Atmel AT42QT1481

48-Key QMatrix FMEA IEC/EN60730 Touch Sensor IC

DATASHEET

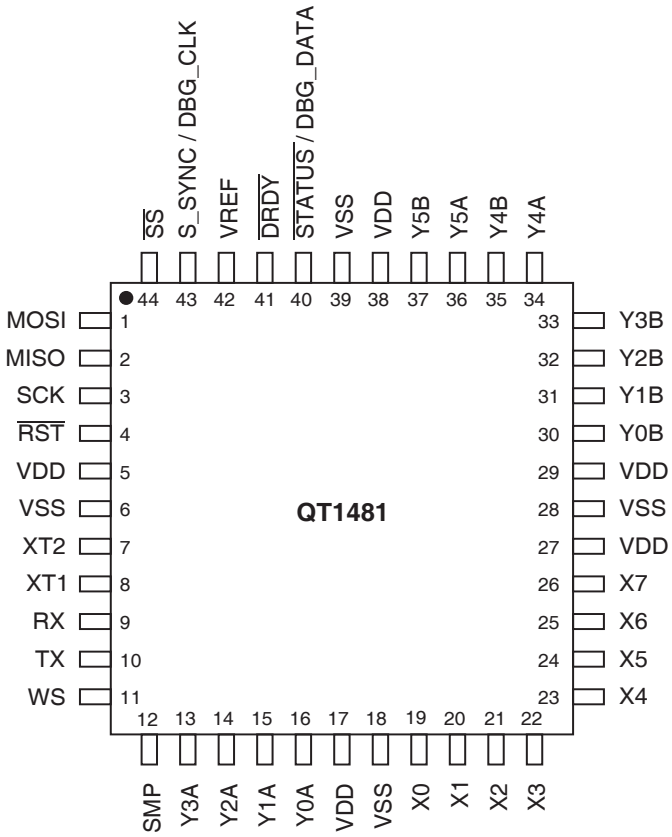
Features

- Number of keys:
  - Up to 48
- Technology:
  - Patented charge-transfer (transverse mode), with frequency hopping
- Key outline sizes:
  - 6 mm × 6 mm or larger (panel thickness dependent); widely different sizes and shapes possible
- Key spacings:
  - 8 mm or wider, center to center (panel thickness dependent)
- Electrode design:
  - Two-part electrode shapes (drive-receive); wide variety of possible layouts
- Layers required:
  - One layer (with jumpers), two layers (no jumpers)
- Electrode materials:
  - PCB, FPCB, silver or carbon on film, ITO on film
- Panel materials:
  - Plastic, glass, composites, painted surfaces (low particle density metallic paints possible)
- Adjacent Metal:
  - Compatible with grounded metal immediately next to keys
- Panel thickness:
  - Up to 50 mm glass, 20 mm plastic (key size dependent)
- Key sensitivity:
  - Individually settable via simple commands over serial interface
- Signal processing:
  - Self-calibration, auto drift compensation, noise filtering, Adjacent Key Suppression®
- Interfaces:
  - UART
  - SPI slave (4 MHz maximum clock rate)
  - STATUS indication pin
  - Debug output
- FMEA compliant design features
- IEC/EN/UL60730 compliant design features
  - UL approval
  - VDE compliance
  - For use in both class B and class C safety-critical products

- Detects and Reports Key Failure
- Power:
  - +4.75 to 5.25 V
- Package:
  - 44-pin 10 × 10 mm TQFP RoHS compliant

1. Pinout and Schematic

1.1 Pinout Configuration



1.2 Pin Descriptions

Table 1-1. Pin Listing

Pin	Name	Type	Description	If Unused...
1	MOSI	I	SPI data input	Leave open
2	MISO	O	SPI data output	Leave open
3	SCK	I	SPI clock input	Vdd
4	$\overline{RST}$	I	Reset low; has internal 30 kΩ – 60 kΩ pull-up resistor. This pin should be controlled by the host.	Vdd
5	VDD	P	Power	–
6	VSS	P	Ground	–
7	XT2	O	Ceramic resonator or crystal,16 MHz	–
8	XT1	I		–
9	RX	I	UART receive data input	Vdd
10	TX	O	UART transmit data; has internal 20 kΩ – 50 kΩ pull-up resistor	Leave open
11	WS	I	Wake-up from sleep input and/or sync input	Vdd
12	SMP	I/O	Sample output	–
13	Y3A	I/O	Y line connection	Leave open
14	Y2A	I/O	Y line connection	Leave open
15	Y1A	I/O	Y line connection	Leave open
16	Y0A	I/O	Y line connection	Leave open
17	VDD	P	Power	–
18	VSS	P	Ground	–
19	X0	O	X matrix drive line	Leave open
20	X1	O	X matrix drive line	Leave open
21	X2	O	X matrix drive line	Leave open
22	X3	O	X matrix drive line	Leave open
23	X4	O	X matrix drive line	Leave open
24	X5	O	X matrix drive line	Leave open
25	X6	O	X matrix drive line	Leave open
26	X7	O	X matrix drive line/	Leave open
27	VDD	P	Power	–
28	VSS	P	Ground	–
29	VDD	P	Power	–
30	Y0B	I/O	Y line connection	Leave open

