Atmel mXT337TPTP
Revision 1.0

maXTouch 336-node Touchpad Controller

DATASHEET

Features

- Atmel® maXTouch® Adaptive Sensing Touchpad Technology
  - Up to 24 X (transmit) lines and 14 Y (receive) lines
  - A maximum of 336 nodes can be allocated to the touchpad
  - Screen sizes of 5.3 inches diagonal are supported (subject to configuration) while meeting Microsoft® Phone touch performance requirements
  - Larger screen sizes for Android applications supported (subject to configuration)
  - Multi-touch support with up to 16 concurrent touches tracked in real time
- Dual-boot OS support for Microsoft® Windows® and Android
- Advanced Touch Handling
  - Moisture/Water Compensation
    - No false touch with condensation or water drop up to 22 mm diameter
    - One-finger tracking with condensation or water drop up to 22 mm diameter
  - Stylus Support
    - Supports passive stylus with 1 mm contact diameter, subject to configuration, stack up, and sensor design
    - Supports single-touch gloved operation with various materials up to 5 mm thickness
- Touch Performance
  - Mutual capacitance and self capacitance measurements supported for touch detection
  - Response Times
    - Initial latency <10 ms for first touch from idle, subject to configuration
  - Atmel maXCharger® technology to combat ambient, charger noise, and power-line noise:
    - Up to 240 Vpp between 1 Hz and 1 kHz sinusoidal waveform
    - Up to 20 Vpp between 1 kHz and 1 MHz sinusoidal waveform
  - Scan Speed
    - Typical report rate for 10 touches ≥100 Hz
- Enhanced Algorithms
  - Lens bending algorithms to remove signal distortions
  - Touch suppression algorithms to remove unintentional touches
  - Palm Recovery Algorithm for quick restoration to normal state
- Panel / Cover Glass Support
  - Supports fully-laminated sensors, touch-on-lens stack-ups and on-cell designs
  - Works with PET or glass, including curved profiles
    - Glass from 0.55 mm to 2.5 mm, dependent on screen size, touch size, and stack-up
    - Plastic from 0.2 mm to 1.2 mm, dependent on screen size and touch size
  - Works with all proprietary sensor patterns recommended by Atmel
  - Compatible with True Single-Layer designs

- Power Saving
  - Programmable timeout for automatic transition from active to idle states
  - Pipelined analog sensing detection and digital processing to optimize system power efficiency
- Application Interfaces
  - I^{2}C-compatible slave mode: Standard/Fast mode 400 kHz, Fast-plus mode 1 MHz, High-speed mode up to 3.4 MHz
  - HID-I^{2}C interface for Microsoft® Windows® 10
  - Interrupt to indicate when a message is available
- Power Supply
  - Digital (Vdd) 3.3 V nominal
  - Analog (AVdd) 3.3 V nominal
  - Host interface I/O voltage (VddIO) 1.8 V to 3.3 V nominal
  - High voltage internal X line drive (XVdd) = 2 × Vdd (6.6 V), with internal voltage doubler
- Packages
  - 56-pin UFQFN 6 × 6 × 0.6 mm, 0.35 mm pitch
  - 64-ball UFBGA 5 × 5 × 0.6 mm, 0.5 mm pitch
  - 72-ball UFBGA 6 × 6 × 0.6 mm, 0.5 mm pitch
- Environmental Conditions
  - Operating temperature –40°C to +85°C

Table of Contents

Features 1
Table of Contents 3
1. Overview of mXT337TPTP 6
   1.1 Introduction 6
2. Connection and Configuration Information 7
   2.1 Pin Configuration – UFQFN 56 Pins 7
   2.2 Pin Configuration – UFBGA 64 Balls 11
   2.3 Pin Configuration – UFBGA 72 Balls 15
3. Schematics 19
   3.1 56-pin UFQFN 19
   3.2 64-ball UFBGA 20
   3.3 72-ball UFBGA 21
   3.4 Schematic Notes 22
4. Circuit Components 24
   4.1 Decoupling Capacitors 24
   4.2 I^{2}C Line Pull-up Resistors 24
   4.3 Supply Quality 24
   4.4 Suggested Voltage Regulators 24
5. Touchpad Basics 26
   5.1 Sensor Construction 26
   5.2 Electrode Configuration 26
   5.3 Scanning Sequence 26
   5.4 Touchpad Sensitivity 26
6. Sensor Layout 28
7. Power-up / Reset Requirements 29
   7.1 Power-up and Reset Sequence – VddIO Enabled after Vdd 30
8. Detailed Operation 32
   8.1 Touch Detection 32
   8.2 Operational Modes 32
   8.3 Detection Integrator 32
   8.4 Sensor Acquisition 32
   8.5 Calibration 32
   8.6 Digital Filtering and Noise Suppression 33
   8.7 Shieldless Support and Display Noise Suppression 33
   8.8 Retransmission Compensation 33
   8.9 Lens Bending 34
   8.10 Stylus Support 34
   8.11 Unintentional Touch Suppression 34
   8.12 GPIO Pins 34

9. Host Communications 35

10. I2C Communications 36
- 10.1 I2C Addresses 36
- 10.2 Writing To the Device 36
- 10.3 I^{2}C Writes in Checksum Mode 36
- 10.4 Reading From the Device 37
- 10.5 Reading Status Messages with DMA 37
- 10.6 $\overline{\text{CHG}}$ Line 39
- 10.7 SDA, SCL 40
- 10.8 Clock Stretching 41

11. HID-I2C Communications 42
- 11.1 I^{2}C Addresses 42
- 11.2 Device 42
- 11.3 HID Descriptor 42
- 11.4 HID-I^{2}C Report IDs 42
- 11.5 Generic HID-I^{2}C 43
- 11.6 Digitizer HID-I^{2}C 48
- 11.7 $\overline{\text{CHG}}$ Line 53
- 11.8 SDA, SCL 53
- 11.9 Clock Stretching 53
- 11.10 Power Control 53
- 11.11 Microsoft Windows Compliance 53

12. PCB Design Considerations 54
- 12.1 Introduction 54
- 12.2 Printed Circuit Board 54
- 12.3 Supply Rails and Ground Tracking 54
- 12.4 Power Supply Decoupling 54
- 12.5 Single Supply Operation 55
- 12.6 Analog I/O 55
- 12.7 Component Placement and Tracking 55
- 12.8 EMC and Other Observations 56

13. Getting Started with mXT337TPTP 57
- 13.1 Establishing Contact 57
- 13.2 Using the Object Protocol 57
- 13.3 Writing to the Device 60
- 13.4 Reading from the Device 60

14. Debugging and Tuning 61
- 14.1 Hardware SPI Debug Interface 61
- 14.2 Object-based Protocol 61
- 14.3 Self Test 61

15. Specifications 62
15.1 Absolute Maximum Specifications 62
15.2 Recommended Operating Conditions 62
15.3 DC Characteristics 62
15.4 Test Configuration 64
15.5 Current Consumption 65
15.6 Deep Sleep Current 67
15.7 Power Supply Ripple and Noise 68
15.8 Timing Specifications 69
15.9 Input/Output Characteristics 70
15.10 I2C Specifications 70
15.11 HID-I^{2}C Specification 71
15.12 Touch Accuracy and Repeatability 71
15.13 Thermal Packaging 71
15.14 ESD Information 72
15.15 Soldering Profile 72
15.16 Moisture Sensitivity Level (MSL) 72

16. Package Information 73
16.1 Part Marking 73
16.2 Orderable Part Numbers 74
16.3 Mechanical Drawings 75

Associated Documents 78

Revision History 79

1. Overview of mXT337TPTP

1.1 Introduction

The Atmel maXTouch family of touch controllers brings industry-leading capacitive touch performance to customer applications. The mXT337TPTP features the latest generation of Atmel Adaptive Sensing technology that utilizes a hybrid mutual- and self-capacitive sensing system in order to deliver unparalleled touch features and a robust user experience.

- **Patented capacitive sensing method** – The mXT337TPTP uses a unique charge-transfer acquisition engine to implement the Atmel-patented QMatrix® capacitive sensing method. Coupled with a state-of-the-art CPU, the entire touchpad sensing solution can measure, classify and track number of individual finger touches with a high degree of accuracy in the shortest response time.
- **Capacitive Touch Engine (CTE)** – The mXT337TPTP features an acquisition engine, which uses an optimal measurement approach to ensure almost complete immunity from parasitic capacitance on the receiver input lines. The engine includes sufficient dynamic range to cope with anticipated touchpad self and mutual capacitances, which allows great flexibility for use with the Atmel proprietary sensor pattern designs. One- and two-layer ITO sensors are possible using glass or PET substrates.
- **Touch detection** – The mXT337TPTP allows for both mutual- and self-capacitance measurements, with the self-capacitance measurements being used to augment the mutual-capacitance measurements to produce reliable touch information.

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

  During idle mode, the device performs self-capacitance touch scans. When a touch is detected, the device starts performing mutual-capacitance touch scans as well as self capacitance scans.

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

  Self-capacitance measurements, on the other hand, allow for the detection of single touches in extreme case, when touches can only be detected by self-capacitance data and may be missed by mutual-capacitance touch detection.
- **Display Noise Cancellation** – A combination of analog circuitry, hardware noise processing, and firmware that combats display noise without requiring additional listening channels or synchronization to display timing. This enables the use of shieldless touch sensor stacks, including touch-on-lens.
- **Noise filtering** – Hardware noise processing in the capacitive touch engine provides enhanced autonomous filtering and allows a broad range of noise profiles to be handled. The result is good performance in the presence of charger and LCD noise.
- **Processing power** – The main CPU has two powerful microsequencer coprocessors under its control consuming low power. This system allows the signal acquisition, preprocessing, postprocessing and housekeeping to be partitioned in an efficient and flexible way.
- **Interpreting user intention** – The Atmel hybrid mutual- and self-capacitance method provides unambiguous multitouch performance. Algorithms in the mXT337TPTP provide optimized touchpad position filtering for the smooth tracking of touches, responding to a user's intended touches while preventing false touch triggered by ambient noise or conductive material on the sensor surface, such as water. The suppression of unintentional touches from the user’s gripping fingers or resting palm also help ensure that the user’s intentions are correctly interpreted.

2. Connection and Configuration Information

2.1 Pin Configuration – UFQFN 56 Pins

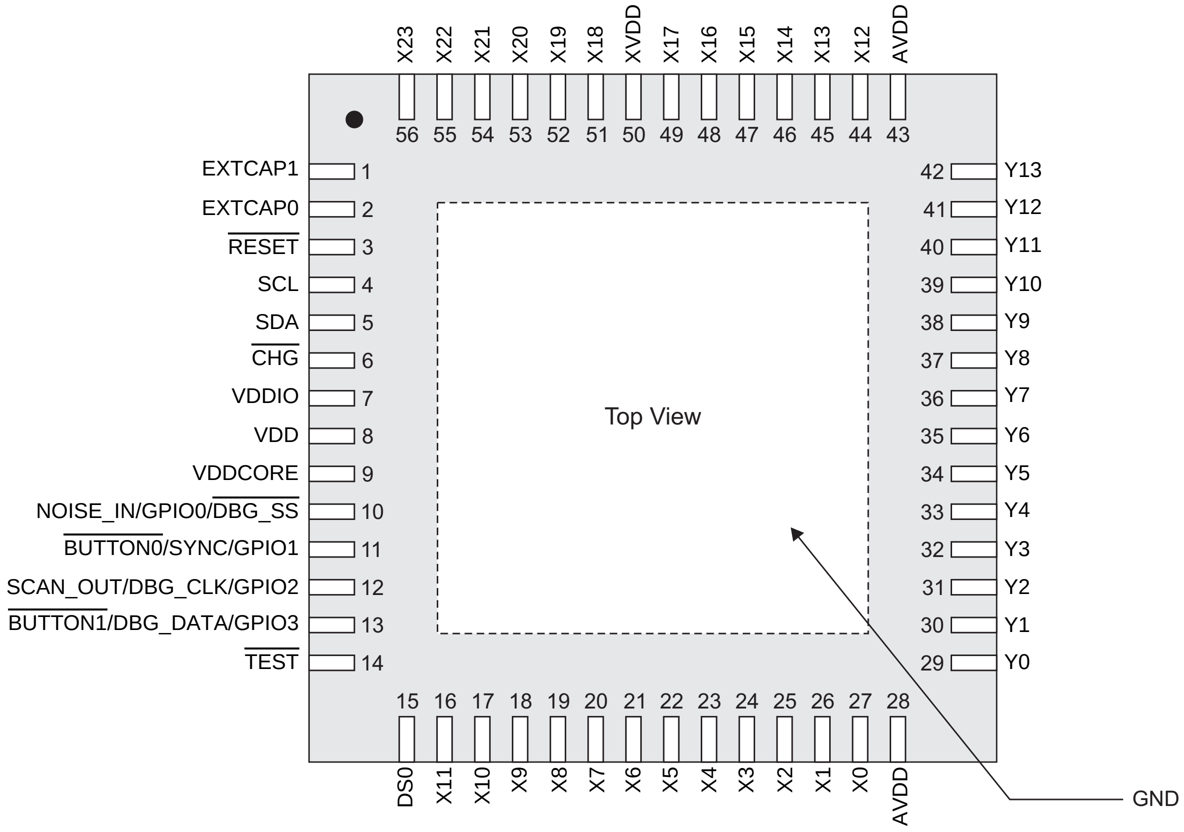


Table 2-1. Pin Listing – UFQFN 56 Pins

Pin	Name	Type	Description	If Unused...
1	EXTCAP1	P	Normal mode – leave open Voltage doubler mode – connect to EXTCAP0 via capacitor	Leave open
2	EXTCAP0	P	Normal mode – leave open Voltage doubler mode – connect to EXTCAP1 via capacitor	Leave open
3	$\overline{\text{RESET}}$	I	Reset low. Connection to host system is recommended	Pull up to VddIO
4	SCL	OD	Serial Interface clock	–
5	SDA	OD	Serial Interface Data	–
6	$\overline{\text{CHG}}$	OD	State change interrupt Note: Briefly set (~100 ms) as an input after power-up/reset for diagnostic purposes	Pull up to VddIO
7	VDDIO	P	Digital IO interface power	–
8	VDD	P	Digital power	–
9	VDDCORE	P	Digital core power	–
10	NOISE_IN GPIO0 $\overline{\text{DBG\_SS}}$	I/O	Noise present input General purpose I/O Debug SS line; pull up to Vdd	Input: GND Output: leave open
11	$\overline{\text{BUTTON0}}$ SYNC GPIO1	I/O	Touchpad button 0, active low External synchronisation General purpose I/O	Input: GND Output: leave open
12	SCAN_OUT DBG_CLK GPIO2	I/O	Indicates touch scanning in progress. Polarity configurable Debug Clock General purpose I/O	Input: GND Output: leave open
13	$\overline{\text{BUTTON1}}$ DBG_DATA GPIO3	I/O	Touchpad button 1, active low Debug Data General purpose I/O	Input: GND Output: leave open
14	$\overline{\text{TEST}}$	–	Reserved for factory use. Pull up to VddIO	Pull up to VddIO
15	DS0	S	Driven Shield signal; used as guard track between X/Y signals and ground	Connect to GND
16	X11	S	X line connection	Leave open
17	X10	S	X line connection	Leave open
18	X9	S	X line connection	Leave open
19	X8	S	X line connection	Leave open
20	X7	S	X line connection	Leave open
21	X6	S	X line connection	Leave open
22	X5	S	X line connection	Leave open

Table 2-1. Pin Listing – UFQFN 56 Pins

Pin	Name	Type	Description	If Unused...
23	X4	S	X line connection	Leave open
24	X3	S	X line connection	Leave open
25	X2	S	X line connection	Leave open
26	X1	S	X line connection	Leave open
27	X0	S	X line connection	Leave open
28	AVDD	P	Analog power	–
29	Y0	S	Y line connection	Leave open
30	Y1	S	Y line connection	Leave open
31	Y2	S	Y line connection	Leave open
32	Y3	S	Y line connection	Leave open
33	Y4	S	Y line connection	Leave open
34	Y5	S	Y line connection	Leave open
35	Y6	S	Y line connection	Leave open
36	Y7	S	Y line connection	Leave open
37	Y8	S	Y line connection	Leave open
38	Y9	S	Y line connection	Leave open
39	Y10	S	Y line connection	Leave open
40	Y11	S	Y line connection	Leave open
41	Y12	S	Y line connection	Leave open
42	Y13	S	Y line connection	Leave open
43	AVDD	P	Analog power	–
44	X12	S	X line connection	Leave open
45	X13	S	X line connection	Leave open
46	X14	S	X line connection	Leave open
47	X15	S	X line connection	Leave open
48	X16	S	X line connection	Leave open
49	X17	S	X line connection	Leave open
50	XVDD	P	X line drive power	Leave open
51	X18	S	X line connection	Leave open
52	X19	S	X line connection	Leave open
53	X20	S	X line connection	Leave open
54	X21	S	X line connection	Leave open

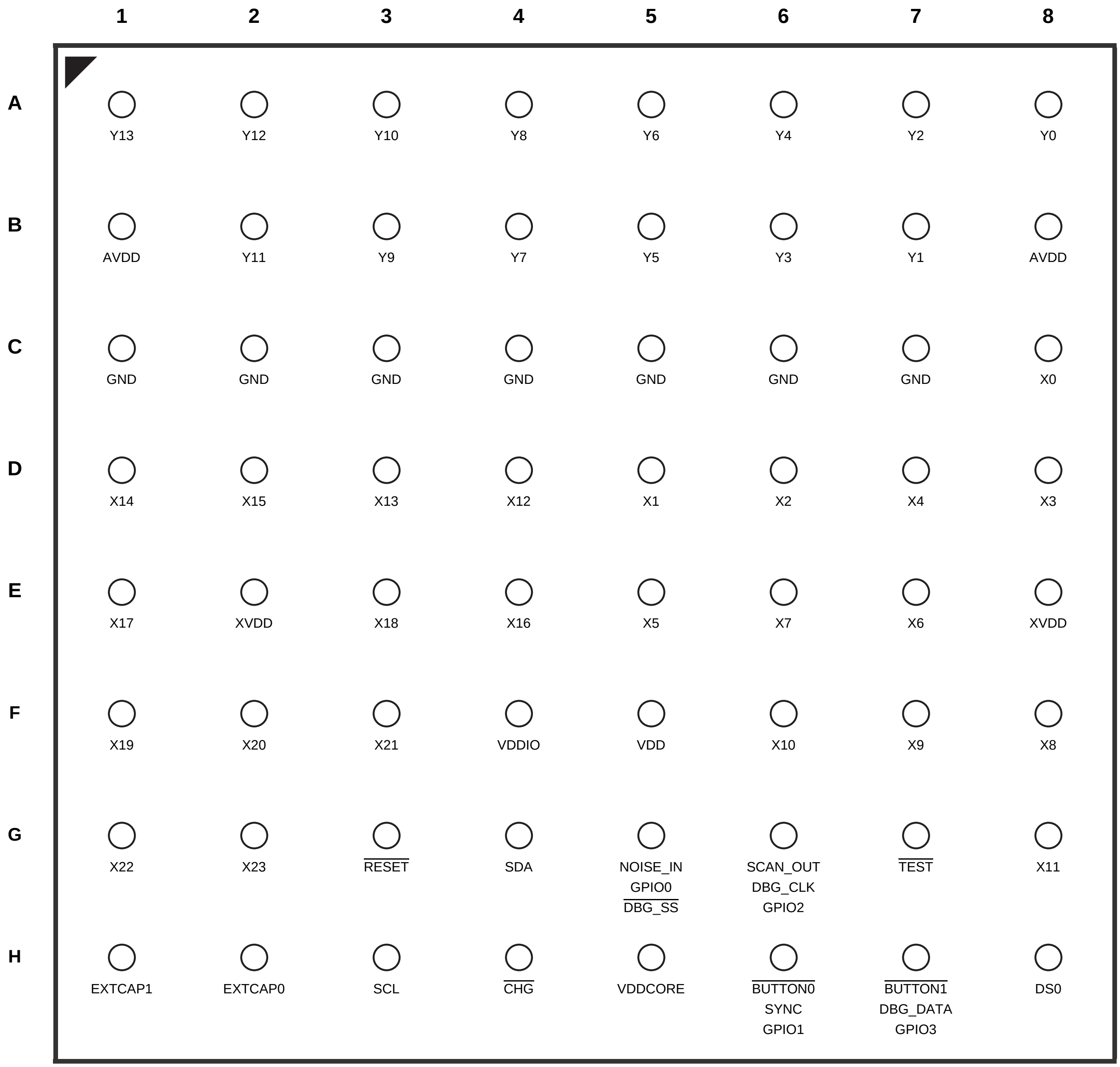
Table 2-1. Pin Listing – UFQFN 56 Pins

Pin	Name	Type	Description	If Unused...
55	X22	S	X line connection	Leave open
56	X23	S	X line connection	Leave open
Pad	GND	P	Exposed pad must be connected to GND	–

