
maXTouch® 3432-channel Touchscreen Controller

[Datasheet](#)

Features

- Chipset Configuration
 - One master mXT3432S-M-AT device
 - Two slave mXT3432S-S-AT devices
- maXTouch® Touchscreen
 - True 12-bit multiple touch reporting and real-time XY tracking for up to 16 concurrent touches per touchscreen
 - Screen sizes up to 17.3 inches diagonal, supported at 5 mm electrode pitch
- Number of Channels
 - Electrode grid configurations of up to 44 X and 78 Y lines supported
 - Touchscreens up to 3432 channels (subject to other configurations)
- Signal Processing
 - Advanced digital filtering using both hardware engine and firmware
 - Self-calibration
 - Auto drift compensation
 - Grip and palm suppression algorithms to remove unintentional touches
 - Reports one-touch and two-touch gestures
 - Down-scaling and clipping support to match LCD resolution
 - Ultra-fast start-up and calibration for best user experience
 - Supports axis flipping and axis switch-over for portrait and landscape modes
 - Supports baseline references for better calibration
 - Supports lens bending algorithms to restore signal distortions
- Environmental Conditions
 - Improved Moisture tolerance with retransmission compensation algorithms
 - Operating temperature -40°C to $+85^{\circ}\text{C}$ for use in a wide range of climates
- Unambiguous Multitouch Performance
 - Advanced algorithms facilitate exceptional multitouch user experience
- Scan Speed
 - Maximum single touch >100 Hz, subject to configuration
 - Configurable to allow power/speed optimization
 - Programmable timeout for automatic transition from active to idle states
- Response Times
 - Initial latency <25 ms for first touch from idle, subject to configuration
- Sensors
 - Works with PET or glass sensors, including curved profiles
 - Works with all proprietary sensor patterns recommended by Atmel
- Glove Support
 - Supports gloved operation with various materials of up to 3 mm thickness

- Panel Thickness
 - Glass up to 5 mm, screen size dependent
 - Plastic up to 3 mm, dependent on screen size, configuration and approval from Atmel
- Interfaces
 - I²C-compatible slave mode, supported up to 200 kHz
 - USB 2.0-compliant composite device, full speed (12 Mbps)
 - HID-I²C interface for Microsoft® Windows® 8.1
- Power
 - Digital 3.3 V nominal
 - Analog 2.7 V to 3.3 V nominal
 - High voltage X-line drive up to 10.0 V nominal
- Packages
 - Master device: 1 × 100-pin TQFP 14 × 14 × 1.0 mm, 0.5 mm pin pitch
 - Slave devices: 2 × 144-pin LQFP 20 × 20 × 1.4 mm, 0.5 mm pin pitch

Table of Contents

| | |
|---|----|
| Features | 1 |
| Table of Contents | 3 |
| 1. Overview of the mXT3432S-AT | 6 |
| 1.1 Introduction | 6 |
| 1.2 Chipset Architecture | 7 |
| 2. Pinouts | 8 |
| 2.1 Pinout Configurations | 8 |
| 2.2 Pin Configurations | 11 |
| 2.3 Schematics | 20 |
| 3. Touchscreen Basics | 22 |
| 3.1 Sensor Construction | 22 |
| 3.2 Electrode Configuration | 22 |
| 3.3 Scanning Sequence | 22 |
| 3.4 Touchscreen Sensitivity | 22 |
| 4. Detailed Operation | 24 |
| 4.1 Power-up/Reset | 24 |
| 4.2 Calibration | 25 |
| 4.3 Operational Modes | 26 |
| 4.4 Touchscreen Layout | 26 |
| 4.5 Signal Processing | 26 |
| 4.6 Circuit Components | 28 |
| 4.7 PCB Layout | 29 |
| 4.8 Debugging | 29 |
| 4.9 Communications | 29 |
| 4.10 Configuring the Chipset | 29 |
| 5. I2C Communications | 30 |
| 5.1 Communications Protocol | 30 |
| 5.2 I ² C-compatible Addresses | 30 |
| 5.3 Writing To the Chipset | 30 |
| 5.4 I ² C-compatible Writes in Checksum Mode | 31 |
| 5.5 Reading From the Chipset | 31 |
| 5.6 Reading Status Messages with DMA | 31 |
| 5.7 $\overline{\text{CHG}}$ Line | 34 |
| 5.8 $\overline{\text{WAKE}}$ Line | 36 |
| 5.9 SDA, SCL | 36 |
| 5.10 Clock Stretching | 36 |
| 6. USB Communications | 37 |
| 6.1 Communications Protocol | 37 |
| 6.2 Endpoint Addresses | 37 |
| 6.3 Composite Device | 38 |
| 6.4 Interface 0 (Digitizer HID) | 38 |
| 6.5 Interface 1 (Generic HID) | 41 |
| 6.6 USB Suspend Mode | 48 |

| | |
|--|-----------|
| 7. HID-I²C-compatible Communications | 49 |
| 7.1 Communications Protocol | 49 |
| 7.2 I ² C-compatible Addresses | 49 |
| 7.3 Device | 49 |
| 7.4 Interface 0 (Digitizer HID-I ² C) | 49 |
| 7.5 Interface 1 (Generic HID-I ² C) | 53 |
| 7.6 $\overline{\text{CHG}}$ Line | 57 |
| 7.7 SDA, SCL | 57 |
| 7.8 Clock Stretching | 57 |
| 7.9 Microsoft Windows 8.x Compliance | 57 |
| 8. Getting Started with mXT3432S-AT | 58 |
| 8.1 Establishing Contact | 58 |
| 8.2 Using the Object Protocol | 58 |
| 8.3 Writing to the Chipset | 58 |
| 8.4 Reading from the Chipset | 59 |
| 8.5 Configuring the Chipset | 59 |
| 9. Specifications | 61 |
| 9.1 Absolute Maximum Specifications | 61 |
| 9.2 Recommended Operating Conditions | 61 |
| 9.3 DC Characteristics | 62 |
| 9.4 ESD Information | 62 |
| 9.5 Current Consumption – I ² C Interface | 63 |
| 9.6 Current Consumption – USB Interface | 66 |
| 9.7 Timing Specifications | 69 |
| 9.8 I ² C Specifications | 70 |
| 9.9 USB Specification | 71 |
| 9.10 HID-I ² C Specification | 71 |
| 9.11 Touch Accuracy and Repeatability | 71 |
| 9.12 Power Supply and Ripple Noise | 71 |
| 9.13 Thermal Packaging | 72 |
| 9.14 Soldering Profile | 72 |
| 9.15 Mechanical Dimensions | 73 |
| 9.16 Part Marking | 75 |
| 9.17 Part Numbers | 75 |
| 9.18 Moisture Sensitivity Level (MSL) | 75 |
| Appendix A. PCB Design Considerations | 76 |
| A.1 Introduction | 76 |
| A.2 Printed Circuit Board | 76 |
| A.3 Supply Rails and Ground Tracking | 76 |
| A.4 Power Supply Decoupling | 76 |
| A.5 Suggested Voltage Regulator Manufacturers | 77 |
| A.6 Crystal Oscillator | 78 |
| A.7 Analog I/O | 78 |
| A.8 Component Placement | 78 |
| A.9 Digital Signals | 78 |
| A.10 EMC and Other Observations | 78 |
| Appendix B. Glossary of Terms | 79 |

| | |
|---|----|
| Appendix C. QMatrix Primer | 81 |
| C.1 Acquisition Technique | 81 |
| C.2 Moisture Resistance | 81 |
| C.3 Interference Sources | 82 |
| Appendix D. I ² C Basics (I ² C-compatible Operation) | 83 |
| D.1 Interface Bus | 83 |
| D.2 Transferring Data Bits | 83 |
| D.3 START and STOP Conditions | 84 |
| D.4 Address Byte Format | 84 |
| D.5 Data Byte Format | 85 |
| D.6 Combining Address and Data Bytes into a Transmission | 85 |
| Revision History | 86 |

1. Overview of the mXT3432S-AT

1.1 Introduction

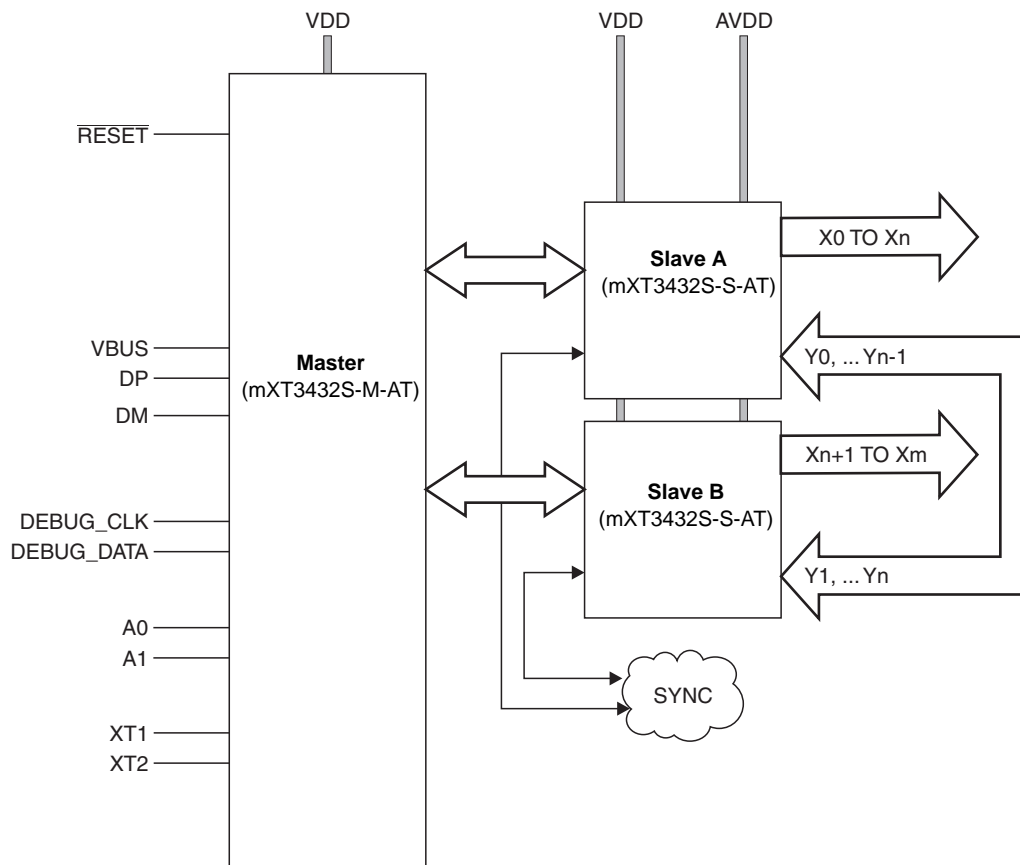
The Atmel mXT3432S-M-AT, together with its two associated mXT3432S-S-AT slave devices, is part of the maXTouch® family of touchscreen controllers. This chipset builds on the success of the maXTouch family to provide a greatly improved user experience:

- **Patented capacitive sensing method** – The mXT3432S-AT uses a unique charge-transfer acquisition engine to implement the QMatrix® capacitive sensing method patented by Atmel. This allows the measurement of up to 3432 mutual capacitance nodes. Coupled with a state-of-the-art CPU, the entire touchscreen sensing solution can measure, classify and track finger touches with a high degree of accuracy.
- **Capacitive Touch Engine (CTE)** – The acquisition engine uses an optimal measurement approach to ensure almost complete immunity from parasitic capacitance on the receiver inputs (Y lines). The engine includes sufficient dynamic range to cope with touchscreen mutual capacitances spanning 0.63 pF to 5 pF. This allows great flexibility for use with the Atmel proprietary ITO pattern designs. One and two layer ITO sensors are possible using glass or PET substrates.
- **Processing power** – The master mXT3432S-M-AT combines with its slave mXT3432S-S-AT devices 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.
- **Noise filtering** – The mXT3432S-AT makes use of the noise filtering algorithms found on the maXTouch solution and copes well with LCD noise and RF noise, but operational enhancements allow the mXT3432S-M-AT to cope even better with severe noise.
- **User experience** – The mXT3432S-AT makes use of the Atmel mutual capacitance method to provide unambiguous multitouch performance and a responsive user experience. Hysteresis algorithms ensure that where a light touch is applied this is reported as a continuous touch, even when close to the touch threshold level, to prevent jitter on the screen. Algorithms also ensure that an on-screen cursor is stationary after the touch is removed, or remains on the edge of the visible area after a drag gesture.
- **Interpreting user intention** – The *Object Protocol* provides enhanced signal processing capabilities. 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, a resting palm, or a touching cheek or ear also help ensure that the user's intentions are correctly interpreted.

1.2 Chipset Architecture

The master mXT3432S-M-AT device controls two slave mXT3432S-S-AT devices, as shown in Figure 1-1.

Figure 1-1. System Block Diagram



Each of the two mXT3432S-S-AT slave devices controls up to 32 X lines and 39 Y lines, with a total of 44 X lines and 78 Y lines available for use.

The X lines are distributed across the two slave devices in two sequential blocks. The Y lines, however, are distributed across the two slave devices in an interleaved manner, such that Y0 connects to Slave A, Y1 connects to Slave B, and so on.

Table 1-1. Sense Lines

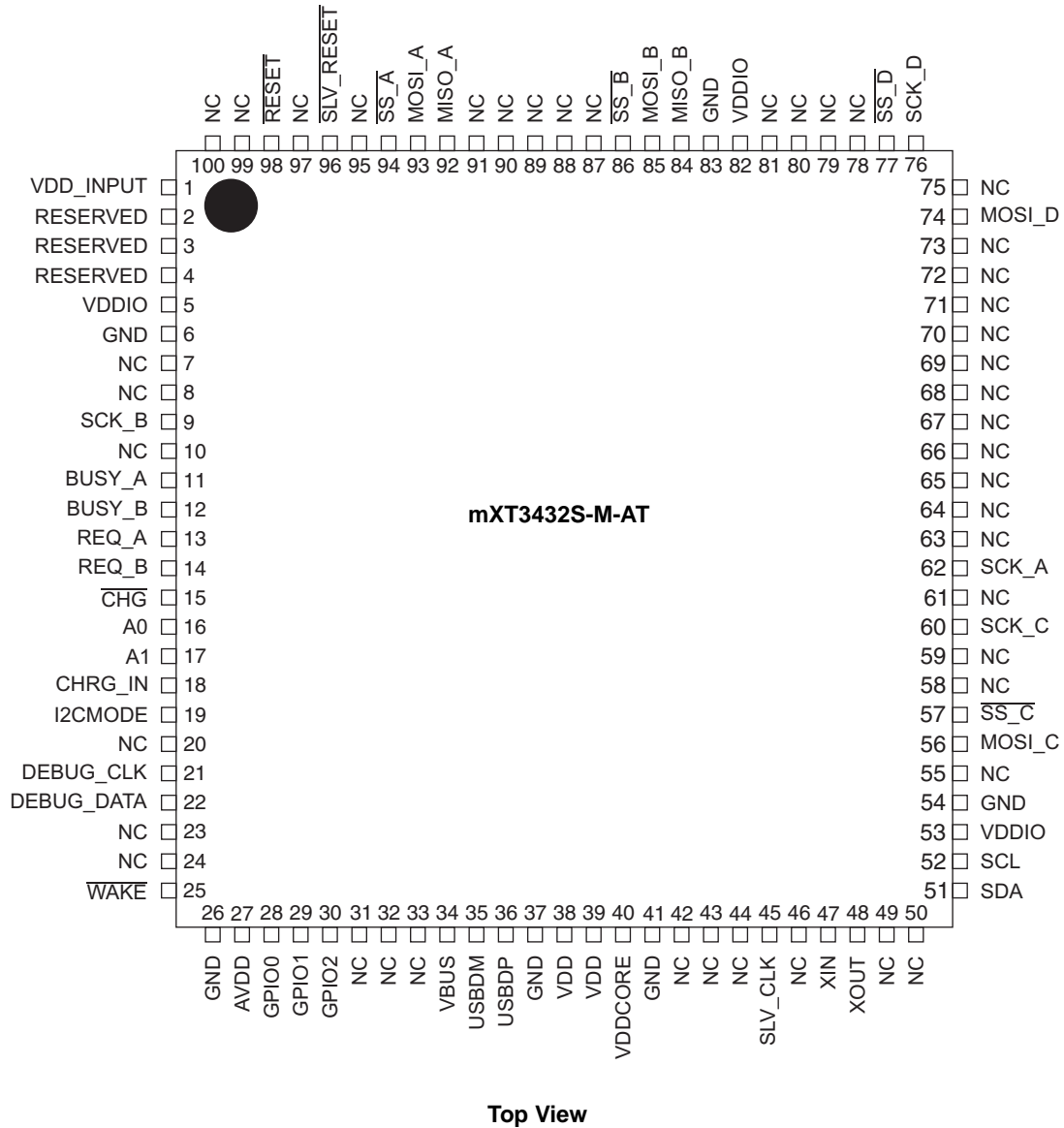
| Slave Device | Controls Sense Lines... | |
|--------------|-------------------------|---------------------|
| | X | Y |
| Slave A | X0 to X31 | Y0, Y2, Y4, ... Y76 |
| Slave B | X32 to X43 | Y1, Y3, Y5, ... Y77 |

The host interfaces with the single master device only; it never needs to deal with the slave devices. It is the responsibility of the master chip to ensure that the configuration and use of the slaves is carried out in a uniform and consistent manner. Communication with the host is achieved using the USB or I²C-compatible interface.

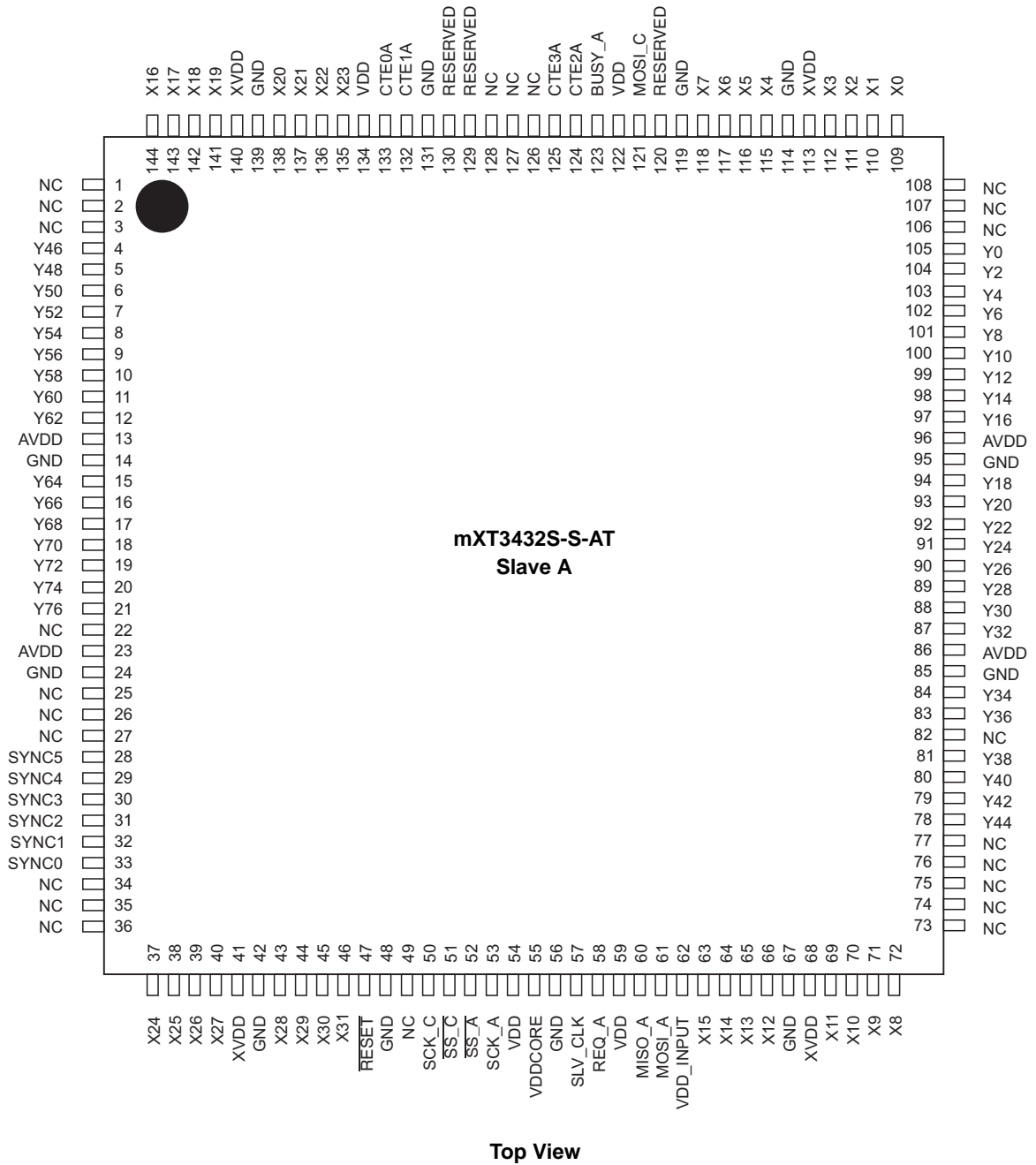
2. Pinouts

2.1 Pinout Configurations

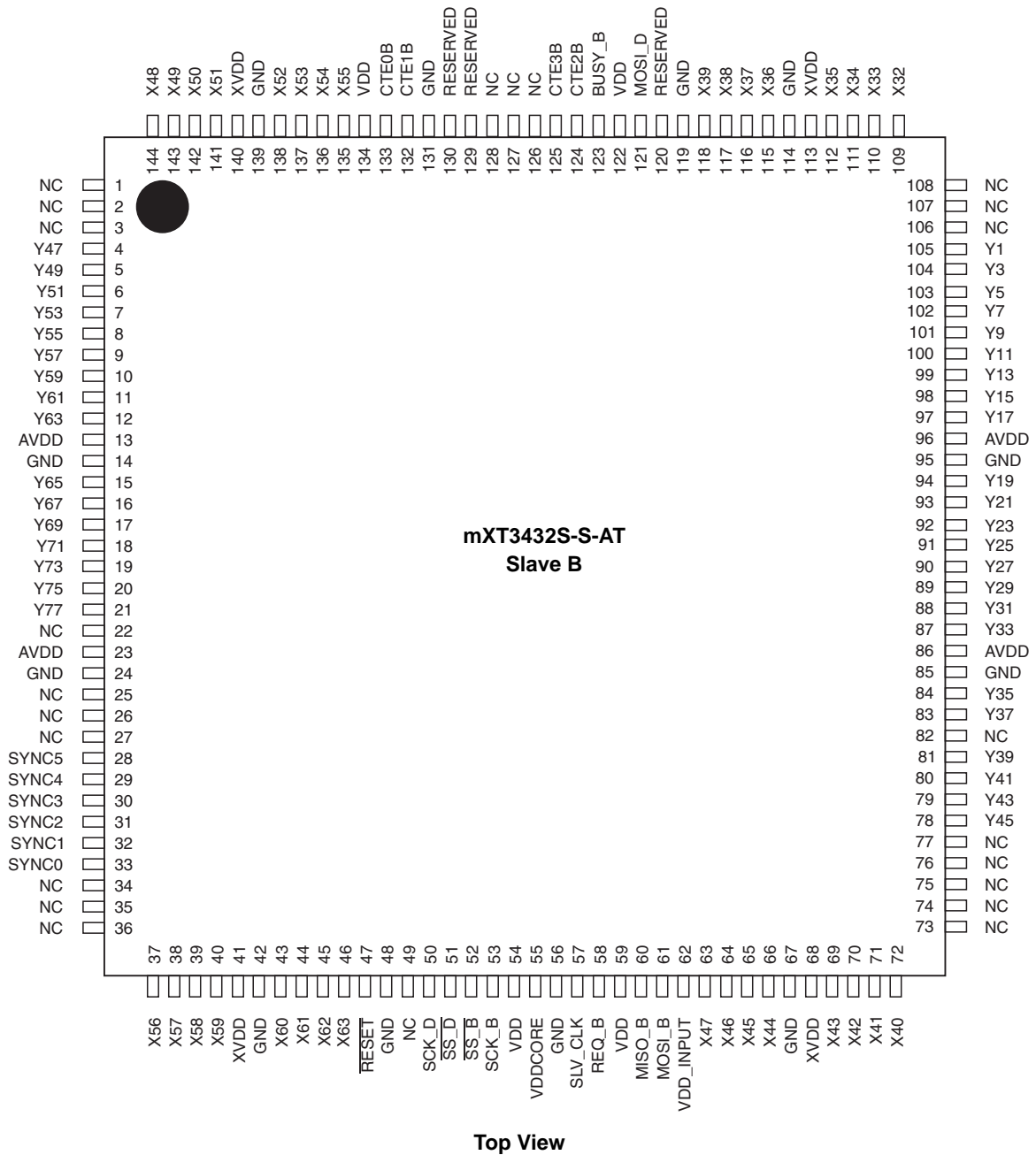
2.1.1 Master mXT3432S-M-AT – 100-pin TQFP