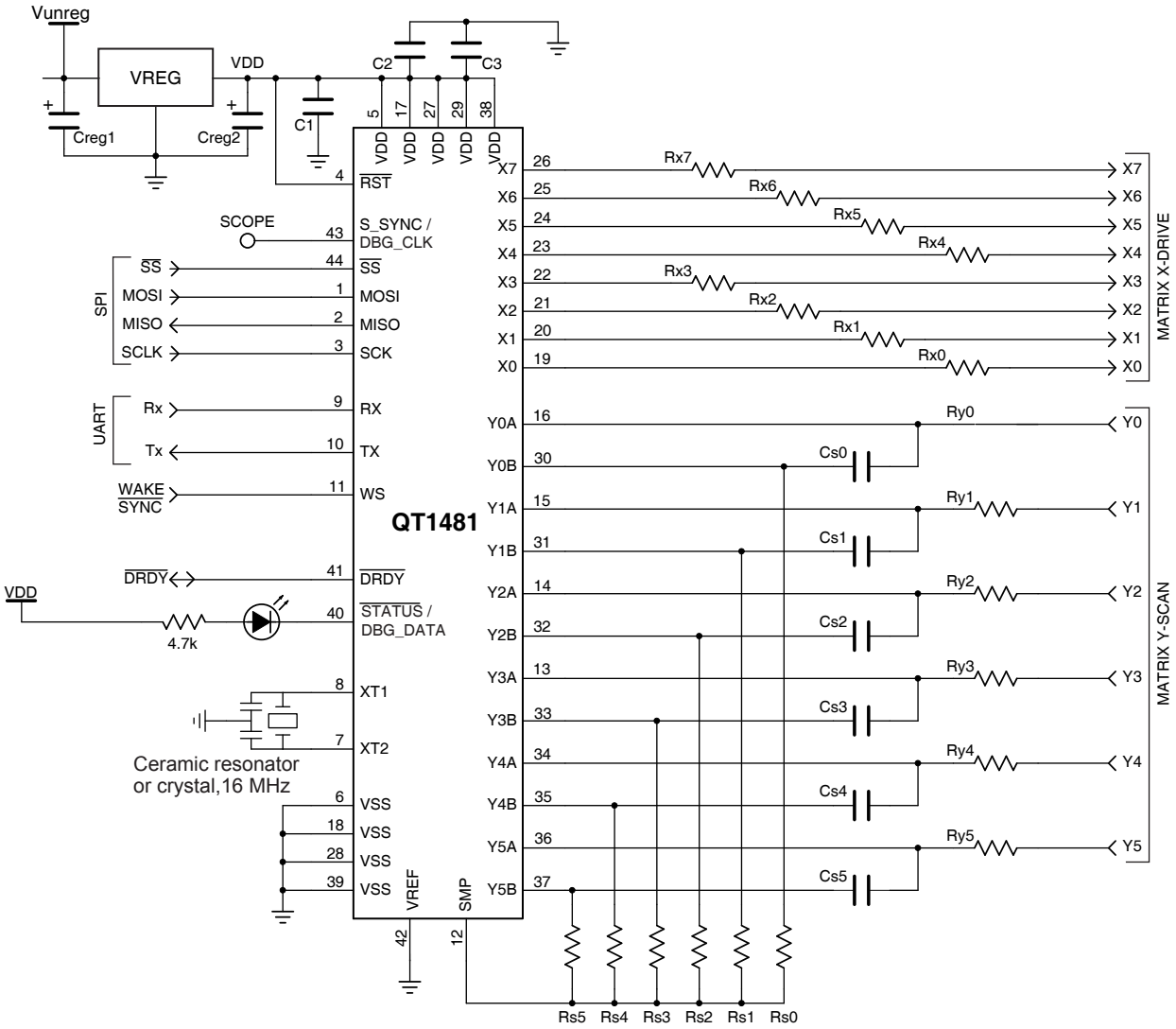
Table 1-1. Pin Listing (Continued)

Pin	Name	Type	Description	If Unused...
31	Y1B	I/O	Y line connection	Leave open
32	Y2B	I/O	Y line connection	Leave open
33	Y3B	I/O	Y line connection	Leave open
34	Y4A	I/O	Y line connection	Leave open
35	Y4B	I/O	Y line connection	Leave open
36	Y5A	I/O	Y line connection	Leave open
37	Y5B	I/O	Y line connection	Leave open
38	VDD	P	Power	–
39	VSS	P	Ground	–
40	$\overline{\text{STATUS}}$ / DBG_DATA	O	Status output (active low) or Debug Data; has internal 20 kΩ – 50 kΩ pull-up resistor	Leave open
41	$\overline{\text{DRDY}}$	I/O	This pin MUST be used. 1 = comms ready; needs a 100 µs grace period before checking. Open-drain with internal 20 kΩ – 50 kΩ pull-up resistor	–
42	VREF	I	Connect to Vss	–
43	S_SYNC / DBG_CLK	O	Scope Synchronization output or Debug Clock	Leave open
44	$\overline{\text{SS}}$	I	SPI slave select; has internal 20 kΩ – 50 kΩ pull-up resistor	Leave open

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

1.3 Schematic

Figure 1-1. Typical Circuit



For component values in Figure 1-1 check the following sections:

- Section 2.7 on page 10: Cs capacitors (Cs0 – Cs5)
- Section 2.8 on page 11: Sample resistors (Rs0 – Rs5)
- Section 2.10 on page 12: Matrix resistors (Rx0 – Rx7, Ry0 – Ry5)
- Section 2.13 on page 14: Power Supply

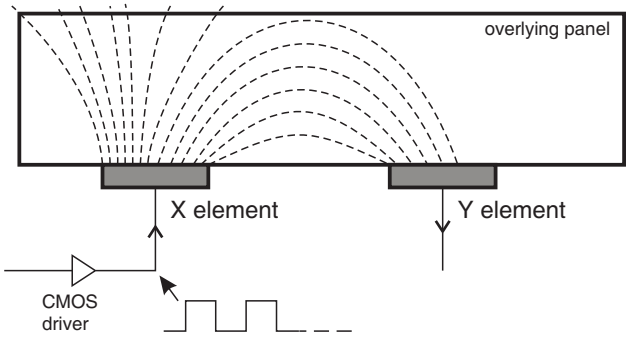
2. Hardware and Functional

2.1 Introduction

The AT42QT1481 (QT1481) is a digital burst mode sensor, designed specifically for QMatrix layout touch controls; it includes all signal processing functions necessary to provide stable sensing under a wide variety of changing conditions. Only a few external parts are required for operation. The entire circuit can be built within a few square centimeters of single-sided PCB area. CEM-1 and FR1 punched, single-sided materials can be used for the lowest possible cost. The PCB's rear can be mounted flush on the back of a glass or plastic panel using a conventional adhesive, such as 3M VHB two-sided adhesive acrylic film.