Key:

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

2.2 Pin Configuration – UFBGA 64 Balls



	1	2	3	4	5	6	7	8
A	Y13	Y12	Y10	Y8	Y6	Y4	Y2	Y0
B	AVDD	Y11	Y9	Y7	Y5	Y3	Y1	AVDD
C	GND	GND	GND	GND	GND	GND	GND	X0
D	X14	X15	X13	X12	X1	X2	X4	X3
E	X17	XVDD	X18	X16	X5	X7	X6	XVDD
F	X19	X20	X21	VDDIO	VDD	X10	X9	X8
G	X22	X23	$\overline{\text{RESET}}$	SDA	NOISE_IN GPIO0 $\overline{\text{DBG\_SS}}$	SCAN_OUT DBG_CLK GPIO2	$\overline{\text{TEST}}$	X11
H	EXTCAP1	EXTCAP0	SCL	$\overline{\text{CHG}}$	VDDCORE	$\overline{\text{BUTTON0}}$ SYNC GPIO1	$\overline{\text{BUTTON1}}$ DBG_DATA GPIO3	DS0

Top View

Table 2-2. Pin Listing – UFBGA 64 Balls

Pin	Name	Type	Description	If Unused...
A1	Y13	S	Y line connection	Leave open
A2	Y12	S	Y line connection	Leave open
A3	Y10	S	Y line connection	Leave open
A4	Y8	S	Y line connection	Leave open
A5	Y6	S	Y line connection	Leave open
A6	Y4	S	Y line connection	Leave open
A7	Y2	S	Y line connection	Leave open
A8	Y0	S	Y line connection	Leave open
B1	AVDD	P	Analog power	–
B2	Y11	S	Y line connection	Leave open
B3	Y9	S	Y line connection	Leave open
B4	Y7	S	Y line connection	Leave open
B5	Y5	S	Y line connection	Leave open
B6	Y3	S	Y line connection	Leave open
B7	Y1	S	Y line connection	Leave open
B8	AVDD	P	Analog power	–
C1	GND	P	Ground	–
C2	GND	P	Ground	–
C3	GND	P	Ground	–
C4	GND	P	Ground	–
C5	GND	P	Ground	–
C6	GND	P	Ground	–
C7	GND	P	Ground	–
C8	X0	S	X line connection	Leave open
D1	X14	S	X line connection	Leave open
D2	X15	S	X line connection	Leave open
D3	X13	S	X line connection	Leave open
D4	X12	S	X line connection	Leave open
D5	X1	S	X line connection	Leave open
D6	X2	S	X line connection	Leave open
D7	X4	S	X line connection	Leave open
D8	X3	S	X line connection	Leave open

Table 2-2. Pin Listing – UFBGA 64 Balls

Pin	Name	Type	Description	If Unused...
E1	X17	S	X line connection	Leave open
E2	XVDD	P	X line drive power	Leave open
E3	X18	S	X line connection	Leave open
E4	X16	S	X line connection	Leave open
E5	X5	S	X line connection	Leave open
E6	X7	S	X line connection	Leave open
E7	X6	S	X line connection	Leave open
E8	XVDD	P	X line drive power	Leave open
F1	X19	S	X line connection	Leave open
F2	X20	S	X line connection	Leave open
F3	X21	S	X line connection	Leave open
F4	VDDIO	P	Digital IO interface power	–
F5	VDD	P	Digital power	–
F6	X10	S	X line connection	Leave open
F7	X9	S	X line connection	Leave open
F8	X8	S	X line connection	Leave open
G1	X22	S	X line connection	Leave open
G2	X23	S	X line connection	Leave open
G3	$\overline{\text{RESET}}$	I	Reset low. Connection to host system is recommended	Pull up to VddIO
G4	SDA	OD	Serial Interface Data	–
G5	NOISE_IN GPIO0 $\overline{\text{DBG\_SS}}$	I/O	Noise present input General purpose I/O Debug SS line; pull up to Vdd	Input: GND Output: leave open
G6	SCAN_OUT DBG_CLK GPIO2	I/O	Indicates touch scanning in progress. Polarity configurable Debug Clock General purpose I/O	Input: GND Output: leave open
G7	$\overline{\text{TEST}}$	–	Reserved for factory use. Pull up to VddIO	Pull up to VddIO
G8	X11	S	X line connection	Leave open
H1	EXTCAP1	P	Connect to EXTCAP0 via capacitor; see schematic notes	Leave open
H2	EXTCAP0	P	Connect to EXTCAP1 via capacitor; see schematic notes	Leave open
H3	SCL	OD	Serial Interface Clock	–
H4	$\overline{\text{CHG}}$	OD	State change interrupt Note: Briefly set (~100 ms) as an input after power-up/reset for diagnostic purposes	Pull up to VddIO

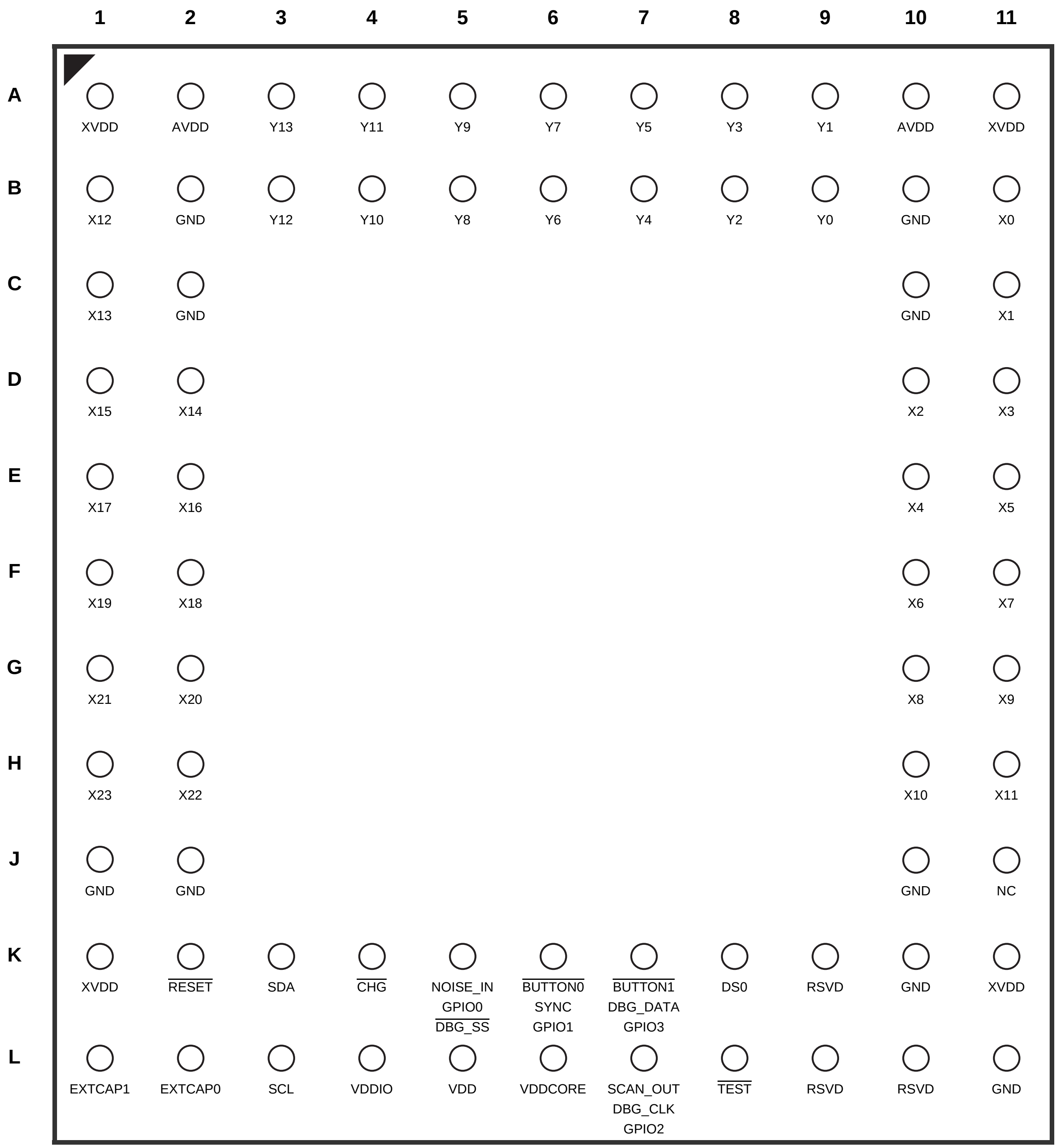
Table 2-2. Pin Listing – UFBGA 64 Balls

Pin	Name	Type	Description	If Unused...
H5	VDDCORE	P	Digital core power	–
H6	$\overline{\text{BUTTON0}}$ SYNC GPIO1	I/O	Touchpad button 0, active low External synchronisation General purpose I/O	Input: GND Output: leave open
H7	$\overline{\text{BUTTON1}}$ DBG_DATA GPIO3	I/O	Touchpad button 1, active low Debug Data General purpose I/O	Input: GND Output: leave open
H8	DS0	S	Driven Shield signal; used as guard track between X/Y signals and ground	Connect to GND

Key:

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

2.3 Pin Configuration – UFBGA 72 Balls



	1	2	3	4	5	6	7	8	9	10	11
A	XVDD	AVDD	Y13	Y11	Y9	Y7	Y5	Y3	Y1	AVDD	XVDD
B	X12	GND	Y12	Y10	Y8	Y6	Y4	Y2	Y0	GND	X0
C	X13	GND								GND	X1
D	X15	X14								X2	X3
E	X17	X16								X4	X5
F	X19	X18								X6	X7
G	X21	X20								X8	X9
H	X23	X22								X10	X11
J	GND	GND								GND	NC
K	XVDD	$\overline{\text{RESET}}$	SDA	$\overline{\text{CHG}}$	NOISE_IN GPIO0 $\overline{\text{DBG\_SS}}$	$\overline{\text{BUTTON0}}$ SYNC GPIO1	$\overline{\text{BUTTON1}}$ DBG_DATA GPIO3	DS0	RSVD	GND	XVDD
L	EXTCAP1	EXTCAP0	SCL	VDDIO	VDD	VDDCORE	SCAN_OUT DBG_CLK GPIO2	$\overline{\text{TEST}}$	RSVD	RSVD	GND

Top View

Table 2-3. Pin Listing – UFBGA 72 Balls

Pin	Name	Type	Description	If Unused...
A1	XVDD	P	X line drive power	Leave open
A2	AVDD	P	Analog power	–
A3	Y13	S	Y line connection	Leave open
A4	Y11	S	Y line connection	Leave open
A5	Y9	S	Y line connection	Leave open
A6	Y7	S	Y line connection	Leave open
A7	Y5	S	Y line connection	Leave open
A8	Y3	S	Y line connection	Leave open
A9	Y1	S	Y line connection	Leave open
A10	AVDD	P	Analog power	–
A11	XVDD	P	X line drive power	Leave open
B1	X12	S	X line connection	Leave open
B2	GND	P	Ground	–
B3	Y12	S	Y line connection	Leave open
B4	Y10	S	Y line connection	Leave open
B5	Y8	S	Y line connection	Leave open
B6	Y6	S	Y line connection	Leave open
B7	Y4	S	Y line connection	Leave open
B8	Y2	S	Y line connection	Leave open
B9	Y0	S	Y line connection	Leave open
B10	GND	P	Ground	–
B11	X0	S	X line connection	Leave open
C1	X13	S	X line connection	Leave open
C2	GND	P	Ground	–
…				
C10	GND	P	Ground	–
C11	X1	S	X line connection	Leave open
D1	X15	S	X line connection	Leave open
D2	X14	S	X line connection	Leave open
…				
D10	X2	S	X line connection	Leave open
D11	X3	S	X line connection	Leave open

Table 2-3. Pin Listing – UFBGA 72 Balls (Continued)

Pin	Name	Type	Description	If Unused...
E1	X17	S	X line connection	Leave open
E2	X16	S	X line connection	Leave open
…				
E10	X4	S	X line connection	Leave open
E11	X5	S	X line connection	Leave open
F1	X19	S	X line connection	Leave open
F2	X18	S	X line connection	Leave open
…				
F10	X6	S	X line connection	Leave open
F11	X7	S	X line connection	Leave open
G1	X21	S	X line connection	Leave open
G2	X20	S	X line connection	Leave open
…				
G10	X8	S	X line connection	Leave open
G11	X9	S	X line connection	Leave open
H1	X23	S	X line connection	Leave open
H2	X22	S	X line connection	Leave open
…				
H10	X10	S	X line connection	Leave open
H11	X11	S	X line connection	Leave open
J1	GND	P	Ground	–
J2	GND	P	Ground	–
…				
J10	GND	P	Ground	–
J11	NC	–	No connection	–
K1	XVDD	P	X line drive power	Leave open
K2	$\overline{\text{RESET}}$	I	Reset low. Connection to host system is recommended	Pull up to VddIO
K3	SDA	OD	Serial Interface Data	–
K4	$\overline{\text{CHG}}$	OD	State change interrupt Note: Briefly set (~100 ms) as an input after power-up/reset for diagnostic purposes	Pull up to VddIO
K5	NOISE_IN GPIO0 $\overline{\text{DBG\_SS}}$	I/O	Noise present input General purpose I/O Debug SS line; pull up to Vdd	Input: GND Output: leave open

Table 2-3. Pin Listing – UFBGA 72 Balls (Continued)

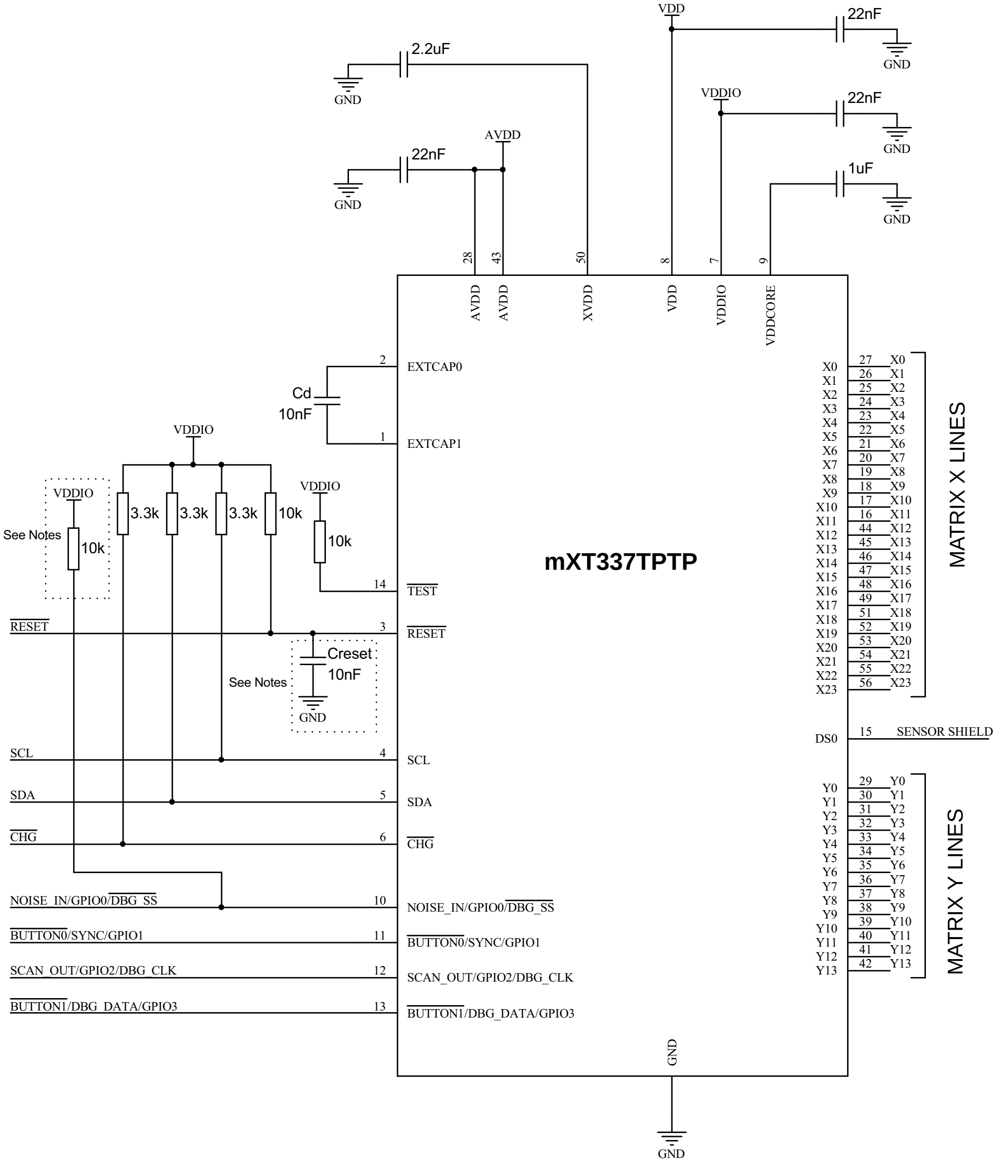
Pin	Name	Type	Description	If Unused...
K6	$\overline{\text{BUTTON0}}$ SYNC GPIO1	I/O	Touchpad button 0, active low External synchronisation General purpose I/O	Input: GND Output: leave open
K7	$\overline{\text{BUTTON1}}$ DBG_DATA GPIO3	I/O	Touchpad button 1, active low Debug Data General purpose I/O	Input: GND Output: leave open
K8	DS0	O	Driven Shield signal; used as guard track between X/Y signals and ground	Connect to GND
K9	RSVD	I/O	Reserved for future use	Leave open
K10	GND	P	Ground	–
K11	XVDD	P	X line drive power	Leave open
L1	EXTCAP1	P	Normal mode – leave open Voltage doubler mode – connect to EXTCAP0 via capacitor	Leave open
L2	EXTCAP0	P	Normal mode – leave open Voltage doubler mode – connect to EXTCAP1 via capacitor	Leave open
L3	SCL	OD	Serial Interface clock	–
L4	VDDIO	P	Digital IO interface power	–
L5	VDD	P	Digital power	–
L6	VDDCORE	P	Digital core power	–
L7	SCAN_OUT DBG_CLK GPIO2	I/O	Indicates touch scanning in progress. Polarity configurable Debug Clock General purpose I/O	Input: GND Output: leave open
L8	$\overline{\text{TEST}}$	–	Reserved for factory use. Pull up to VddIO	Pull up to VddIO
L9	RSVD	I/O	Reserved for future use	Leave open
L10	RSVD	I/O	Reserved for future use	Leave open
L11	GND	P	Ground	–

