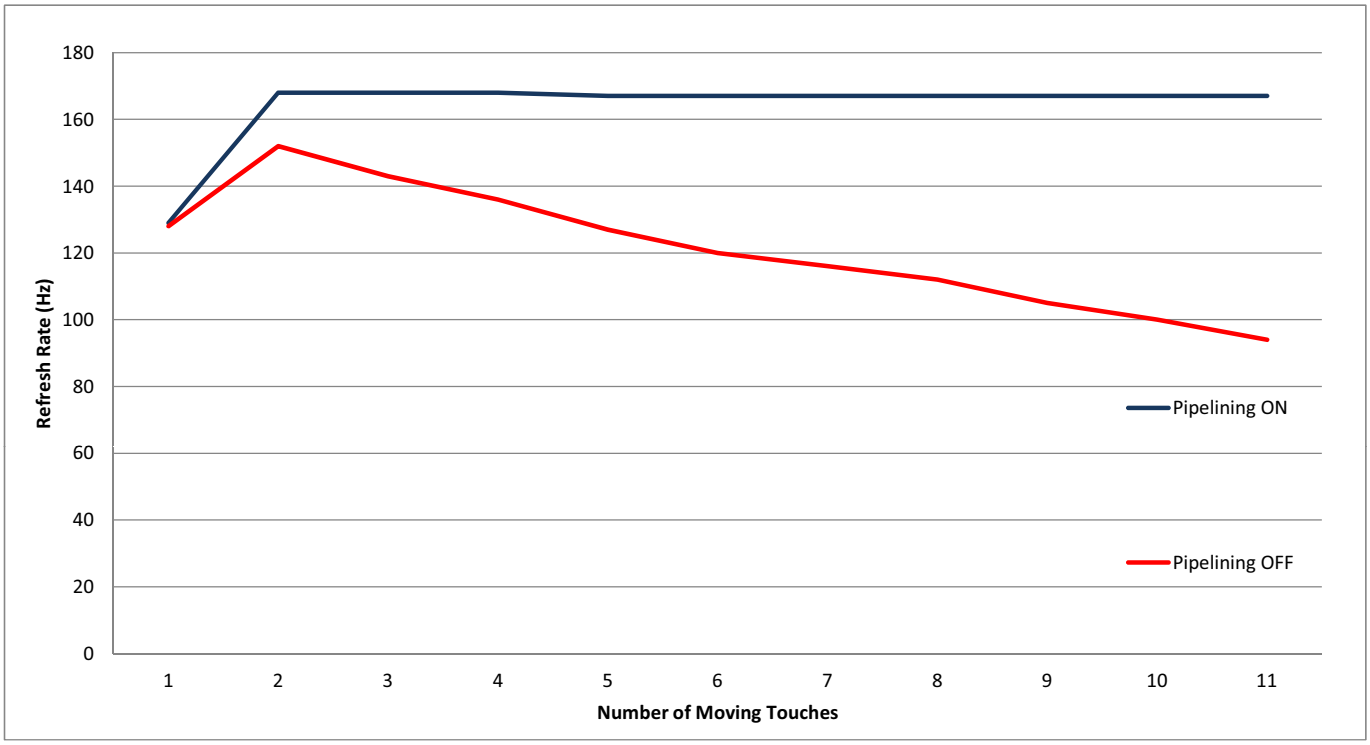
ACTVSYNCSPERX_T46	16

Acquisition Rate (ms)	0 Touches (mA)	1 Touch (mA)	2 Touches (mA)	5 Touches (mA)	10 Touches (mA)
Free-run	1.08	1.13	1.13	1.13	1.13
10	1.00	0.99	1.01	0.99	0.99
16	0.87	0.86	0.87	0.86	0.86
32	0.76	0.75	0.76	0.75	0.78
64	0.70	0.73	0.71	0.71	0.73
128	0.68	0.72	0.73	0.74	0.74
254	0.66	0.66	0.67	0.67	0.69



8.6 Timing Specifications

8.6.1 Speed



8.6.2 Touch Latency

For Cpk = 1.66.

Parameter	Description	Min	Typ	Max	Units	Notes
Tlatency	83 Hz	–	–	37.5	ms	TCHDI = 0 IDLESYNCSPERX/ ACTSYNCSPERX = 8
	100 Hz	–	–	35.5	ms	
	143 Hz	–	–	29.6	ms	

8.7 Reset Timings

For Cpk = 1.66.