The QT1481 employs transverse charge-transfer (QT™) sensing, a technology that senses changes in electrical charge forced across two electrode elements by a pulse edge (see Figure 2-1).

Figure 2-1. Field Flow Between X and Y Elements



The QT1481 allows a wide range of key sizes and shapes to be mixed together in a single touch panel. The QT1481 is designed for use with up to 48 keys.

The QT1481 uses both UART and SPI interfaces (only one at a time) to allow key data to be extracted and to permit individual key parameter setup. The interface protocol uses simple single byte commands and responds with single byte responses in most cases. The command structure is designed to minimize the amount of data traffic while maximizing the amount of information conveyed.

In addition to normal operating and setup functions the QT1481 can also report back actual signal strengths and error codes.

QmBtn software for the PC can be used to program the operation of the IC as well as read back key status and signal levels in real time.

A Debug output interface is also supported, which can be used to monitor many operating variables during product development.

The QT1481 incorporates many tests and checks to enable a product to achieve FMEA and EN60730 compliance. The results of some tests need to be checked by the host. To achieve a compliant design, the host must read back the test results and confirm their validity.

The QT1481 is able to scan the touch matrix twice as fast as previous generation devices; it can take twice the number of samples in a given time frame. This means the QT1481 is much better equipped to continue normal operation in the face of heavy noise.

See Appendix C. on page 68 for information on conducted noise immunity.

2.2 Key Numbers

The keys are numbered from 0 – 47.Table 2-1 shows the key numbering.

Table 2-1. Key Numbers

	X7	X6	X5	X4	X3	X2	X1	X0	
Y0	7	6	5	4	3	2	1	0	Key numbers
Y1	15	14	13	12	11	10	9	8	
Y2	23	22	21	20	19	18	17	16	
Y3	31	30	29	28	27	26	25	24	
Y4	39	38	37	36	35	34	33	32	
Y5	47	46	45	44	43	42	41	40	

2.3 Matrix Scan Sequence

Key scanning begins with location X = 0, Y = 0 (key 0). All keys on X0 are scanned first, then X1 and finishing with all keys on X7 (for example, the sequence X0Y0, X0Y1 – X0Y5, X1Y0, X1Y1...). Table 2-1 shows the key numbering.

All keys on the same X line are excited together in a burst of acquisition pulses whose length is determined by the Setups parameter BL (see Section 5.9 on page 43); this can be set to a different value for each key. A burst is completed entirely before the next X line is excited. At the end of each burst the resulting signals, one for each Y line, are converted to digital form and processed. The burst length directly impacts key gain. Each key can have a different burst length in order to allow tailoring of key sensitivity. Although all keys on an entire X line are excited simultaneously, the charge is selectively captured at each Y line according to the burst length selected.

2.4 Enabling/Disabling Keys – Burst Paring

Unused keys are always pared from the computation sequence in order to optimize speed. If all keys are disabled on any given X, the entire X line is also pared from the burst sequence. If only two X lines have enabled keys, only two timeslots are used for scanning.

The NDIL parameter is used to enable and disable keys in the matrix. Setting NDIL = 0 for a key disables it (Section 5.5 on page 41). Keys that are disabled are eliminated from the scan sequence to save scan time and thus power. If all keys on an X line are disabled, the burst for the entire X line is removed from the scan sequence, further saving time and power. This has the consequence of affecting the scan rate of the entire matrix as well as the time required for initial matrix calibration. It does not affect the time required to calibrate an individual key once the matrix is initially calibrated after power-up or reset.

It is very important that only those keys that physically exist are enabled. All non-existent keys must be disabled (NDIL = 0) otherwise other keys in the matrix can incorrectly report their signal as zero.

2.5 Response Time

The response time of the QT1481 depends on:

- the burst spacing
- the number of enabled X lines (Section 5.5 on page 41)
- the detect integrator settings (Section 5.5 on page 41)
- Mains Sync
- and the serial polling rate by the host microcontroller

Example, **without mains sync**:

- NXE = Number of X lines enabled = 8
- NDIL = Norm detect integrator limit = 2
- FDIL = Fast detect integrator limit = 5
- BS = Burst spacing = 1 ms
- FMEA = FMEA test slot = 1
- HPR = Host polling rate = 10 ms
- TMS = Time to perform one Matrix Scan

The worst case response time is computed as:

Tr = (TMS × NDIL) + HPR

TMS = ((NXE + FMEA + (FDIL – 1)) × BS)

Tr = (((NXE + FMEA + (FDIL – 1)) × BS) × NDIL) + HPR

For the above example values:

Tr = (((8 + 1 + (5 – 1)) × 1 ms) × 2) + 10 ms = 36 ms

The use of the $\overline{\text{STATUS}}$ pin to trigger host sampling can reduce this to approximately 26 ms by eliminating the majority of the host polling time (see Section 5.20 on page 47).

TMS varies with the configurations of Burst Length (see Section 5.9 on page 43) and Dwell (see Section 5.13 on page 45), and should be measured using an oscilloscope.

Example, **with mains sync**:

The value calculated for TMS needs to be rounded up to the nearest multiple of the mains periods before proceeding with the rest of the calculation. Continuing with the above example, TMS = ((8 + 1 + (5 – 1)) × 1 ms) = 13 ms.

Rounded up to a multiple of whole mains periods, this becomes 20 ms (assuming a mains frequency of 50 Hz).

The worst case response time is then computed as:

Tr = (20 ms × 2) + 10 ms = 50 ms

An X line is considered enabled if any key on that X line is enabled. An X line is disabled if all keys on that X line are disabled.

Note: TMS will be stretched by 15 ms if STS_DEBUG is enabled.