Key:

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

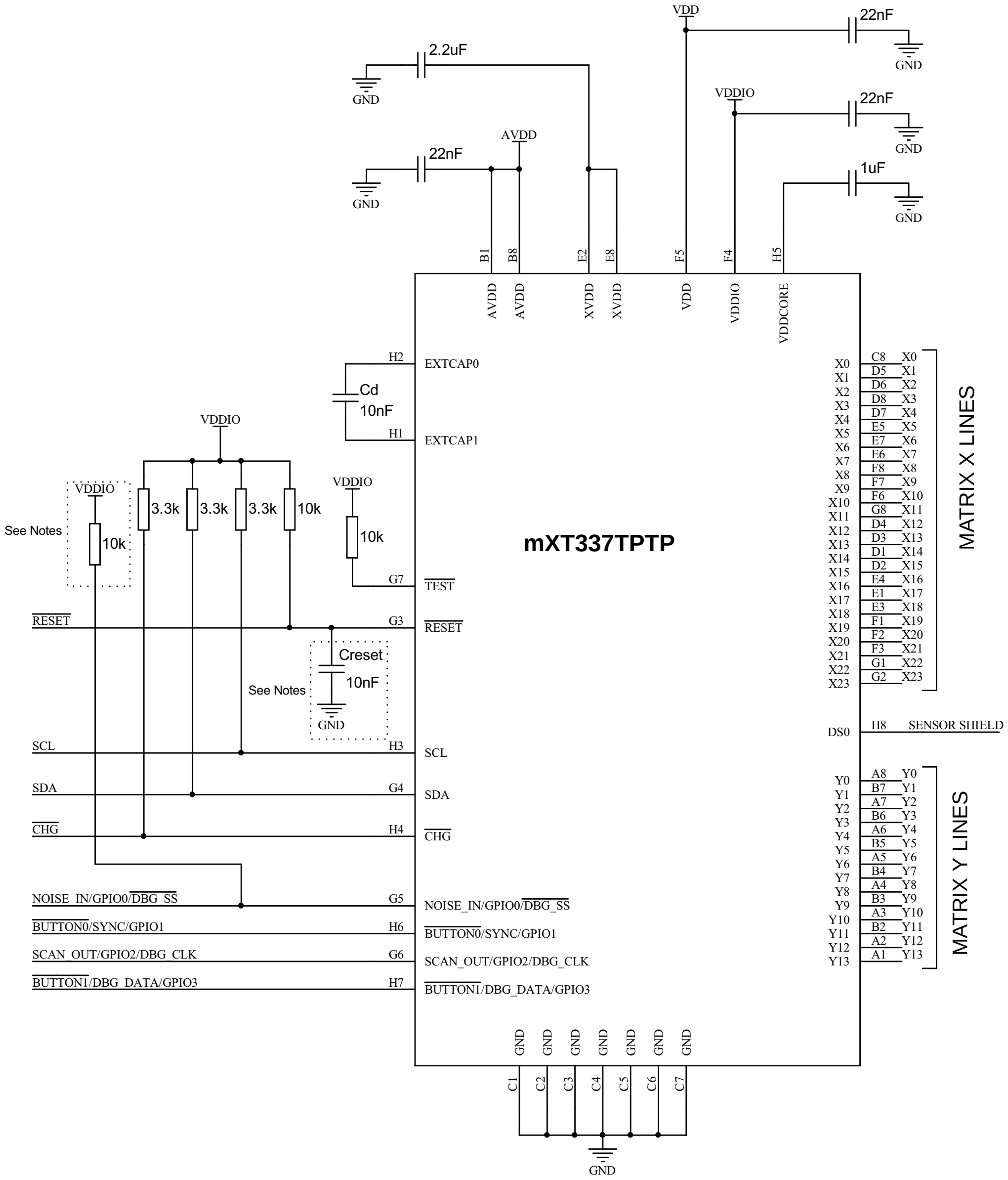
3. Schematics

3.1 56-pin UFQFN



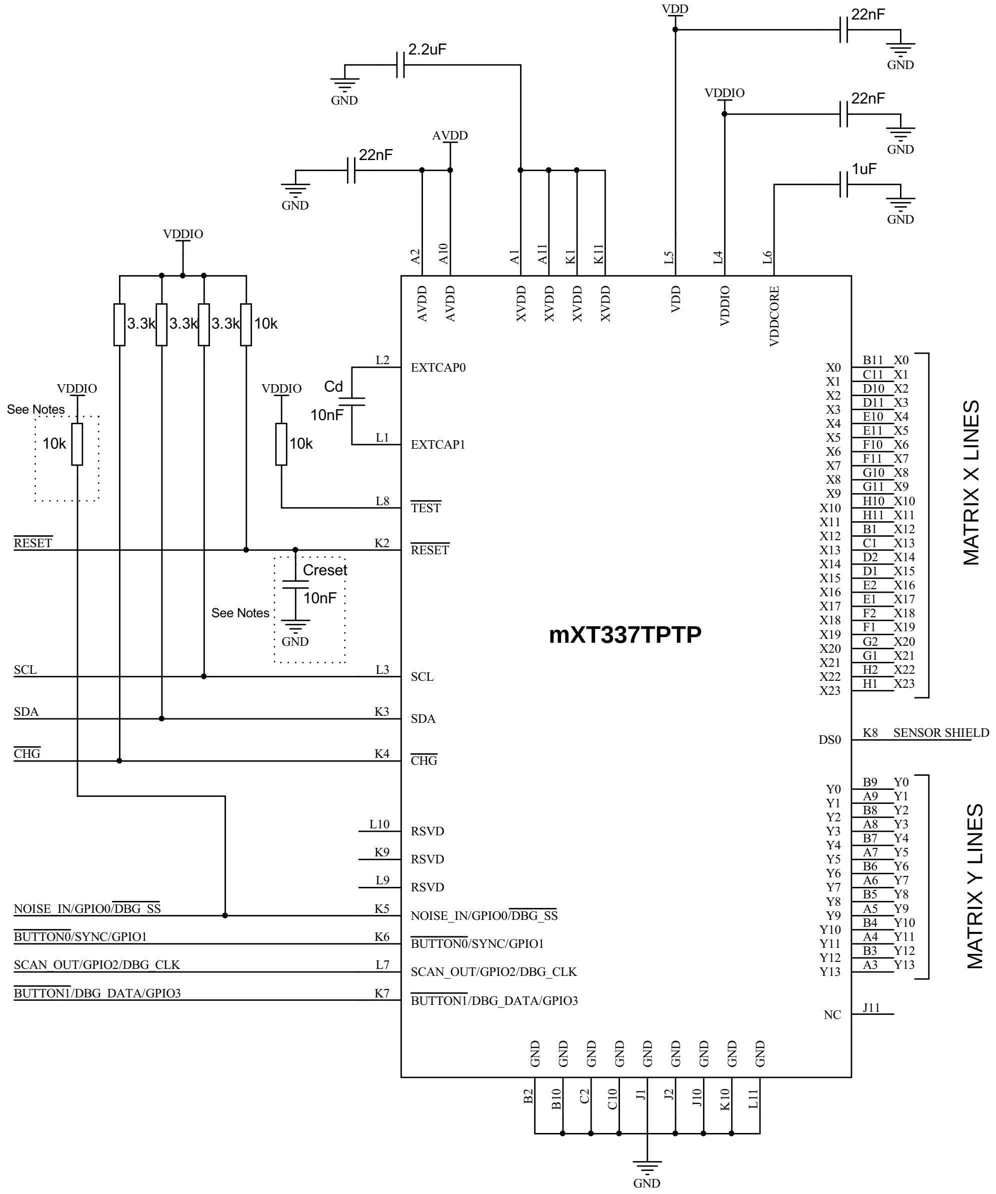
See Section 3.4 “Schematic Notes” on page 22

3.2 64-ball UFBGA



See Section 3.4 “Schematic Notes” on page 22

3.3 72-ball UFBGA

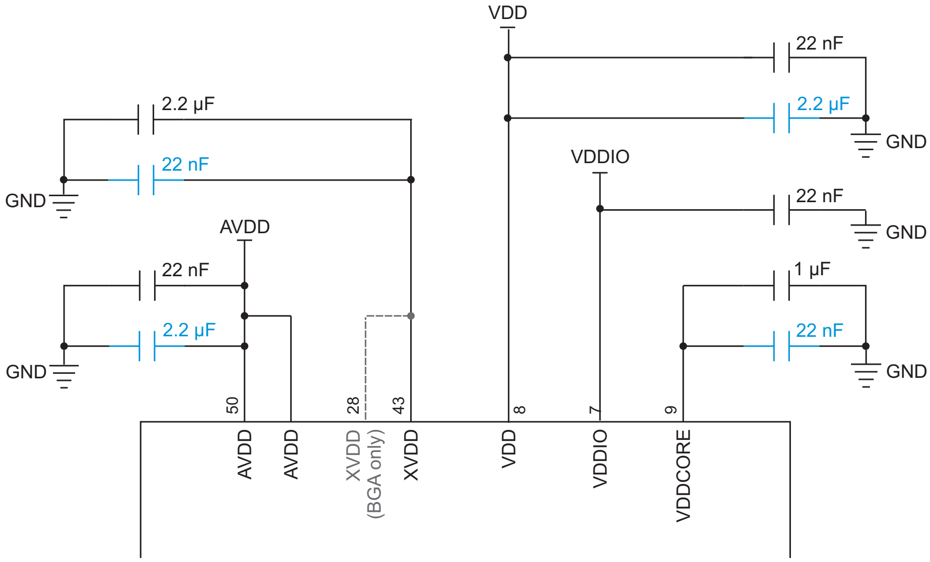


3.4 Schematic Notes

3.4.1 Decoupling capacitors:

1. Decoupling capacitors must be X7R or X5R and placed <5 mm away from the pins for which they act as bypass capacitors.
2. The schematics show the minimum capacitors required. If the ball configuration means that sharing a bypass capacitor is not possible, then the number of capacitors should be increased.
3. If the device is placed on the system board, the minimum number of capacitors required is as shown in the schematics on page 20 and page 19. Note that this requires that the voltage regulator supplies for Avdd/Vdd and VddIO are clean and noise free. It also assumes that the track length between the capacitors and on-board power supplies is < 50 mm.
4. If an active tail design is used, the voltage regulators are likely to be some distance from the device and it may be necessary to implement additional decoupling. In this case, a parallel combination of capacitors is recommended to give high and low frequency filtering, as shown in Figure 3-1.

Figure 3-1. Additional Recommended Decoupling Capacitors



NOTE: Recommended additional decoupling capacitors are shown in blue

3.4.2 VDDCORE

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

3.4.3 DBG_SS Line

The DBG_SS line shares the same pin as GPIO0. The GPIO function cannot be used when the SPI Debug port is in use. See Section 14. on page 61 for further information. A similar restriction applies to DBG_CLK and DBG_DATA.

The pull-up resistor in the schematics is optional and should be present only if the ball/pin is used as DBG_SS.

For more information refer to Application Note: *QTAN0050 Using the maXTouch Debug Port*.

3.4.4 RESET Line

The RESET line is shown on the schematics with a 10 nF capacitor (Creset) to ground. This capacitor is optional but may help if ESD issues are encountered.

3.4.5 Voltage Doubler

To use XVdd voltage doubler:

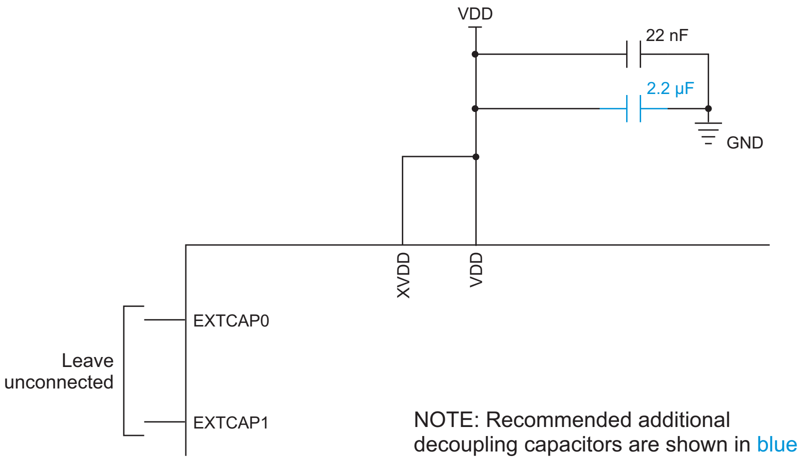
- EXTCAP0 must be connected to EXTCAP1 via a capacitor (Cd) to provide XVdd voltage doubler mode.
- The recommended value of the capacitor is 10 nF. Other values can be used if necessary after consultation with Atmel.
- The capacitor on XVDD should be rated at least 10 V if the voltage doubler is used.

If XVdd voltage doubler is not required:

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

These modifications are shown in Figure 3-2.

Figure 3-2. No voltage Doubler

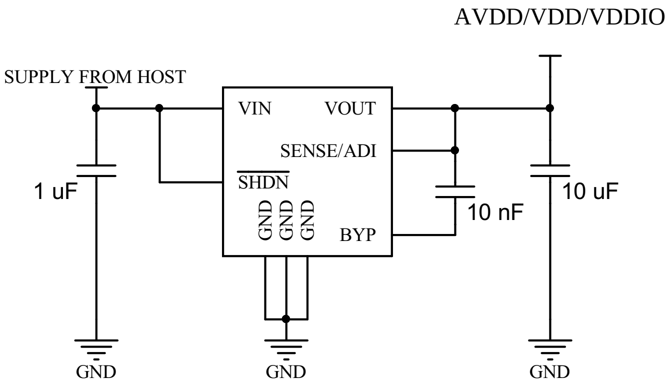


NOTE: Recommended additional decoupling capacitors are shown in blue

3.4.6 Low Drop-Out Voltage Regulators (LDOs)

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

Figure 3-3. Low Drop-Out Regulators



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

See Section 4.4 on page 24 for further details. A list of approved regulators is given in Table 4-1 on page 25.

4. Circuit Components

4.1 Decoupling Capacitors

Each power supply pin requires decoupling as described in Section 3.4 on page 22. The capacitors should be ceramic X7R or X5R.

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

4.2 I^{2}C Line Pull-up Resistors

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

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

4.3 Supply Quality

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

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

4.4 Suggested Voltage Regulators

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

Suitable fixed output LDO devices are shown in Table 4-1 on page 25.

With a single regulator, PCB layout is more critical than with multiple LDO regulators, and special care with the PCB layout should be taken. See Section 12.5 on page 55 for information concerning PCB design with a single LDO.

4.4.1 Multiple Voltage Regulator Supply

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

4.4.2 Suggested Voltage Regulators

The voltage regulators listed in Table 4-1 have been tested and found to work well with the mXT337TPTP.

Table 4-1. Suitable LDO Regulators

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

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

4.4.3 LDO Selection Criteria

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

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

5. Touchpad Basics

5.1 Sensor Construction

A touchpad is usually constructed from a number of transparent electrodes. These are typically on a glass or plastic substrate. They can also be made using non-transparent electrodes, such as copper or carbon. Electrodes are constructed from Indium Tin Oxide (ITO) or metal mesh. Thicker electrodes yield lower levels of resistance (perhaps tens to hundreds of Ω/square) at the expense of reduced optical clarity. Lower levels of resistance are generally more compatible with capacitive sensing. Thinner electrodes lead to higher levels of resistance (perhaps hundreds to thousands of Ω/square) with some of the best optical characteristics.

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

5.2 Electrode Configuration

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

The device supports various configurations of electrodes as summarized in Section 6. on page 28.

5.3 Scanning Sequence

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

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

5.4 Touchpad Sensitivity

5.4.1 Adjustment

Sensitivity of touchpads can vary across the extents of the electrode pattern due to natural differences in the parasitic capacitance of the interconnections, control chip, and so on. An important factor in the uniformity of sensitivity is the electrode design itself. It is a natural consequence of a touchpad 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 touchpad. This adjustment is a basic algorithmic threshold that defines when a node is considered to have enough signal change to qualify as being in detect.

5.4.2 Mechanical Stackup

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

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

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

6. Sensor Layout

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

The device supports various configurations of electrodes as summarized below:

- Touchpad: 24 X × 14 Y maximum (subject to other configurations)

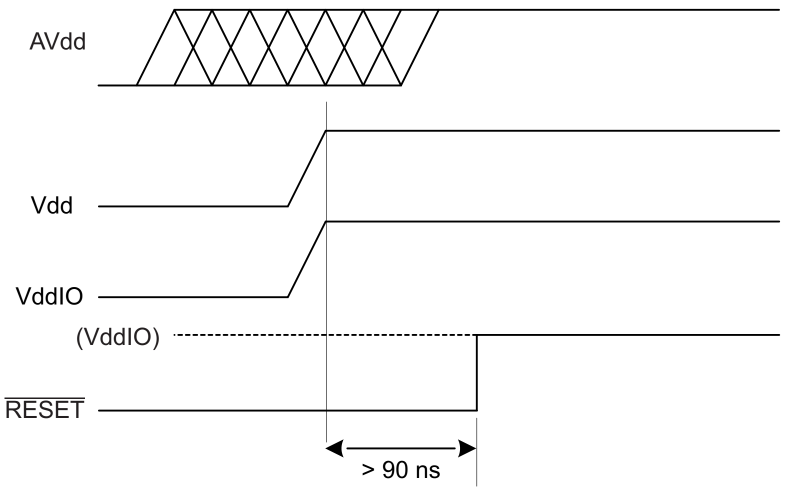
For optimal performance in terms of cycle time overhead, it is recommended that the number of X (drive) lines used for the key array is kept to the minimum and designs should favor using Y lines where possible.

7. Power-up / Reset Requirements

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

If an external reset is to be used the device must be held in $\overline{\text{RESET}}$ (active low) while the digital (Vdd) analog (AVdd) and I/O (VddIO) power supplies are powering up. The supplies must have reached their nominal values before the $\overline{\text{RESET}}$ signal is deasserted (that is, goes high). This is shown in Figure 7-1. See Section 15.2 on page 62 for nominal values for Vdd, VddIO, and AVdd.