Parameter	Min	Typ	Max	Units	Notes
Power on to $\overline{\text{CHG}}$ line low	–	–	78.5	ms	
Hardware reset to $\overline{\text{CHG}}$ line low	–	–	78	ms	
Software reset to $\overline{\text{CHG}}$ line low	–	–	121	ms	

8.8 I²C-compatible Specifications

Parameter	Value (1)
Addresses	0x4A or 0x4B
Maximum bus speed (SCL)	1.7 MHz
I²C specification	Version 2.1
Required pull-up resistance for standard mode (100 kHz)	1 kΩ to 10 kΩ
Required pull-up resistance for fast mode (400 kHz)	1 kΩ to 3 kΩ
Required pull-up resistance for high-speed mode (1.7 MHz)	0.5 kΩ to 0.75 kΩ

1. Refer to *mXT1664S-AT 1.0 Protocol Guide* for bootloader-specific information.

8.9 HID-I²C Specification

Parameter	Operation
Vendor ID	0x03EB (Atmel)
Product ID	0x212C (mXT1664S-AT)
HID-I²C specification	0.9

8.10 Touch Accuracy and Repeatability

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

8.11 Power Supply and Ripple Noise

Parameter	Min	Typ	Max	Units	Notes
Vdd	–	–	±50	mV	Across frequency range 1 Hz to 1 MHz
XVdd and AVdd	–	–	±25	mV	Across frequency range 1 Hz to 1 MHz
XVdd and AVdd	–	–	± 40	mV	Across frequency range 1 Hz to 1 MHz, with Noise Suppression enabled

8.12 Thermal Packaging

8.12.1 Thermal Data

Parameter	Typ	Unit	Condition	Package
Junction to ambient thermal resistance	44.9	°C/W	Still air	LQFP 144, 20 x 20 mm
Junction to case thermal resistance	7.5	°C/W	–	LQFP 144, 20 x 20 mm

8.12.2 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).

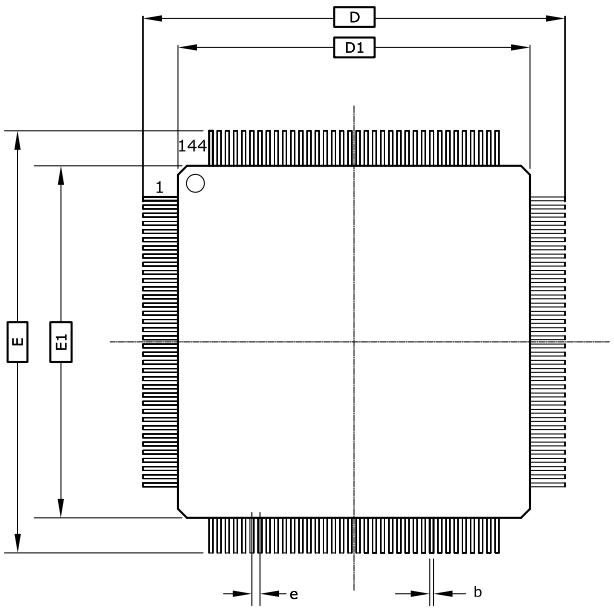
8.13 Soldering Profile

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

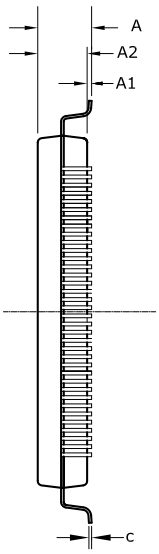
8.14 Mechanical Dimensions

8.14.1 144-pin LQFP 20 x 20 x 1.4 mm

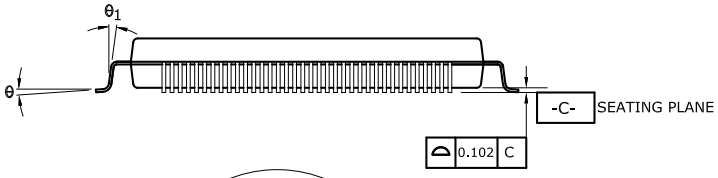
DRAWINGS NOT SCALED

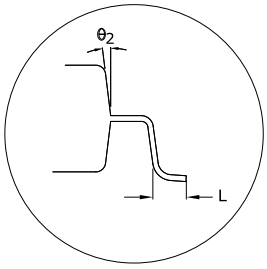


TOP VIEW



SIDE VIEW





DETAIL VIEW

COMMON DIMENSIONS

(Unit of Measure = mm)