2.6 Oscillator

The oscillator can use either a quartz crystal or a ceramic resonator. In all cases, XT1 and XT2 must both be loaded with low-value capacitors to ground. These capacitors should be in the range 12 pF to 22 pF. Follow the manufacturer's recommendations for the appropriate value within this range. Resonators and crystals requiring loading capacitors outside this range are unsuitable for operation with the QT1481.

A resistor of value 1MΩ is connected internally between XT1 and XT2.

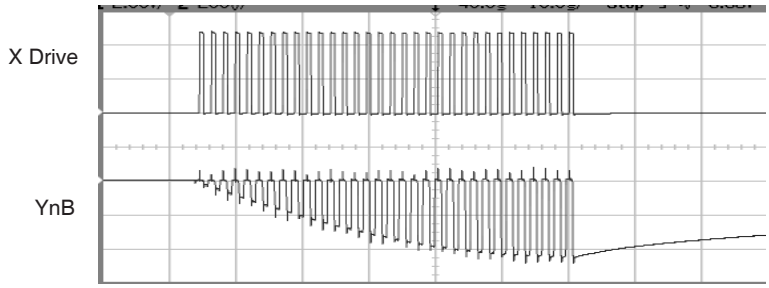
The frequency of oscillation should be 16 MHz ±1% for accurate UART transmission timing.

2.7 Sample Capacitor; Saturation Effects

The charge sampler capacitors on the Y pins (Cs0 – Cs5) should be NPO (preferred), X7R ceramics or PPS film; NPO offers the best stability. The value of these capacitors is not critical but 4.7 nF is recommended for most cases.

Cs voltage saturation is shown in Figure 2-2. This nonlinearity is caused by excessive voltage accumulation on Cs inducing conduction in the pin protection diodes. This badly saturated signal destroys key gain and introduces a strong thermal coefficient which can cause phantom detection.

Figure 2-2. VCs – Nonlinear During Burst (Burst too long, or Cs too small, or X-Y transcapacitance too large)



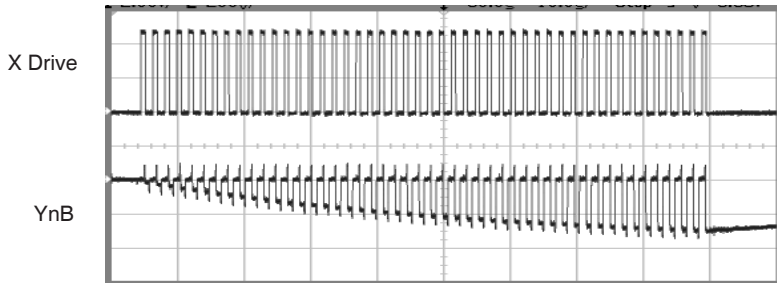
The cause of this is either from the burst length being too long, the Cs value being too small, or the X-Y transfer coupling being too large. Solutions include loosening up the interdigitation of key structures, greater separation of the X and Y lines on the PCB, increasing Cs, and decreasing the burst length.

Increasing Cs makes the part slower; decreasing burst length makes it less sensitive. A better PCB layout and a looser key structure (up to a point) have no negative effects.

Cs voltages should be observed on an oscilloscope with the matrix layer bonded to the panel material; if the Rs side of any Cs ramps is more negative than –0.25 V during any burst (not counting overshoot spikes which are probe artifacts), there is a potential saturation problem.

Figure 2-3 on page 11 shows a defective waveform similar to that of Figure 2-2, but in this case the distortion is caused by excessive stray capacitance coupling from the Y line to AC ground; for example, from running too near and too far alongside a ground trace, ground plane, or other traces. The excess coupling causes the charge-transfer effect to dissipate a significant portion of the received charge from a key into the stray capacitance.

Figure 2-3. VCs – Poor Gain, Nonlinear During Burst (Excess capacitance from Y line to Gnd)



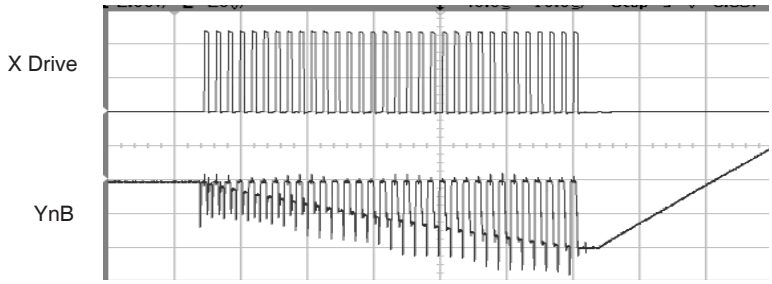
This phenomenon is more subtle; it can be best detected by increasing BL to a high count and watching what the waveform does as it descends towards and below –0.25 V. The waveform appears deceptively straight, but it slowly starts to flatten even before the –0.25 V level is reached.

A correct waveform is shown in Figure 2-4. Note that the bottom edge of the bottom trace is substantially straight (ignoring the downward spikes).

Unlike other QT circuits, the Cs capacitor values on QT1481 have no effect on conversion gain. However, they do affect conversion time.

Unused Y lines should be left open.

Figure 2-4. VCs – Correct



2.8 Sample Resistors

The sample resistors (Rs0 – Rs5) are used to perform single-slope analog-to-digital (ADC) conversion of the acquired charge on each Cs capacitor. These resistors directly control acquisition gain; larger values of Rs proportionately increase signal gain. Values of Rs can range from 220 kΩ to 4.7 MΩ. 470 kΩ is a typical value for most purposes.

Larger values for Rs also increase conversion time and may reduce the fastest possible key sampling rate, which can impact response time especially with larger numbers of enabled keys.

Unused Y lines do not require an Rs resistor.