Figure 7-1. Power Sequencing on the mXT337TPTP



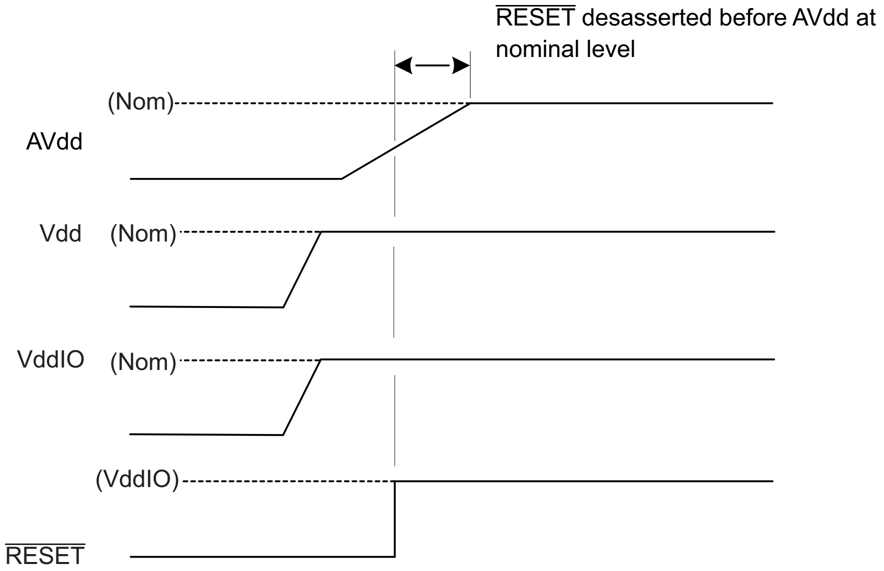
Note: When using external $\overline{\text{RESET}}$ at power-up,
VddIO must not be enabled after Vdd

After power-up, the device takes 79 ms before it is ready to start communications.

If the $\overline{\text{RESET}}$ line is released before the AVdd supply has reached its nominal voltage (see Figure 7-2 on page 30), then some additional operations need to be carried out by the host. There are two options open to the host controller:

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

Figure 7-2. Power Sequencing on the mXT337TPTP – Late rise on AVdd or XVdd



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 83 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.

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

Note that the voltage level on the $\overline{\text{RESET}}$ pin of the device must never exceed VddIO (digital supply voltage).

A software reset command can be used to reset the chip (refer to the Command Processor T6 object in the *mXT337TPTP 1.0 Protocol Guide*. A software reset takes typically 96 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 T6 object (refer to the *mXT337TPTP 1.0 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 Table 15.8.3 on page 70).

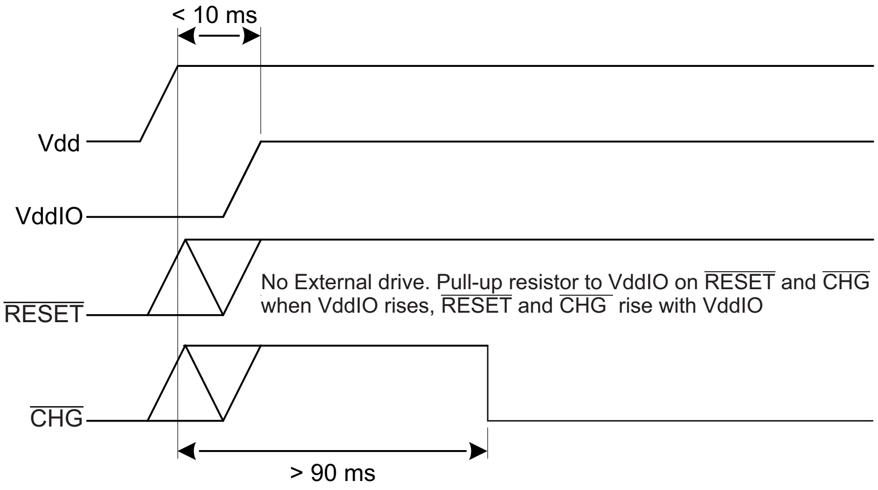
At power-on, the device performs a self-test routine to check for shorts that might cause damage to the device. Refer to the Self Test T25 object in the *mXT337TPTP 1.0 Protocol Guide* for more details about this process.

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

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

In this case the communication interface to the maXTouch device is not driven by the host system. The $\overline{\text{RESET}}$ and $\overline{\text{CHG}}$ pins are connected to VddIO using suitable pull-up resistors. Vdd is powered up, followed by VddIO, no more than 10 ms after Vdd. Due to the pull-up resistors, $\overline{\text{RESET}}$ and $\overline{\text{CHG}}$ will rise with VddIO. The internal POR system ensures reliable boot up of the device and the $\overline{\text{CHG}}$ line will go low approximately 79 ms after Vdd to notify the host that the device is ready to start communication.

Figure 7-3. Power-up Sequence



7.1.1 Summary

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

Table 7-1. Power-up and Reset Requirements

Condition	External $\overline{\text{RESET}}$	VddIO Delay (After Vdd)	AVdd Power-Up	Comments
1	Low at Power-up	0 ms	Before $\overline{\text{RESET}}$ is released	If AVdd bring-up is delayed then additional actions will be required by the host. See notes in Figure 7-1 on page 29
2	Not driven	<10 ms	Before VddIO	

8. Detailed Operation

8.1 Touch Detection

The mXT337TPTP allows for both mutual and self capacitance measurements, with the self capacitance measurements being used to augment the mutual capacitance measurements to produce reliable touch information.

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

Mutual capacitance touch data is used wherever possible to classify touches as this has greater granularity than self capacitance measurements and provides positional information on touches. Refer to the *mXT337TPTP 1.0 Protocol Guide* for more information on measurements.

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

8.2 Operational Modes

The device operates in two modes: Active (touch detected) and Idle (no touches detected). Both modes operate as a series of burst cycles. Each cycle consists of a short burst (during which measurements are taken) followed by an inactive sleep period. The difference between these modes is the length of the cycles. Those in idle mode typically have longer sleep periods. The cycle length is configured using the IDLEACQINT and ACTVACQINT settings in the Power Configuration T7. In addition, an *Active to Idle Timeout* setting is provided.

Refer to the *mXT337TPTP 1.0 Protocol Guide* for full information on how these modes operate, and how to use the settings provided.

8.3 Detection Integrator

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

The detection integrator is configured using the appropriate touch objects (Multiple Touch Touchscreen T100). Refer to the *mXT337TPTP 1.0 Protocol Guide* for more information.

8.4 Sensor Acquisition

The maximum acquisition time for one X line on the mXT337TPTP is 5 µs. Care should be taken to ensure that the total time for one X line configured by the Acquisition Configuration T8 and CTE Configuration T46 objects do not exceed this (refer to the *mXT337TPTP 1.0 Protocol Guide* for details on these objects).

8.5 Calibration

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

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

or

- The node is already enabled and one of the following applies:
  - The node is held in detect for longer than the Touch Automatic Calibration setting (refer to the *mXT337TPTP 1.0 Protocol Guide* for more information on TCHAUTOCAL setting in the Acquisition Configuration object).

- The signal delta on a node is at least the touch threshold (TCHTHR) in the anti-touch direction, while it meets the criteria in the Touch Recovery Processes that results in a recalibration. (Refer to the *mXT337TPTP 1.0 Protocol Guide* for objects Acquisition Configuration T8 and Self Capacitance Configuration T111).
- The host issues a recalibrate command.
- Certain configuration settings are changed.

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

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

8.6 Digital Filtering and Noise Suppression

The mXT337TPTP supports on-chip filtering of the acquisition data received from the sensor. Specifically, the maXCharger T72 object provides an algorithm to suppress the effects of noise (for example, from a noisy charger plugged into the user's product). This algorithm can automatically adjust some of the acquisition parameters on-the-fly to filter the analog-to-digital conversions (ADCs) received from the sensor.

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

Noise suppression is triggered when a noise source is detected.

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

Refer to the *mXT337TPTP 1.0 Protocol Guide* for more information on the maXCharger T72 and Self Capacitance maXCharger T108 objects.

8.7 Shieldless Support and Display Noise Suppression

The mXT337TPTP can support shieldless sensor design even with a noisy LCD by using the following features.

- **Optimal Integration:** This feature is not filtering as such, but enables the user to use a shorter integration window. The integration window optimizes the amount of charge collected against the amount of noise collected, to ensure an optimal SNR. This feature also benefits the system in the presence of an external noise source. This feature is configured using the Shieldless T56 object. Refer to the *mXT337TPTP 1.0 Protocol Guide* for more information
- **Display noise suppression:** This feature is based on filtering provided by the Lens Bending T65 object (See Section 8.9 on page 34). This feature allows the device to overcome display noise simultaneously with external noise. Refer to the *mXT337TPTP 1.0 Protocol Guide* for more information

8.8 Retransmission Compensation

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

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

Refer to the *mXT337TPTP 1.0 Protocol Guide* for more information on the Retransmission Compensation T80 object.

8.9 Lens Bending

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

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

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

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

Refer to the *mXT337TPTP 1.0 Protocol Guide* for more information on the Lens Bending T65 object.

8.10 Stylus Support

The mXT337TPTP allows for the particular characteristics of passive stylus touches, whilst still allowing conventional finger touches to be detected. The touch sensitivity and threshold controls for stylus touches are configured separately from those for conventional finger touches so that both types of touches can be accommodated.

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

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

Refer to the *mXT337TPTP 1.0 Protocol Guide* for more information on configuring a stylus.

8.11 Unintentional Touch Suppression

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

Refer to the *mXT337TPTP 1.0 Protocol Guide* for more information on the Touch Suppression T42 object.

8.12 GPIO Pins

The mXT337TPTP has 4 GPIO pins. The pins can be set to be either an input or an output, as required. Note that unused GPIO pins can be left externally unconnected as long as they are given a defined state by using the GPIO/PWM Configuration T19 object. With the GPIO/PWM Configuration T19 object, an unused GPIO pin can be either set to Input mode, with internal pull-up, or Output mode. If the GPIO/PWM Configuration T19 is not enabled for use, all the GPIO pins are unused.

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

9. Host Communications

Communication with the host is achieved using one of the following interfaces:

- I^2C (see Section 10. on page 36)
- HID-I^2C (see Section 11. on page 42)

Either interface can be used, depending on the needs of the project, but only one interface should be used in any one design.

The device will listen on the I^2C line and automatically select the protocol to be used depending on the address contained in the first message received:

Address detected	Communications Mode
0x4B	I^2C
0x4A	HID-I2C

10. I^{2}C Communications

The device can use an I^{2}C interface for communication.

The I^{2}C interface is used in conjunction with the $\overline{\text{CHG}}$ line. The $\overline{\text{CHG}}$ line going active signifies that a new data packet is available. This provides an interrupt-style interface and allows the device to present data packets when internal changes have occurred.

10.1 I^{2}C Addresses

The I^{2}C address is shifted left to form the SLA+W or SLA+R address when transmitted over the I^{2}C interface, as shown in Table 10-1.

Table 10-1. Format of an I^{2}C Address

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

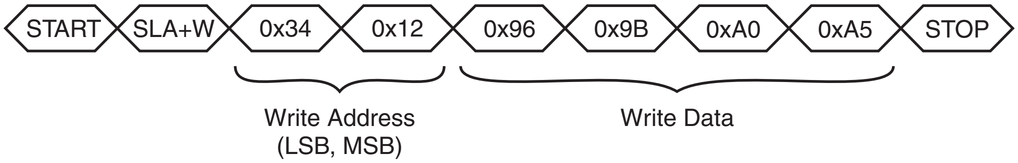
10.2 Writing To the Device

A WRITE cycle to the device consists of a START condition followed by the I^{2}C address of the device (SLA+W). The next two bytes are the address of the location into which the writing starts. The first byte is the Least Significant Byte (LSByte) of the address, and the second byte is the Most Significant Byte (MSByte). This address is then stored as the address pointer.

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

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

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

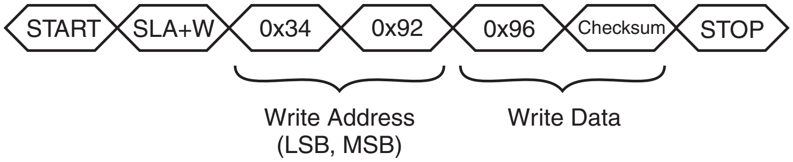


10.3 I^{2}C Writes in Checksum Mode

In I^{2}C checksum mode an 8-bit CRC is added to all I^{2}C writes. The CRC is sent at the end of the data write as the last byte before the STOP condition. All the bytes sent are included in the CRC, including the two address bytes. Any command or data sent to the device is processed even if the CRC fails.

To indicate that a checksum is to be sent in the write, the most significant bit of the MSByte of the address is set to 1. For example, the I^{2}C command shown in Figure 10-2 writes a value of 150 (0x96) to address 0x1234 with a checksum. The address is changed to 0x9234 to indicate checksum mode.

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



10.4 Reading From the Device

Two I^2C bus activities must take place to read from the device. The first activity is an I^2C write to set the address pointer (LSByte then MSByte). The second activity is the actual I^2C read to receive the data. The address pointer returns to its starting value when the read cycle NACK is detected.

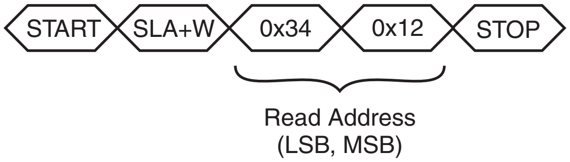
It is not necessary to set the address pointer before every read. The address pointer is updated automatically after every read operation. The address pointer will be correct if the reads occur in order. In particular, when reading multiple messages from the Message Processor T5 object, the address pointer is automatically reset to allow continuous reads (see Section 10.5).

The WRITE and READ cycles consist of a START condition followed by the I^2C address of the device (SLA+W or SLA+R respectively).

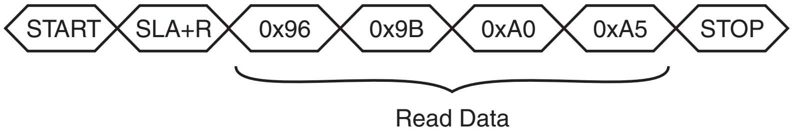
Figure 10-3 shows the I^2C commands to read four bytes starting at address 0x1234.

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

Set Address Pointer



Read Data



10.5 Reading Status Messages with DMA

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

1. The host uses a write operation to set the address pointer to the start of the Message Count T44 object, if necessary (1). If a checksum is required on each message, the most significant bit of the MSByte of the read address must be set to 1.
2. The host starts the read operation of the message by sending a START condition.
3. The host reads the Message Count T44 object (one byte) to retrieve a count of the pending messages (refer to the *mXT337TPTP 1.0 Protocol Guide* for details).
4. The host calculates the number of bytes to read by multiplying the message count by the size of the Message Processor T5 object (2).
5. Note that the size of the Message Processor T5 object as recorded in the Object Table includes a checksum byte. If a checksum has not been requested, one byte should be deducted from the size of the object. That is: number of bytes = count × (size – 1).
6. The host reads the calculated number of message bytes. It is important that the host does *not* send a STOP condition during the message reads, as this will terminate the continuous read operation and reset the address pointer. No START and STOP conditions must be sent between the messages.

---

(1) The STOP condition at the end of the read resets the address pointer to its initial location, so it may already be pointing at the Message Count T44 object following a previous message read.

(2) The host should have already read the size of the Message Processor T5 object in its initialization code.

7. The host sends a STOP condition at the end of the read operation after the last message has been read. The NACK condition immediately before the STOP condition resets the address pointer to the start of Message Count T44 object.

Figure 10-4 shows an example of using a continuous read operation to read three messages from the device without a checksum. Figure 10-5 on page 39 shows the same example with a checksum.

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

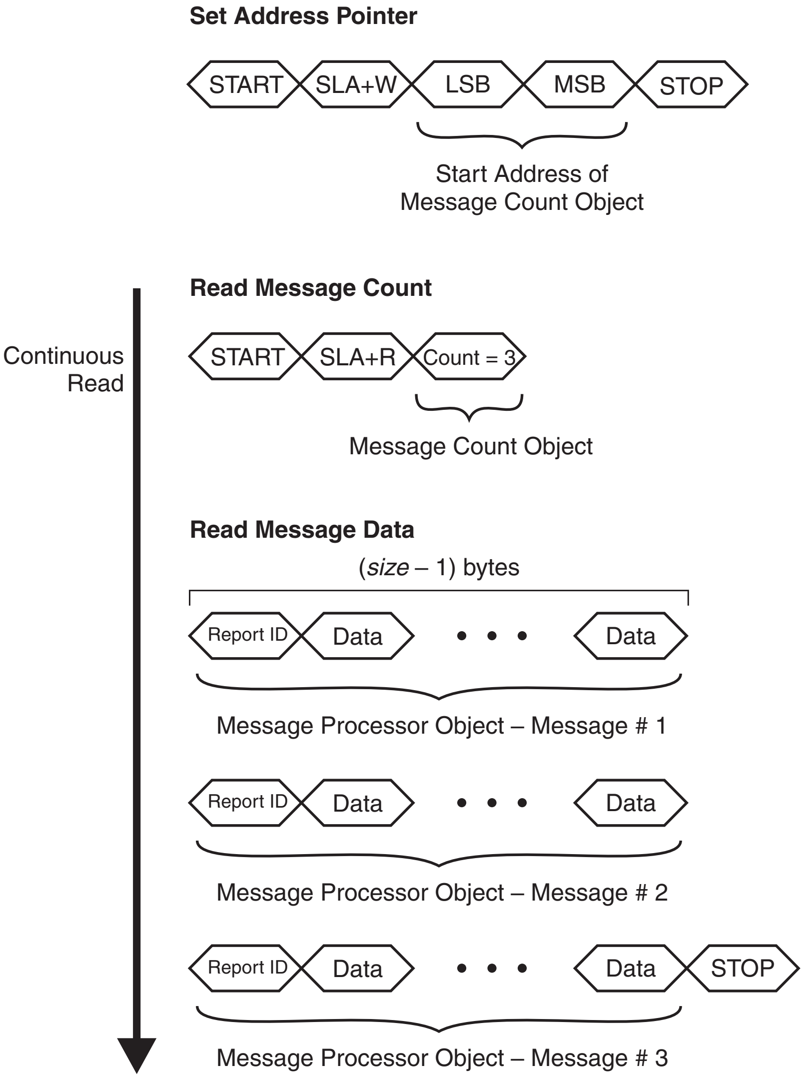
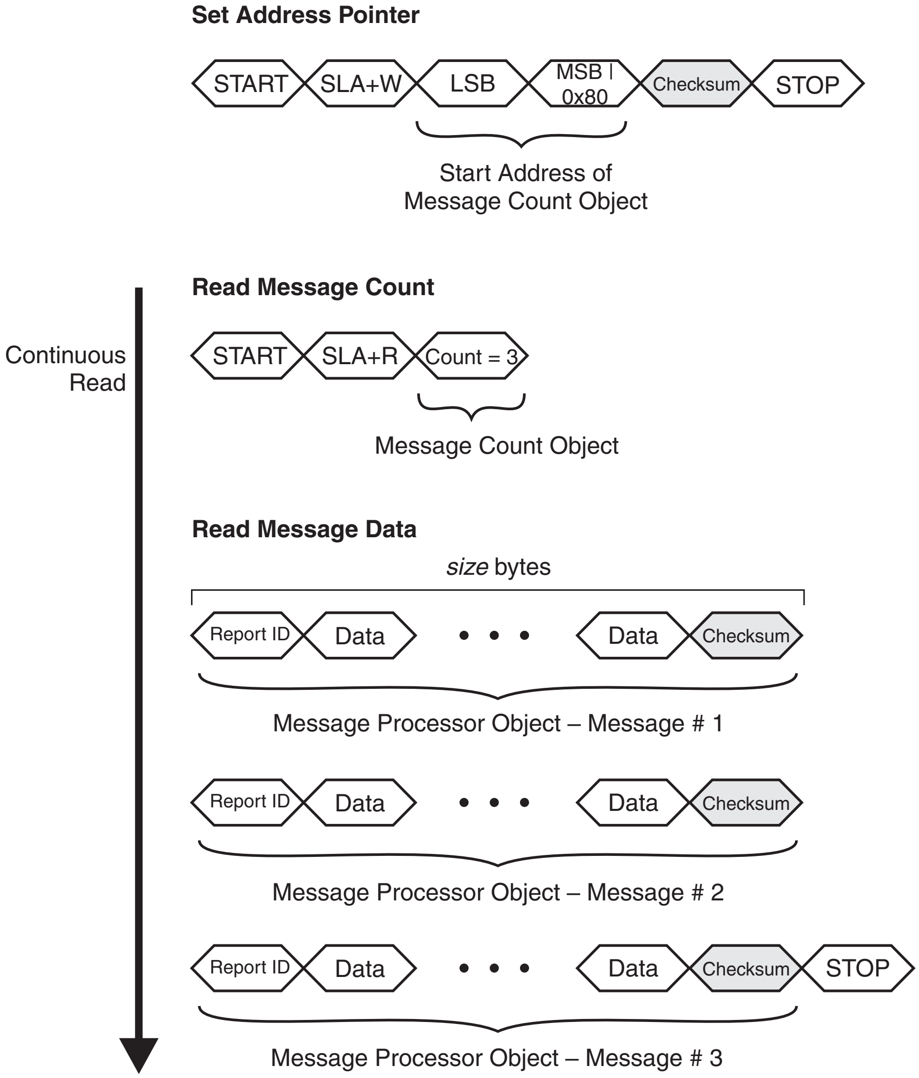


Figure 10-5. Continuous Message Read Example – I^{2}C Checksum Mode



There are no checksums added on any other I^{2}C reads. An 8-bit CRC can be added, however, to all I^{2}C writes, as described in Section 10.3 on page 36.