2.1.2 Slave A Connections – 144-pin LQFP



2.1.3 Slave B Connections – 144-pin LQFP



2.2 Pin Configurations

2.2.1 Master mXT3432S-M-AT – 100-pin TQFP

Table 2-1. Pin Listing mXT3432S-M-AT –100-pin TQFP

| Pin | Name | Type | Comments | If Unused, Connect To... |
|-----|--------------------------|------|--|--------------------------|
| 1 | VDD_INPUT | P | Digital power | – |
| 2 | RESERVED | – | Reserved for future use | – |
| 3 | RESERVED | – | Reserved for future use | – |
| 4 | RESERVED | – | Reserved for future use | – |
| 5 | VDDIO | P | Digital power | – |
| 6 | $\overline{\text{GND}}$ | P | Ground | – |
| 7 | $\overline{\text{NC}}$ | – | No connection | Leave open |
| 8 | $\overline{\text{NC}}$ | – | No connection | Leave open |
| 9 | SCK_B | O | Inter-chip signal | – |
| 10 | $\overline{\text{NC}}$ | – | No connection | Leave open |
| 11 | BUSY_A | I | Inter-chip signal | – |
| 12 | BUSY_B | I | Inter-chip signal | – |
| 13 | REQ_A | O | Inter-chip signal | – |
| 14 | REQ_B | O | Inter-chip signal | – |
| 15 | $\overline{\text{CHG}}$ | OD | State change interrupt | – |
| 16 | A0 | I | I ² C-compatible address select | Leave open |
| 17 | A1 | I | I ² C-compatible address select | Leave open |
| 18 | CHRG_IN | I | Charger present input | GND |
| 19 | I2CMODE ⁽²⁾ | I | I ² C-compatible protocol selection | Leave open |
| 20 | $\overline{\text{NC}}$ | – | No connection | Leave open |
| 21 | DEBUG_CLK | I | Debug clock | Leave open |
| 22 | DEBUG_DATA | O | Debug data | Leave open |
| 23 | $\overline{\text{NC}}$ | – | No connection | Leave open |
| 24 | $\overline{\text{NC}}$ | – | No connection | Leave open |
| 25 | WAKE | I | External wake-up pin | – |
| 26 | GND | P | Ground | – |
| 27 | $\overline{\text{AVDD}}$ | P | Analog power | – |
| 28 | GPIO0 | I/O | General purpose IO | Leave open |
| 29 | GPIO1 | I/O | General purpose IO | Leave open |

Table 2-1. Pin Listing mXT3432S-M-AT –100-pin TQFP (Continued)

| Pin | Name | Type | Comments | If Unused, Connect To... |
|-----|---------------------------|------|------------------------------------|--------------------------|
| 30 | $\overline{\text{GPIO2}}$ | I/O | General purpose IO | Leave open |
| 31 | $\overline{\text{NC}}$ | – | No connection | Leave open |
| 32 | $\overline{\text{NC}}$ | – | No connection | Leave open |
| 33 | $\overline{\text{NC}}$ | – | No connection | Leave open |
| 34 | VBUS | USB | USB VBUS monitor | GND |
| 35 | USBDM | USB | USB device port– | GND |
| 36 | USBDP | USB | USB device port + | GND |
| 37 | GND | P | Ground | – |
| 38 | VDD | P | Digital power | – |
| 39 | VDD | P | Digital power | – |
| 40 | VDDCORE | P | Core digital power | – |
| 41 | GND | P | Ground | – |
| 42 | $\overline{\text{NC}}$ | – | No connection | Leave open |
| 43 | $\overline{\text{NC}}$ | – | No connection | Leave open |
| 44 | $\overline{\text{NC}}$ | – | No connection | Leave open |
| 45 | SLV_CLK | O | Inter-chip signal | – |
| 46 | $\overline{\text{NC}}$ | – | No connection | Leave open |
| 47 | XIN | I | External crystal oscillator input | Leave open |
| 48 | XOUT | O | External crystal oscillator output | Leave open |
| 49 | $\overline{\text{NC}}$ | – | No connection | Leave open |
| 50 | $\overline{\text{NC}}$ | – | No connection | Leave open |
| 51 | SDA | OD | Serial interface data | – |
| 52 | SCL | OD | Serial interface clock | – |
| 53 | VDDIO | P | Digital power | – |
| 54 | GND | P | Ground | – |
| 55 | $\overline{\text{NC}}$ | – | No connection | Leave open |
| 56 | MOSI_C | O | Inter-chip signal | – |
| 57 | SS_C | O | Inter-chip signal | – |
| 58 | $\overline{\text{NC}}$ | – | No connection | Leave open |
| 59 | $\overline{\text{NC}}$ | – | No connection | Leave open |
| 60 | $\overline{\text{SCK_C}}$ | O | Inter-chip signal | – |

Table 2-1. Pin Listing mXT3432S-M-AT –100-pin TQFP (Continued)

| Pin | Name | Type | Comments | If Unused, Connect To... |
|-----|------------------------|------|-------------------|--------------------------|
| 61 | $\overline{\text{NC}}$ | – | No connection | Leave open |
| 62 | SCK_A | O | Inter-chip signal | – |
| 63 | $\overline{\text{NC}}$ | – | No connection | Leave open |
| 64 | $\overline{\text{NC}}$ | – | No connection | Leave open |
| 65 | $\overline{\text{NC}}$ | – | No connection | Leave open |
| 66 | $\overline{\text{NC}}$ | – | No connection | Leave open |
| 67 | $\overline{\text{NC}}$ | – | No connection | Leave open |
| 68 | $\overline{\text{NC}}$ | – | No connection | Leave open |
| 69 | $\overline{\text{NC}}$ | – | No connection | Leave open |
| 70 | $\overline{\text{NC}}$ | – | No connection | Leave open |
| 71 | $\overline{\text{NC}}$ | – | No connection | Leave open |
| 72 | $\overline{\text{NC}}$ | – | No connection | Leave open |
| 73 | $\overline{\text{NC}}$ | – | No connection | Leave open |
| 74 | MOSI_D | O | Inter-chip signal | – |
| 75 | $\overline{\text{NC}}$ | – | No connection | Leave open |
| 76 | SCK_D | O | Inter-chip signal | – |
| 77 | SS_D | O | Inter-chip signal | – |
| 78 | $\overline{\text{NC}}$ | – | No connection | Leave open |
| 79 | $\overline{\text{NC}}$ | – | No connection | Leave open |
| 80 | $\overline{\text{NC}}$ | – | No connection | Leave open |
| 81 | $\overline{\text{NC}}$ | – | No connection | Leave open |
| 82 | VDDIO | P | Digital power | – |
| 83 | GND | P | Ground | – |
| 84 | MISO_B | I | Inter-chip signal | – |
| 85 | MOSI_B | O | Inter-chip signal | – |
| 86 | SS_B | O | Inter-chip signal | – |
| 87 | $\overline{\text{NC}}$ | – | No connection | Leave open |
| 88 | $\overline{\text{NC}}$ | – | No connection | Leave open |
| 89 | $\overline{\text{NC}}$ | – | No connection | Leave open |
| 90 | $\overline{\text{NC}}$ | – | No connection | Leave open |
| 91 | $\overline{\text{NC}}$ | – | No connection | Leave open |
| 92 | MISO_A | I | Inter-chip signal | – |
| 93 | MOSI_A | O | Inter-chip signal | – |

2.2.2 Slaves mXT3432S-S-AT – 144-pin LQFP

Table 2-2. Pin Listing mXT3432S-S-AT Slaves – 144-pin LQFP

| Pin | Name | | Type | Comments | If Unused, Connect To... |
|-----|---------|---------|------|------------------------|--------------------------|
| | Slave A | Slave B | | | |
| 1 | NC | | – | No connection | Leave open |
| 2 | NC | | – | No connection | Leave open |
| 3 | NC | | – | No connection | Leave open |
| 4 | Y46 | Y47 | I | Y sense line | Leave open |
| 5 | Y48 | Y49 | I | Y sense line | Leave open |
| 6 | Y50 | Y51 | I | Y sense line | Leave open |
| 7 | Y52 | Y53 | I | Y sense line | Leave open |
| 8 | Y54 | Y55 | I | Y sense line | Leave open |
| 9 | Y56 | Y57 | I | Y sense line | Leave open |
| 10 | Y58 | Y59 | I | Y sense line | Leave open |
| 11 | Y60 | Y61 | I | Y sense line | Leave open |
| 12 | Y62 | Y63 | I | Y sense line | Leave open |
| 13 | AVDD | | P | Analog power | – |
| 14 | GND | | P | Ground | – |
| 15 | Y64 | Y65 | I | Y sense line | Leave open |
| 16 | Y66 | Y67 | I | Y sense line | Leave open |
| 17 | Y68 | Y69 | I | Y sense line | Leave open |
| 18 | Y70 | Y71 | I | Y sense line | Leave open |
| 19 | Y72 | Y73 | I | Y sense line | Leave open |
| 20 | Y74 | Y75 | I | Y sense line | Leave open |
| 21 | Y76 | Y77 | I | Y sense line | Leave open |
| 22 | NC | | – | No connection | Leave open |
| 23 | AVDD | | P | Analog power | – |
| 24 | GND | | – | No connection | Leave open |
| 25 | NC | | – | No connection | Leave open |
| 26 | NC | | – | No connection | Leave open |
| 27 | NC | | – | No connection | Leave open |
| 28 | SYNC5 | | I | Inter-slave connection | – |
| 29 | SYNC4 | | I | Inter-slave connection | – |
| 30 | SYNC3 | | I | Inter-slave connection | – |

Table 2-2. Pin Listing mXT3432S-S-AT Slaves – 144-pin LQFP (Continued)

| Pin | Name | | Type | Comments | If Unused, Connect To... |
|-----|---------------------------|---------|------|----------------------------------|--------------------------|
| | Slave A | Slave B | | | |
| 31 | SYNC2 | | I | Inter-slave connection | – |
| 32 | SYNC1 | | I | Inter-slave connection | – |
| 33 | SYNC0 | | I | Inter-slave connection | – |
| 34 | NC | | – | No connection | Leave open |
| 35 | NC | | – | No connection | Leave open |
| 36 | NC | | – | No connection | Leave open |
| 37 | X24 | X56 | O | X drive line | Leave open |
| 38 | X25 | X57 | O | X drive line | Leave open |
| 39 | X26 | X58 | O | X drive line | Leave open |
| 40 | X27 | X59 | O | X drive line | Leave open |
| 41 | XVDD | | P | X line drive voltage | – |
| 42 | GND | | P | Ground | – |
| 43 | X28 | X60 | O | X drive line | Leave open |
| 44 | X29 | X61 | O | X drive line | Leave open |
| 45 | X30 | X62 | O | X drive line | Leave open |
| 46 | X31 | X63 | O | X drive line | Leave open |
| 47 | $\overline{\text{RESET}}$ | | I | Inter-chip signal | – |
| 48 | GND | | P | Ground | – |
| 49 | NC | | – | No connection | Leave open |
| 50 | SCK_C | SCK_D | I | Inter-chip signal | – |
| 51 | SS_C | SS_D | I | Inter-chip signal | – |
| 52 | SS_A | SS_B | I | Inter-chip signal | – |
| 53 | SCK_A | SCK_B | I | Inter-chip signal | – |
| 54 | VDD | | P | Digital power | – |
| 55 | VDDDCORE | | P | Core digital power | – |
| 56 | GND | | P | Ground | – |
| 57 | SLV_CLK | | I | Inter-chip signal | – |
| 58 | REQ_A | REQ_B | I | Inter-chip signal | – |
| 59 | VDD | | P | Digital power | – |
| 60 | MISO_A | MISO_B | O | Inter-chip signal | – |
| 61 | MOSI_A | MOSI_B | I | Inter-chip signal | – |
| 62 | VDD_INPUT | | I | Inter-chip; for factory use only | – |

Table 2-2. Pin Listing mXT3432S-S-AT Slaves – 144-pin LQFP (Continued)

| Pin | Name | | Type | Comments | If Unused, Connect To... |
|-----|---------|---------|------|----------------------|--------------------------|
| | Slave A | Slave B | | | |
| 63 | X15 | X47 | O | X drive line | Leave open |
| 64 | X14 | X46 | O | X drive line | Leave open |
| 65 | X13 | X45 | O | X drive line | Leave open |
| 66 | X12 | X44 | O | X drive line | Leave open |
| 67 | GND | | P | Ground | – |
| 68 | XVDD | | P | X line drive voltage | – |
| 69 | X11 | X43 | O | X drive line | Leave open |
| 70 | X10 | X42 | O | X drive line | Leave open |
| 71 | X9 | X41 | O | X drive line | Leave open |
| 72 | X8 | X40 | O | X drive line | Leave open |
| 73 | NC | | – | No connection | Leave open |
| 74 | NC | | – | No connection | Leave open |
| 75 | NC | | – | No connection | Leave open |
| 76 | NC | | – | No connection | Leave open |
| 77 | NC | | – | No connection | Leave open |
| 78 | Y44 | Y45 | I | Y sense line | Leave open |
| 79 | Y42 | Y43 | I | Y sense line | Leave open |
| 80 | Y40 | Y41 | I | Y sense line | Leave open |
| 81 | Y38 | Y39 | I | Y sense line | Leave open |
| 82 | NC | | – | No connection | Leave open |
| 83 | Y36 | Y37 | I | Y sense line | Leave open |
| 84 | Y34 | Y35 | I | Y sense line | Leave open |
| 85 | GND | | P | Ground | – |
| 86 | AVDD | | P | Analog power | – |
| 87 | Y32 | Y33 | I | Y sense line | Leave open |
| 88 | Y30 | Y31 | I | Y sense line | Leave open |
| 89 | Y28 | Y29 | I | Y sense line | Leave open |
| 90 | Y26 | Y27 | I | Y sense line | Leave open |
| 91 | Y24 | Y25 | I | Y sense line | Leave open |
| 92 | Y22 | Y23 | I | Y sense line | Leave open |
| 93 | Y20 | Y21 | I | Y sense line | Leave open |
| 94 | Y18 | Y19 | I | Y sense line | Leave open |

Table 2-2. Pin Listing mXT3432S-S-AT Slaves – 144-pin LQFP (Continued)

| Pin | Name | | Type | Comments | If Unused, Connect To... |
|-----|----------|---------|------|-------------------------|--------------------------|
| | Slave A | Slave B | | | |
| 95 | GND | | P | Ground | – |
| 96 | AVDD | | P | Analog power | – |
| 97 | Y16 | Y17 | I | Y sense line | Leave open |
| 98 | Y14 | Y15 | I | Y sense line | Leave open |
| 99 | Y12 | Y13 | I | Y sense line | Leave open |
| 100 | Y10 | Y11 | I | Y sense line | Leave open |
| 101 | Y8 | Y9 | I | Y sense line | Leave open |
| 102 | Y6 | Y7 | I | Y sense line | Leave open |
| 103 | Y4 | Y5 | I | Y sense line | Leave open |
| 104 | Y2 | Y3 | I | Y sense line | Leave open |
| 105 | Y0 | Y1 | I | Y sense line | Leave open |
| 106 | NC | | – | No connection | Leave open |
| 107 | NC | | – | No connection | Leave open |
| 108 | NC | | – | No connection | Leave open |
| 109 | X0 | X32 | O | X drive line | Leave open |
| 110 | X1 | X33 | O | X drive line | Leave open |
| 111 | X2 | X34 | O | X drive line | Leave open |
| 112 | X3 | X35 | O | X drive line | Leave open |
| 113 | XVDD | | P | X line drive voltage | – |
| 114 | GND | | P | Ground | – |
| 115 | X4 | X36 | O | X drive line | Leave open |
| 116 | X5 | X37 | O | X drive line | Leave open |
| 117 | X6 | X38 | O | X drive line | Leave open |
| 118 | X7 | X39 | O | X drive line | Leave open |
| 119 | GND | | P | Ground | – |
| 120 | RESERVED | | – | Reserved for future use | – |
| 121 | MOSI_C | MOSI_D | O | Inter-chip signal | – |
| 122 | VDD | | P | Digital power | – |
| 123 | BUSY_A | BUSY_B | O | Inter-chip signal | – |
| 124 | CTE2A | CTE2B | I/O | Inter-slave connection | – |
| 125 | CTE3A | CTE3B | I/O | Inter-slave connection | – |
| 126 | NC | | – | No connection | Leave open |

Table 2-2. Pin Listing mXT3432S-S-AT Slaves – 144-pin LQFP (Continued)

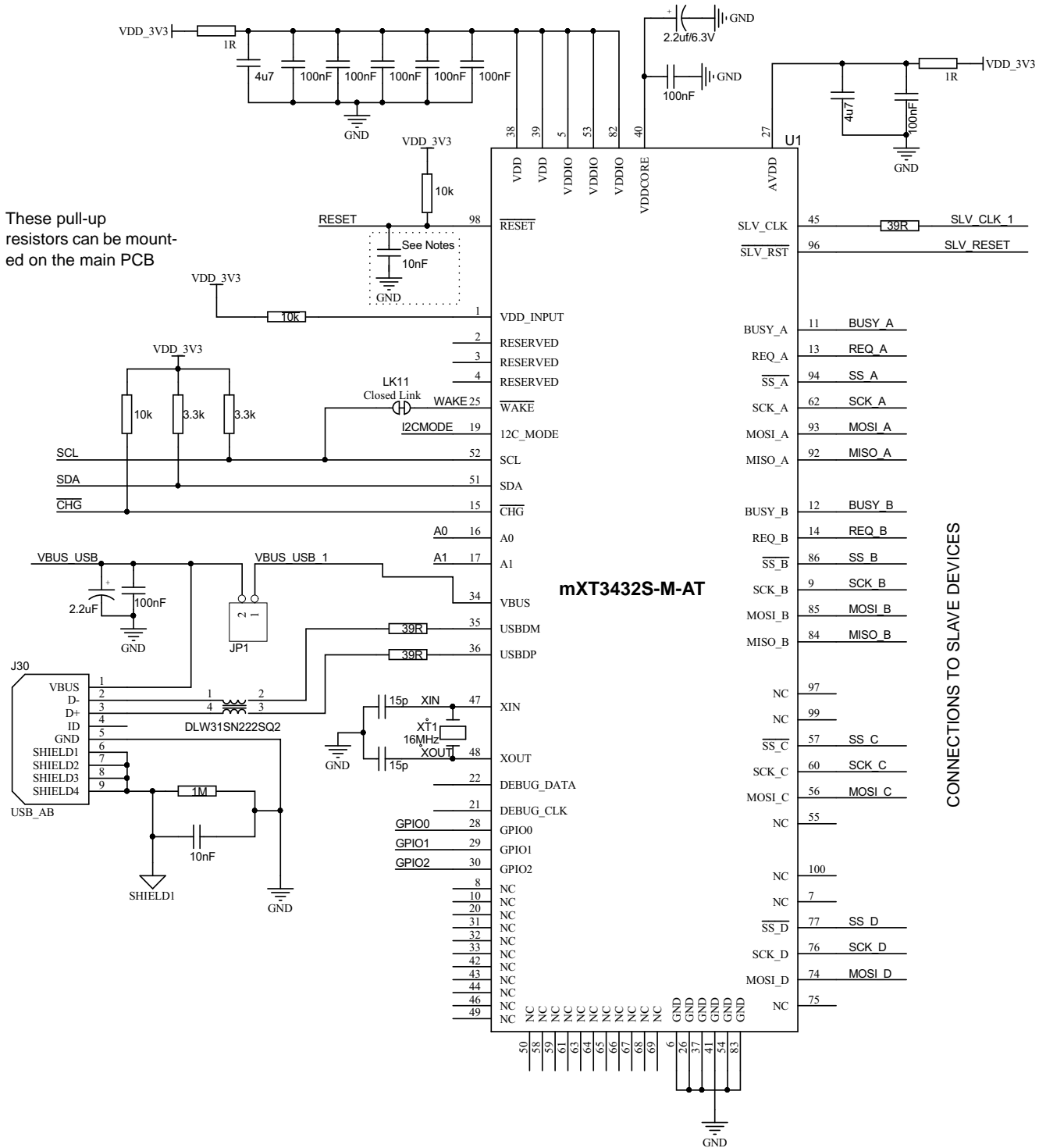
| Pin | Name | | Type | Comments | If Unused, Connect To... |
|-----|----------|---------|------|-------------------------|--------------------------|
| | Slave A | Slave B | | | |
| 127 | NC | | – | No connection | Leave open |
| 128 | NC | | – | No connection | Leave open |
| 129 | RESERVED | | – | Reserved for future use | – |
| 130 | RESERVED | | – | Reserved for future use | – |
| 131 | GND | | P | Ground | – |
| 132 | CTE1A | CTE1B | I/O | Inter-slave connection | – |
| 133 | CTE0A | CTE0B | I/O | Inter-slave connection | – |
| 134 | VDD | | P | Digital power | – |
| 135 | X23 | X55 | O | X drive line | Leave open |
| 136 | X22 | X54 | O | X drive line | Leave open |
| 137 | X21 | X53 | O | X drive line | Leave open |
| 138 | X20 | X52 | O | X drive line | Leave open |
| 139 | GND | | P | Ground | – |
| 140 | XVDD | | P | X line drive voltage | – |
| 141 | X19 | X51 | O | X drive line | Leave open |
| 142 | X18 | X50 | O | X drive line | Leave open |
| 143 | X17 | X49 | O | X drive line | Leave open |
| 144 | X16 | X48 | O | X drive line | Leave open |