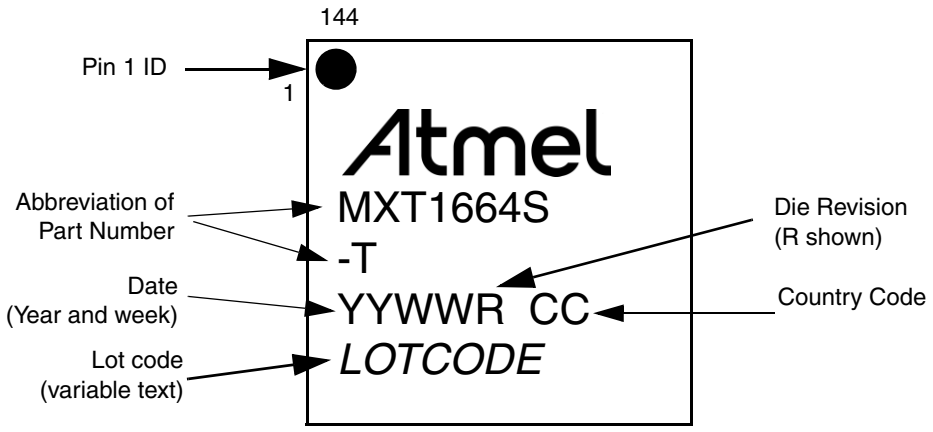
SYMBOL	MIN	NOM	MAX	NOTE
A	-----	-----	1.60	
A1	0.05	-----	0.15	
A2	1.35	1.40	1.45	
D/E		22.00 BSC		
D1/E1		20.00 BSC		2
L	0.45	0.60	0.75	
b		0.22 bsc		
c	0.09	-----	0.20	
e		0.50 bsc		
θ	0°	3.5°	7°	
θ_1	0°	-----	-----	
θ_2	11°	12°	13°	
n		144		

Notes : 1. This drawing is for general information only. Refer to JEDEC Drawing MS-026, Variation BFB.
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.

03/04/2013

	TITLE	GPC	DRAWING NO.	REV.
Atmel **Package Drawing Contact:** **packagedrawings@atmel.com**	RU, 144 Lds - 0.5mm Pitch, 20x20x1.4mm Body Size Low Profile Plastic Quad Flat Package (LQFP)	AEI	RU	H

8.15 Part Marking



8.16 Part Numbers

Part Number	Reference Number	Firmware Revision	Description
ATMXT1664S-ATR (Supplied in tape and reels)	HGH	1.0	144-pin 20 x 20 mm LQFP RoHS compliant

8.17 Moisture Sensitivity Level (MSL)

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

Appendix A. PCB Design Considerations

A.1 Introduction

The following sections give the design considerations that should be adhered to when designing a PCB layout for use with the mXT1664S-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.

A.2 Printed Circuit Board

Atmel recommends the use of a four-layer printed circuit board for mXT1664S-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.

A.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.

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

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

A.4 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 5 mm away). The ground connection of these capacitors should be tracked to 0V by the shortest, heaviest traces possible.

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

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

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

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

Refer to the application note QTAN0094: *mXT1664S PCB/FPCB Layout Guidelines* for more information on PCB/FPCB layout.

A.5 Suggested Voltage Regulator Manufacturers

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

A single low value series resistor (approximately 1Ω) is required from the regulator output to the analog supply input on the device. This, together with the regulator output capacitor, and the capacitors at the DC input to the device, forms a simple filter on the supply rail.

A low noise device should be chosen for the regulator. If possible this should have provision for adding a capacitor across the internal reference for further noise reduction. Reference should be made to the manufacturer's datasheet.