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

10.6 $\overline{\text{CHG}}$ Line

The $\overline{\text{CHG}}$ line is an active-low, open-drain output that is used to alert the host that a new message is available in the Message Processor T5 object. This provides the host with an interrupt-style interface with the potential for fast response times. It reduces the need for wasteful I^{2}C communications.

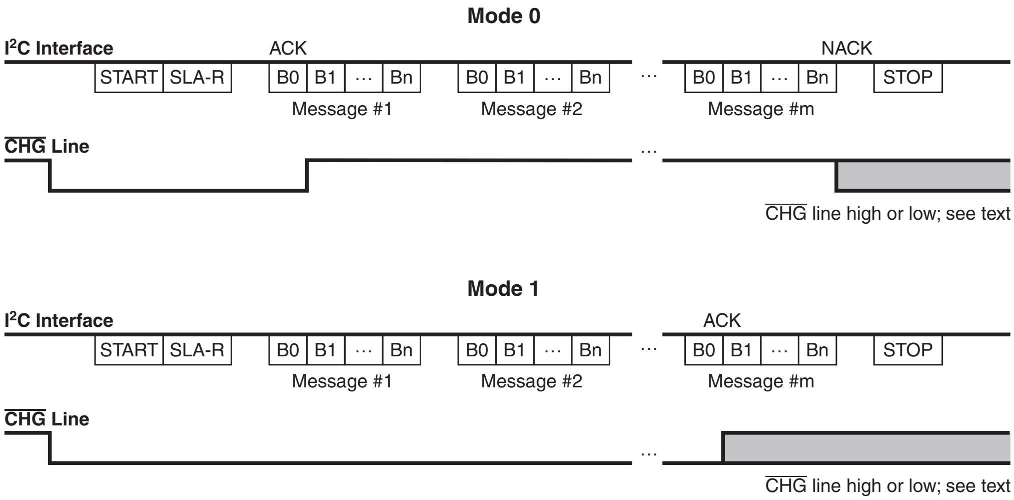
The $\overline{\text{CHG}}$ line remains low as long as there are messages to be read. The host should be configured so that the $\overline{\text{CHG}}$ line is connected to an interrupt line that is level-triggered. The host should not use an edge-triggered interrupt as this means adding extra software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

The $\overline{\text{CHG}}$ line should be allowed to float during normal usage. This is particularly important after power-up or reset (see Section 7. on page 29).

A pull-up resistor is required, typically 3.3 kΩ to VddIO.

The $\overline{\text{CHG}}$ line operates in two modes, as defined by the Communications Configuration T18 object (refer to the *mXT337TPTP 1.0 Protocol Guide*).

Figure 10-6. $\overline{\text{CHG}}$ **Line Modes for I^{2}C-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 I^{2}C 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 I^{2}C read.

In Mode 1:

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

Mode 1 allows the host to continually read the messages until the $\overline{\text{CHG}}$ line goes high, and the state of the $\overline{\text{CHG}}$ line determines whether or not the host should continue receiving messages from the device.

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

The Communications Configuration T18 object can be used to configure the behavior of the $\overline{\text{CHG}}$ line. In addition to the $\overline{\text{CHG}}$ line operation modes described above, this object allows the use of edge-based interrupts, as well as direct control over the state of the $\overline{\text{CHG}}$ line. Refer to the *mXT337TPTP 1.0 Protocol Guide* for more information.

10.7 SDA, SCL

The I^{2}C bus transmits data and clock with SDA and SCL, respectively. These are open-drain. The device can only drive these lines low or leave them open. The termination resistors (Rp) pull the line up to Vdd if no I^{2}C device is pulling it down.

The termination resistors should be chosen so that the rise times on SDA and SCL meet the I^{2}C specifications for the interface speed being used, bearing in mind other loads on the bus (see Section 15.10 on page 70).

10.8 Clock Stretching

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

The device has an internal bus monitor that can reset the internal I^2C hardware if SDA or SCL is stuck low for more than 200 ms. This means that if a prolonged clock stretch of more than 200 ms is seen by the device, then any ongoing transfers with the device may be corrupted. The bus monitor is enabled or disabled using the Communications Configuration T18 object. Refer to the *mXT337TPTP 1.0 Protocol Guide* for more information.

11. HID-I^2C Communications

The device is an HID-I^2C device presenting two Top-level Collections (TLCs):

- **Generic HID-I^2**C – Provides a generic HID-I^2C interface that allows the host to communicate with the device using the object-based protocol (OBP).
- **Digitizer HID-I^2**C – Supplies touch information to the host. This interface is supported by Microsoft Windows without the need for additional software.

See Section 9. on page 35 for information on selecting HID-I^2C mode.

Other features are identical to standard I^2C communication described in Section 10. on page 36.

Refer to the Microsoft HID-I^2C documentation, *HID Over I^2C Protocol Specification – Device Side*, for information on the HID-I^2C specification.

11.1 I^2C Addresses

See Section 10.1 on page 36.

11.2 Device

The device is compliant with HID-I^2C specification V1.0. It has the following specification:

Vendor ID:	0x03EB (Atmel)
Product ID:	0x2168 (mXT337TPTP)
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

11.3 HID Descriptor

The host should read the HID descriptor on initialization to ascertain the key attribute of the HID device. These include the report description and the report ID to be used for communication with the HID device. The HID descriptor address is 0x0000.

Note that the host driver must not make any assumptions about the report packet formats, data locations or report IDs. These must be read from the HID descriptor as they may change in future versions of the firmware.

For more information on how to read the HID descriptor, refer to the Microsoft HID-I^2C documentation.

11.4 HID-I^2C Report IDs

Table 11-1 describes the HID-I^2C report IDs used in reports sent to the host.

Table 11-1. HID-I^2C Report IDs

Report ID	Description	Top-level Collection
0x06	Object Protocol (OBP) command and response (see Section 11.5 on page 43)	Generic HID-I^2C
0x07	Touch report for Touchpad Mode (see Section 11.6.1 on page 48)	Digitizer HID-I^2C
0x08	Maximum Touches (Surface Contacts) report (see Section 11.6.3 on page 50)	Digitizer HID-I^2C
0x0A	Input Mode command and report (see Section 11.6.4 on page 51)	Digitizer HID-I^2C
0x0B	Touch Hardware Quality Assurance (THQA) report (see Section 11.6.5 on page 51)	Digitizer HID-I^2C

Table 11-1. HID-I^{2}C Report IDs

Report ID	Description	Top-level Collection
0x0C	Mouse report for Mouse Mode (see Section 11.6.2 on page 49)	Digitizer HID-I^{2}C
0x0D	Reporting Mode command and report (see Section 11.6.6 on page 51)	Digitizer HID-I^{2}C
0x0E	Latency Mode command and report (see Section 11.6.7 on page 52)	Digitizer HID-I^{2}C

11.5 Generic HID-I^{2}C

The Generic HID-I^{2}C TLC 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

The HID-I^{2}C Report ID used is that for Object Protocol commands and responses; see Table 11-1 on page 42 for the value.

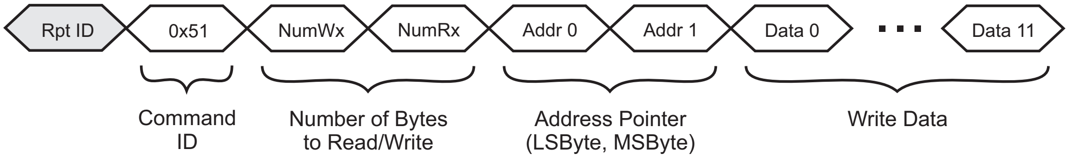
11.5.1 Read/Write Memory Map

11.5.1.1 Introduction

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

The command packet has the generic format given in Figure 11-1. The following sections give examples on using the command to write to the memory map and to read from the memory map.

Figure 11-1. Generic Command Packet Format

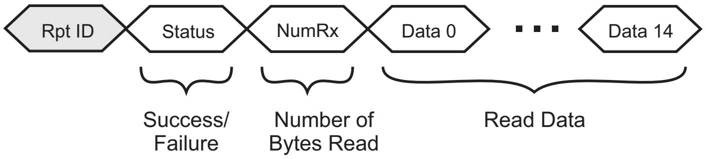


In Figure 11-1:

- **Rpt ID** is the HID-I^2**C** Report ID used for Object Protocol commands and responses (see Table 11-1 on page 42).
- **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 11-2.

Figure 11-2. Response Packet Format



In Figure 11-2:

- **Rpt ID** is the HID-I^2**C** Report ID used for Object Protocol commands and responses (see Table 11-1 on page 42).
- **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.

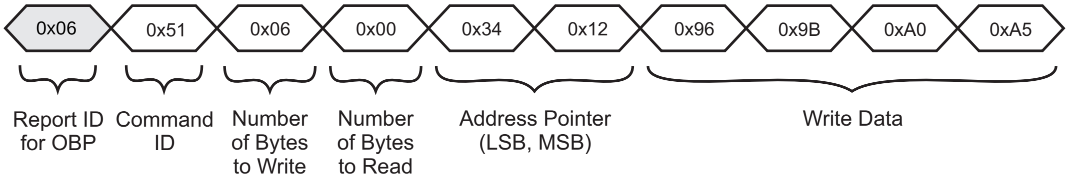
11.5.1.2 Writing To the Device

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

Subsequent bytes in a 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.

Figure 11-3 shows an example command packet to write four bytes of data to contiguous addresses starting at 0x1234.

Figure 11-3. Example of a Four-byte Write Starting at Address 0x1234

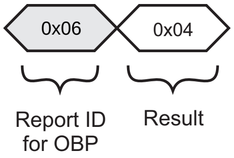


In Figure 11-3:

- **Rpt ID** is the HID-I^2**C** Report ID used for Object Protocol commands and responses (see Table 11-1 on page 42).
- 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 11-4 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). Note also that the Report ID is the same one used in the command packet.

Figure 11-4. Response to Example Four-byte Write

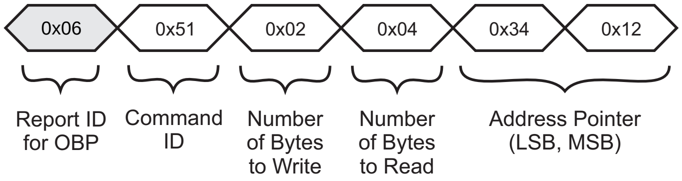


11.5.1.3 Reading From the Device

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

Figure 11-5 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.

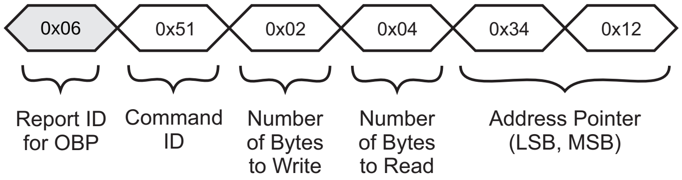
Figure 11-5. Example of a Four-byte Read Starting at Address 0x1234



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 11-6 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).

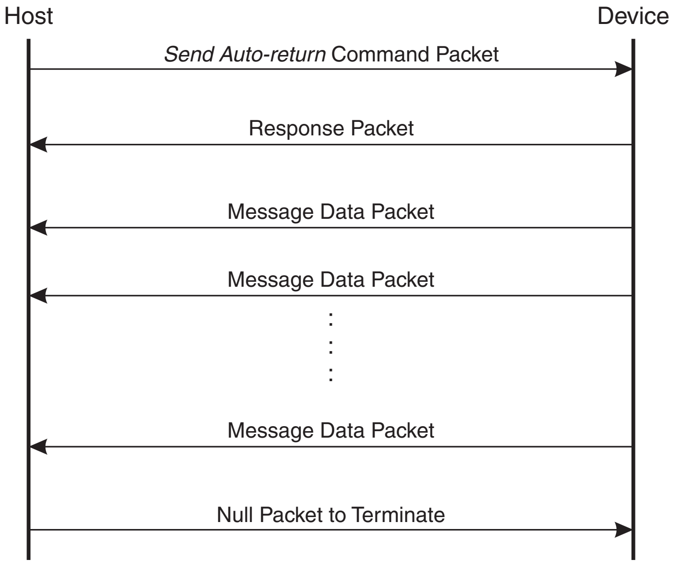
Figure 11-6. Response to Example Four-byte Read



11.5.2 Send Auto-return Messages

With this command the device can be configured to return new messages from the Message Processor T5 object autonomously. The packet sequence to do this is shown in Figure 11-7.

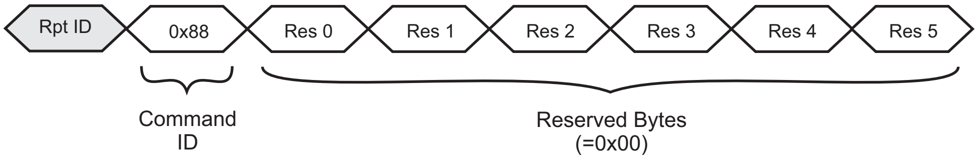
Figure 11-7. Packet Sequence for “Send Auto-return” Command



The HID-I^{2}C Report ID used is that for Object Protocol commands and responses; see Table 11-1 on page 42 for the value.

The command packet has the format given in Figure 11-8.

Figure 11-8. Command Packet Format

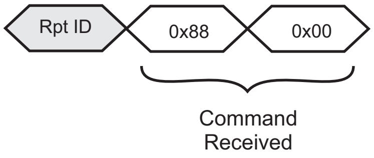


In Figure 11-8:

- **Rpt ID** is the HID-I^2**C** Report ID used for Object Protocol commands and responses (see Table 11-1 on page 42).
- **Res 0** to **Res 5** are reserved bytes with a value of 0x00.

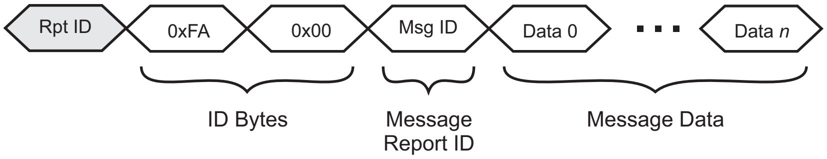
The response packet has the format given in Figure 11-9. Note that with this command, the command packet does not include an address pointer as the device already knows the address of the Message Processor T5 object.

Figure 11-9. Response Packet Format



Once the device has responded to the command, it starts sending message data. Each time a message is generated in the Message Processor T5 object, the device automatically sends a message packet to the host with the data. The message packets have the format given in Figure 11-10.

Figure 11-10. Message Packet Format

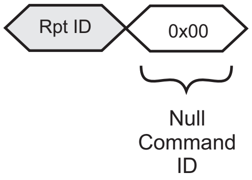


In Figure 11-10:

- **Rpt ID** is the HID-I^{2}C Report ID used for Object Protocol commands and responses (see Table 11-1 on page 42).
- **ID Bytes** identify the packet as an auto-return message packet.
- **Msg ID** is the Report ID returned by the Message Processor T5 object.
- **Message Data** bytes are the bytes of data returned by the Message Processor T5 object. The size of the data depends on the source object for which this is the message data. Refer to the *mXT337TPTP 1.0 Protocol Guide* for more information.

To stop the sending of the messages, the host can send a null command packet. This consists of two bytes: a HID-I^{2}C report ID and a command byte of 0x00 (see Figure 11-11).

Figure 11-11. Null Command Packet Format

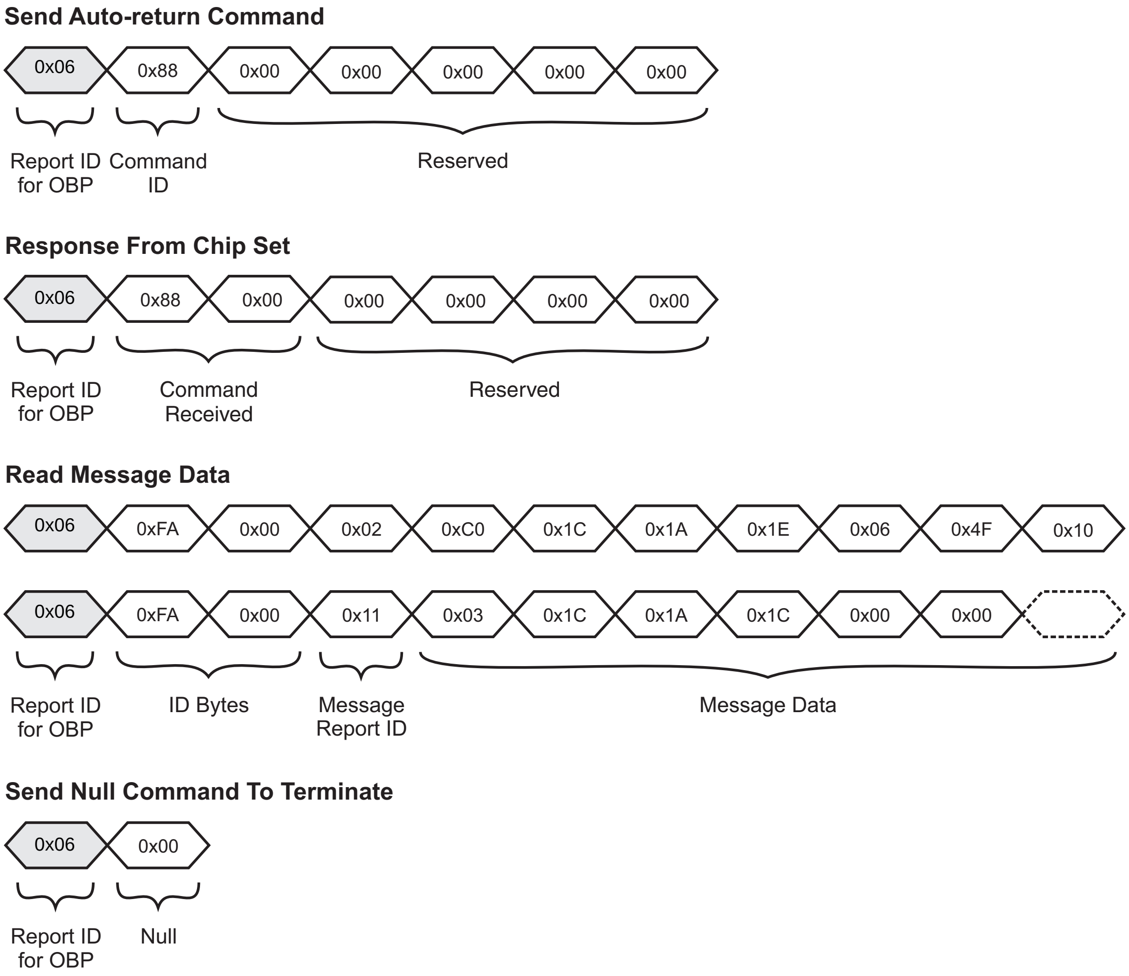


Note that any read or write will also terminate any currently enabled auto-return mode (see Section 11.5.1.2 on page 44).