2.9 Signal Levels

Using Atmel QmBtn software it is easy to observe the absolute level of signal received by the sensor on each key. The signal values should normally be in the range of 250 to 750 counts with properly designed key shapes (see the *Touch Sensors Design Guide*, available on the Atmel website). However, long adjacent runs of X and Y lines can also artificially boost the signal values, and induce signal saturation: this is to be avoided. The X-to-Y coupling should come mostly from intra-key electrode coupling, not from stray X-to-Y trace coupling.

QmBtn software is available free of charge on the Atmel website.

The signal swing from the smallest finger touch should preferably exceed 10 counts, with 15 being a reasonable target. The signal threshold setting (NTHR) should be set to a value guaranteed to be less than the signal swing caused by the smallest touch.

Increasing the burst length (BL) parameter increases the signal strengths as will increasing the sampling resistor (Rs) values.

2.10 Matrix Series Resistors

The X and Y matrix scan lines should use series resistors (Rx0 – Rx7 and Ry0 – Ry5 respectively) for improved EMC performance (Figure 1-1 on page 6).

X drive lines require Rx in most cases to reduce edge rates and thus reduce RF emissions. Values range from 1 kΩ to 100 kΩ, typically 1 kΩ.

Y lines need Ry to reduce EMC susceptibility problems and in some extreme cases, ESD. Values range from 1 kΩ to 100 kΩ, typically 1 kΩ. Y resistors act to reduce noise susceptibility problems by forming a natural low-pass filter with the Cs capacitors.

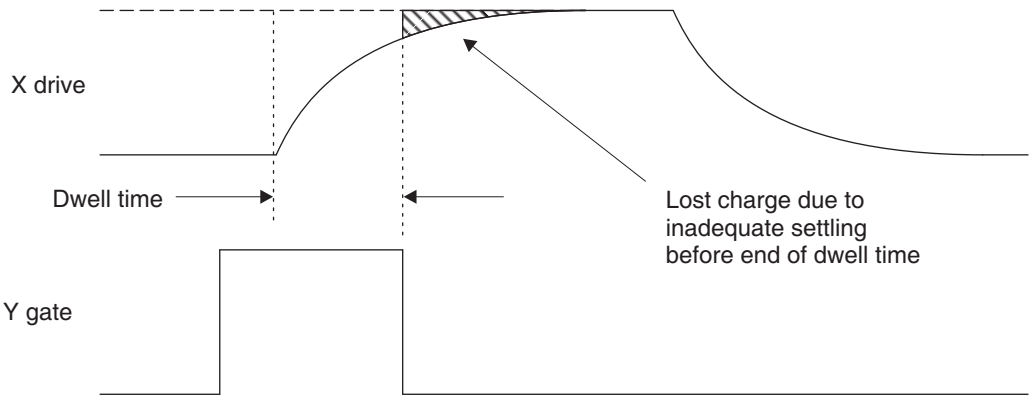
It is essential that the Rx and Ry resistors and Cs capacitors be placed very close to the chip. Placing these parts more than a few millimeters away opens the circuit up to high frequency interference problems (above 20 MHz) as the trace lengths between the components and the chip start to act as RF antennas.

The upper limits of Rx and Ry are reached when the signal level and hence key sensitivity are clearly reduced. The limits of Rx and Ry depend on key geometry and stray capacitance, and thus an oscilloscope is required to determine optimum values of both.

Dwell time is the duration in which charge coupled from X to Y is captured (Figure 2-5 on page 12). Increasing the dwell time increases the signal levels lost to higher values of Rx and Ry, as shown in Figure 2-5. Too short a dwell time causes charge to be 'lost', if there is too much rising edge roll-off. Lengthening the dwell time causes this lost charge to be recaptured, thereby restoring key sensitivity. In the QT1481 dwell time is adjustable (see Section 5.13 on page 45).

Dwell time problems can also be solved by either reducing the stray capacitance on the X line(s) (by a layout change – for example, by reducing X line exposure to nearby ground planes or traces) or the Rx resistor needs to be reduced in value (or a combination of both approaches).

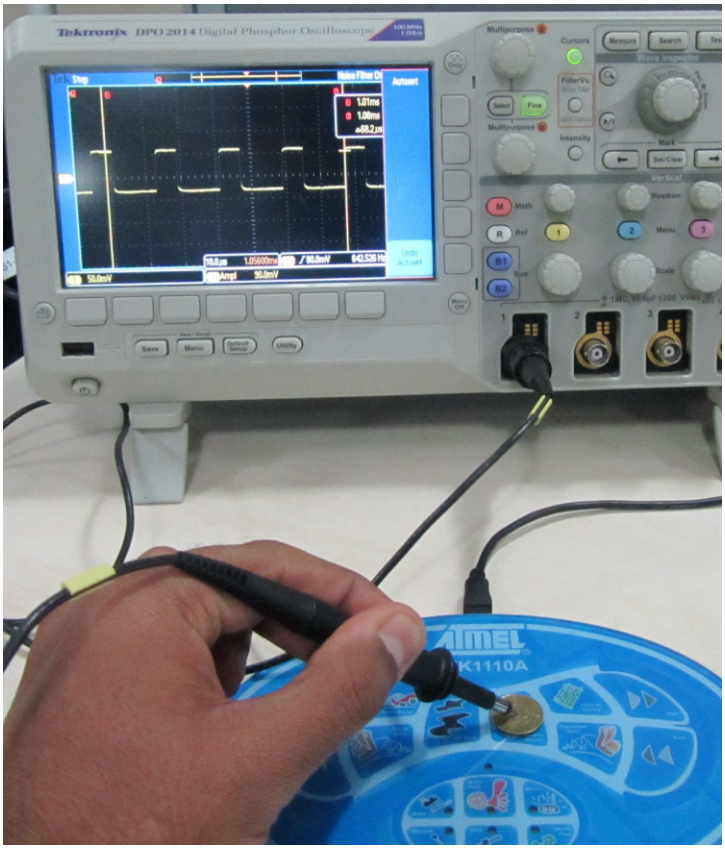
Figure 2-5. Drive Pulse Roll-off and Dwell Time



Note: The Dwell time is a minimum of approximately 125 ns – see Section 5.13 on page 45

One way to determine X-line settling time is to monitor the fields using a patch of metal foil or a small coin over the key (see Figure 2-6). Only one key along a particular X line needs to be observed, as each of the keys along a particular X line are identical. The dwell time should exceed the observed 95% settling of the X-pulse by 25% or more.