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

- Notes:
1. Inter-chip connections are used for signals between the master and slave devices
 2. Inter-slave connections are used for signals between the two slave devices

2.3 Schematics

2.3.1 Master Device (mXT3432S-M-AT) – 100-pin TQFP



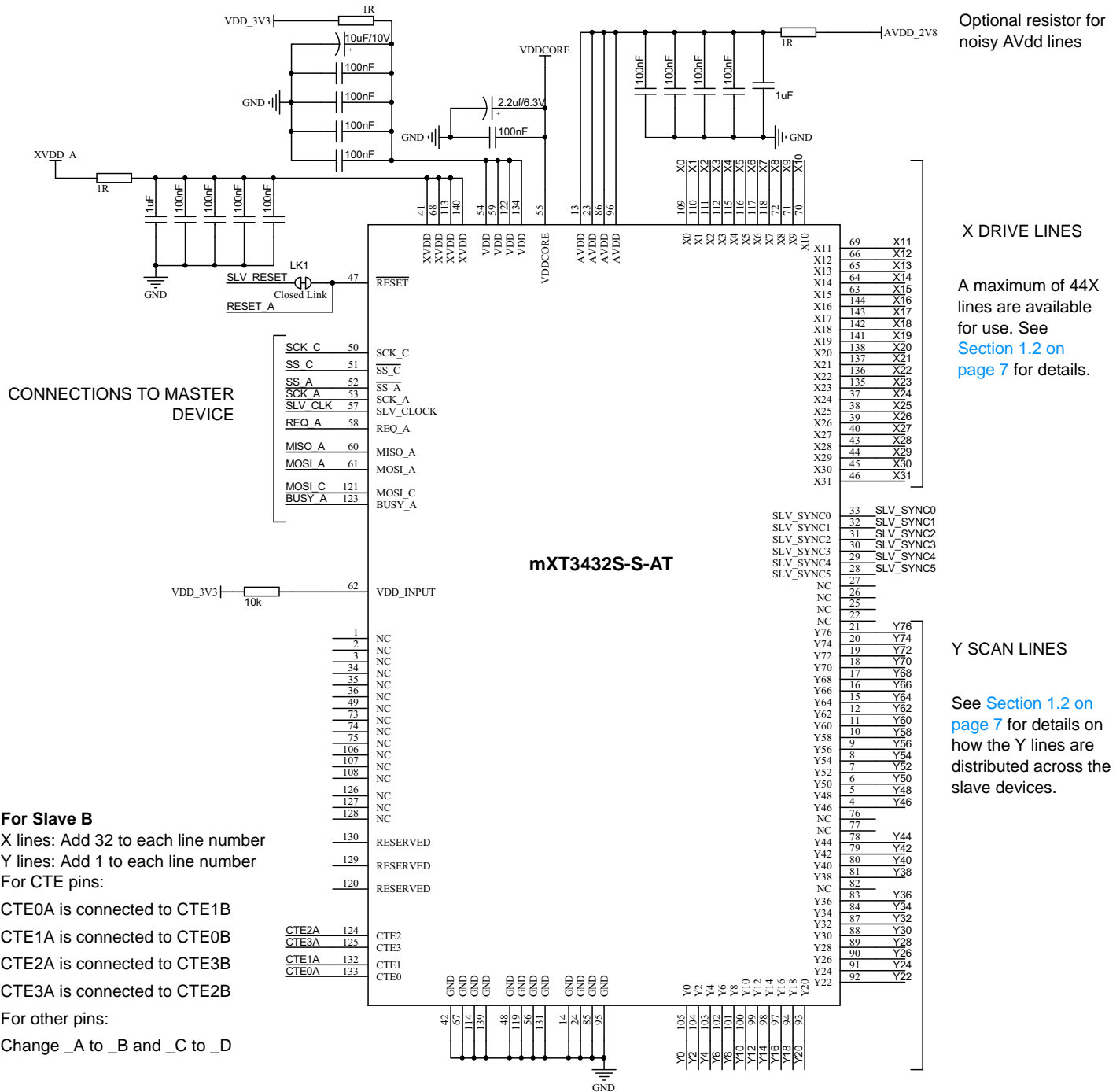
CONNECTIONS TO SLAVE DEVICES

- Notes:
1. Decoupling capacitors must be X7R or X5R.
 2. The RESET line is shown on the schematics with a 10 nF capacitor to ground. The capacitor is optional but may help if ESD issues are encountered.

2.3.2 Slave Devices

2.3.2.1 Slave Devices (2 x mXT3432S-S-AT) – 144-pin LQFP

- Notes:
1. Slave A only Shown; Slave B is omitted for simplicity.
 2. Bypass capacitors must be X7R or X5R.



For Slave B

X lines: Add 32 to each line number

Y lines: Add 1 to each line number

For CTE pins:

CTE0A is connected to CTE1B

CTE1A is connected to CTE0B

CTE2A is connected to CTE3B

CTE3A is connected to CTE2B

For other pins:

Change _A to _B and _C to _D

3. Touchscreen Basics

3.1 Sensor Construction

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

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

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

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

3.2 Electrode Configuration

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

The chipset supports various configurations of electrodes as summarized below:

Touchscreens: 1 Touchscreens allowed
 3X x 3Y minimum (depends on screen resolution)
 44X x 78Y maximum (subject to other configurations)
Note: The origin of the touchscreen must be at X0,Y0.

3.3 Scanning Sequence

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

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

3.4 Touchscreen Sensitivity

3.4.1 Adjustment

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

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

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 3.0 mm, and glass up to about 2.5 mm (dependent upon the screen size and layout). The thicker the front panel, the lower the signal-to-noise ratio of the measured capacitive changes and hence the lower the resolution of the touchscreen. In general, glass front panels are near optimal because they conduct electric fields almost twice as easily as plastic panels.

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

4. Detailed Operation

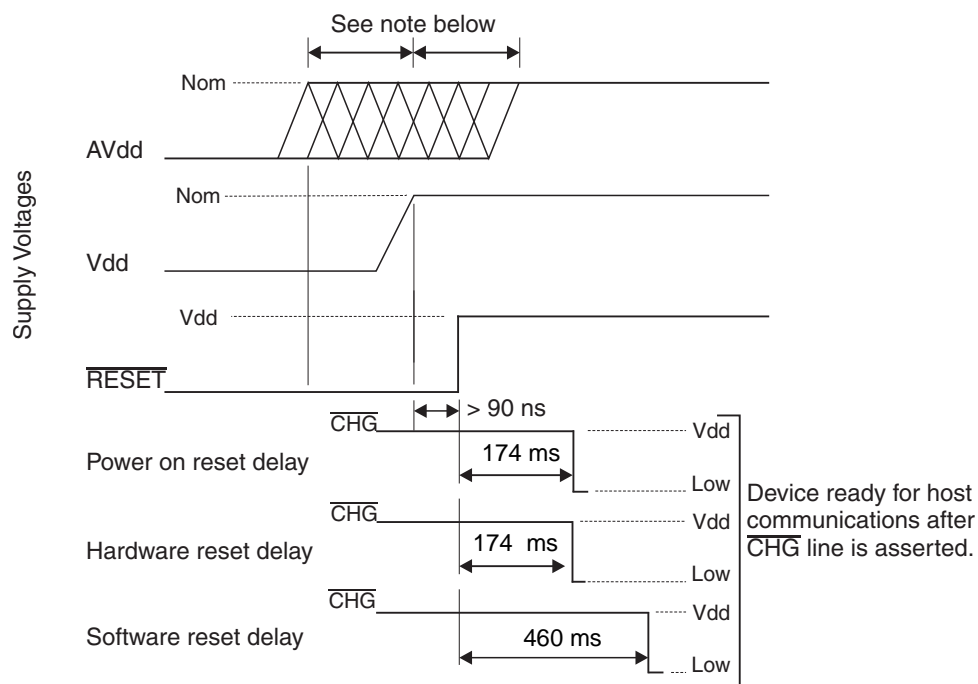
4.1 Power-up/Reset

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

The device must be held in $\overline{\text{RESET}}$ (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 $\overline{\text{RESET}}$ signal is de-asserted (that is, goes high). This is shown in Figure 4-1. See Section 9.2 on page 61 for nominal values for Vdd, AVdd and XVdd.

Note: The XVdd rail has a maximum rate of rise specification (see Section 9.3.3 on page 62), that is, a soft-start XVdd supply must be used.

Figure 4-1. Power Sequencing on the mXT3432S-AT

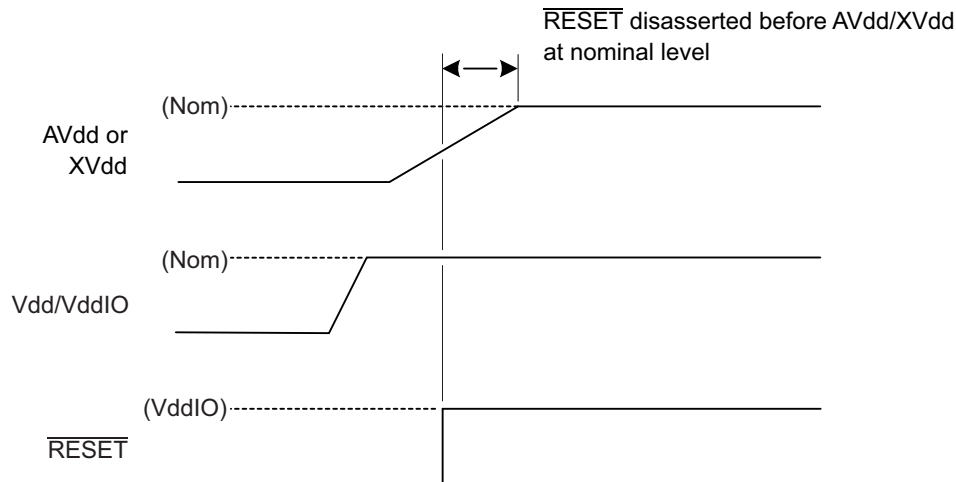


Note: Vdd and Avdd can be powered up in either order.
There is no prerequisite for the length of time between Vdd and Avdd powering up.

The digital or analog (AVdd) supplies can be applied independently and in any order on the mXT3432S-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 k Ω 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 <174 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 3. on page 22 for further specifications.

If the $\overline{\text{RESET}}$ 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 mXT3432S-AT – Late rise on AVDD

The $\overline{\text{RESET}}$ pin can be used to reset the device whenever necessary. The $\overline{\text{RESET}}$ pin must be asserted low for at least 90 ns to cause a reset. After releasing the $\overline{\text{RESET}}$ pin the device takes ~174 ms before it is ready to start communications. It is recommended to connect the $\overline{\text{RESET}}$ pin to a host controller to allow it to initiate a full hardware reset without requiring a power-down.

Note that the voltage level on the $\overline{\text{RESET}}$ 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 *mXT3432S-AT Protocol Guide*). A software reset takes 460 ms. After the chip has finished it asserts the $\overline{\text{CHG}}$ line to signal to the host that a message is available. The reset flag is set in the Message Processor object to indicate to the host that it has just completed a reset cycle. This bit can be used by the host to detect any unexpected brownout events. This allows the host 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 *mXT3432S-AT Protocol Guide* for more information).

Note that the $\overline{\text{CHG}}$ line is briefly set as an input during power-up or reset. It is therefore particularly important that the line should be allowed to float high via the $\overline{\text{CHG}}$ line pull-up resistor during this period. It should not be driven by the host. See [Section 9.7.3 on page 70](#).

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 *mXT3432S-AT Protocol Guide* for more details about this process.

4.2 Calibration

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

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

OR

- The channel is already enabled and one of the following applies:
 - The channel is held in detect for longer than the Touch Automatic Calibration setting (refer to the *mXT3432S-AT Protocol Guide* for more information on TCHAUTOCAL setting in the Acquisition Configuration object).
 - The signal delta on a channel is at least the touch threshold (TCHTHR) in the anti-touch direction, while no other touches are present on the channel matrix (refer to the *mXT3432S-AT Protocol Guide* for more information on the TCHTHR field in the Multiple Touch Touchscreen object).

- 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 chipset performs a global calibration; that is, all the channels are calibrated together.

4.3 Operational Modes

The chipset 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 *mXT3432S-AT Protocol Guide* for full information on how these modes operate, and how to use the settings provided.

4.4 Touchscreen Layout

The mXT3432S-AT support one Multiple Touch Touchscreen T9 object. Refer to the *mXT3432S-AT Protocol Guide* for more information on configuring the Multiple Touch Touchscreen T9 object.

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

- Each touch object should be in rectangular shape in terms of the lines it uses.
- The design of the touch object does not physically need to be on a strict XY grid pattern.

4.5 Signal Processing

4.5.1 Detection Integrator

The chipset 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 object (Multiple Touch Touchscreen T9). Refer to the *mXT3432S-AT Protocol Guide* for more information.

4.5.2 Digital Filtering and Noise Suppression

The mXT3432S-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's 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 *mXT3432S-AT Protocol Guide* for more information on the maXCharger T62 object.

4.5.3 Gestures

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

Gestures are configured using the One-touch Gesture Processor and the Two-touch Gesture Processor objects. Refer to the *mXT3432S-AT Protocol Guide* for more information on gestures and their configuration.

4.5.4 GPIO Pins

The mXT3432S-AT has three GPIO pins. This can be set to be either an input or an output pin, as required. The GPIO pin is configured using the GPIO/PWM Configuration T19 object.

4.5.5 Grip Suppression

The mXT3432S-AT 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. . Refer to the *mXT3432S-AT Protocol Guide* for more information.

4.5.6 Baseline References

The mXT3432S-AT supports using a set of known-good baseline 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 *mXT3432S-AT Protocol Guide* for more information.

4.5.7 Lens Bending

The mXT3432S-AT 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 *mXT3432S-AT Protocol Guide* for more information.

4.5.8 Shieldless Support

The mXT3432S-AT 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.

- The main noise suppression method for display noise is the noise canceller. The noise canceller measures the noise generated by the display and subtracts it from the noise cancellation feature measurement. When the noise canceller is enabled the maXCharger T62 object cannot use the Grass Cut filter.

Refer to the *mXT3432S-AT Protocol Guide* Protocol Guide for more information on the Shieldless T56 object.

4.5.9 Stylus Support

The mXT3432S-AT allows for the particular characteristics of stylus touches, whilst still allowing conventional finger touches to be detected. Stylus touches are configured by the Stylus T47 object.

The mXT3432S-AT also supports active stylus through the Active Stylus T63 object.

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

The touch sensitivity and threshold controls for stylus touches are configured separately from those for conventional finger touches so that both types of touches can be accommodated.

4.5.10 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. Refer to the *mXT3432S-AT Protocol Guide* for more information.

4.6 Circuit Components

4.6.1 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 9.2 on page 61](#).

4.6.2 Bypass Capacitors

Each power supply (Vdd, XVdd and AVdd) requires a 1 μ F bypass capacitor. If the internal 1.8V VDDCORE regulator is used, the Vdd 1 μ F should be replaced with a 10 μ F capacitor and a 2.2 μ F 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 20](#) 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 9.12 on page 71](#).

4.6.3 Supply Quality

While the chipset has good Power Supply Rejection Ratio properties, poorly regulated and/or noisy power can significantly reduce performance. See [Section 9.12 on page 71](#).

Always operate the chipset with a well-regulated and clean AVdd (and XVdd, if used) supply. It supplies the sensitive analog stages in the chipset. See [Figure A-1 on page 77](#) for an example XVdd supply.

4.6.4 Supply Sequencing

Vdd and AVdd can be powered independently of each other without damage to the chipset. Vdd must be applied to the device before XVdd to ensure proper initialization of the device. All voltage ranges should be used within the limits specified in [Section 9.2 on page 61](#).

Make sure that any lines connected to the chipset are below or equal to V_{dd} during power-up. For example, if $\overline{\text{RESET}}$ is supplied from a different power domain to the mXT3432S-AT VDD pin, make sure that it is held low when V_{dd} is off. If this is not done, the $\overline{\text{RESET}}$ signal could parasitically couple power via the mXT3432S-AT $\overline{\text{RESET}}$ pin into the V_{dd} supply.

4.6.5 Decoupling Requirements

See also the schematics in [Section 2.3 on page 20](#).

4.6.6 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 chipset devices, or to any associated traces or components, be sure that you fully understand the nature of the flux used during the rework process. Leakage currents from hygroscopic ionic residues can stop capacitive sensors from functioning. If you have any doubts, a thorough cleaning after rework may be the only safe option.

4.7 PCB Layout

See [Appendix A. on page 76](#) for general advice on PCB layout.

4.8 Debugging

The chipset provides a mechanism for obtaining raw data for development and testing purposes by reading data from the Diagnostic Debug T37 object. Refer to the *mXT3432S-AT Protocol Guide* for more information on this object.

A second mechanism is provided that allows the host to read the real-time raw data using the low-level debug port. This can be accessed via the SPI interface or the USB interface. Note that if both the (HID) I²C-compatible and the USB interfaces are connected to the host at the same time, the debug data is output on the USB interface. Refer to QTAN0050, *Using the maXTouch Debug Port*, for more information on the debug port.

There is also a Self Test T25 object that runs self-test routines in the mXT3432S-AT 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 *mXT3432S-AT Protocol Guide* and QTAN0059, *Using the maXTouch Self Test Feature*, for more information on the Self Test T25 object.

4.9 Communications

Communication with the host is achieved using either the I²C-compatible interface (see [Section 5. on page 30](#)), the HID-I²C-compatible interface (see [Section 7. on page 49](#)) or the USB interface (see [Section 6. on page 37](#)). 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 I²C or the HID-I²C protocol of the I²C or the HID-I²C protocol of the I²C interface is determined by the I2CMODE pin.

The interface is selected using the VBUS pin. Connect the VBUS to V_{dd} to select the USB interface, or to GND to select one of the two I²C-compatible interfaces.

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 11](#) for details on what should be done with the unconnected pins.

4.10 Configuring the Chipset

The chipset has an object-based protocol that organizes the features of the chipset into objects that can be controlled individually. This is configured using the Object Protocol common to many of Atmel's touch sensor devices. For more information on the Object Protocol and its implementation on the chipset, refer to the *mXT3432S-AT Protocol Guide*.

5. I²C Communications

5.1 Communications Protocol

The chipset can use an I²C interface for communication. See [Appendix on page 80](#) for details of the I²C protocol. The I²C interface is used in conjunction with the CHG line. The CHG line going active signifies that a new data packet is available. This provides an interrupt-style interface and allows the chipset to present data packets when internal changes have occurred.

5.2 I²C-compatible Addresses

The chipset supports four I²C device addresses. These are selected at start-up using the A0 and A1 pins on the mXT3432S-M-AT master device (see [Table 5-1](#)). The address pins should be connected to GND to signal a logic 0, and either left open or connected to VDD_3v3 to signal a logic 1⁽¹⁾.