11.5.2.1 Reading Status Messages

Figure 11-12 shows an example sequence of packets to receive messages from the Message Processor T5 object using the “Send Auto-return” command.

Figure 11-12.Example Auto-return Command Packet



11.6 Digitizer HID-I^{2}C

This is a digitizer class HID.

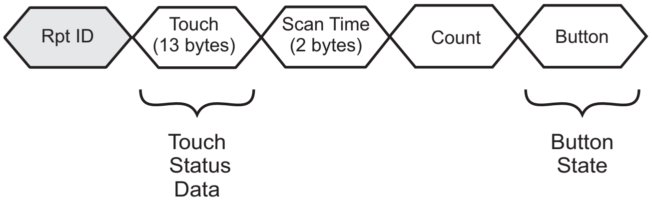
11.6.1 Touch Report

The format of a Touch report packet is shown in Figure 11-13. This is sent when the touchpad is in Touchpad Mode (see Section 11.6.4 on page 51).

Each Touch report starts with a report ID and contains the data for one touch.

The HID-I^{2}C Report ID used is that for Touch reports; see Table 11-1 on page 42 for the value.

Figure 11-13.Touch Report Packet Format

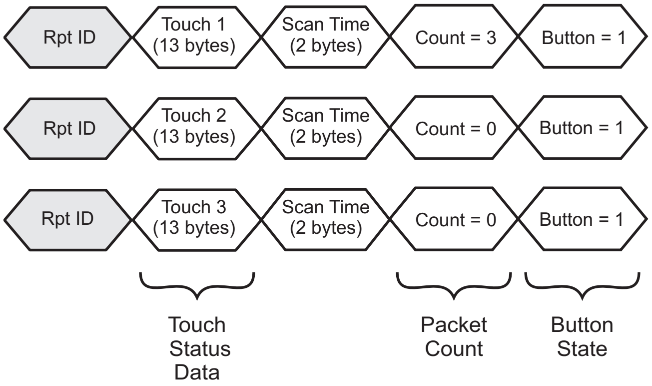


In Figure 11-13:

- **Rpt ID** is the HID-I^{2}C Report ID used for Touch reports (see Table 11-1 on page 42).
- **Touch** is the data for the touch.
- **Scan Time** is the Timestamp for the report packet.
- **Count** is used to identify the report packets for current active touches that are to be reported as a single package. The Count in the first packet for the first touch is set to the number of active touches to be sent in one package. Subsequent packets for subsequent active touches have a Count of 0.
- **Button** is the status of the touchpad button (BUTTON0): 0 = not pressed, 1 = pressed

An example of the input report packets for 3 active touches is shown in Figure 11-14. In this example there are three touches, with a press on the touchpad button.

Figure 11-14.Example Touch Report Packets for 3 Active Touches



Each input report consists of a HID-I^{2}C report ID followed by 17 bytes of that describe the status of one active touch. Table 11-2 explains the detailed format of an input report packet.

Table 11-2. Touch Report Format

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	HID-I^{2}C Touch Report ID							
1	Reserved			Touch ID			Confidence	Status
2 – 3	X Position							
4 – 5	X’ Position							
6 – 7	Y Position							
8 – 9	Y’ Position							
10	Touch Width							
11	Reserved							
12	Touch Height							
13	Reserved							
14 – 15	Scan Time							
16	Count							
17	Reserved							Button

- **Byte 0:**

  The HID-I^{2}C Report ID for Touch reports (see Table 11-1 on page 42).
- **Byte 1:**

  Status: Status of the touch detection. This bit is set to 1 if touch is detected, and set to 0, if no touches are detected.

  Confidence: Specifies the confidence of the detected touch. 1 = normal touch, 0 = large touch.

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

  X’ and Y’ Positions: X and Y centre positions.
- **Byte 10:**

  Touch Width: Reports the width of the detected touch.
- **Byte 12:**

  Touch Height: Reports the height of the detected touch.
- **Byte 14 to 15:**

  Scan Time: Timestamp associated with the current report packet with a 10 kHz resolution.
- **Byte 16:**

  Count: For the first touch, the number of active touches to be sent in one package. Subsequent packets for subsequent active touches have a Count of 0.
- **Byte 17:**

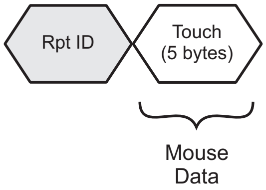
  Button: Reports the status of the touchpad button: 0 = not pressed, 1 = pressed.

11.6.2 Mouse Mode Report

The format of a Mouse Mode report packet is shown in Figure 11-15 on page 50. This is sent when the touchpad is in Mouse Mode. See Section 11.6.1 on page 48 for the format of Touchpad Mode reports.

The HID-I^{2}C Report ID used is that for Mouse Mode reports; see Table 11-1 on page 42 for the value.

Figure 11-15.Mouse Mode Report Packet Format



In Figure 11-15:

- **Rpt ID** is the HID-I^{2}C Report ID used for Mouse reports (see Table 11-1 on page 42).
- **Touch** is the data for the Mouse touch.

Each input report consists of a HID-I^{2}C report ID followed by five bytes of that describe the status of the mouse Table 11-3 gives the detailed format of a mouse report packet.

Table 11-3. Detailed Format of a Mouse Report

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	HID-I^{2}C Mouse Report ID							
1	Reserved						Button1	Button0
2	Relative X Position							
3								
4	Relative Y Position							
5								

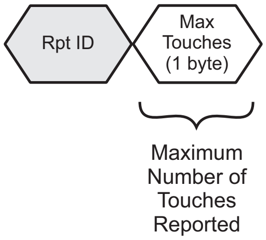
- **Byte 1:**
  Button0: 1 = Button 0 is pressed, 0 = Button 0 is not pressed.
  Button1: 1 = Button 1 is pressed, 0 = Button 1 is not pressed.
- **Byte 2 to 3:**
  X position of the mouse relative to the previous X position (capped to the range –127 to +127).
- **Byte 4 to 5:**
  Y position of the mouse relative to the previous Y position (capped to the range –127 to +127)

11.6.3 Maximum Touches (Surface Contacts) Report

The format of the Maximum Touches report packet is shown in Figure 11-16.

The HID-I^{2}C Report ID used is that for Maximum Touches reports; see Table 11-1 on page 42 for the value.

Figure 11-16.Example Maximum Touches Report



In Figure 11-16 on page 50:

- **Rpt ID** is the HID-I^{2}C Report ID used for Maximum Touches reports (see Table 11-1 on page 42).
- **Max Touches** is the maximum number of touches that can be reported by the device.

Read this report to receive the maximum number of touches that can currently be reported.

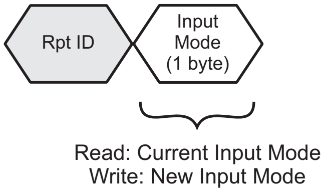
Write this report to set the maximum number of touches to be reported. Note that the number of touches cannot be set to more than the maximum number of touches defined by Multiple Touch Touchscreen T100 NUMTCH.

11.6.4 Input Mode Report

The format of the Input Mode report packet is shown in Figure 11-17.

The HID-I^{2}C Report ID used is that for Input Mode reports; see Table 11-1 on page 42 for the value.

Figure 11-17. Input Mode Packet Format



In Figure 11-17:

- **Rpt ID** is the HID-I^{2}C Report ID used for Input Mode reports (see Table 11-1 on page 42).
- **Input Mode** is the current Input Mode.

Read this report to receive the current Input Mode.

Write this report to change the Input Mode.

The valid values are given in Table 11-4.

Table 11-4. Input Modes

Value	Input Mode
0	Mouse mode
3	Touchpad mode
All other values	Not used; will be ignored by the device

11.6.5 Touch Hardware Quality Assurance (THQA) Report

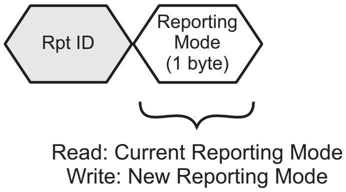
The THQA data is reported to Windows using the THQA Report ID (see Table 11-1 on page 42 for the value). The content of this data is defined by Microsoft.

11.6.6 Reporting Mode Report

The format of the Reporting Mode packet is shown in Figure 11-18.

The HID-I^{2}C Report ID used is that for Reporting Mode reports; see Table 11-1 on page 42 for the value.

Figure 11-18. Reporting Mode Packet Format



In Figure 11-17 on page 51:

- **Rpt ID** is the HID-I^{2}C Report ID used for Reporting Mode reports (see Table 11-1 on page 42).
- **Reporting Mode** is the Reporting Mode to be used.

Read this report to receive the current reporting mode.

Write this report to change the reporting mode.

The valid values are given in Table 11-5.

Table 11-5. Reporting Mode

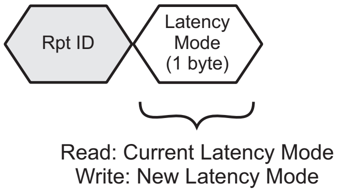
Value	Reporting Mode	Effect
0	No input reports	–
1	Report button states only	Surface contacts are not reported. The Touch fields in the Touch and Mouse Mode reports are set to zero.
2	Report surface contacts (touches) only	Button states are not reported. The Button fields in the Touch and Mouse Mode reports are set to zero.
3	Report both button states and surface contacts	Surface contacts and button states are reported in the Touch and Mouse Mode reports.

11.6.7 Latency Mode

The format of the Latency Mode packet is shown in Figure 11-19.

The HID-I^{2}C Report ID used is that for Latency reports; see Table 11-1 on page 42 for the value.

Figure 11-19.Latency Mode Packet Format



In Figure 11-17 on page 51:

- **Rpt ID** is the HID-I^{2}C Report ID used for Latency Mode reports (see Table 11-1 on page 42).
- **Latency Mode** is the Latency Mode to be used.

Read this report to receive the current latency mode.

Write this report to change the latency mode.

The valid values are given in Table 11-6 on page 52.

Table 11-6. Latency Mode

Value	Latency Mode
0	Low latency
1	High latency

11.7 $\overline{\text{CHG}}$ Line

The $\overline{\text{CHG}}$ line is used to implement the HID-I^{2}C interrupt line. It provides a level triggered interrupt to the host to indicate when there is one or more reports to be read. The $\overline{\text{CHG}}$ line will be pulled low when a report is ready and will remain low as long as there are further reports to be read. Once the last report is read the $\overline{\text{CHG}}$ line will go high.

Note: In order to comply with the HID-I^{2}C specification, Communications Configuration T18 MODE should be set to 0. Refer to the *mXT337TPTP 1.0 Protocol Guide* for more information.

11.8 SDA, SCL

Identical to standard I^{2}C operation. See Section 10.7 on page 40.

11.9 Clock Stretching

Identical to standard I^{2}C operation. See Section 10.8 on page 41.

11.10 Power Control

The mXT337TPTP supports the use of the HID-I^{2}C *SET POWER* commands to put the device into a low power state

11.11 Microsoft Windows Compliance

The mXT337TPTP has algorithms within the Multiple Touch Touchscreen T100 object specifically to ensure Microsoft Windows compliance.

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

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

12. PCB Design Considerations

12.1 Introduction

The following sections give the design considerations that should be adhered to when designing a PCB layout for use with the mXT337TPTP. Of these, power supply and ground tracking considerations are the most critical.

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

12.2 Printed Circuit Board

Atmel recommends the use of a four-layer printed circuit board for mXT337TPTP applications. This, together with careful layout, will ensure that the board meets relevant EMC requirements for both noise radiation and susceptibility, as laid down by the various national and international standards agencies.

12.2.1 PCB Cleanliness

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

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

12.3 Supply Rails and Ground Tracking

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

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

12.4 Power Supply Decoupling

As a rule, a suitable decoupling capacitor should be placed on each and every supply pin on all digital devices. It is important that these capacitors are placed as close to the chip supply pins as possible (less than 10 mm away). The ground connection of these capacitors should be tracked to 0 V by the shortest, heaviest traces possible.

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

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

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

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

12.5 Single Supply Operation

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

12.6 Analog I/O

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

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

12.7 Component Placement and Tracking

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

12.7.1 Digital Signals

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

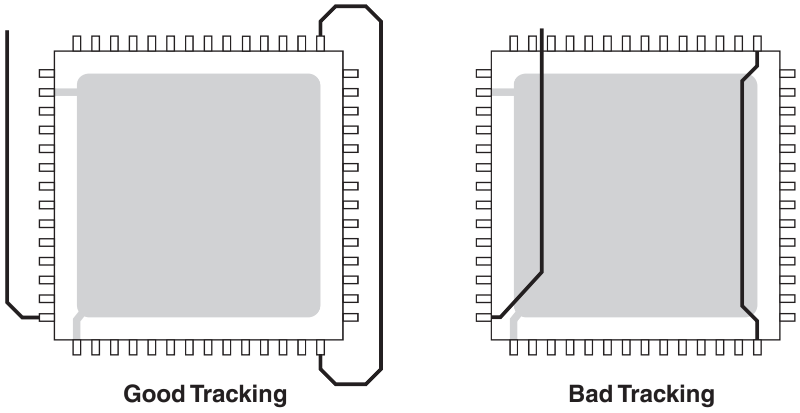
A good ground return path for all signals should be provided, where possible, to ensure that there are no discontinuities in the ground return path.

12.7.2 UQFN Package Restrictions

The central pad on the underside of the UQFN device should be connected to ground. Do not run any tracks underneath the body of the device, only ground. Figure 12-1 shows an example of good/bad tracking.

Figure 12-1. Examples of Good and Bad Tracking

Note: The number of pins and their functions is shown for example purposes only and may not reflect the actual number or function on the device.



12.8 EMC and Other Observations

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

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

13. Getting Started with mXT337TPTP

13.1 Establishing Contact

13.1.1 Communication with the Host

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

- I^{2}C interface (see Section 10. on page 36)
- HID-I^{2}C interface (see Section 11. on page 42)

13.1.2 Power-up Sequence

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

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

13.2 Using the Object Protocol

The device has an object-based protocol that is used to communicate with the device. Typical communication includes configuring the device, sending commands to the device, and receiving messages from the device. Refer to the *mXT337TPTP 1.0 Protocol Guide* for more information.

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

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

13.2.1 Classes of Objects

The mXT337TPTP contains the following classes of objects:

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

13.2.2 Object Instances

Table 13-1. Objects on the mXT337TPTP

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

Table 13-1. Objects on the mXT337TPTP (Continued)

Object	Description	Number of Instances	Usage
Command Processor T6	Performs a command when written to. Commands include reset, calibrate and backup settings	1	No configuration necessary.
Power Configuration T7	Controls the sleep mode of the device. Power consumption can be lowered by controlling the acquisition frequency and the sleep time between acquisitions.	1	Must be configured before use.
Acquisition Configuration T8	Controls how the device takes each capacitive measurement.	1	Must be configured before use.
Touch Objects			
Multiple Touch Touchscreen T100	Creates a Touchscreen that supports the tracking of more than one touch.	1	Enable and configure as required.
Signal Processing Objects			
One-touch Gesture Processor T24	Operates on the data from a Touchscreen object. A One-touch Gesture Processor T24 converts touches into one-touch finger gestures (for example, taps, double taps and drags).	1	Enable and configure as required.
Touch Suppression T42	Suppresses false detections caused, for example, by the user placing their face too near the touchscreen on a mobile phone.	1	Enable and configure as required.
Passive Stylus T47	Processes passive stylus input.	1	Enable and configure as required.
Shieldless T56	Allows a sensor to use true single-layer co-planar construction.	1	Enable and configure as required.
Lens Bending T65	Compensates for lens deformation (lens bending) by attempting to eliminate the disturbance signal from the reported deltas.	3	Enable and configure as required.
maXCharger T72	Performs various noise reduction techniques during touchscreen signal acquisition.	1	Enable and configure as required.
Retransmission Compensation T80	Limits the negative effects on touch signals caused by poor device coupling to ground.	1	Enable and configure as required.
Self Capacitance maXCharger T108	Suppresses the effects of external noise within the context of self capacitance touch measurements.	1	Enable and configure as required.
Support Objects			
Communications Configuration T18	Configures additional communications behavior for the device.	1	Check and configure as necessary.
GPIO/PWM Configuration T19	Creates a bank of digital I/O pins. These pins can then be controlled from the host by writing to the object's configuration memory space.	1	Enable and configure as required.
Self Test T25	Configures and performs self-test routines to find faults on a touch sensor.	1	Pin test commands only. No configuration/tuning necessary. Not for use in production.
User Data T38	Provides a data storage area for user data.	1	Read-only object. Configure as required.

Table 13-1. Objects on the mXT337TPTP (Continued)

Object	Description	Number of Instances	Usage
Digitizer HID Configuration T43	Configures the Digitizer HID interface and the Descriptors associated with it.	1	Enable and configure as required.
Message Count T44	Provides a count of pending messages.	1	Read only object.
CTE Configuration T46	Controls the capacitive touch engine for the device.	1	Must be configured.
Timer T61	Provides control of a timer.	6	Enable and configure as required.
Serial Data Command T68	Provides an interface for the host driver to deliver various data sets to the device.	1	Enable and configure as required.
Dynamic Configuration Controller T70	Allows rules to be defined that respond to system events.	20	Enable and configure as required.
Dynamic Configuration Container T71	Allows the storage of user configuration on the device that can be selected in run-time based on rules defined in the T70 object.	1	Configure if Dynamic Configuration Controller T70 is in use.
CTE Scan Configuration T77	Configures enhanced X line scanning features.	1	Enable and configure as required.
Auxiliary Touch Configuration T104	Allows the setting of self capacitance gain and thresholds for a particular measurement to generate auxiliary touch data for use by other objects.	1	Enable and configure if using self capacitance measurements
Self Capacitance Global Configuration T109	Provides configuration for a self capacitance measurements employed on the device.	1	Check and configure as required (if using self capacitance measurements).
Self Capacitance Tuning Parameters T110	Provides configuration space for a generic set of settings for self capacitance measurements.	6	Use under the guidance of Atmel field engineers only.
Self Capacitance Configuration T111	Provides configuration for a self capacitance measurements employed on the device.	2	Check and configure as required (if using self capacitance measurements).
Self Capacitance Measurement Configuration T113	Configures self capacitance measurements to generate data for use by other objects.	1	Enable and configure as required.
Touchpad T123	Configures the Touchpad and Mouse modes to work with touchpad applications.	1	Enable and configure as required.