Figure 2-6. Probing X-Drive Waveforms With a Coin



2.11 Key Design

For information about key design refer to the *Touch Sensors Design Guide* on the Atmel website.

2.12 PCB Layout, Construction

2.12.1 Overview

It is best to place the chip near the touch keys on the same PCB so as to reduce X and Y trace lengths, thereby reducing the chances for EMC problems. Long connection traces act as RF antennas. The Y (receive) lines are much more susceptible to noise pickup than the X (drive) lines.

Even more importantly, all signal related discrete parts (resistors and capacitors) should be very close to the body of the chip. Wiring between the chip and the various resistors and capacitors should be as short and direct as possible to suppress noise pickup.

Ground planes and traces should NOT be used around the keys and the Y lines from the keys. Ground areas, traces, and other adjacent signal conductors that act as AC ground (such as Vdd) absorb the received key signals and reduce signal-to-noise ratio (SNR) and thus are counterproductive. Ground planes around keys also make water film effects worse.

Ground planes, if used, should be placed under or around the QT1481 chip itself and the associated resistors and capacitors in the circuit, under or around the power supply, and back to a connector, but nowhere else.

2.12.2 LED Traces and Other Switching Signals

Digital switching signals near the Y lines induce transients into the acquired signals, deteriorating the SNR performance of the QT1481. Such signals should be routed away from the Y lines, or the design should be such that these lines are not switched during the course of signal acquisition (bursts).

LED terminals which are multiplexed or switched into a floating state and which are within or physically very near a key structure (even if on another nearby PCB) should be bypassed to either Vss or Vdd with at least a 10 nF capacitor of any type, to suppress capacitive coupling effects which can induce false signal shifts. LED terminals which are constantly connected to Vss or Vdd do not need further bypassing.

2.12.3 PCB Cleanliness

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

CAUTION: If a PCB is reworked in any way, it is highly likely that the behavior of the no-clean flux will change. This can mean that the flux changes from an inert material to one that can absorb moisture and dramatically affect capacitive measurements due to additional leakage currents. If so, the circuit can become erratic and exhibit poor environmental stability.

If a PCB is reworked in any way, clean it thoroughly to remove all traces of the flux residue around the capacitive sensor components. Dry it thoroughly before any further testing is conducted.

2.13 Power Supply Considerations

For Vdd information see Section 6.1 and Section 6.2 on page 58.

As the QT1481 uses the power supply as an analog reference, the power should be very clean and come from a separate regulator. A standard inexpensive Low Dropout (LDO) type regulator should be used; it should not also be used to power other loads such as relays or other high current devices. Load shifts on the output of the LDO can cause Vdd to fluctuate enough to cause false detection or sensitivity shifts.

Ceramic 0.1 µF bypass capacitors should be placed very close and routed with short traces to all power pins of the IC. There should be at least three such capacitors around the part.

2.14 Startup/Calibration Times

The QT1481 employs a rigorous initialization and self-check sequence for EN60730 compliance. If the self-tests are passed, the last step in this sequence enables the serial communication interfaces. The communication interfaces are not enabled if a safety critical fault is detected during the startup sequence. The QT1481 requires initialization times as follows:

1. Normal reset to ability to communicate: 110 ms.
2. From very first power-up to ability to communicate:

   2,200 ms (one time event to initialize all of EEPROM, or to recover EEPROM copy from Flash in the event of EEPROM corruption).
3. From power-up to ability to communicate:

   140 ms in the event the setups have been changed and the part needs to back up the EEPROM to Flash.

The QT1481 determines a reference level for each key by calibrating all the keys immediately after initialization. Each key is calibrated independently and in parallel with all other enabled keys. Calibration takes between 11 and 62 keyscan cycles; each cycle being made up of one sample from each enabled key. The QT1481 ends calibration for a key if its reference has converged with the signal DC level. The calibration time is shortest when the keys signals are stable, typically increasing with increasing noise levels to the maximum of 62 keyscan cycles.

An error is reported for each key where calibration continues for the maximum number of keyscan cycles and the key's reference does not appear to have converged with the signals DC level. Noise levels can vary from key to key such that some keys may take longer to calibrate than others. However, the QT1481 can report during this interval that the key(s) affected are still in calibration via the QT1481 status bits. Table 2-2 shows keyscan cycle times and calibration times per key versus dwell time and burst length for all 48 keys enabled. The values given assume that MSYNC = off, SDC = 0 and STS_DEBUG = 0.

Table 2-2. Keyscan Cycle and Calibration Times

Setups	Keyscan Cycle Time	Calibration Time (min)	Calibration Time (max)
BL = 0 (16 pulses) DWELL = 0 (125 ns) FREQ0 = 0 Signal level = 200 counts	6 ms	66 ms (11 × 6)	372 ms (62 × 6)
BL = 3 (64 pulses) DWELL = 15 (9.9 µs) Signal level = 400 counts	17 ms	187 ms (11 × 17)	1054 ms (62 × 17)

2.15 Reset Input

Should communications with the QT1481 be lost the $\overline{\text{RST}}$ pin can be used to reset the QT1481 to simulate a power-down cycle, in order to then bring the QT1481 up into a known state. The pin is active low, and a low pulse lasting at least 10 µs must be applied to this pin to cause a reset.

To provide for proper operation during power transitions the QT1481 has an internal brownout detector set to 4 V.

The reset pin has an internal 30 kΩ – 60 kΩ resistor. A 2.2 µF capacitor plus a diode to Vdd can be connected to this pin as a traditional reset circuit, but this is not necessary.