Table 5-1. I²C-compatible Device Addresses

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

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

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

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

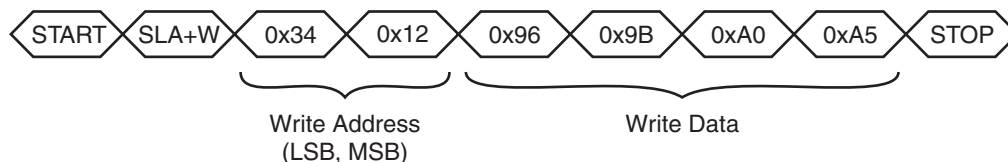
5.3 Writing To the Chipset

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

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

[Figure 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



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

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 chipset 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 Chipset

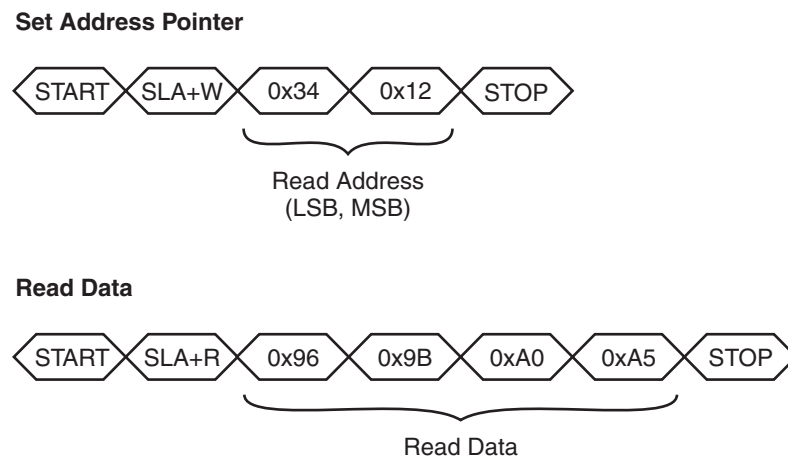
Two I²C-compatible bus activities must take place to read from the chipset. The first activity is an I²C-compatible write to set the address pointer (LSByte then MSByte). The second activity is the actual I²C-compatible read to receive the data. The address pointer returns to its starting value when 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 *mXT3432S-AT 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.⁽¹⁾

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

1. 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

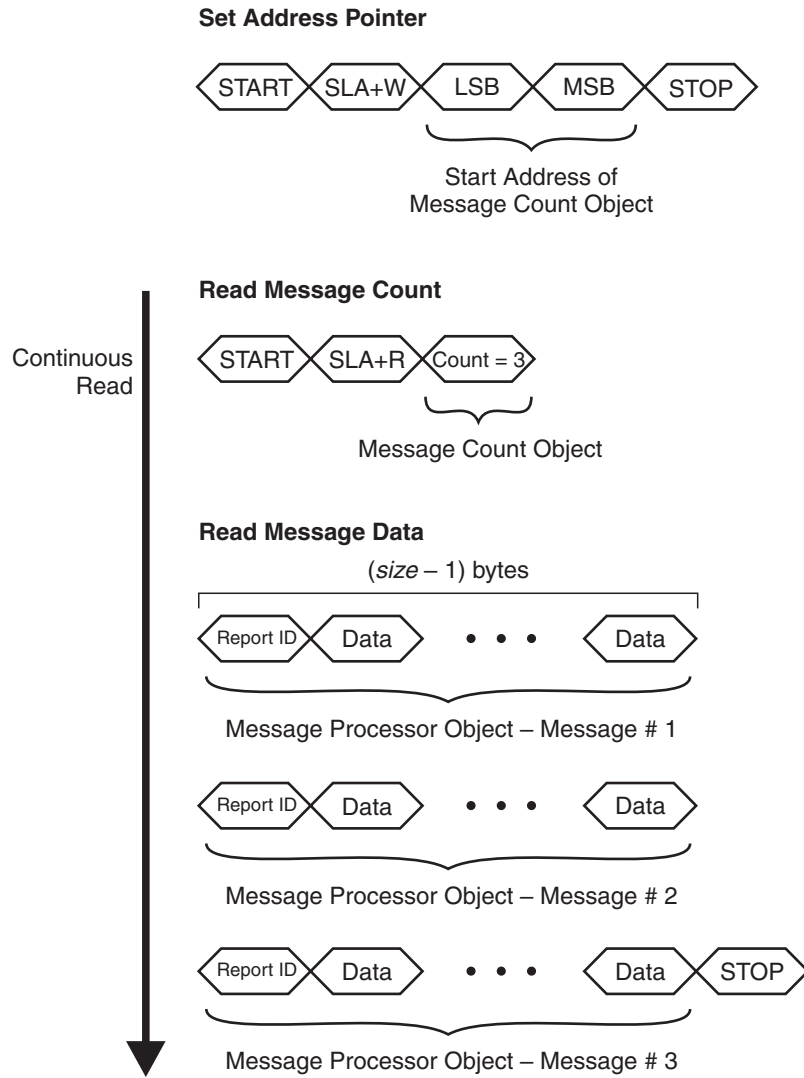
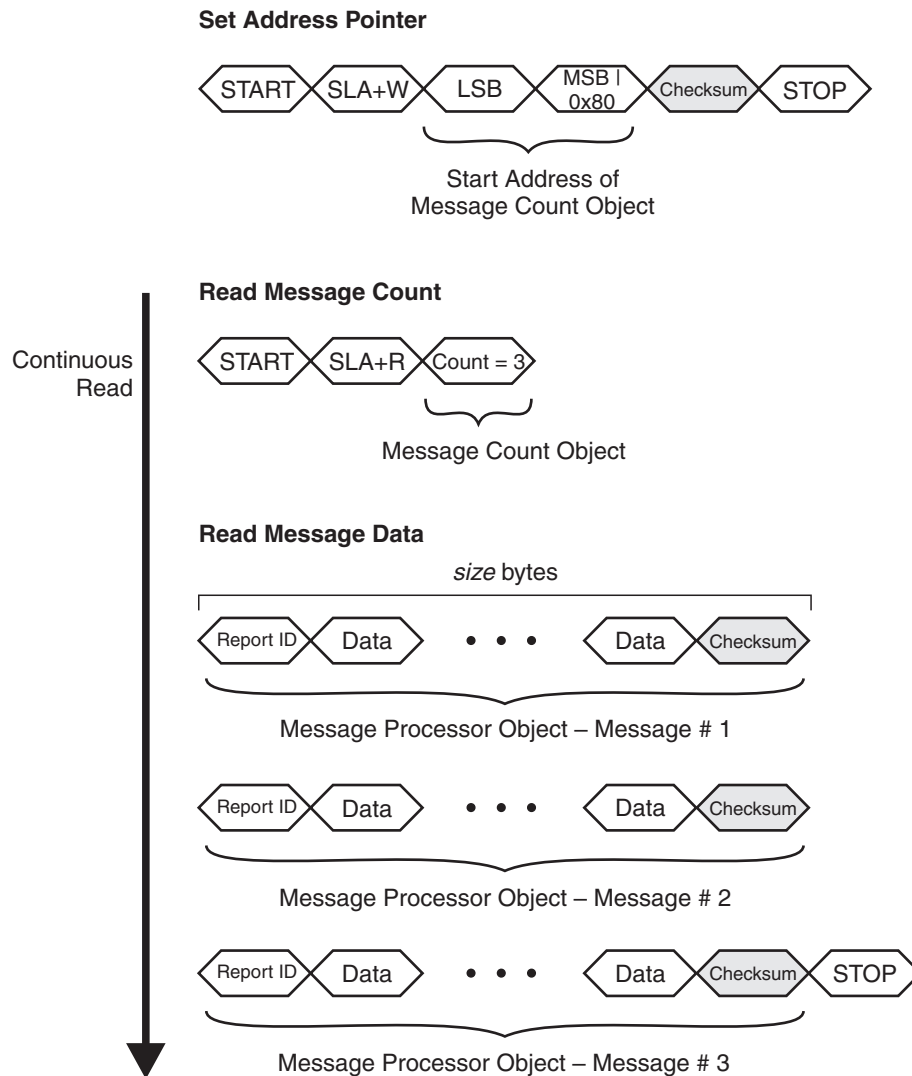


Figure 5-5. Continuous Message Read Example – I²C-compatible Checksum Mode

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 31](#).

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

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

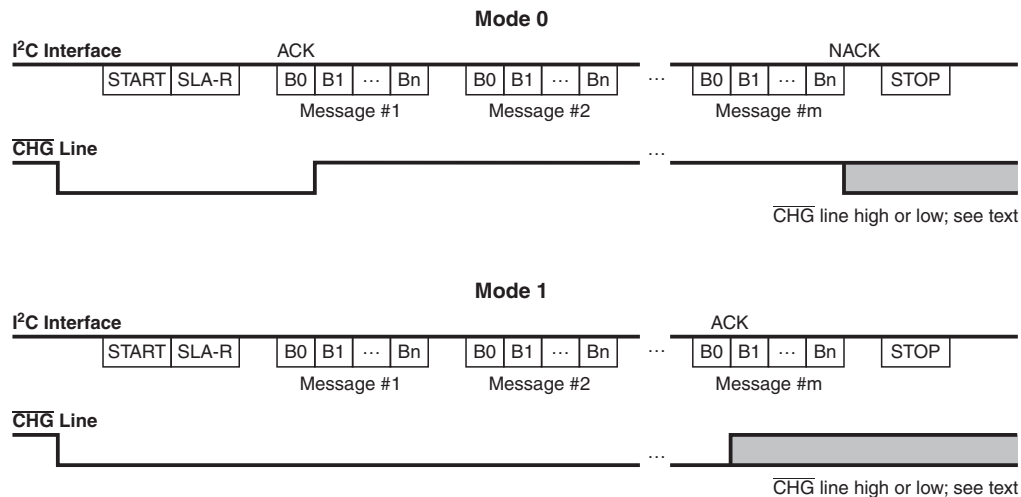
The $\overline{\text{CHG}}$ line is an active-low, open-drain output that is used to alert the host that a new message is available in the Message Processor T5 object. This provides the host with an interrupt-style interface with the potential for fast response times. It reduces the need for wasteful 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 24](#)).

A pull-up resistor is required, typically 10 k Ω to V_{dd}.

The $\overline{\text{CHG}}$ line operates in two modes, as defined by the Communications Configuration T18 object (refer to the *mXT3432S-AT 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 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 chipset.

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 *mXT3432S-AT Protocol Guide* for more information.

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

The $\overline{\text{WAKE}}$ line is an active-low input that is used to wake the mXT3432S-AT up from deep sleep mode before communicating with it via the I²C-compatible interface. It can be used to minimize current consumption when the mXT3432S-AT is in deep sleep mode. Refer to the *mXT3432S-AT Protocol Guide* for information on deep sleep mode. Note that the $\overline{\text{WAKE}}$ line is not used when the mXT3432S-AT is not in deep sleep mode.

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

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

The mXT3432S-AT is ready to accept I²C-compatible communications 25 ms after the $\overline{\text{WAKE}}$ line is asserted. This means that if the $\overline{\text{WAKE}}$ line is connected to a GPIO line, the line must be asserted 25 ms before the host attempts to communicate with the mXT3432S-AT.

If the $\overline{\text{WAKE}}$ line is connected to the SCL pin, the mXT3432S-AT will send a NACK on the first attempt to address it; the host must then retry 25 ms later.

The mXT3432S-AT remains ready to accept I²C-compatible communications for 2 seconds after the $\overline{\text{WAKE}}$ line is asserted, after which time the chip will timeout and return to deep sleep mode. This timeout period is reset every time there is an I²C-compatible communication with the mXT3432S-AT, or if the $\overline{\text{WAKE}}$ line is held asserted.

Note that when the mXT3432S-AT is sent into deep sleep mode, it goes to sleep immediately. In this case the two-second timeout does not apply until the $\overline{\text{WAKE}}$ pin is asserted.

In USB mode, (that is, when the I²C-compatible interface is not being used), the $\overline{\text{WAKE}}$ pin should be connected to Vdd.

5.9 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 (R_p) 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 9.8 on page 70](#)).

5.10 Clock Stretching

The chipset supports clock stretching in accordance with the I²C-compatible specification. It may also instigate a clock stretch if a communications event happens during a period when the chipset is busy internally.

The chipset 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 chipset, then any ongoing transfers with the chipset may be corrupted. The bus monitor is enabled or disabled using the Communications Configuration T18 object. Refer to the *mXT3432S-AT Protocol Guide* for more information.

6. USB Communications

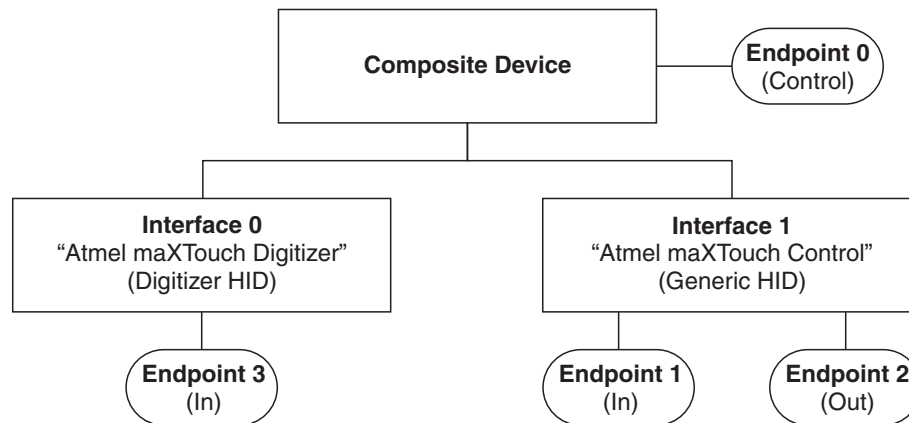
6.1 Communications Protocol

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

- **Interface 0** – This interface provides a Digitizer HID that supplies touch information to the Host for passing on to a PC’s operating system. This interface is supported by Microsoft® Windows® 8.1 without the need for additional software. The HID identifier string is “Atmel maXTouch Digitizer”.
- **Interface 1** – This interface provides a Generic HID that allows the host to communicate with the chipset using the Object Protocol. The HID identifier string is “Atmel maXTouch Control”.

The topography of the USB device is shown in [Figure 6-1](#).

Figure 6-1. USB Topography



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

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

6.2 Endpoint Addresses

The endpoint addresses are listed in the table below:

Table 6-1. Endpoint Addresses

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

6.3 Composite Device

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

Vendor ID: 0x03EB (Atmel)
 Product ID: 0x2136 (mXT3432S-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

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

6.4 Interface 0 (Digitizer HID)

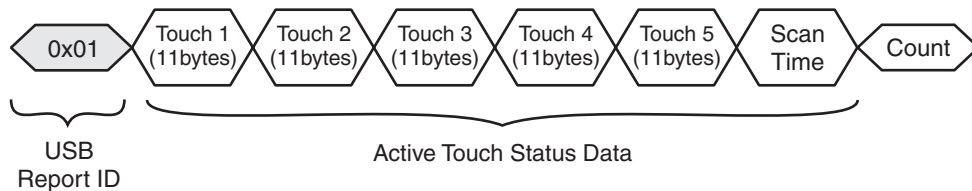
6.4.1 Normal Touch Report

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

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

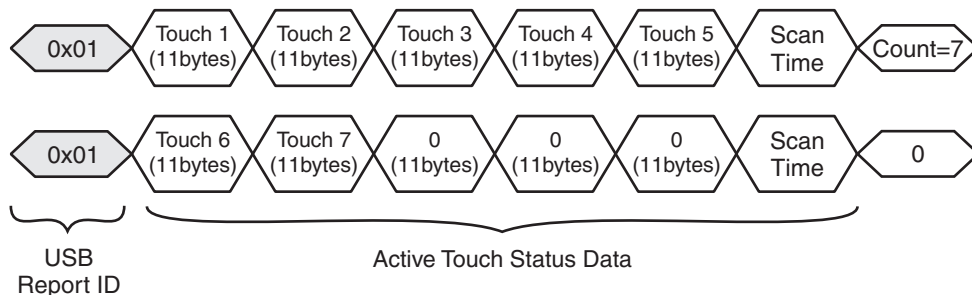
The format of an input report is shown in Figure 6-2. Each input report start with a USB Report ID ⁽²⁾ (value 0x01). This is followed by 6 sets of data (11 bytes each) that describe the status of up to 5 active touches. The input report is terminated by a single byte that contains the number of active touches.

Figure 6-2. Input Report Packet



Any unused touch data bytes are set to zero (for example, the data for one active touch would be followed by 56 zeroed bytes). If there are more than five active touches to be reported, a further input report is sent with the remaining touch data. In this case, the count (for all touches) is sent in the last count byte and the count byte in the first report is zero. An example of the input report packets for 7 active touches is shown in Figure 6-3 on page 38.

Figure 6-3. Example Input Report Packets for 7 Active Touches



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

The input report format depends on the geometry calculation field (TCHGEOMEN) of the Digitizer HID Configuration T43 object. [Table 6-2](#) and [Table 6-3](#) gives the detailed format of an input report packet..

Table 6-2. Input Report Format when TCHGEOMEN is Enabled

| Byte | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---------|---|-------|-------|-------|----------------------------------|----------|-------|--------|
| 0 | USB Report ID | | | | | | | |
| 1 | Touch ID (first touch) | | | | | Reserved | | Status |
| 2 | X Position LSByte (first touch) | | | | | | | |
| 3 | 0 | 0 | 0 | 0 | X Position MSBits (first touch) | | | |
| 4 | X' Position LSByte (first touch) | | | | | | | |
| 5 | 0 | 0 | 0 | 0 | X' Position MSBits (first touch) | | | |
| 6 | Y Position LSByte (first touch) | | | | | | | |
| 7 | 0 | 0 | 0 | 0 | Y Position MSBits (first touch) | | | |
| 8 | Y' Position LSByte (first touch) | | | | | | | |
| 9 | 0 | 0 | 0 | 0 | Y' Position MSBits (first touch) | | | |
| 10 | Touch width | | | | | | | |
| 11 | Touch height | | | | | | | |
| 12 – 22 | Touch data for second touch – same format as bytes 1 – 11 | | | | | | | |
| 23 – 33 | Touch data for third touch – same format as bytes 1 – 11 | | | | | | | |
| 34 – 44 | Touch data for fourth touch – same format as bytes 1 – 11 | | | | | | | |
| 45 – 55 | Touch data for fifth touch – same format as bytes 1 – 11 | | | | | | | |
| 56 – 57 | Scan time | | | | | | | |
| 58 | Contact count | | | | | | | |

Table 6-3. Input Report Format when TCHGEOMEN is Disabled

| Byte | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---------|---|-------|-------|-------|---------------------------------|----------|-------|--------|
| 0 | USB Report ID | | | | | | | |
| 1 | Touch ID (first touch) | | | | | Reserved | | Status |
| 2 | X Position LSByte (first touch) | | | | | | | |
| 3 | 0 | 0 | 0 | 0 | X Position MSBits (first touch) | | | |
| 4 – 5 | Reserved | | | | | | | |
| 6 | Y Position LSByte (first touch) | | | | | | | |
| 7 | 0 | 0 | 0 | 0 | Y Position MSBits (first touch) | | | |
| 8 – 9 | Reserved | | | | | | | |
| 10 – 11 | Reserved | | | | | | | |
| 12 – 22 | Touch data for second touch – same format as bytes 1 – 11 | | | | | | | |
| 23 – 33 | Touch data for third touch – same format as bytes 1 – 11 | | | | | | | |
| 34 – 44 | Touch data for fourth touch – same format as bytes 1 – 11 | | | | | | | |
| 45 – 55 | Touch data for fifth touch – same format as bytes 1 – 11 | | | | | | | |
| 56 – 57 | Scan time | | | | | | | |
| 58 | Contact count | | | | | | | |

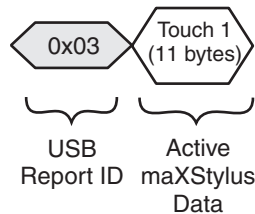
In [Table 6-3](#):

- Byte 1:
Status: 1 = In detect, 0 = Not in detect.
Touch ID: Identifies the touch for which this is a status report (starting from 1).
- Bytes 2 to 9:
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 4, 5, 8, 9, 10 and 11 are Reserved when TCHGEOMEN field is set to 0.
- Byte 10:
Touch Width: Reports the width of the detected touch.
- Byte 11:
Touch Height: Reports the height of the detected touch.
- Byte 56 to 57:
Scan Time: Timestamp associated with the current report frame with a 10 kHz resolution. Note that the scan time for each report packet of a single frame is same.
- Byte 58:
Contact Count: Non-zero value in the first report packet of a frame indicating the total number of report packets in the frame. Zeros in the subsequent report packets within the frame.

6.4.2 Active maXStylus Report

The format of an active maXStylus report is shown in [Figure 6-4](#). Each input report start with a USB Report ID⁽¹⁾ (value 0x03).

Figure 6-4. Active maXStylus Report Packet



[Table 6-4](#) gives the detailed format of an active stylus report packet

Table 6-4. Active maXStylus Report Format

| Byte | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|------|-------------------|-------|-------|----------|----------|--------|--------|-------|
| 0 | USB Report ID | | | | | | | |
| 1 | Reserved | | | In Range | Reserved | Eraser | Barrel | Tip |
| 2 | X Position | | | | | | | |
| 3 | | | | | | | | |
| 4 | X Center Position | | | | | | | |
| 5 | | | | | | | | |
| 6 | Y Position | | | | | | | |
| 7 | | | | | | | | |

1. The term USB Report ID should not be confused with the term Report Id as used in the Object Protocol; the two are entirely different concepts.

Table 6-4. Active maXStylus Report Format (Continued)

| Byte | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|------|-------------------|-------|-------|-------|-------|-------|-------|-------|
| 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 to 7:**
 Y position
- **Byte 8 to 9:**
 Y center position
- **Byte 10:**
Tip Pressure: Force exerted against the touch screen surface by the stylus.

There are two update conditions:

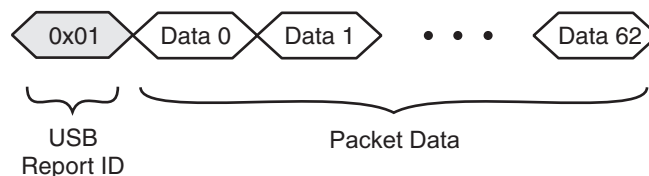
- **Change:** A change in status of any contact (touch) triggers a touch update message to be sent to the host.
- **Idle:** The idle delay of the Digitizer Interface may be controlled via the Control Endpoint as per the HID 1.11 specification (Set Idle command). By default this is set to a delay of 2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

6.5 Interface 1 (Generic HID)

Interface 1 is a Generic Human Interface Device, compliant with HID specification 1.11 with amendments.⁽¹⁾

It consists of two endpoints: an interrupt-in endpoint (Endpoint 1) and an interrupt-out endpoint (Endpoint 2). The data packet in each case contains a 1-byte USB Report ID followed by 63 bytes of data, totalling 64 bytes (see [Figure 6-5](#)).