13.2.3 Configuring and Tuning the Device

The objects are designed such that a default value of zero in their fields is a “safe” value that typically disables functionality. The objects must be configured before use and the settings written to the nonvolatile memory using the Command Processor T6 object.

Perform the following actions for each object:

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

For information on configuring and tuning the mXT337TPTP, refer to the following document:

- *maXTouch T Series Tuning Guide*

Refer also to the *mXT337TPTP 1.0 Protocol Guide* for detailed information on the configuration parameters for the objects.

13.3 Writing to the Device

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

- Using an I^{2}C write operation (see Section 10.2 on page 36).
- Using the Generic HID-I^{2}C write operation (see Section 11.5.1.2 on page 44).

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

- To send a command to the device, you write the appropriate command to the Command Processor T6 object (refer to the *mXT337TPTP 1.0 Protocol Guide*).
- To configure the device, you write to an object. For example, to configure the device power consumption you write to the global Power Configuration T7 object, and to set up a touchpad you write to a Multiple Touch Touchscreen T100 object. Some objects are optional and need to be enabled before use. Refer to the *mXT337TPTP 1.0 Protocol Guide* for more information on the objects.

13.4 Reading from the Device

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

- The $\overline{\text{CHG}}$ line is asserted whenever a new message is available in the Message Processor T5 object (see Section 10.6 on page 39). See Section 10.4 on page 37 for information on the format of the I^{2}C read operation.
- When using the HID-I^{2}C interface, the interface provides an interrupt-driven interface that sends the messages automatically (see Section 11.5.1.3 on page 44)

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

14. Debugging and Tuning

14.1 Hardware SPI Debug Interface

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

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

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

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

Refer to the following documents for more information:

- *mXT337TPTP 1.0 Protocol Guide* for information on the Command Processor T6 object
- QTAN0050, *Using the maXTouch Debug Port*, for information on using the debug port
- *maXTouch T Series Tuning Guide* (distributed with Atmel approval only) for guidance on using the debug interface for tuning purposes

14.2 Object-based Protocol

The device provides a mechanism for obtaining raw data for development and testing purposes by reading data from the Diagnostic Debug T37 object. Refer to the *mXT337TPTP 1.0 Protocol Guide* for more information on this object.

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

14.3 Self Test

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

Refer to the *mXT337TPTP 1.0 Protocol Guide* for more information on the Self Test T25 object.

15. Specifications

15.1 Absolute Maximum Specifications

Vdd	3.6 V
VddIO	3.6 V
AVdd	3.6 V
XVdd	9.0 V
Voltage forced onto any pin	–0.3 V to (Vdd, VddIO or AVdd) + 0.3 V
Configuration parameters maximum writes (flash memory write cycles)	10,000

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

15.2 Recommended Operating Conditions

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

15.3 DC Characteristics

15.3.1 Analog Voltage Supply – AVdd

Parameter	Min	Typ	Max	Units	Notes
Operating limits	2.7	3.3	3.47	V	
Supply Rise Rates	–	–	0.25	V/µs	

15.3.2 Digital Voltage Supply

Parameter	Min	Typ	Max	Units	Notes
VddIO					
Operating limits	1.71	–	3.47	V	I^2C-compatible. For compatibility with future devices in the same chip family, VddIO should be at the same level as Vdd.
Supply Rise Rates	–	–	0.25	V/µs	
Vdd					
Operating limits	3.0	3.3	3.47	V	
Supply Rise Rates	–	–	0.25	V/µs	

15.4 Test Configuration

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

See *mXT337TPTP 1.0 Protocol Guide* for information about the individual objects and their fields.

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

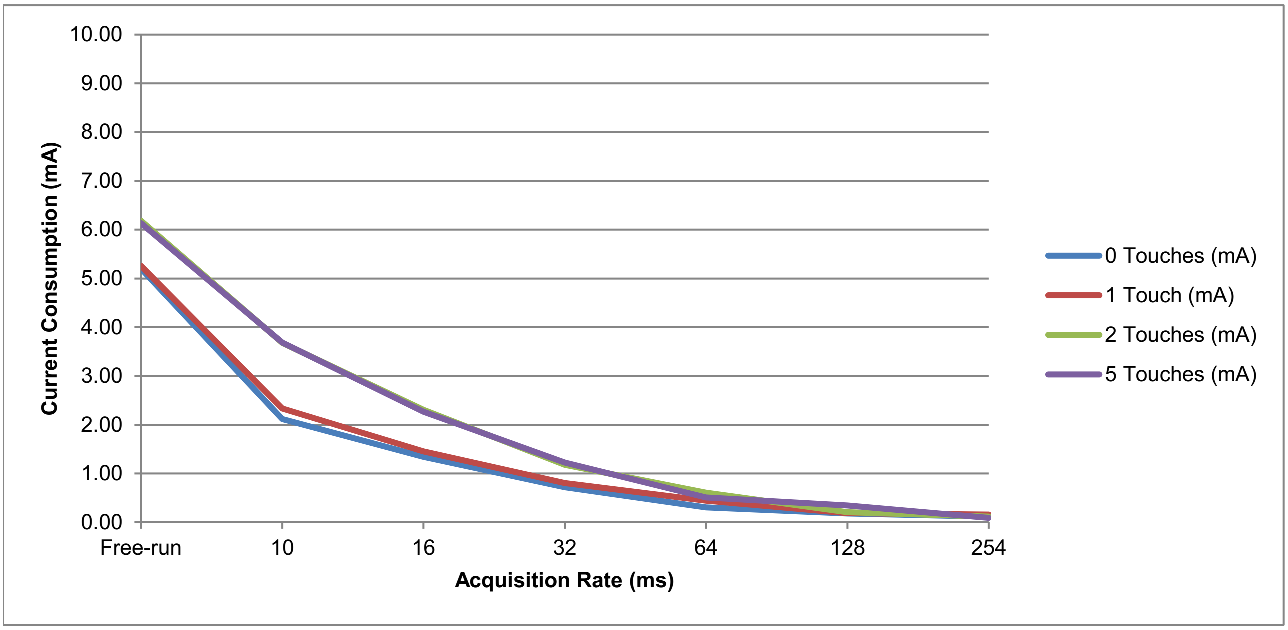
Table 15-1. Test Configuration

Object/Parameter	Description/Setting (Numbers in Decimal)
Acquisition Configuration T8	
CHRGTIME	35
MEASALLOW	15
MEASIDLEDEF	2
MEASACTVDEF	2
CTE Configuration T46	
IDLESYNCSPERX	16
ACTVSYNCSPERX	16
Shieldless T56	Object Enabled
INTTIME	30
Lens Bending T65 Instance 0	Object Enabled
maXCharger T72	Object Enabled
Retransmission Compensation T80	Object Enabled
Multiple Touch Touchscreen T100	Object Enabled
XSIZE	24
YSIZE	14
Auxiliary Touch Configuration T104	Object Enabled
Self Capacitance maXCharger T108	Object Enabled
Self Capacitance Configuration T111 Instance 0	
INTTIME	40
IDLESYNCSPERL	24
ACTVSYNCSPERL	24
Self Capacitance Configuration T111 Instance 1	
INTTIME	40
IDLESYNCSPERL	84
ACTVSYNCSPERL	84

15.5 Current Consumption

15.5.1 Analog Supply

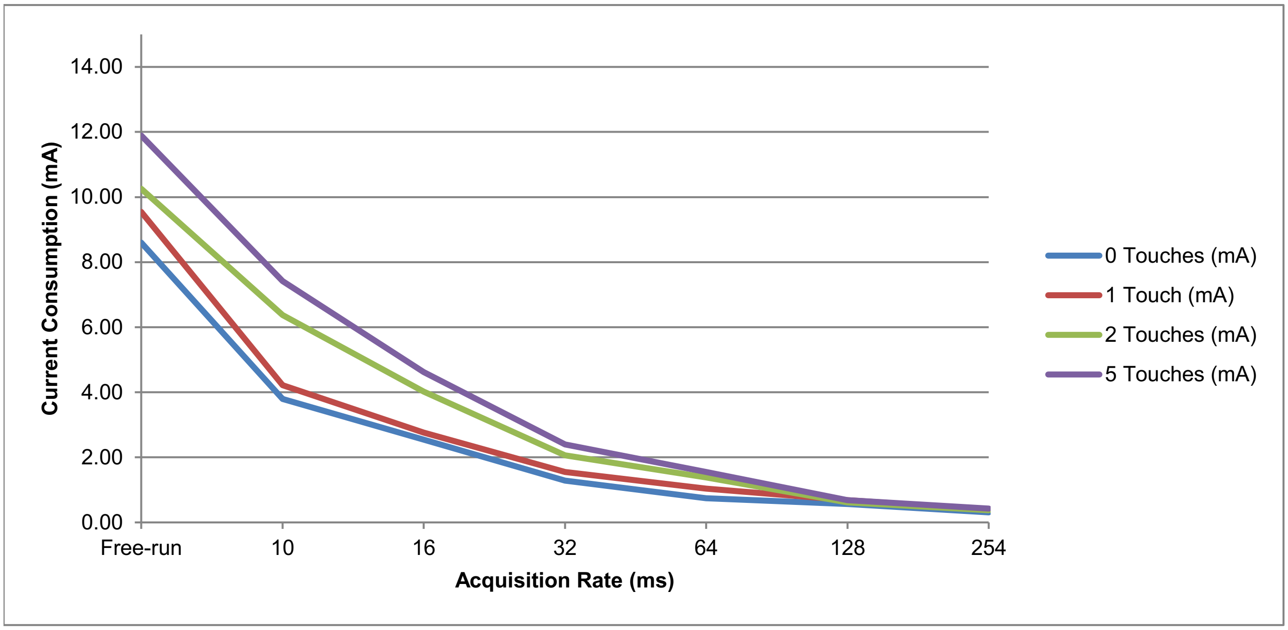
Acquisition Rate (ms)	0 Touches (mA)	1 Touch (mA)	2 Touches (mA)	5 Touches (mA)
Free-run	5.20	5.26	6.19	6.14
10	2.12	2.33	3.68	3.68
16	1.34	1.45	2.31	2.27
32	0.72	0.80	1.18	1.22
64	0.31	0.44	0.61	0.51
128	0.18	0.19	0.21	0.35
254	0.12	0.16	0.12	0.09



15.5.2 Digital Supply

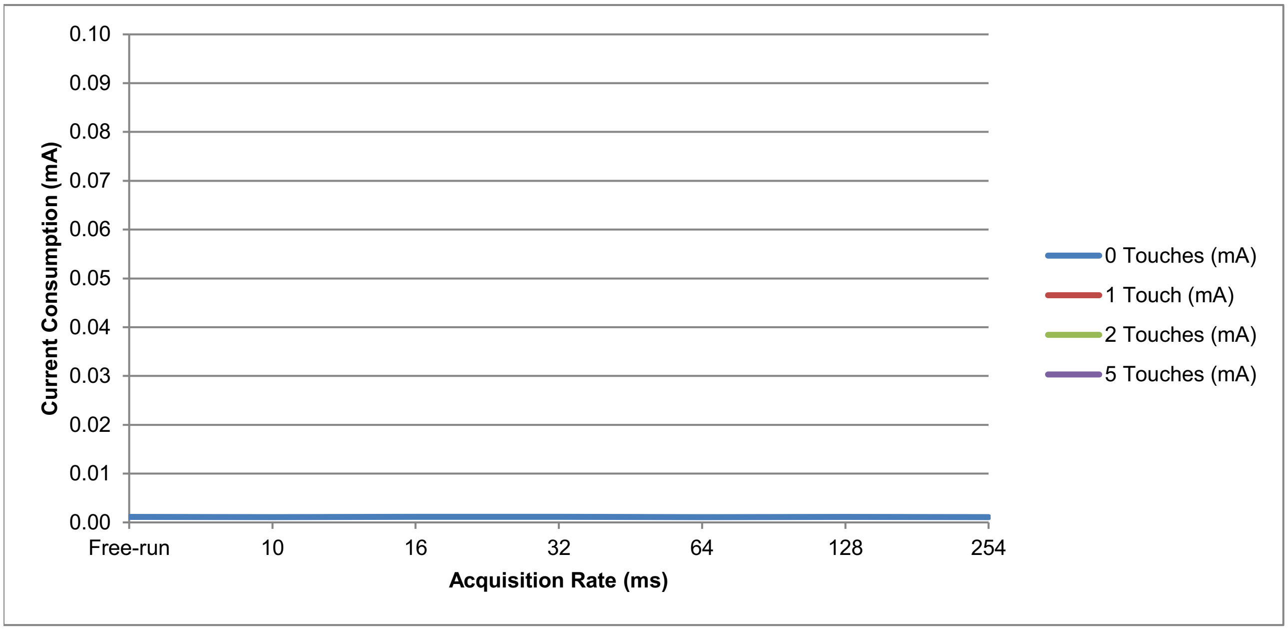
15.5.2.1 Vdd

Acquisition Rate (ms)	0 Touches (mA)	1 Touch (mA)	2 Touches (mA)	5 Touches (mA)
Free-run	8.60	9.55	10.26	11.89
10	3.79	4.22	6.37	7.42
16	2.54	2.76	4.03	4.63
32	1.29	1.55	2.06	2.40
64	0.74	1.04	1.39	1.56
128	0.57	0.68	0.62	0.68
254	0.31	0.39	0.37	0.43



15.5.2.2 VddIO

Acquisition Rate (ms)	0 Touches (mA)	1 Touch (mA)	2 Touches (mA)	5 Touches (mA)
Free-run	0.00	-0.82	-0.82	-0.82
10	0.00	-0.82	-0.82	-0.82
16	0.00	-0.82	-0.82	-0.82
32	0.00	-0.82	-0.82	-0.82
64	0.00	-0.82	-0.82	-0.82
128	0.00	-0.82	-0.82	-0.82
254	0.00	-0.82	-0.82	-0.82



15.6 Deep Sleep Current

$T_A = 25°C$

Parameter	Min	Typ	Max	Units	Notes
Deep Sleep Current	–	135	–	µA	Vdd = 3.3 V, AVdd = 3.3 V
Deep Sleep Power	–	445	–	µW	Vdd = 3.3 V, AVdd = 3.3 V

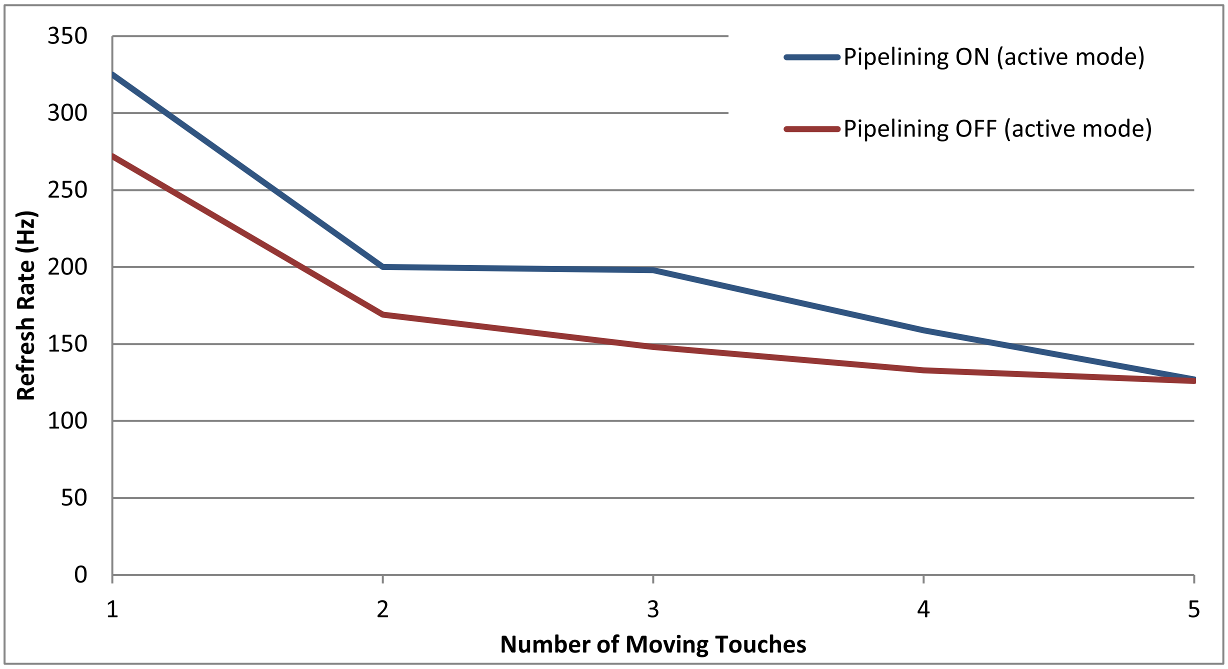
15.7 Power Supply Ripple and Noise

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

15.8 Timing Specifications

15.8.1 Touch Latency

15.8.2 Speed



15.8.3 Reset Timing

Parameter	Min	Typ	Max	Units	Notes
Power on to $\overline{\text{CHG}}$ line low	-	79	-	ms	Vdd supply for POR VddIO supply for external reset
Hardware reset to $\overline{\text{CHG}}$ line low	-	83	-	ms	
Software reset to $\overline{\text{CHG}}$ line low	-	96	-	ms	

1. Any $\overline{\text{CHG}}$ line activity before the power-on or reset period has expired should be ignored by the host. Operation of this signal cannot be guaranteed before the power-on/reset periods have expired.
2. The mXT337TPTP meets Microsoft Windows 10 requirements.

15.9 Input/Output Characteristics

Parameter	Description	Min	Typ	Max	Units	Notes
Input ($\overline{\text{RESET}}$, GPIO, SDA, SCL)						
Vil	Low input logic level	–0.3	–	0.3 × VddIO	V	VddIO = 1.8 V to Vdd
Vih	High input logic level	0.7 × VddIO	–	VddIO	V	VddIO = 1.8 V to Vdd
Iil	Input leakage current	–	–	1	μA	Pull-up resistors disabled
$\overline{\text{RESET}}$ pin	Internal pull-up resistor	20	40	60	kΩ	
Output ($\overline{\text{CHG}}$, GPIO, SDA, SCL)						
Vol	Low output voltage	0	–	0.2 × VddIO	V	VDDIO = 1.8 V to VDD. Iol = -2 mA
Voh	High output voltage	0.8 × VddIO	–	VddIO	V	VDDIO = 1.8 V to VDD. Ioh = 2 mA

15.10 I^{2}C Specifications

Parameter	Value
Addresses	0x4B or 0x4A
Maximum bus speed (SCL)	3.4 MHz
I^{2}C specification	Version 6.0
Required pull-up resistance for standard mode (100 kHz)	1 kΩ to 10 kΩ (1)
Required pull-up resistance for fast mode (400 kHz)	1 kΩ to 3 kΩ (1)
Required pull-up resistance for fast+ mode (1 MHz)	0.7 kΩ (max) (1)
Required pull-up resistance for high-speed mode (3.4 MHz)	0.5 kΩ to 0.75 kΩ (1)

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

More detailed information on I^{2}C operation is available from www.nxp.com/documents/user_manual/UM10204.pdf.