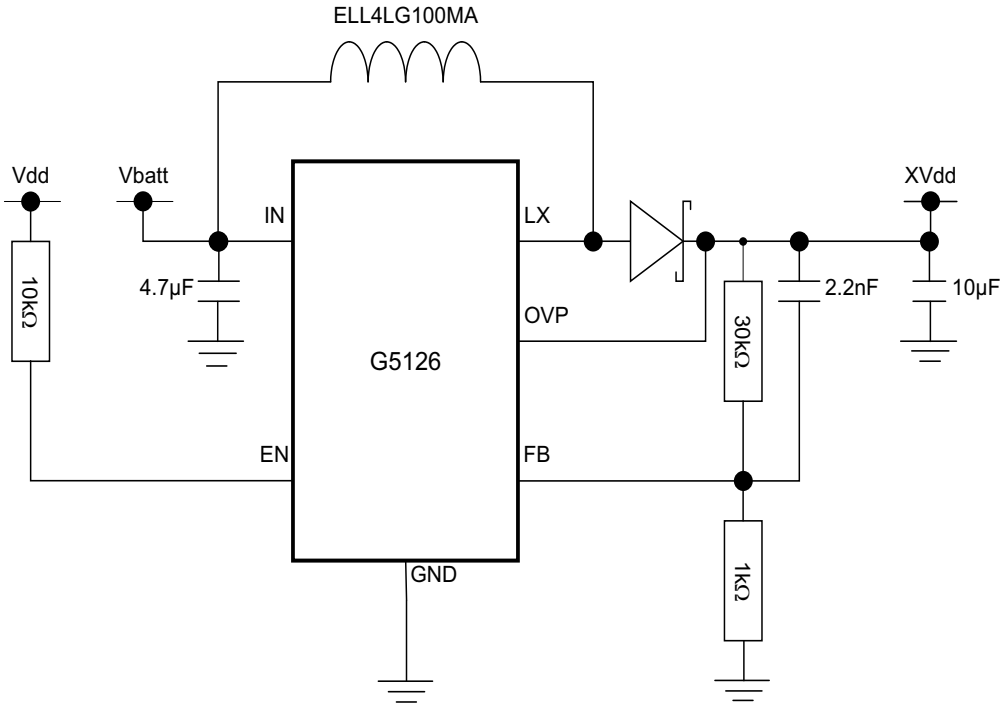
The voltage regulators listed in Table A-1 have been tested and found to work well with the mXT1664S-AT. They have compatible footprints and pin-out specifications, and are available in the SOT-23 package.

Table A-1. Recommended Voltage Regulators

Manufacturer	Pin	Part Number
Texas Instruments	AVdd	TLV70028
Linear Technology	Vdd	LT1761
Micrel	Vdd	MIC5255
National Semiconductor	Vdd	LP2981
Torex	Vdd	XC6204
AMS	XVdd	AS1340
GMMT	XVdd	G5126

Note that 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 datasheet should always be referred to when selecting capacitors for these devices and the typical recommended values, types and dielectrics adhered to.

Figure A-1. Example Regulator Circuit



Note that a “soft-start” regulator with excellent noise and load step regulation will be needed to satisfy the XVdd supply requirements. 1% resistors should be used to define the nominal output voltage. If 5% resistors are used, the nominal XVdd voltage must be reduced accordingly to ensure that the recommended voltage range is adhered to. Figure A-1 provides an example circuit for the XVdd supply.

A.6 Crystal Oscillator

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

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

A.7 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.

A.8 Component Placement

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

A.9 Digital Signals

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

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

A.10 EMC and Other Observations

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

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

Appendix B. Glossary of Terms

Channel

See *Node.*

Jitter

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

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

Linearity

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

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

Multitouch

The ability of a touchscreen to report multiple concurrent touches. The touches are reported as separate sets of XY coordinates.