A Force Reset command, `0x04`, also generates an equivalent hardware reset where the device is still in communication with the host. Where the QT1481 has detected a failure of one of the internal EN60730 checks and has subsequently locked up in an infinite loop, only a power cycle or an external hardware reset can restore normal operation. It is strongly recommended that the host has control over the $\overline{\text{RST}}$ pin.

If an external hardware reset is not used, this pin may be connected to Vdd or left floating.

2.16 Detection Integrators

See also Section 5.5 on page 41.

The QT1481 features a detection integration mechanism, which acts to confirm a detection in a robust fashion. A per-key counter is incremented each time the key has exceeded its threshold and stayed there for a number of acquisitions. When this counter reaches a preset limit the key is finally declared to be touched.

For example, if the limit value is 10, then the QT1481 has to exceed its threshold and stay there for a minimum of 10 acquisitions before the key is declared to be touched.

The QT1481 uses a two-tier confirmation mechanism having two such counters for each key. These can be thought of as inner loop and outer loop confirmation counters.

The inner counter is referred to as the fast-DI. This acts to attempt to confirm a detection via rapid successive acquisition bursts, at the expense of delaying the sampling of the next key. Each key has its own fast-DI counter and limit value. These limits can be changed via the Setups block on a per-key basis.

The outer counter is referred to as the normal-DI. This DI counter increments whenever the fast-DI counter has reached its limit value. The normal-DI counter also has a limit value which is settable on a per-key basis.

If a normal-DI counter reaches its terminal count, the corresponding key is declared to be touched and becomes active. Note that the normal-DI can only be incremented once per complete keyscan cycle (more slowly), whereas the fast-DI is incremented on the spot without interruption (at the same burst spacing timing).

The net effect of this mechanism is a multiplication of the inner and outer counters and hence a highly noise-resistant sensing method. If the inner limit is set to 5, and the outer to 3, the net effect is a minimum of 5 × 3 = 15 threshold crossings to declare a key as active.

2.17 Sleep

The QT1481 can be configured for automatic sleep using the Sleep Drift Compensation (SDC) setup, and woken with a low pulse applied to the WS pin.

If the sleep feature is enabled using SDC (see Section 5.16 on page 46), and the sleep command (`0x16`) has been issued, the QT1481 sleeps whenever possible to conserve power. Periodically, it should be woken by the host using the WS pin. Upon being woken, the matrix is scanned and the QT1481 returns to sleep unless there is activity which demands further attention. The QT1481 returns to sleep automatically after a period of inactivity, the duration of which is defined by the AWAKE feature.

At least one full matrix scan is always performed after waking up and before returning to sleep. At the end of each matrix scan, the part returns to sleep unless recent activity, such as a touch event, demands further attention. If there has been recent activity, the part performs another complete matrix scan before attempting to sleep once again. This process is repeated indefinitely until the activity stops and the part returns to sleep.

Key touch activity forces the matrix scanning into free run whereby each matrix scan is not interleaved with sleep. The part will not sleep while any key is calibrating or if any touch events are detected at any key in the most recent scan of the key matrix. If the sleep feature is disabled in the setups, the QT1481 never sleeps.

Sleep should be used with caution if the QT1481 is being used in an FMEA or EN60730 compliant design because all operations are stopped within the QT1481 while the part is asleep and the host might have difficulty distinguishing between the EN60730 counters appearing to run slow because the part is intermittently sleeping, and faulty operation. However, in the knowledge it has configured the QT1481 for sleep, the host can take this into account. For example, the host could wake the QT1481 at suitable intervals, check for correct operation and then return the QT1481 to sleep.

Also see “Mains Sync – MSYNC” , Section 5.14 on page 45.

2.18 FMEA Tests

Failure Modes and Effects Analysis (FMEA) is a tool used to determine critical failure problems in control systems. FMEA analysis is being applied increasingly to a wide variety of applications including domestic appliances. To survive FMEA testing the control board must survive any single problem in a way that the overall product can either continue to operate in a safe way, or shut down.

The most common FMEA requirements regard opens and shorts analysis of adjacent pins on components and connectors. However, other criteria must usually be taken into account, for example complete QT1481 failure.

The QT1481 incorporates a number of special self-test features which allow products to pass such FMEA tests easily, and enable key failure to be detected. These tests are performed in an extra burst slot after the last enabled key.

The sequence of tests are performed repeatedly during normal running once all initialization is complete. During initialization, all FMEA error flags are cleared. Any FMEA errors are reported as the tests are performed for the first time.

The FMEA testing is done on all enabled keys in the matrix, and results are reported via the serial interface. Disabled keys are not tested.

Assuming the part does not sleep, the real time that elapses from the start of one sequence of FMEA tests to the start of the next, or the FMEA sequence time, never exceeds 2 s.

Also, since the QT1481 only communicates in slave mode, the host can determine immediately if the QT1481 has suffered a catastrophic failure. The QT1481 can also participate in cross-checking the integrity of the host controller, and even reset the host if no communications have been heard from it in a short while (via the $\overline{\text{STATUS}}$ pin output).

The FMEA tests performed are:

- X drive line shorts to Vdd and Vss
- X drive line shorts to other pins
- X drive signal deviation
- Y line shorts to Vdd and Vss
- Y line shorts to other pins
- X to Y line shorts
- Cs capacitor checks including shorts and opens
- Vref test
- Key gain (see Section 5.19 on page 47)

Other tests incorporated into the QT1481 include:

- A test for signal levels against a preset minimum value (Lower Signal Limit (LSL) setup, see Section 5.18 on page 47). If any signal level falls below this level, an error flag is generated.
- 16-bit CRC communications checks on all data returns.
- *Last-command* command to verify that an instruction was properly received.
- Loss of communications reset of the host controller.