Figure 6-5. Data Packet for Interface 1



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

The supported commands are as follows:

- Read/write Memory Map
- Send Auto-return messages

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

- Start debug monitoring
- End debug monitoring

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

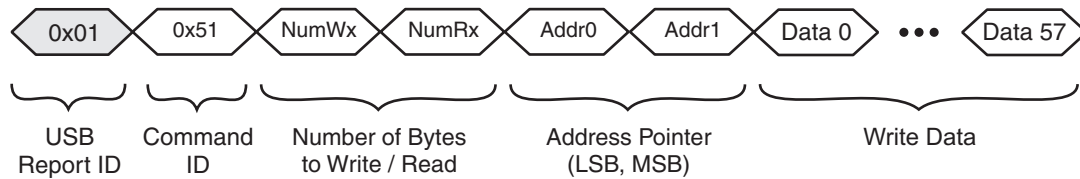
6.5.1 Read/Write Memory Map

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

The USB Report ID is 0x01.

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

Figure 6-6. Generic Command Packet Form

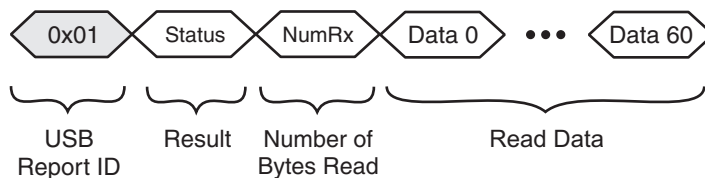


In Figure 6-6:

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

The response packet has the generic format given in Figure 6-7.

Figure 6-7. Response Packet Format



In Figure 6-7:

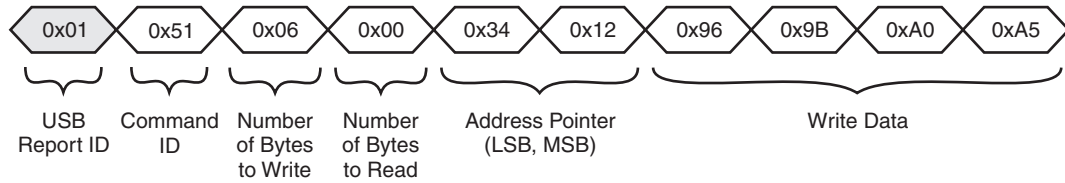
- **Status** indicates the result of the command:
 0x00 = read and write completed; read data returned
 0x04 = write completed; no read data requested
- **NumRx** is the number of bytes following that have been read from the memory map (in the case of a read). This will be the same value as NumRx in the command packet.
- **Data 0** to **Data 60** are the data bytes read from the memory map.

6.5.1.1 Writing To the Chipset

A write operation cycle to the chipset consists of sending a packet that contains six header bytes. These specify the USB report ID, the Command ID, the number of bytes to read, the number of bytes to write, and the 16-bit address pointer. Subsequent bytes in a multibyte transfer form the actual data. These are written to the location of the address pointer, location of the address pointer +1, location of the address pointer + 2, and so on.

Figure 6-8 shows an example command packet to write four bytes of data to contiguous addresses starting at 0x1234.

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

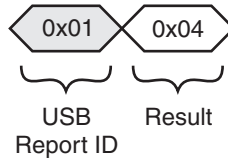


In Figure 6-8:

- 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 6-9 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 6-9. Response to Example Four-byte Write

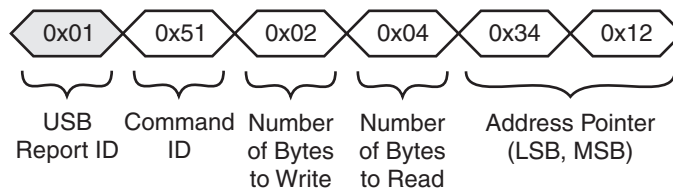


6.5.1.2 Reading From the Chipset

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

Figure 6-10 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.

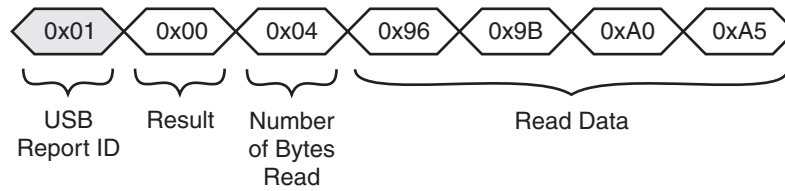
Figure 6-10. Example of a Four-byte Read Starting at Address 0x1234



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 6-11 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).

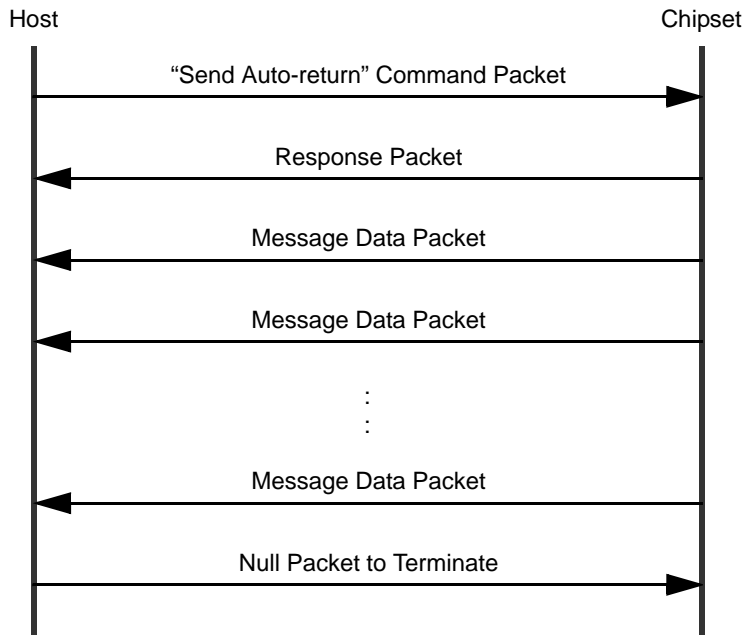
Figure 6-11. Response to Example Four-byte Read



6.5.2 Send Auto-return Messages

With this command the chipset can be configured to return new messages from the Message Processor T5 object autonomously. The packet sequence to do this is shown in [Figure 6-12](#).

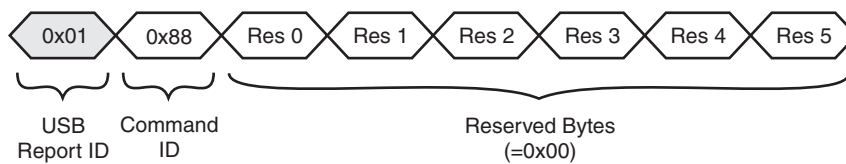
Figure 6-12. Packet Sequence for “Send Auto-return” Command.



The USB Report ID is 0x01.

The command packet has the format given in [Figure 6-13](#).

Figure 6-13. Command Packet Format

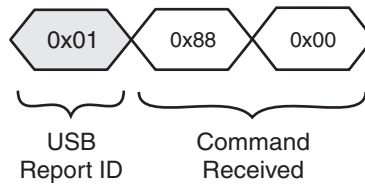


In [Figure 6-13](#):

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

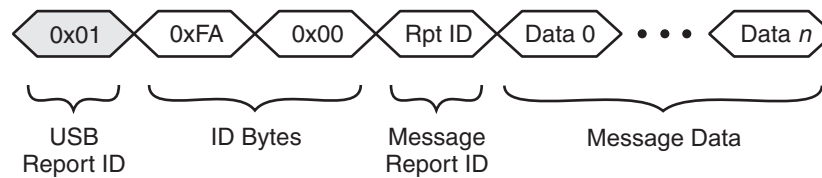
The response packet has the format given in [Figure 6-14](#). Note that with this command, the command packet does not include an address pointer as the chipset already knows the address of the Message Processor T5 object.

Figure 6-14. Response Packet Format



Once the chipset has responded to the command, it starts sending message data. Each time a message is generated in the Message Processor T5 object, the chipset automatically sends a message packet to the host with the data. The message packets have the format given in [Figure 6-15](#).

Figure 6-15. Message Packet Format

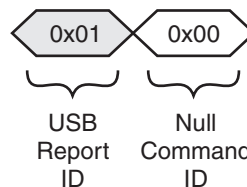


In [Figure 6-15](#):

- ID Bytes identify the packet as an auto-return message packet.
- **Rpt ID** is the Report ID returned by the Message Processor T5 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 *mXT3432S-AT 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 6-16](#)).

Figure 6-16. Null Command Packet Format



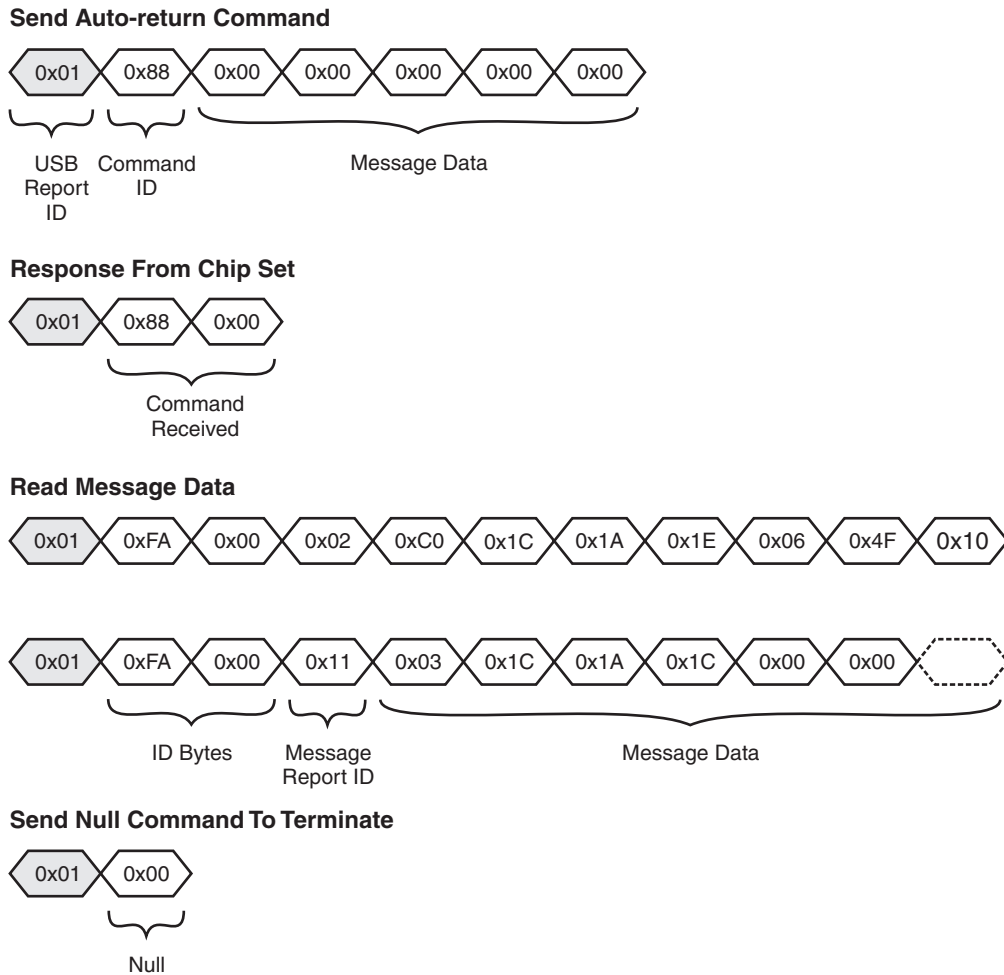
Note that the “Start Debug Monitoring” command may also terminate any currently enabled auto-return mode (see [Section 6.5.3](#)).

1. This is the Report ID used in the Object Protocol and should not be confused with the USB Report ID. Refer to the *mXT3432S-AT Protocol Guide* for more information on the use of Report IDs in the Object Protocol.

6.5.2.1 Reading Status Messages

Figure 6-17 shows an example sequence of packets to receive messages from the Message Processor T5 object using the “Send Auto-return” command.

Figure 6-17. Example Auto-return Command Packet



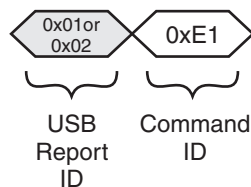
6.5.3 Start Debug Monitoring

This command instructs the device to return debug-monitoring data packets using the debug port, if this feature has been enabled in the Command Processor T6 object.

The USB Report ID can be either 0x01 or 0x02. This allows the source of the request to be identified. The main difference is that a USB Report ID of 0x01 will terminate any currently enabled auto-return mode (see Section 6.5.2 on page 44).

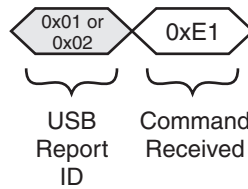
The command packet has the format given in Figure 6-18.

Figure 6-18. Command Packet Format



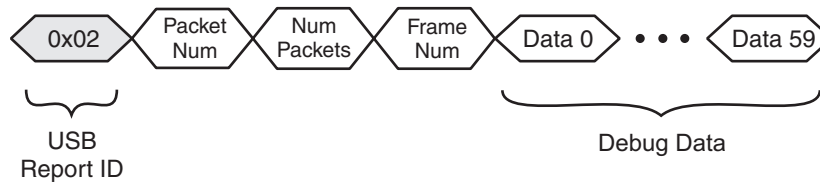
The response packet has the format given in [Figure 6-19](#). Note that the USB Report ID will be the same as that used in the command packet.

Figure 6-19. Response Packet Format



The debug data packet has the format given in [Figure 6-20](#).

Figure 6-20. Debug Data Packet Format



In [Figure 6-20](#):

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

6.5.4 Stop Debug monitoring

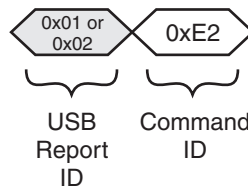
This command instructs the device to cease returning debug-monitoring data packets.

The command packet has the following format:

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

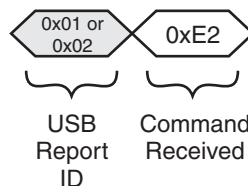
The command packet has the format given in [Figure 6-21](#).

Figure 6-21. Command Packet Format



The response packet has the format given in [Figure 6-22](#).

Figure 6-22. Response Packet Format



6.6 USB Suspend Mode

When the mXT3432S-AT is used in USB configuration, the USB “System Suspend” event can be used to minimize current consumption. Note that it is possible to put the mXT3432S-AT into deep sleep mode without also sending a “System Suspend” event on the USB bus, but the current consumption is not as low. The USB controller must send a USB “System Wake Up” event on the bus to bring the mXT3432S-AT out of suspend mode.

The mXT3432S-AT can also be configured to respond to USB “Remote Wakeup” requests. In this case, if the operating system enables remote wakeup and the mXT3432S-AT is suspended, the chipset will continue to scan at a preset sensor refresh rate. Use of the remote wake up feature and the sensor refresh rate are configured using the Digitizer HID Configuration T43 object (refer to the *mXT3432S-AT Protocol Guide* for more information).

7. HID-I²C-compatible Communications

7.1 Communications Protocol

The chipset 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 30](#).

7.2 I²C-compatible Addresses

See [Section 5.2 on page 30](#).

7.3 Device

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

Vendor ID: 0x03EB (Atmel)
 Product ID: 0x2136 (mXT3432S-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

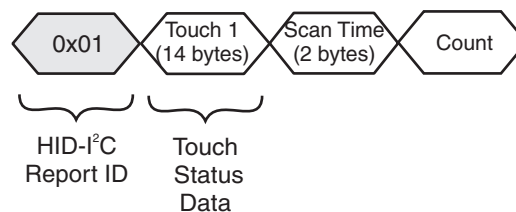
7.4 Interface 0 (Digitizer HID-I²C)

Interface 0 is a digitizer class HID.

7.4.1 Normal Touch Report

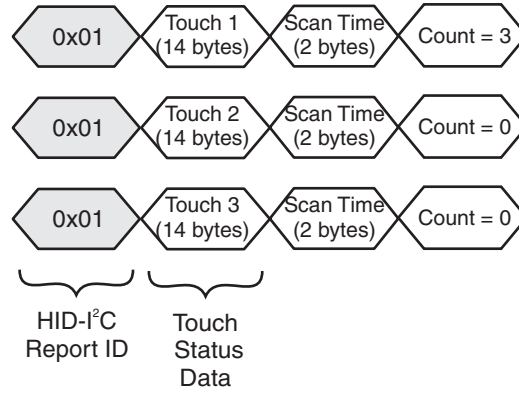
The format of an input report is shown in [Figure 7-1](#). Each input report starts with a report ID and each input report message report contains data of one touch.

Figure 7-1. Input Report Packet



An example of the input report packets for 3 active touches is shown in [Figure 7-3 on page 51](#).

Figure 7-2. Example Input Report Packets for 3 Active Touches



Each input report consists of a HID-I²C report ID followed by 17 bytes 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 7-1](#) and [Table 7-2](#) explain the detailed format of an input report packet.

Table 7-1. 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 7-2. Input Report Format when TCHGEOMEN = 0

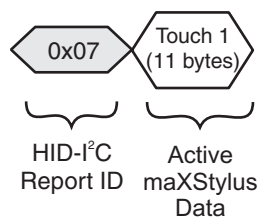
| Byte | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---------|-------------------|-------|-------|-------|-------|-------|-------|--------|
| 0 | HID-I2C Report ID | | | | | | | |
| 1 | Reserved | | | | | | | Status |
| 2 | Touch ID | | | | | | | |
| 3 – 4 | X Position | | | | | | | |
| 5 – 6 | Reserved | | | | | | | |
| 7 – 8 | Y Position | | | | | | | |
| 9 – 10 | Reserved | | | | | | | |
| 11 | Reserved | | | | | | | |
| 12 | Reserved | | | | | | | |
| 13 | Reserved | | | | | | | |
| 14 | Reserved | | | | | | | |
| 15 – 16 | Scan Time | | | | | | | |
| 17 | Contact Count | | | | | | | |

In [Table 7-1](#) and [Table 7-2](#)

- 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.
- Byte 17:
Contact Count: Number of Active Touches to be sent.

7.4.2 Active maXStylus Report

The format of an active maXStylus report packet is shown in [Figure 7-3](#).

Figure 7-3. Example Active maXStylus Report

Each active maXStylus report start with a HID-I²C Report ID (value 0x06). [Table 7-3](#) gives the detailed format of an active stylus report packet.

Table 7-3. 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 | | | | | | | |

In [Table 7-3](#)

- **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 to 7:**
 - Y position
- **Byte 8 to 9:**
 - Y center position
- **Byte 10:**
 - Tip Pressure:** Force exerted against the touch screen surface by the stylus.

7.5 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

7.5.1 Read/Write Memory Map

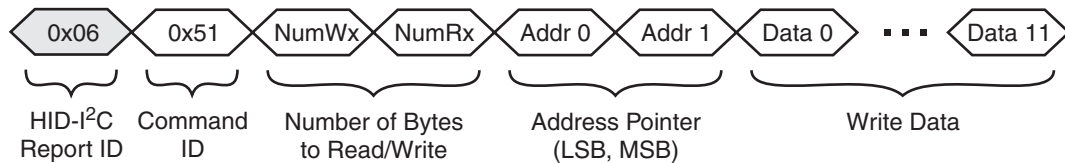
This command is used to carry out a write/read operation on the memory map of the chipset.

The HID-I²C Report ID is 0x06.

Note: This value may change.

The command packet has the generic format given in [Figure 7-4](#). The following sections give examples on using the command to write to the memory map and to read from the memory map.

Figure 7-4. Generic Command Packet Format

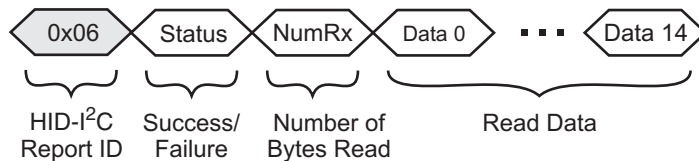


In [Figure 7-4](#):

- **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 7-5](#).

Figure 7-5. Response Packet Format



In [Figure 7-5](#):

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

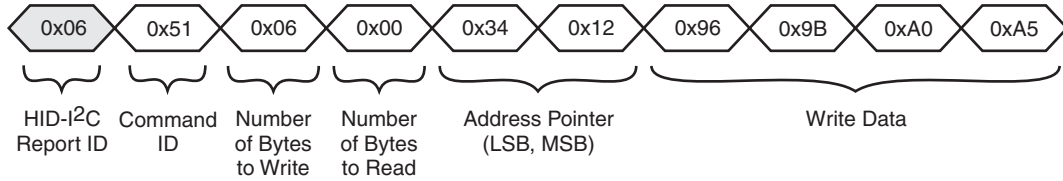
7.5.1.1 Writing To the Chipset

A write operation cycle to the chipset consists of sending a packet that contains six header bytes. These specify the HID-I²C report ID, the Command ID, the number of bytes to read, the number of bytes to write, and the 16-bit address pointer.

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

Figure 7-6 shows an example command packet to write four bytes of data to contiguous addresses starting at 0x1234.

Figure 7-6. Example of a Four-byte Write Starting at Address 0x1234

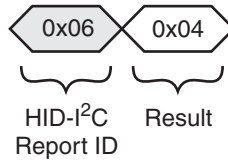


In Figure 7-6:

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

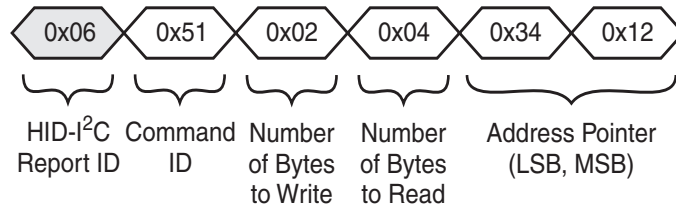


7.5.1.2 Reading From the Chipset

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

Figure 7-8 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.

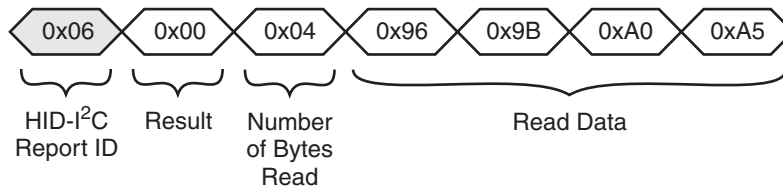
Figure 7-8. Example of a Four-byte Read Starting at Address 0x1234



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 7-9 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).