Node

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

One-touch Gesture

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

Resolution

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

1. Sometimes called Bell-shaped or Normal distribution.

Touchscreen

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

Two-touch Gesture

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

Appendix C. QMatrix PrimerQMatrix Primer

C.1 Acquisition Technique

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

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

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

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

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

C.2 Moisture Resistance

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

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

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

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

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

C.3 Interference Sources

C.3.1 Power Supply

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

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

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

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

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

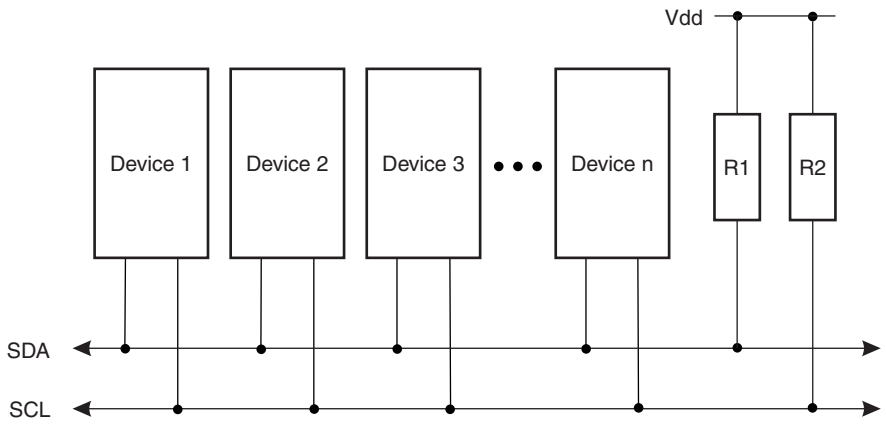
C.3.2 Other Noise Sources

Refer to QTAN0079, *Buttons, Sliders and Wheels Sensor Design Guide*, for information (downloadable from the Touch Technology area of the Atmel website).

C.4 I²C Basics (I²C-compatible Operation)Interface Bus

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

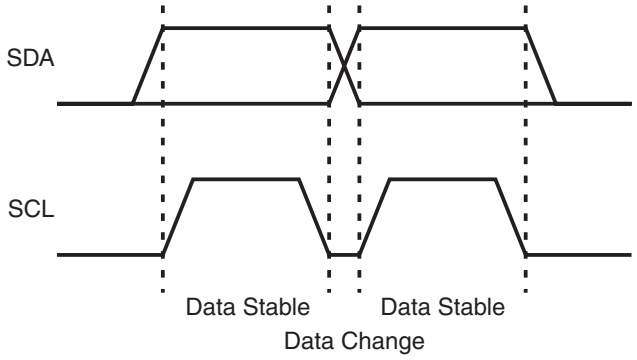
Figure C-1. I²C Interface Bus



C.5 Transferring Data Bits

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

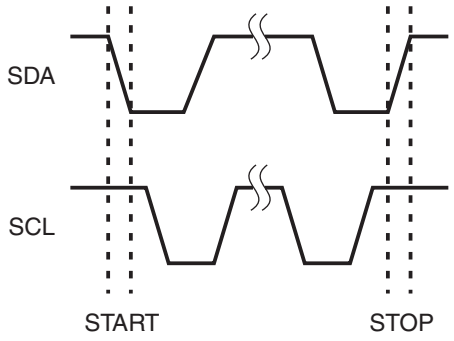
Figure C-2. Data Transfer



C.6 START and STOP Conditions

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

Figure C-3. START and STOP Conditions

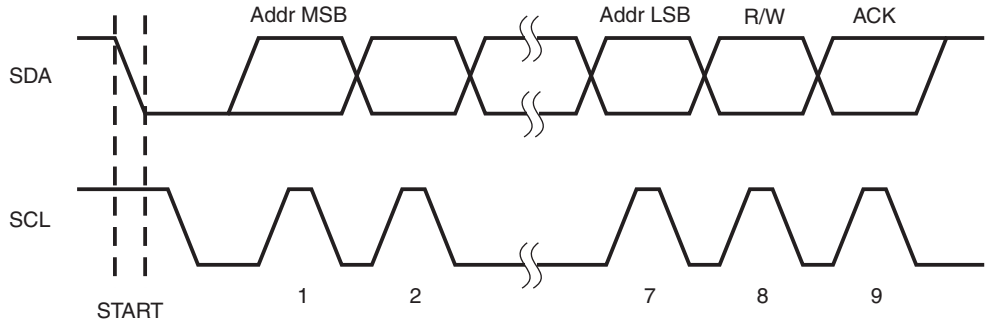


C.7 Address Byte Format

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

The most significant bit of the address byte is transmitted first. The address sent by the host must be consistent with that selected with the option jumpers.