15.11 HID-I^{2}C Specification

Parameter	Value
Vendor ID	0x03EB (Atmel)
Product ID	0x2168 (mXT337TPTP)
HID-I^{2}C specification	1.0

15.12 Touch Accuracy and Repeatability

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

15.13 Thermal Packaging

15.13.1 Thermal Data

Parameter	Typ	Unit	Condition	Package
Junction to ambient thermal resistance	46.5	°C/W	Still air	UFBGA 64, 5 × 5 mm
Junction to case thermal resistance	7.3	°C/W		UFBGA 64, 5 × 5 mm
Junction to ambient thermal resistance	48.4	°C/W	Still air	UFBGA 72, 6 × 6 mm
Junction to case thermal resistance	5.3	°C/W		UFBGA 72, 6 × 6 mm
Junction to ambient thermal resistance	22.4	°C/W	Still air	UFQFN 56, 6 × 6 mm
Junction to case thermal resistance	5.2	°C/W	–	UFQFN 56, 6 × 6 mm

15.13.2 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).

15.14 ESD Information

Parameter	Value	Reference standard
Human Body Model (HBM)	±2000 V	JEDEC JS–001
Charge Device Model (CDM)	±250 V	

15.15 Soldering Profile

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

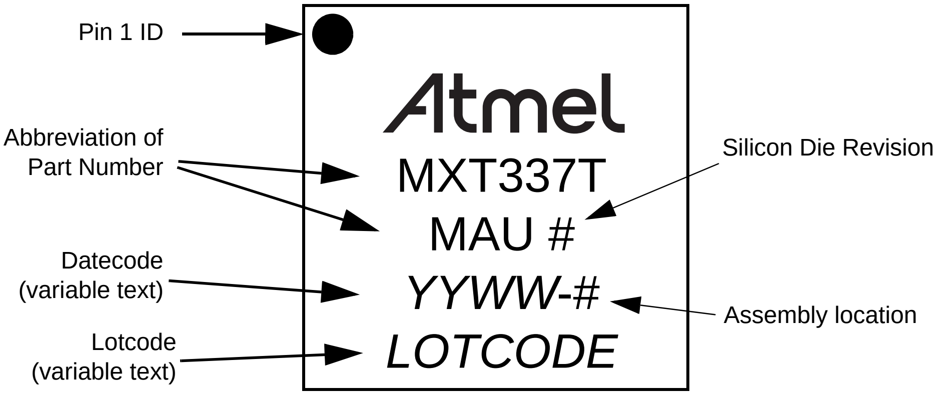
15.16 Moisture Sensitivity Level (MSL)

MSL Rating	Package Type(s)	Peak Body Temperature	Specifications
MSL1	UQFN	260°C	IPC/JEDEC J-STD-020
MSL3	BGA	260°C	IPC/JEDEC J-STD-020

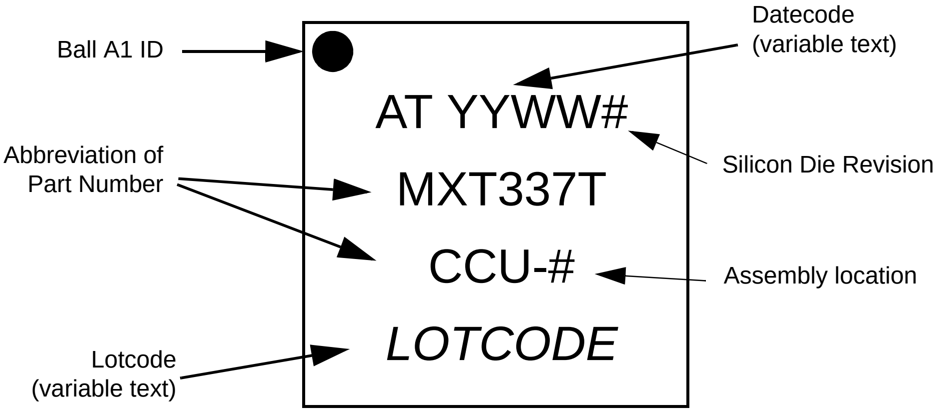
16. Package Information

16.1 Part Marking

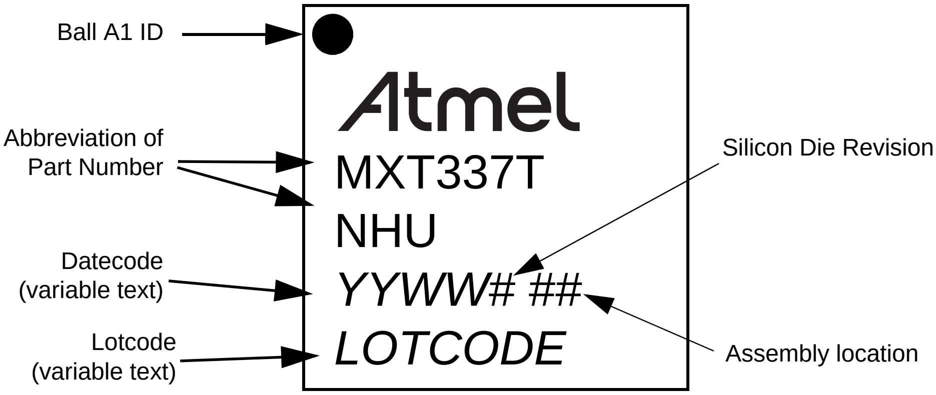
16.1.1 56-Pin UFQFN Package



16.1.2 64-Ball UFBGA Package



16.1.3 72-Ball UFBGA Package



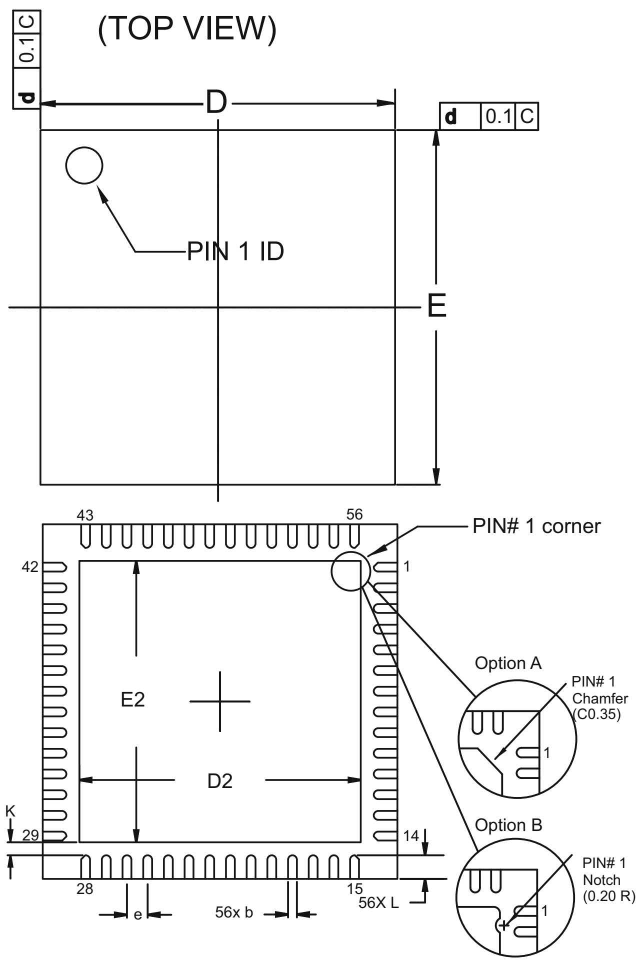
16.2 Orderable Part Numbers

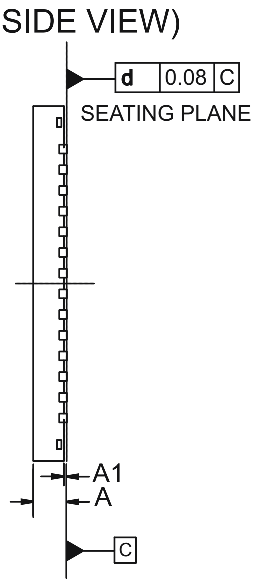
Orderable Part Number	SL Code[(1)]	Firmware Revision	Description
ATMXT337TPTP-MAU (trays)	Q1094	1.0	56-pin UQFN 6 × 6 × 0.6 mm, 0.35 mm pin pitch, RoHS compliant
ATMXT337TPTP-MAUR (tape and reels)	Q1094	1.0	56-pin UQFN 6 × 6 × 0.6 mm, 0.35 mm pin pitch, RoHS compliant
ATMXT337TPTP-CCU (trays)	Q1094	1.0	64-ball UFBGA 5 × 5 × 0.6 mm, 0.5 mm ball pitch, RoHS compliant
ATMXT337TPTP-CCUR (tape and reels)	Q1094	1.0	64-ball UFBGA 5 × 5 × 0.6 mm, 0.5 mm ball pitch, RoHS compliant
ATMXT337TPTP-NHU (trays)	Q1094	1.0	72-ball UFBGA 6 × 6 × 0.6 mm, 0.5 mm ball pitch, RoHS compliant
ATMXT337TPTP-NHUR (tape and reels)	Q1094	1.0	72-ball UFBGA 6 × 6 × 0.6 mm, 0.5 mm ball pitch, RoHS compliant

Notes: 1. All purchase orders must include the Orderable Part Number and the SL Code. (Note: SL Code was formerly known as QS Number.)

16.3 Mechanical Drawings

16.3.1 UFQFN 56 Pins





COMMON DIMENSIONS
(UNIT OF MEASURE=MM)

SYMBOL	MIN	NOM	MAX	NOTE
A	-	0.55	0.60	
A1	0.00	-	0.05	
b	0.13	0.18	0.23	1
D	6.00 BSC			
D2	4.40	4.50	4.60	
E	6.00 BSC			
E2	4.40	4.50	4.60	
e	-	0.35	-	
L	0.35	0.40	0.45	
K	0.30	-	-	

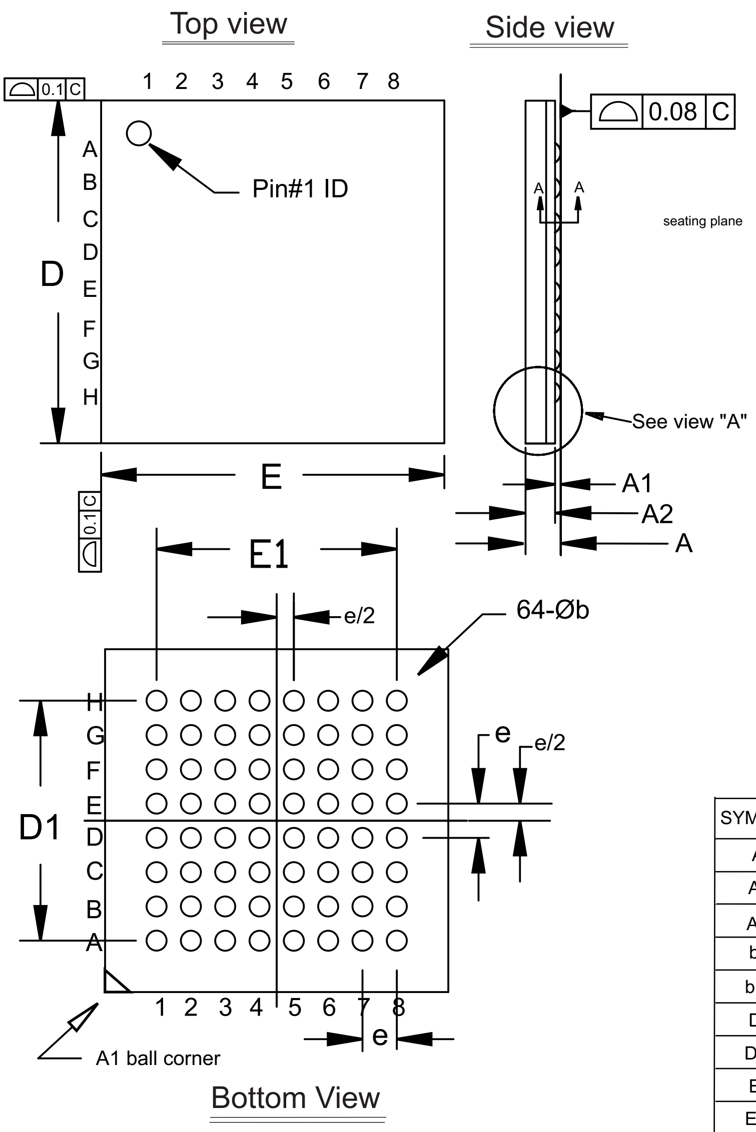
NOTE :

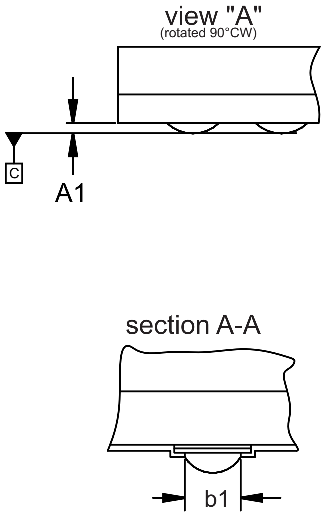
1. Dimension "b" applies to metalized terminal and is measured between 0.15mm and 0.30mm from the terminal tip. If the terminal has the optional radius on the other end of the terminal, the dimension should not be measured in that radius area.

03/26/14

Package Drawing Contact: packagedrawings@atmel.com	**TITLE** 56MA2, 56-pad, 6x6x0.6mm body,0.35mm pitch, (4.5mmSq Thermal ePAD) Ultra-Thin, Fine-Pitch Quad Flat No Lead Package (Sawn)(UFQFN)	**GPC** ZEK	**DRAWING NO.** 56MA2	**REV.** B

16.3.2 UFBGA 64 Balls





COMMON DIMENSIONS
(UNIT OF MEASURE=MM)

SYMBOL	MIN	NOM	MAX	NOTE
A	-	0.53	0.60	
A1	0.12	-	-	
A2	0.38 REF			
b	0.25	0.30	0.35	1
b1	0.25	-	-	2
D	4.90	5.00	5.10	
D1	3.50 BSC			
E	4.90	5.00	5.10	
E1	3.50 BSC			
e	0.50BSC			

Note 1: Dimension "b" is measured at the maximum ball dia. in a plane parallel to seating plane.

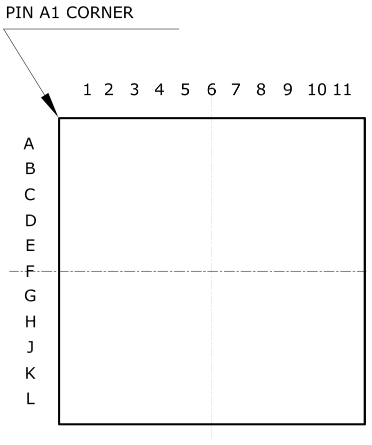
Note 2: Dimension "b1" is the solderable surface defined by the opening of the solder resist layer.

9/15/2011

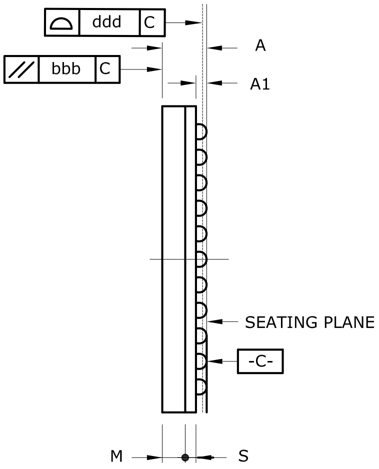
	TITLE	GPC	DRAWING NO.	REV.
Atmel Package Drawing Contact: packagedrawings@atmel.com	64CC1, 64-ball, 5x5x0.6mm body,0.50mm ball pitch, (8 x 8 Array) Ultra-Thin, Fine-Pitch Ball Grid Array (UFBGA)	CAJ	64CC1	A

16.3.3 UFBGA 72 Balls

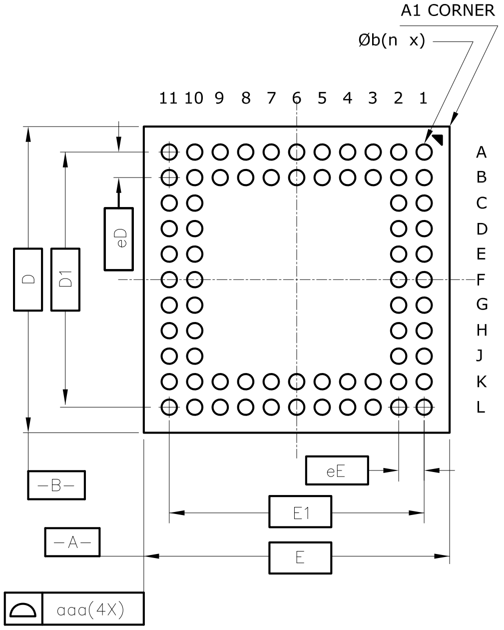
DRAWINGS NOT SCALED



TOP VIEW



SIDE VIEW



BOTTOM VIEW

COMMON DIMENSIONS

(Unit of Measure = mm)

SYMBOL	MIN	NOM	MAX	NOTE
A	-----	-----	0.600	
A1	0.105	-----	0.205	
E/D	6.00 / 6.00			
E1/D1	5.00 / 5.00			
b	0.225	-----	0.325	
eD/eE	Ball pitch : 0.500			
M	Mold thickness : 0.250 ref			
S	Subst thickness : 0.136 ref			
aaa	Pack edge tolerance : 0.100			
bbb	Mold flatness : 0.100			
ddd	Copla : 0.080			
ball diam	0.250			
n	72			

Notes :
1. This drawing is for general information only. Refer to JEDEC Drawing MO-280, for proper dimensions, tolerances, datums, etc.
2. Array as seen from the bottom of the package.
3. Dimension A includes stand-off height A1, package body thickness, and lid height, but does not include attached features.
4. Dimension b is measured at the maximum ball diameter, parallel to primary datum C.

05/02/2013

	TITLE	GPC	DRAWING NO.	REV.
Atmel **Package Drawing Contact:** **packagedrawings@atmel.com**	7C, 72-ball (11x11 array, 2 Out. rings), 0.5mm pitch, 6x6x0.6mm Ultra Thin Fine-Pitch Ball Grid Array Package (UFBGA)	CAD	7C	B

Associated Documents

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

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

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

- *maXTouch T Series Tuning Guide* (distributed with Atmel approval only)
- *mXT337TPTP 1.0 Protocol Guide* (distributed with Atmel approval only)

The following documents may also be useful:

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

Revision History

Revision Number	History
Revision AX – September 2015	Initial edition for firmware revision 1.0 – Release

Enabling Unlimited Possibilities®

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

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

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

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

© 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. Microsoft® and Windows® are registered trademarks of Microsoft Corporation in the United States and other countries. 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. Atmel products are not intended, authorized, or warranted for use as components in applications intended to support or sustain life.