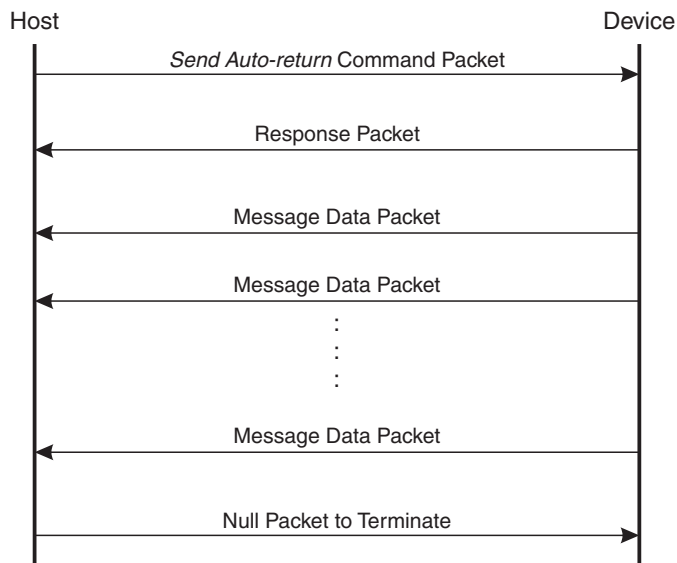
Figure 7-9. Response to Example Four-byte Read



7.5.2 Send Auto-return Messages

With this command the chipset can be configured to return new messages from the Message Processor object autonomously. The packet sequence to do this is shown in [Figure 7-10](#).

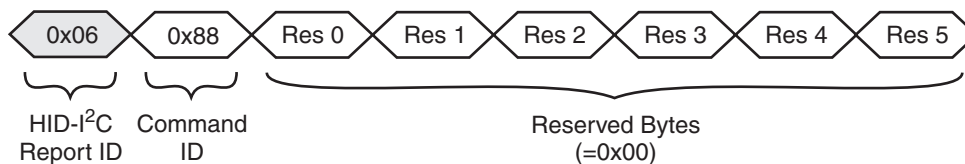
Figure 7-10. Packet Sequence for “Send Auto-return” Command



The HID-I²C Report ID is 0x06.

The command packet has the format given in [Figure 7-11](#).

Figure 7-11. Command Packet Format

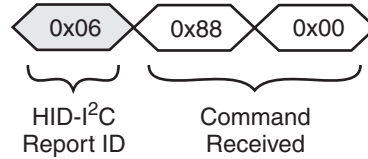


In [Figure 7-11](#):

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

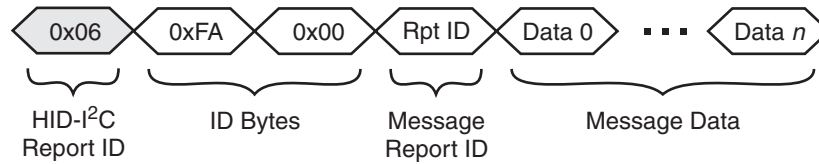
The response packet has the format given in [Figure 7-12](#). Note that with this command, the command packet does not include an address pointer as the chipset already knows the address of the Message Processor object.

Figure 7-12. Response Packet Format



Once the chipset has responded to the command, it starts sending message data. Each time a message is generated in the Message Processor object, the chipset automatically sends a message packet to the host with the data. The message packets have the format given in [Figure 7-13](#).

Figure 7-13. Message Packet Format

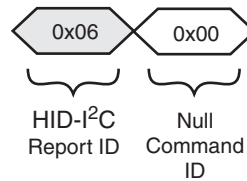


In [Figure 7-13](#):

- 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 *mXT3432S-AT 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 7-14](#)).

Figure 7-14. Null Command Packet Format



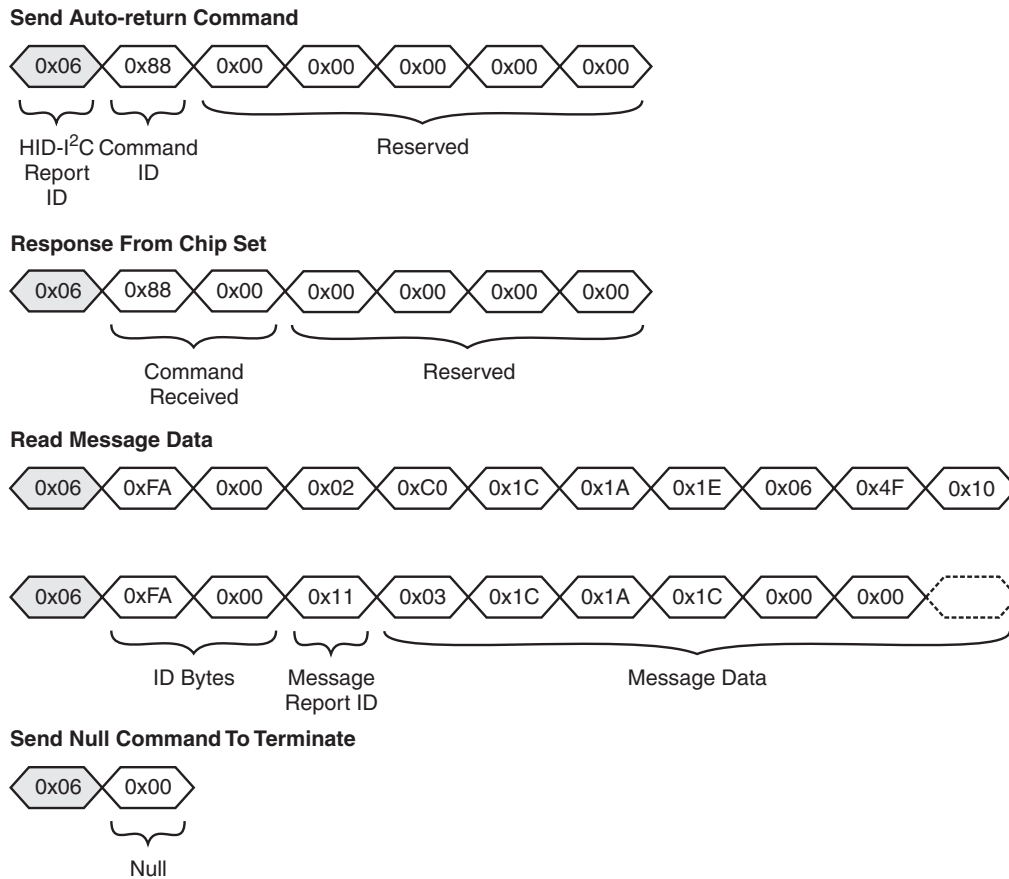
Note that any read or write will also terminate any currently enabled auto-return mode (see [“Start Debug Monitoring” on page 46](#)).

1. This is the Report ID used in the Object Protocol and should not be confused with the USB Report ID. Refer to the *mXT3432S-AT Protocol Guide* for more information on the use of Report IDs in the Object Protocol.

7.5.2.1 Reading Status Messages

Figure 7-15 shows an example sequence of packets to receive messages from the Message Processor object using the “Send Auto-return” command.

Figure 7-15. Example Auto-return Command Packet



7.6 $\overline{\text{CHG}}$ 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 34](#).

7.7 SDA, SCL

Identical to standard I²C operation. Refer to [Section 5.9 on page 36](#).

7.8 Clock Stretching

Identical to standard I²C operation. Refer to [Section 5.10 on page 36](#).

7.9 Microsoft Windows 8.x Compliance

The mXT3432S-AT has algorithms within the Digitizer HID Configuration T43 and Multiple Touch Touchscreen T9 specifically to ensure Microsoft Windows 8.x compliance.

The mXT3432S-AT also supports Microsoft Touch Hardware Quality Assurance (THQA) in the Serial Data Command T68 object. Refer to the Microsoft whitepaper on “*How to Design and Test Multitouch Hardware Solutions for Windows 8*”. These, and other device features, may need specific tuning.

8. Getting Started with mXT3432S-AT

8.1 Establishing Contact

8.1.1 Communication with the Host

The host can use either the I²C-compatible interface (see [Section 5.1 on page 30](#)), USB interface (see [Section 6.1 on page 37](#)) or the HID-I²C interface (see [Section 7.1 on page 49](#)) to communicate with the chipset.

8.1.2 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 chipset.

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

8.1.3 USB Interface

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

8.1.4 HID-I²C Interface

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

8.2 Using the Object Protocol

The chipset has an object-based protocol that is used to communicate with the chipset. Typical communication includes configuring the chipset, sending commands to the chipset, and receiving messages from the chipset. Refer to the *mXT3432S-AT Protocol Guide* for more information.

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

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

8.3 Writing to the Chipset

There are three mechanisms for writing to the chipset:

- Using an I²C-compatible write operation (see [Section 5.3 on page 30](#)).
- Using the USB Generic HID write operation (see [Section 6.5.1.1 on page 43](#)).
- Using the Generic HID-I²C write operation (see [Section 7.5.1.1 on page 54](#)).

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

- To send a command to the chipset, you write the appropriate command to the Command Processor T6 object (refer to the *mXT3432S-AT Protocol Guide*).
- To configure the chipset, you write to an object. For example, to configure the chipset 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 *mXT3432S-AT Protocol Guide* for more information on the objects.

8.4 Reading from the Chipset

Status information is stored in the Message Processor T5 object. This object can be read to receive any status information from the chipset. 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 34](#)). See [Section 5.5 on page 31](#) for information on the format of the I²C-compatible read operation.
- When using the USB interface, the Generic HID interface provides an interrupt-driven interface that sends the messages automatically (see [Section 6.5.2 on page 44](#)).
- When using the HID-I²C interface, the Generic HID-I²C interface provides an interrupt-driven interface that sends the messages automatically (see [Section 7.5.1.2 on page 54](#))

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

The USB Digitizer HID provides a third alternative interrupt-style mechanism for reading a subset of the touch data. See [Section 6.4 on page 38](#) for more information.

8.5 Configuring the Chipset

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 *mXT3432S-AT 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
 - CTE Configuration T46

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

- Touch objects:
 - Multiple Touch Touchscreen T9
- Signal processing objects:
 - Grip Suppression T40
 - Stylus T47
 - One-touch Gesture Processor T24
 - Two-touch Gesture Processor T27
 - Touch Suppression T42
 - Shieldless T56
 - Extra Touchscreen Data T57
 - maXCharger T62
 - Active Stylus T63
 - Lens Bending T65
- 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

9. Specifications

9.1 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.

9.2 Recommended Operating Conditions

| | |
|---|--|
| Operating temp | -40°C to +85°C |
| Storage temp | -60°C to +150°C |
| Vdd External | 3.3 V \pm 5% |
| AVdd | 2.7 V to 3.3 V \pm 5% |
| XVdd | 2.7 V to 10.0 V \pm 5% |
| Vdd vs AVdd power sequencing | No sequencing required |
| Vdd vs XVdd power sequencing | XVdd must not be powered before Vdd |
| Vdd supply ripple | \pm 50 mV 1 Hz to 1 MHz |
| XVdd supply ripple | \pm 25 mV 1 Hz to 1MHz |
| AVdd supply ripple (Noise suppression disabled) | \pm 25 mV 1 Hz to 1 MHz |
| Cx transverse load capacitance per channel | 0.95 pF to 4.8 pF, when XVdd = 2.5 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 |

9.3 DC Characteristics

9.3.1 Digital Voltage Supply

| Parameter | Description | Min | Typ | Max | Units | Notes |
|--------------|------------------|-------|-----|------|------------|------------------------------------|
| Vdd | Operating limits | 3.14 | 3.3 | 3.47 | V | Common to Master and Slave devices |
| Rate of rise | Rate of rise | 0.002 | – | 2.5 | V/ μ s | |

9.3.2 Analog Voltage Supply

| Parameter | Description | Min | Typ | Max | Units | Notes |
|--------------|------------------|-------|-----|------|------------|--|
| AVdd | Operating limits | 2.57 | – | 3.47 | V | See Section 4.6.3 on page 28 |
| Rate of rise | Rate of rise | 0.002 | – | 2.5 | V/ μ s | |

9.3.3 XVdd Supply

| Parameter | Description | Min | Typ | Max | Units | Notes |
|--------------|------------------|------|-----|------|-------|-------|
| XVdd | Operating limits | 2.57 | – | 10.5 | 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 devices.

9.3.4 Input/Output Characteristics

| Parameter | Description | Min | Typ | Max | Units | Notes |
|-------------------------|------------------------|---------------------|-----|---------------------|---------|---|
| Input (RESET, SDA, SCL) | | | | | | |
| Vil | Low input logic level | –0.3 | – | $0.3 \times V_{dd}$ | V | $V_{dd} = 2.5 \text{ V to } 3.3 \text{ V}$ |
| Vih | High input logic level | $0.7 \times V_{dd}$ | – | $V_{dd} + 0.3$ | V | $V_{dd} = 2.5 \text{ V to } 3.3 \text{ V}$ |
| Iil | Input leakage current | – | – | 1 | μ A | Pull-up resistors disabled |
| Output (CHG) | | | | | | |
| Vol | Low output voltage | – | – | $0.2 \times V_{dd}$ | V | $V_{dd} = 3 \text{ V}$, IOL = 2.7 mA, High Drive Enabled $V_{dd} = 3 \text{ V}$, IOL = 1.35 mA |
| Voh | High output voltage | $0.8 \times V_{dd}$ | – | – | V | $V_{dd} = 3 \text{ V}$, IOH = 2.7 mA, High Drive Enabled $V_{dd} = 3 \text{ V}$, IOH = 1.35 mA |

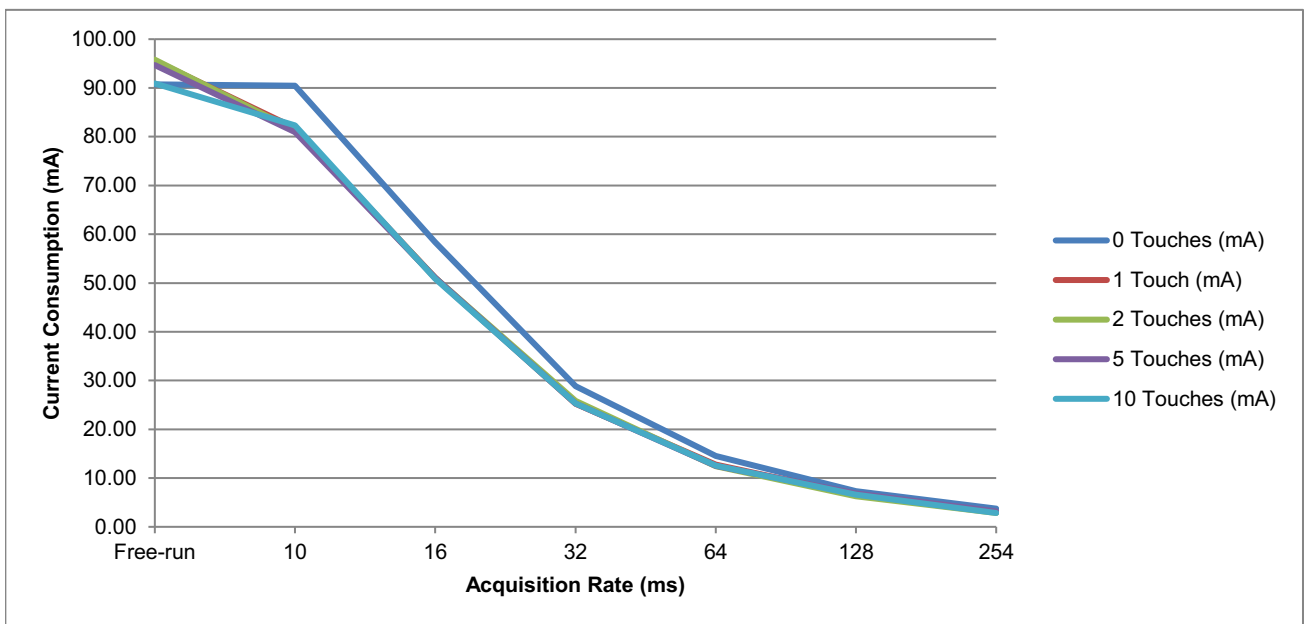
9.4 ESD Information

| Parameter | Value | Reference standard |
|-----------------------------|----------------------|---------------------------|
| Electrostatic Discharge HBM | $\pm 2000 \text{ V}$ | MIL– STD883 Method 3015.7 |

9.5 Current Consumption – I²C Interface

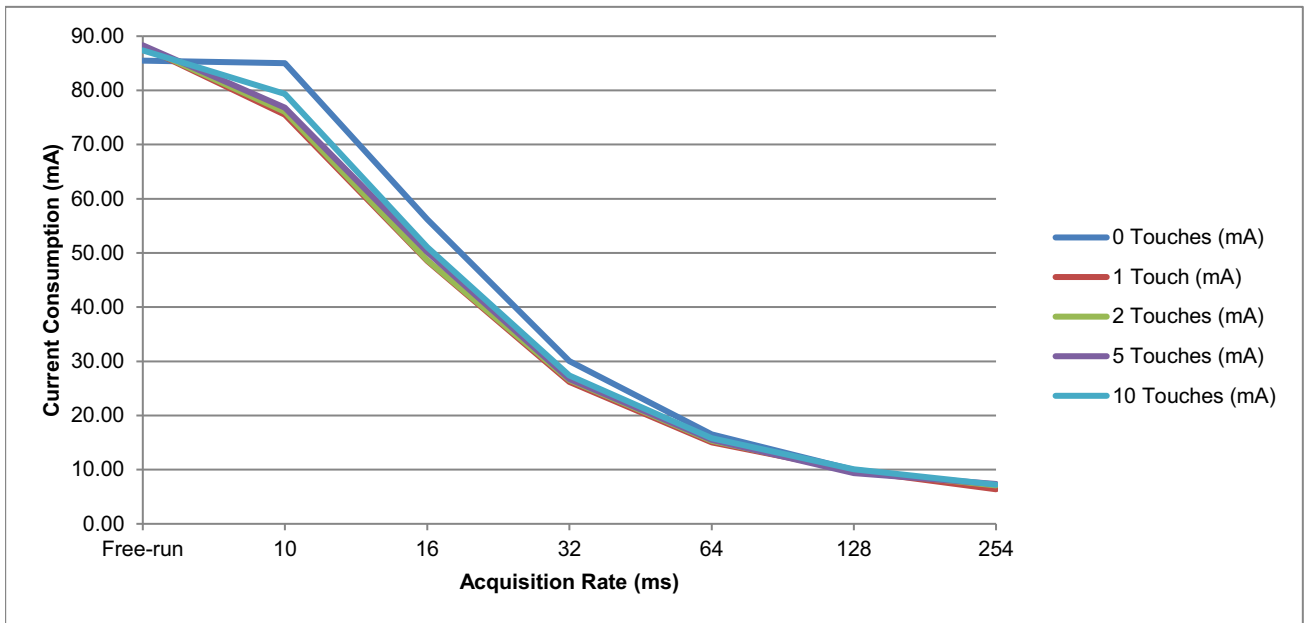
9.5.1 Avdd

| Acquisition Rate (ms) | 0 Touches (mA) | 1 Touch (mA) | 2 Touches (mA) | 5 Touches (mA) | 10 Touches (mA) |
|-----------------------|----------------|--------------|----------------|----------------|-----------------|
| Free-run | 90.74 | 95.59 | 95.80 | 94.69 | 90.92 |
| 10 | 90.46 | 81.79 | 81.36 | 80.92 | 82.29 |
| 16 | 58.38 | 51.09 | 51.10 | 51.08 | 50.89 |
| 32 | 28.85 | 25.21 | 25.80 | 25.31 | 25.28 |
| 64 | 14.57 | 12.77 | 12.42 | 12.51 | 12.56 |
| 128 | 7.30 | 6.60 | 6.25 | 6.85 | 6.62 |
| 254 | 3.74 | 3.02 | 2.85 | 3.15 | 2.84 |



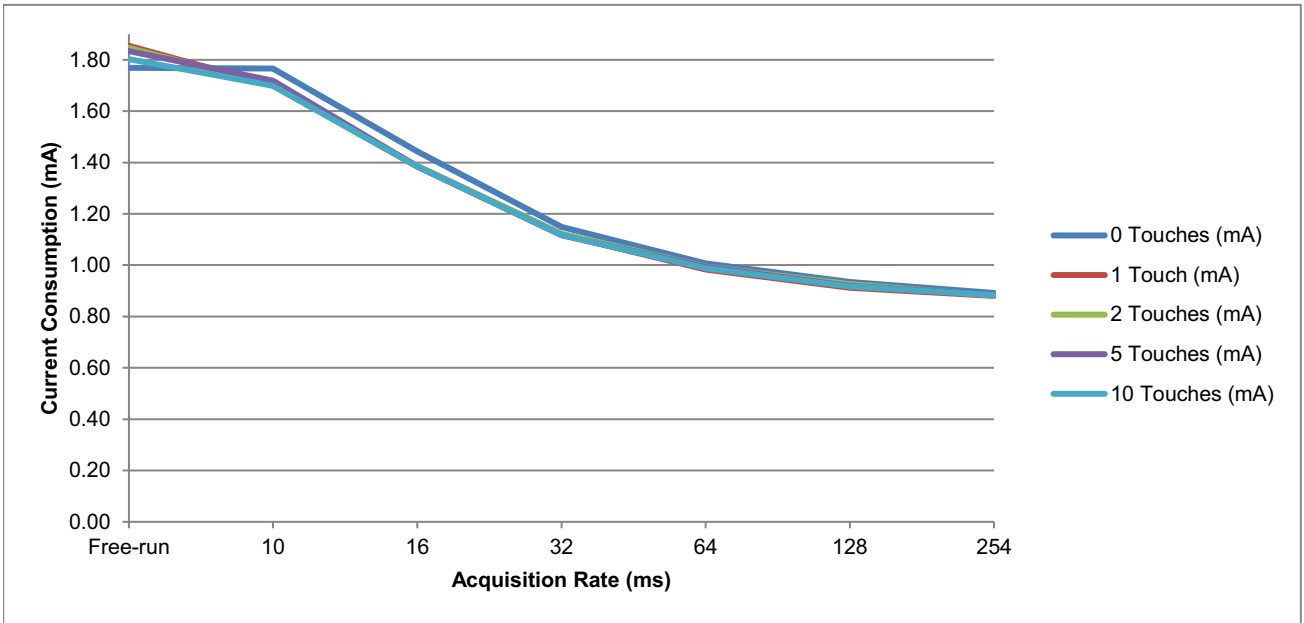
9.5.2 Vdd

| Acquisition Rate (ms) | 0 Touches (mA) | 1 Touch (mA) | 2 Touches (mA) | 5 Touches (mA) | 10 Touches (mA) |
|-----------------------|----------------|--------------|----------------|----------------|-----------------|
| Free-run | 85.49 | 88.09 | 88.14 | 88.32 | 87.38 |
| 10 | 85.03 | 75.48 | 76.04 | 76.80 | 79.33 |
| 16 | 56.22 | 48.54 | 48.61 | 50.16 | 51.05 |
| 32 | 30.05 | 26.18 | 26.56 | 26.72 | 27.35 |
| 64 | 16.50 | 15.06 | 15.32 | 15.48 | 15.81 |
| 128 | 9.98 | 9.86 | 9.75 | 9.34 | 10.07 |
| 254 | 6.99 | 6.38 | 7.09 | 7.36 | 7.23 |