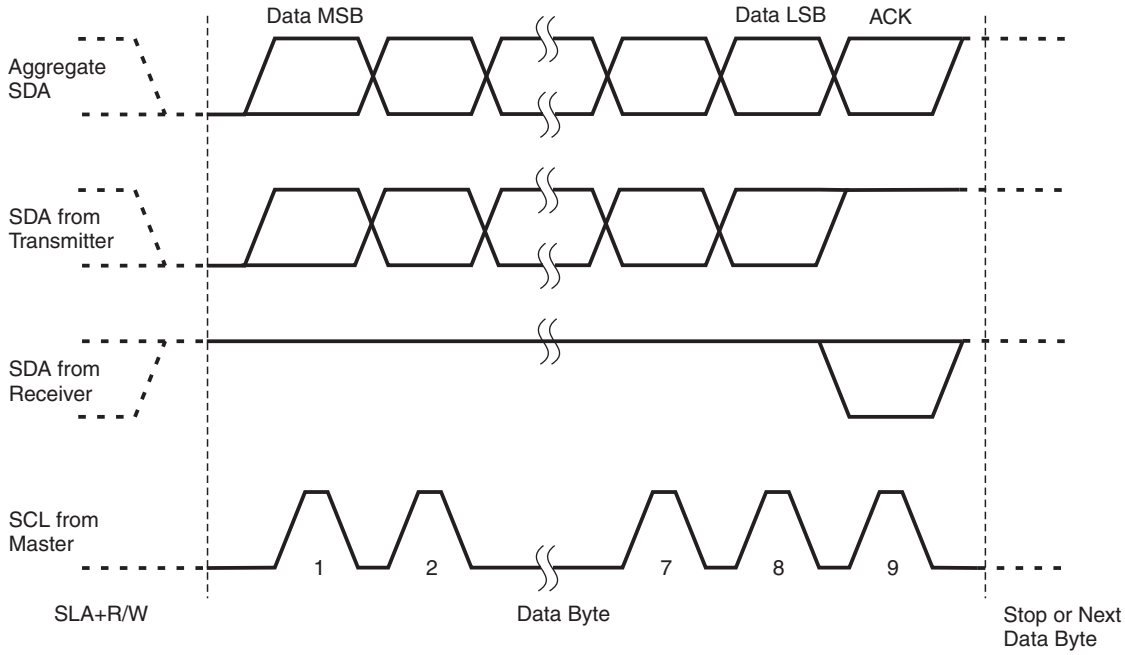
Figure C-4. Address Byte Format



C.8 Data Byte Format

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

Figure C-5. Data Byte Format



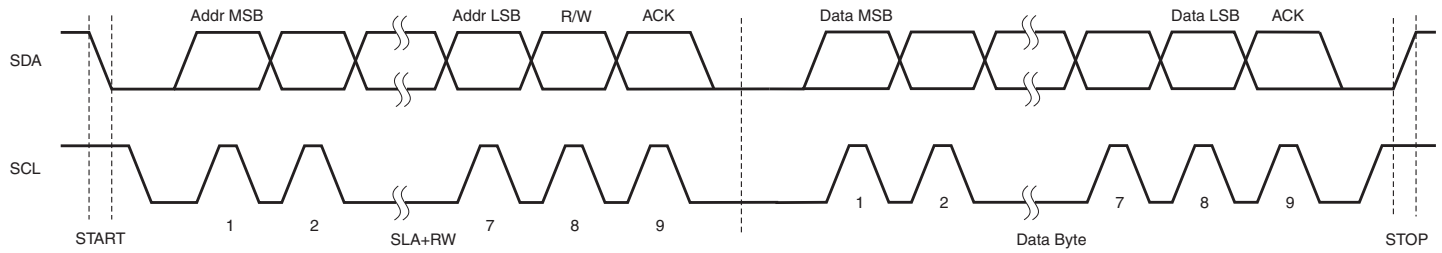
C.9 Combining Address and Data Bytes into a Transmission

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

Note: Each write or read cycle must end with a stop condition. The device may not respond correctly if a cycle is terminated by a new start condition.

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

Figure C-6. Byte Transmission



Revision History

Revision Number	History
Revision ASX – March 2013	Initial release – Summary
Revision BX– November 2013	Initial release for firmware revision 2.0 – Preliminary
Revision CX– January 2014	• Correction to XVdd values in “Recommended Operating Conditions” on page 44. • Redesignation of *channels* to *nodes* throughout.
Revision DX – August 2014	Updated for firmware revision 1.0 – Release • Firmware revision changed to 1.0 • Touch latency figures updated • Reset timings updated
Revision EX – August 2014	• Updated Reference Number

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 Roa
Kwun Tong, Kowloon
HONG KONG
Tel: (+852) 2245-6100
Fax: (+852) 2722-1369

Atmel Munich 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