9.5.3 XVdd

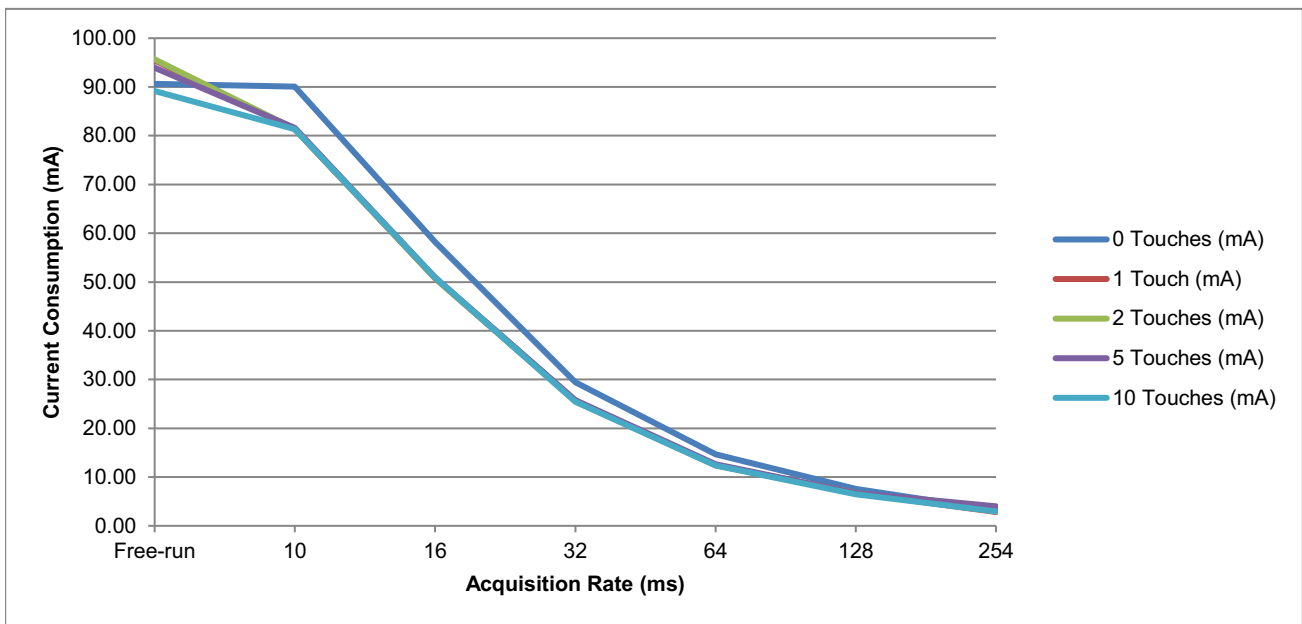
| Acquisition Rate (ms) | 0 Touches (mA) | 1 Touch (mA) | 2 Touches (mA) | 5 Touches (mA) | 10 Touches (mA) |
|-----------------------|----------------|--------------|----------------|----------------|-----------------|
| Free-run | 1.77 | 1.85 | 1.85 | 1.83 | 1.80 |
| 10 | 1.77 | 1.70 | 1.70 | 1.72 | 1.70 |
| 16 | 1.44 | 1.38 | 1.39 | 1.38 | 1.38 |
| 32 | 1.15 | 1.12 | 1.12 | 1.12 | 1.12 |
| 64 | 1.01 | 0.98 | 0.99 | 0.99 | 0.99 |
| 128 | 0.93 | 0.91 | 0.93 | 0.92 | 0.92 |
| 254 | 0.89 | 0.88 | 0.88 | 0.89 | 0.88 |



9.6 Current Consumption – USB Interface

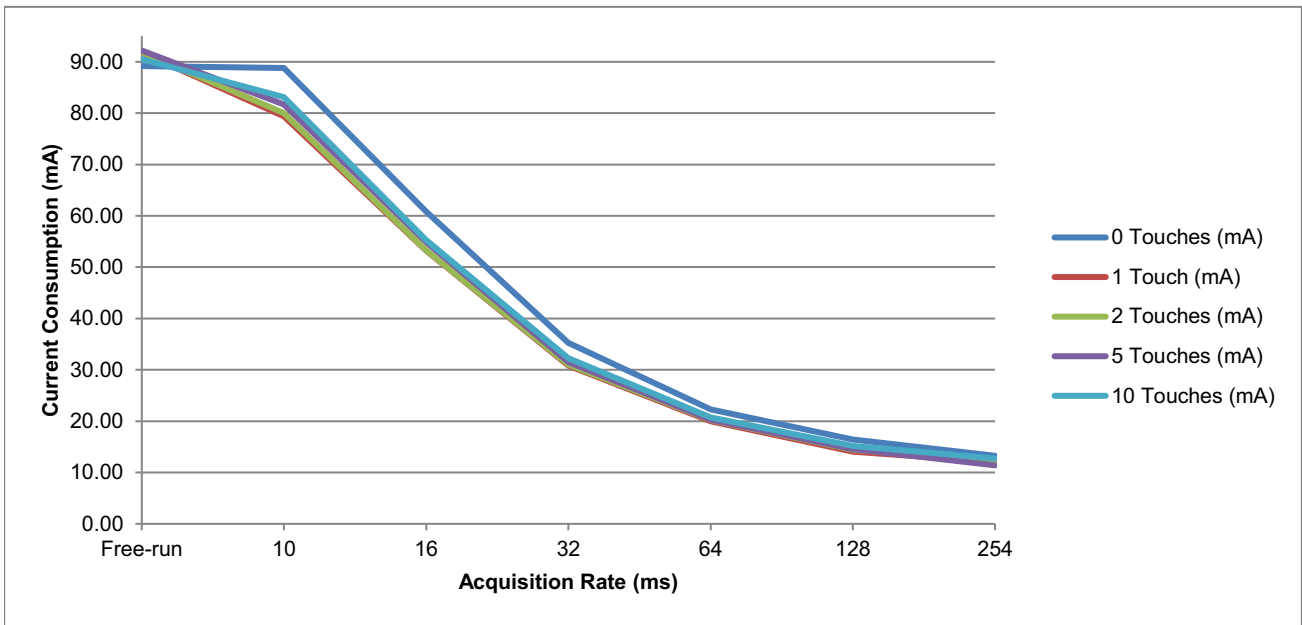
9.6.1 AVdd

| Acquisition Rate (ms) | 0 Touches (mA) | 1 Touch (mA) | 2 Touches (mA) | 5 Touches (mA) | 10 Touches (mA) |
|-----------------------|----------------|--------------|----------------|----------------|-----------------|
| Free-run | 90.64 | 95.46 | 95.65 | 93.94 | 89.17 |
| 10 | 90.09 | 81.32 | 81.39 | 81.63 | 81.35 |
| 16 | 58.22 | 50.81 | 50.78 | 50.93 | 50.94 |
| 32 | 29.41 | 25.60 | 25.64 | 25.73 | 25.40 |
| 64 | 14.64 | 12.46 | 12.41 | 12.63 | 12.39 |
| 128 | 7.57 | 6.73 | 6.73 | 6.73 | 6.46 |
| 254 | 3.19 | 2.84 | 2.97 | 4.01 | 2.96 |



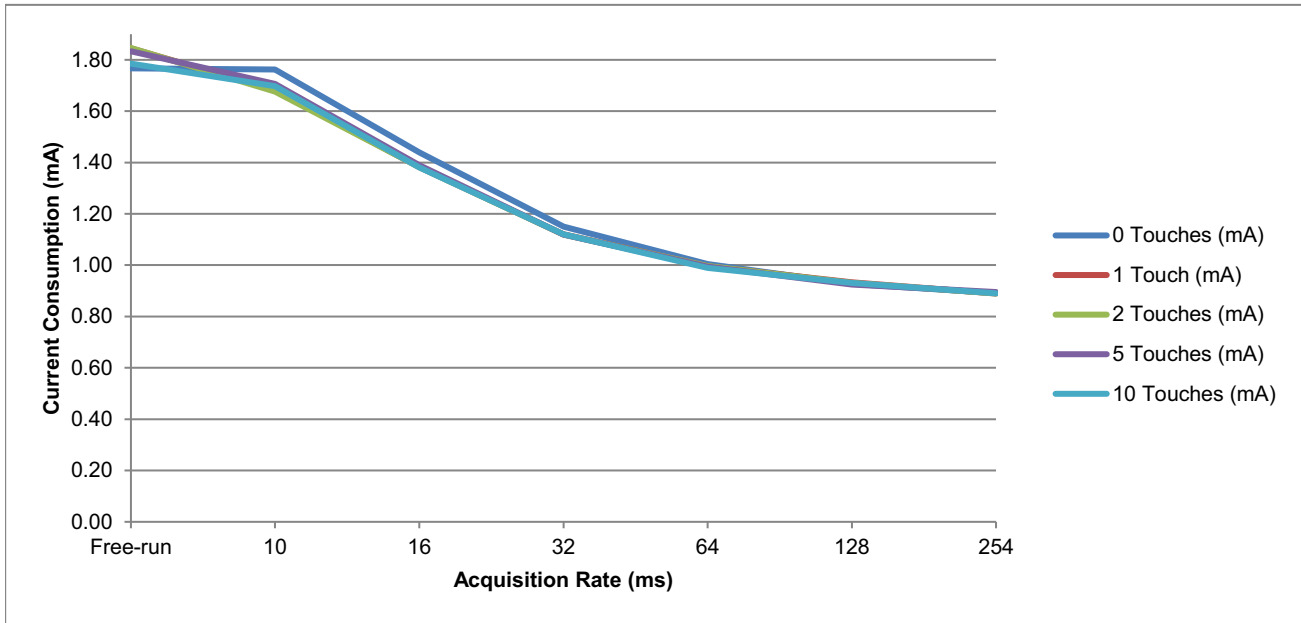
9.6.2 Vdd

| Acquisition Rate (ms) | 0 Touches (mA) | 1 Touch (mA) | 2 Touches (mA) | 5 Touches (mA) | 10 Touches (mA) |
|-----------------------|----------------|--------------|----------------|----------------|-----------------|
| Free-run | 89.17 | 91.84 | 91.90 | 92.21 | 90.52 |
| 10 | 88.78 | 79.37 | 80.00 | 81.68 | 83.15 |
| 16 | 60.83 | 53.22 | 53.27 | 54.92 | 55.25 |
| 32 | 35.24 | 30.85 | 31.05 | 31.52 | 32.27 |
| 64 | 22.32 | 19.99 | 20.26 | 20.32 | 20.70 |
| 128 | 16.45 | 14.05 | 14.46 | 14.53 | 15.21 |
| 254 | 13.27 | 12.08 | 12.25 | 11.40 | 12.73 |



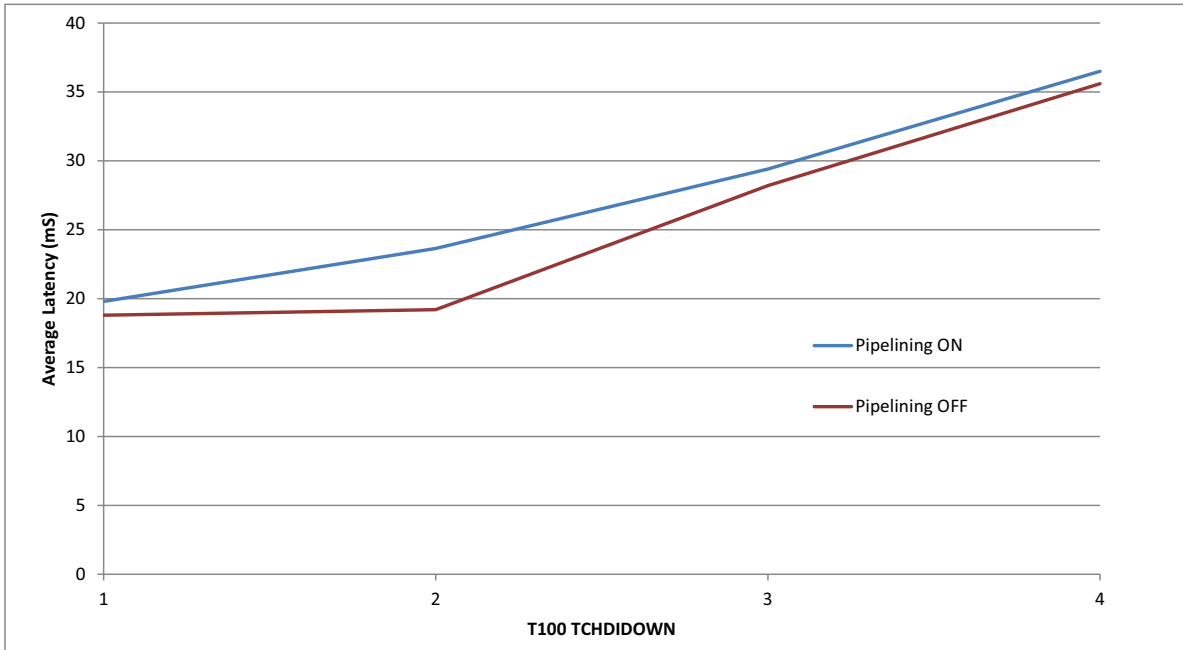
9.6.3 XVdd

| Acquisition Rate (ms) | 0 Touches (mA) | 1 Touch (mA) | 2 Touches (mA) | 5 Touches (mA) | 10 Touches (mA) |
|-----------------------|----------------|--------------|----------------|----------------|-----------------|
| Free-run | 1.77 | 1.84 | 1.85 | 1.83 | 1.79 |
| 10 | 1.76 | 1.68 | 1.68 | 1.71 | 1.70 |
| 16 | 1.44 | 1.38 | 1.38 | 1.39 | 1.38 |
| 32 | 1.15 | 1.12 | 1.12 | 1.12 | 1.12 |
| 64 | 1.00 | 1.00 | 1.00 | 0.99 | 0.99 |
| 128 | 0.93 | 0.93 | 0.93 | 0.92 | 0.93 |
| 254 | 0.89 | 0.89 | 0.89 | 0.90 | 0.89 |

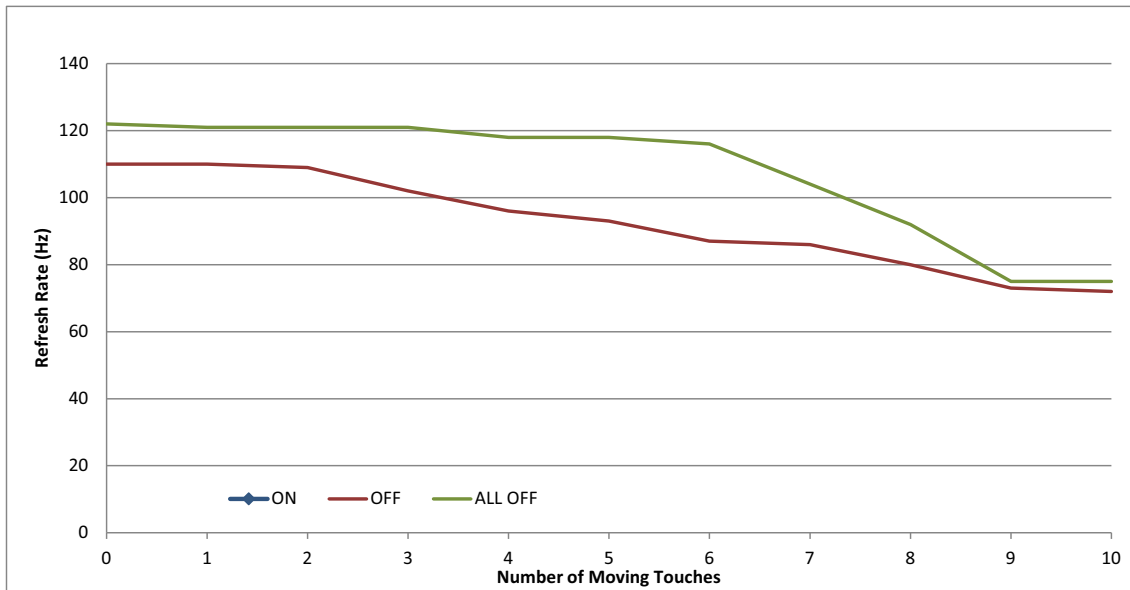


9.7 Timing Specifications

9.7.1 Touch Latency



9.7.2 Speed



9.7.3 Reset Timings

9.7.3.1 I²C Interface

| Parameter | Min | Typ | Max | Units | Notes |
|--|-----|-----|-----|-------|-------|
| Power on to $\overline{\text{CHG}}$ line low | – | 174 | – | ms | |
| Hardware reset to $\overline{\text{CHG}}$ line low | – | 174 | – | ms | |
| Software reset to $\overline{\text{CHG}}$ line low | – | 460 | – | ms | |

9.7.3.2 USB Interface

| Parameter | Min | Typ | Max | Units | Notes |
|--|-----|-----|-----|-------|-------|
| Power on to $\overline{\text{CHG}}$ line low | – | 174 | – | ms | |
| Hardware reset to $\overline{\text{CHG}}$ line low | – | 174 | – | ms | |

9.7.3.3 HID-I²C Interface

| Parameter | Min | Typ | Max | Units | Notes |
|---------------------------------|-----|-----|-----|-------|--|
| Power on to first X burst | – | 174 | – | ms | The maXTouch communication port is ready few ms prior to the first X burst |
| Hardware reset to first X burst | – | 174 | – | ms | |
| Software reset to first X burst | – | 460 | – | ms | |

9.8 I²C Specifications

| Parameter | Value |
|---|-------------------------------|
| Addresses | 0x4C, 0x4D, 0x5A or 0x5B |
| Maximum bus speed (SCL) | 200 kHz |
| I ² C specification | Version 2.1 |
| Required pull-up resistance for standard mode (100 kHz) | 1 k Ω to 10 k Ω |
| Required pull-up resistance for fast mode (200 kHz) | 1 k Ω to 3 k Ω |

9.9 USB Specification

| Parameter | Operation |
|--------------------|---|
| Endpoint Addresses | 0x81 (Endpoint 1) 0x02 (Endpoint 2) 0x83 (Endpoint 3) |
| Maximum bus speed | 12 Mbps |
| Vendor ID | 0x03EB (Atmel) |
| Product ID | 0x2136 (mXT3432S-AT) |
| USB specification | USB 2.0 HID specification 1.11 with amendments for multitouch digitizers |

9.10 HID-I²C Specification

| Parameter | Operation |
|------------------------------------|----------------------|
| Vendor ID | 0x03EB (Atmel) |
| Product ID | 0x2136 (mXT3432S-AT) |
| HID-I ² C specification | 0.9 |

9.11 Touch Accuracy and Repeatability

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

9.12 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 |

9.13 Thermal Packaging

9.13.1 Thermal Data

| Parameter | Typ | Unit | Condition | Package |
|--|------|------|-----------|----------------------|
| Junction to ambient thermal resistance | 49.7 | °C/W | Still air | TQFP 100, 14 x 14 mm |
| Junction to case thermal resistance | 14.2 | °C/W | – | TQFP 100, 14 x 14 mm |
| 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 |

9.13.2 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:

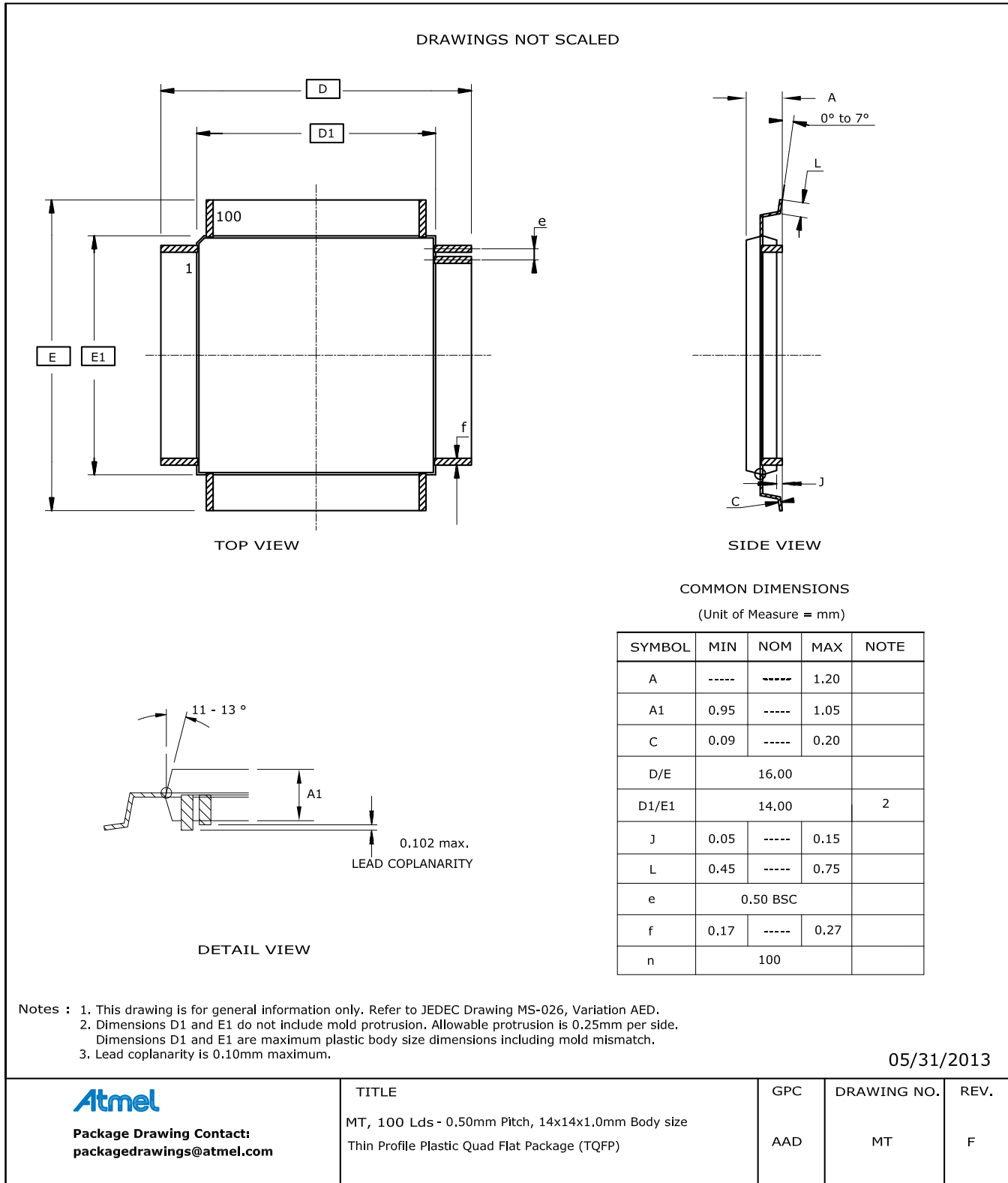
- θ_{JA} = package thermal resistance, Junction to ambient (°C/W).
- θ_{JC} = package thermal resistance, Junction to case thermal resistance (°C/W).
- $\theta_{HEATSINK}$ = cooling device thermal resistance (°C/W), provided in the cooling device datasheet.
- P_D = device power consumption (W).
- T_A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

9.14 Soldering Profile

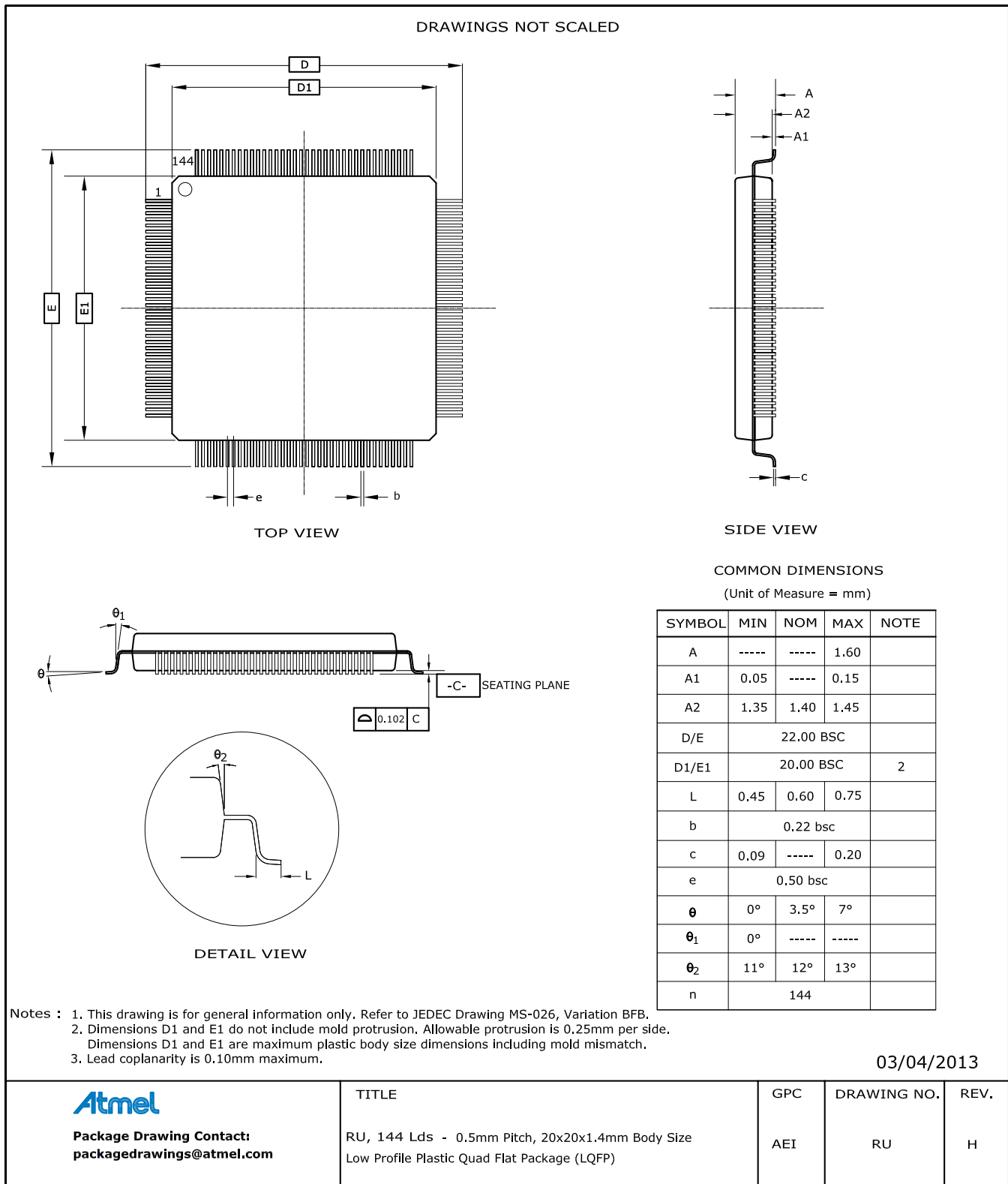
| Profile Feature | Green Package |
|--|---------------|
| Average Ramp-up Rate (217°C to Peak) | 3°C/s max |
| Preheat Temperature 175°C ±25°C | 150 – 200°C |
| Time Maintained Above 217°C | 60 – 150 s |
| Time within 5°C of Actual Peak Temperature | 30 s |
| Peak Temperature Range | 260°C |
| Ramp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

9.15 Mechanical Dimensions

9.15.1 mXT3432S-M-AT – 100-pin TQFP



9.15.2 mXT3432S-S-AT – 144-pin LQFP



9.16 Part Marking

9.16.1 ATMXT3432S-M-AT (Master)



9.16.2 ATMXT3432S-S-AT (Slave)



9.17 Part Numbers

9.17.1 Orderable Individual Parts

| Orderable Part Number | Reference Number | Firmware Revision | Description |
|-----------------------|------------------|-------------------|---|
| ATMXT3432S-M-ATR | HLE | 1.1 AA | 100-pin 14 x 14 x 1.0 mm TQFP, RoHS compliant |
| ATMXT3432S-S-ATR | HHR | 1.1 AA | 144-pin 20 x 20 x 1.4 mm LQFP, RoHS compliant |

9.18 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 mXT3432S-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 mXT3432S-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 M Ω). 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's supply pins as possible (less than 5mm away). The ground connection of these capacitors should be tracked to 0V by the shortest, heaviest traces possible.

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

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

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

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

A.5 Suggested Voltage Regulator Manufacturers

The AVdd supply stability is critical for the mXT3432S-AT 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 (around 1Ω) is required from the regulator output to the analog supply input on the mXT3432S-AT 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.

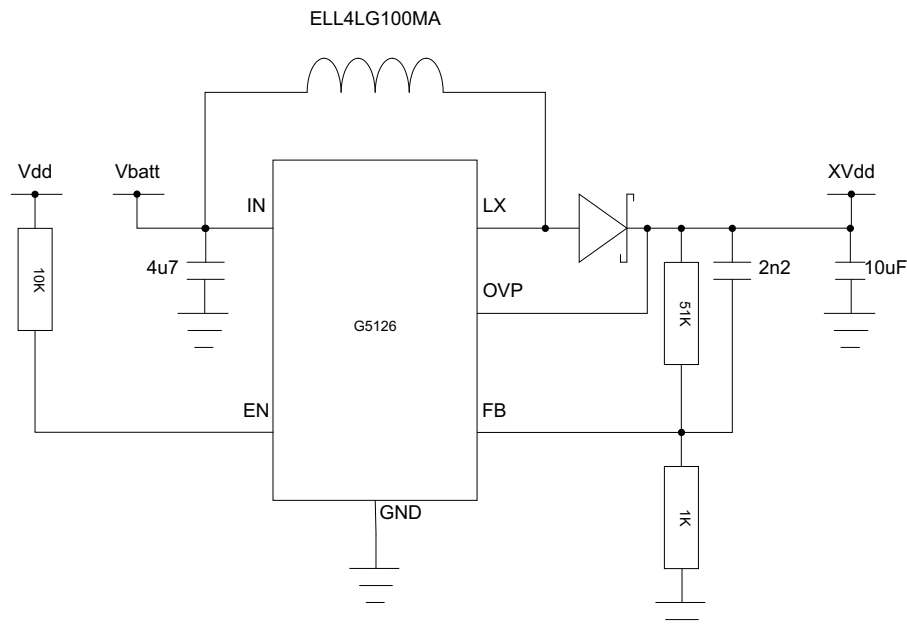
A low noise device should be chosen for the regulator. If possible this should have provision for adding a capacitor across the internal reference for further noise reduction. Reference should be made to the manufacturer's datasheet.

The voltage regulators listed in below have been tested and found to work well with the mXT3432S-AT. They have compatible footprints and pin-out specifications, and are available in the SOT-23 package.

| Manufacturer | Pin | Part Number |
|-------------------|----------|----------------|
| Texas Instruments | AVdd/Vdd | TPS79301DBVRQ1 |
| GMMT | XVdd | G5126 |

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

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

The placement of the crystal oscillator is critical to the performance of the design. The connecting leads between the mXT3432S-AT and the crystal should be as short as possible. These tracks, together with the crystal itself, should be placed above a suitable ground plane. It is also important that no other signal tracks are placed close to, or under, these tracks. The crystal input pins are at a relatively high impedance and cross-talk from other signals will seriously affect oscillator stability and accuracy. The crystal's case should also be connected to ground if possible.

If an oscillator module is used, care still needs to be taken when tracking to the mXT3432S-AT. 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 mXT3432S-AT 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 mXT3432S-AT 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.

Refer to the QTAN0096: *mXT3432S-AT PCB/FPC Layout Guidelines* application note for more details.

Appendix B. Glossary of Terms

Channel

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

Jitter

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

For example a 100 x 100 mm touchpad that shows $\pm 0.5\%$ jitter in X and $\pm 1\%$ 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 $\pm 1\%$ reports a position that is, at most, 1 mm away in either direction from the true position.

Multitouch

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

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.

Touchpad

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 touchpad active region.

1. Sometimes called Bell-shaped or Normal distribution.

Two-touch Gesture

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

Appendix C. QMatrix Primer

C.1 Acquisition Technique

QMatrix capacitive acquisition uses a series of pulses to deposit charge into a sampling capacitor, C_s . The pulses are driven on X lines from the controller. The rising edge of the pulse causes current to flow in the mutual capacitance, C_x , 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 touchpad performance.

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

When a finger is placed in close proximity (a few millimeters) or directly onto the dielectric 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 C_s , and hence the terminal voltage present on C_s , after all the charge transfer pulses are complete, becomes less. In this way, the controller can measure changes in C_x during touch. This means that the measured capacitance C_x 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 C_x . 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 touchpads 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 touchpad suffers due to moisture on the dielectric surface.

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

C.3 Interference Sources

C.3.1 Power Supply

The chipset can tolerate short-term power supply fluctuations. If the power supply fluctuates slowly with temperature, the chipset tracks and compensates 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 chipset 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 μ F and a higher frequency capacitor of around 10 nF to 100 nF in parallel are recommended; both must be X7R or X5R dielectric capacitors.

C.3.2 Other Noise Sources

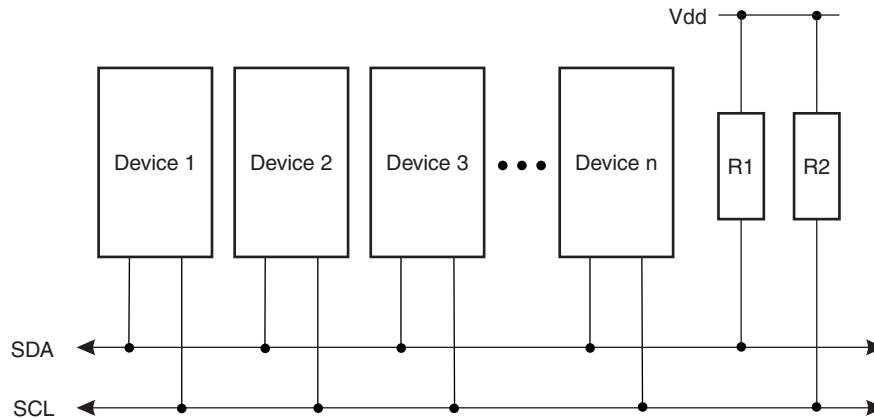
Refer to QTAN0079, *Buttons, Sliders and Wheels Sensor Design Guide*, for information (downloadable from the Touch Technology area of the Atmel website).

Appendix D. I²C Basics (I²C-compatible Operation)

D.1 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 D-1. Both bus lines are connected to V_{dd} 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.

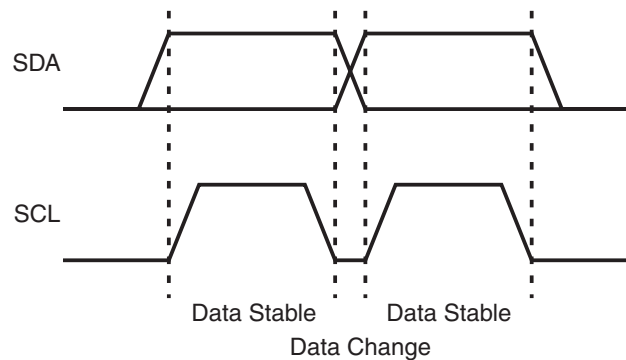
Figure D-1. I²C Interface Bus



D.2 Transferring Data Bits

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

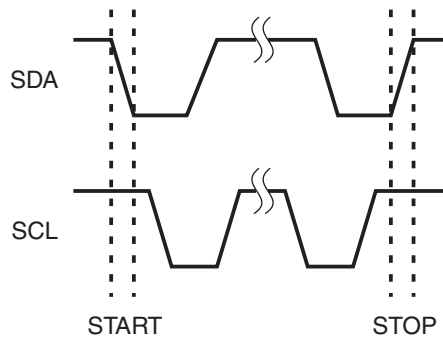
Figure D-2. Data Transfer



D.3 START and STOP Conditions

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

Figure D-3. START and STOP Conditions

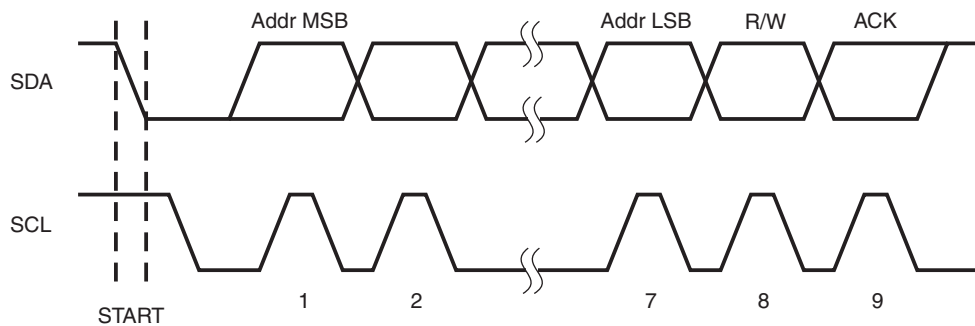


D.4 Address Byte Format

All address bytes are 9 bits long, 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.

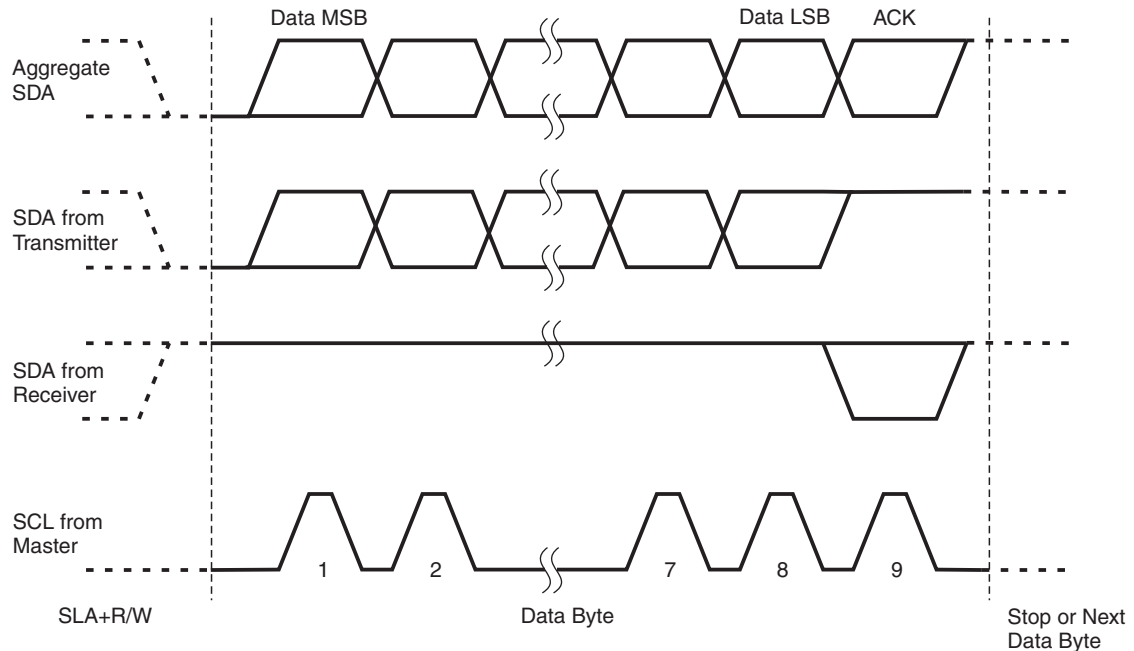
Figure D-4. Address Byte Format



D.5 Data Byte Format

All data bytes are 9 bits long, consisting of 8 data bits and an acknowledge bit. During a data transfer, the host generates the clock and the START and STOP conditions, 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 D-5. Data Byte Format



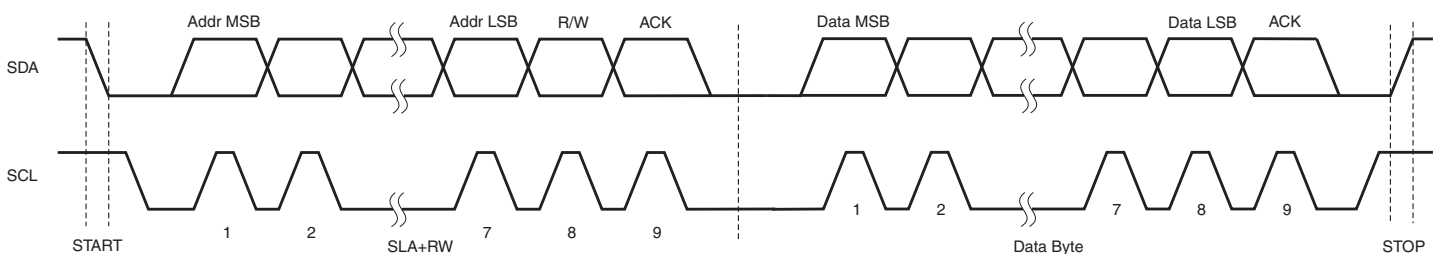
D.6 Combining Address and Data Bytes into a Transmission

A transmission consists of a START condition, an SLA+R/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 D-6 shows a typical data transmission. Note that several data bytes can be transmitted between the SLA+R/W and the STOP.

Figure D-6. Byte Transmission



Revision History

| Revision Number | History |
|------------------------|---|
| Revision AX – May 2015 | Initial release for firmware revision 1.1 – Release |

Notes



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 München GmbH

Business Campus
Parkring 4
D-85748 Garching bei München

GERMANY

Tel: (+49) 89-31970-0

Fax: (+49) 89-3194621

Atmel Japan G.K.

16F Shin-Osaki Kangyo Bldg
1-6-4 Osaki, Shinagawa-ku
Tokyo 141-0032

JAPAN

Tel: (+81) (3) 6417-0300

Fax: (+81) (3) 6417-0370

© 2015 Atmel Corporation. All rights reserved.

Atmel®, Atmel logo and combinations thereof, Enabling Unlimited Possibilities®, maXTouch®, QMatrix® and others are registered trademarks or trademarks of Atmel Corporation or its subsidiaries. Other terms and product names may be trademarks of others.

Disclaimer: The information in this document is provided in connection with Atmel products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of Atmel products. EXCEPT AS SET FORTH IN THE ATMEL TERMS AND CONDITIONS OF SALES LOCATED ON THE ATMEL WEBSITE, ATMEL ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL ATMEL BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS AND PROFITS, BUSINESS INTERRUPTION, OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF ATMEL HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Atmel makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and products descriptions at any time without notice. Atmel does not make any commitment to update the information contained herein. Unless specifically provided otherwise, Atmel products are not suitable for, and shall not be used in, automotive applications. Atmel products are not intended, authorized, or warranted for use as components in applications intended to support or sustain life.