2.19 EN60730 Compliance

The QT1481 also incorporates special test features which, together with the FMEA tests, allow products to achieve IEC/EN/UL60730 compliance with ease. IEC/EN60730 compliance demands dynamic verification of all safety related components and sub-components within a product. The QT1481 is able to verify some sub-components internally, but others require verification by a separate, independent processing unit with another timing source.

To this end the QT1481 exposes a number of internal operating parameters through its serial communications interface and requires the cooperation of a host to check and verify these parameters regularly. It is also necessary for the host to verify the communications by checking and validating the CRC, which the QT1481 appends to data returns. If a CRC check should fail, the host should not rely on the data but retry the transmission.

Occasional CRC failures might be anticipated as a result of noise spikes. Repeated CRC failures might indicate a safety-critical failure. Where the QT1481 is able to verify sub-components internally, but any such verification fails, the QT1481 disables serial communication and locks up in an infinite loop. The host can detect this condition if repeated CRC failures are observed.

During normal operation the host must perform regular reads of the IEC/EN60730 counters (see Section 4.7 on page 28) to verify correct operation of the QT1481. The host must also perform regular reads of the QT1481 status (see Section 4.7 on page 28) and verify there are no errors reported. The FMEA error flag, LSL error flag and Setups CRC error flag must all be considered as part of an IEC/EN60730 compliant design.

The host can try to recover from any safety critical failure by resetting the QT1481 using its $\overline{\text{RST}}$ pin. The host should allow a grace period in consideration of the start-up and initialisation time the QT1481 requires after reset to ability to communicate (see Section 2.14 on page 14).

The sub-components that the QT1481 is able to verify internally are tested repeatedly during the normal running of the device, and the various tests run in parallel. As each test ends the result is recorded and the test is restarted. The real time that elapses from the start of each test to the start of the next iteration of the same test is called the failure detect time, or hazard time, the maximum time for which an error could be undetected.

Each test is broken down into a number of smaller parts, each of which is processed in turn during each matrix scan. Each test is therefore completed either after a number of matrix scans, as shown in Table 2-3.

Table 2-3. Test run times (measured in matrix scans)

Test	Required Matrix Scans to complete test
FMEA	8
Other	18
Variable Memory	2304
Firmware CRC	2000
Setups CRC	44

Table 2-4 shows matrix scan times for Setups that yield the shortest matrix scan time and a much longer scan time resulting from the use of long dwell and low frequency settings.

Table 2-4. Matrix Scan Times

Setups Conditions	Matrix Scan Time (ms)
BL = 0 (16 pulses), DWELL = 0 (0.13 µs), FREQ0 = 1, All keys enabled, FHM = 0, MSYNC = 0 (off), SDC = 0 (sleep disabled), DEBUG=0 (off).	7.5
BL = 3 (64 pulses), DWELL = 13 (5.1 µs), FREQ0 = 25, All keys enabled, FHM = 0, MSYNC = 0 (off), SDC = 0 (sleep disabled), DEBUG = 0 (off).	17

Longer matrix scan times are possible than those shown in Table 2-4 by using even longer dwell times and higher values for FREQ0 (lower burst frequencies), but these are considered extreme settings.

Table 2-5 shows the failure detect times for the internal tests assuming a matrix scan time of 9ms.

Table 2-5. Failure Detect Time

Test	Failure Detect Time (ms)
FMEA	72
Other	162
Variable Memory	20736
Firmware CRC	18000
Setups CRC	396
Conditions: Matrix scan time = 9 ms. QT1481 does not sleep for duration of tests.	

Longer failure detect times are possible than those shown in Table 2-5 where the matrix scan time is longer. The failure detect times are proportional to the matrix scan time. The failure detect time for other setups can therefore be determined by observing the matrix scan time using an oscilloscope and scaling the times given in Table 2-5 accordingly. Alternatively, the failure detect times can be calculated by taking the numbers from Table 2-3 and multiplying them by the matrix scan time.

Unnecessarily long settings of dwell and low burst frequencies should be avoided because these will also result in undesirably long failure detect times.

2.19.1 UL approval / VDE compliance

The QT1481 has been given a compliance test report by VDE and is approved by UL as a component suitable for use in both class B and class C safety critical products. By using this device and following the safety critical information throughout this datasheet, manufacturers can easily add a touch sense interface to their product, and be confident it can also readily pass UL or VDE testing.

2.20 Frequency Hopping

This QT1481 supports frequency hopping, which tries to select a sampling frequency that does not clash with noise at specific frequencies elsewhere in products or product operating environments. It tries to hop away from the noise.

During the acquisition bursts, a sequence of pulses are emitted with a particular spacing, which equates to a particular sampling frequency. If the latter should coincide with significant noise generated elsewhere, touch sensing may be seriously impaired or false detections may occur. To help combat such noise, the burst frequency can either be preset to one specific frequency (with frequency hopping disabled), away from the noisy frequency, or frequency hopping can be enabled and set to switch dynamically between three specific configured frequencies or even set to sweep a configured range of frequencies.

3. Serial Communications

3.1 Introduction

The QT1481 uses either SPI or UART communications modes; it cannot use both at the same time. The QT1481 responds on whichever interface it receives a command. The QT1481 also includes a Debug output interface, which can be used to monitor many operating variables during product development.

The host device always initiates communications sequences; the QT1481 is incapable of chattering data back to the host. This is intentional for FMEA and IEC/EN60730 purposes so that the host always has total control over the communications with the QT1481.

- In **SPI mode** the QT1481 is a slave, so that even return data following a command is controlled by the host.
- In **UART mode**, the QT1481 still only responds to the host after a command, but the responses are not controlled by the host.

A command from the host always ends in a response of some kind from the QT1481. Some transmission types from the host or the QT1481 employ a CRC check byte to provide for robust communications.

A $\overline{\text{DRDY}}$ line that handshakes transmissions is provided. This is needed by the host from the QT1481 to ensure that transmissions are not sent when the QT1481 is busy or has not yet processed a prior command. In UART mode this line is bidirectional, and the QT1481 can use it to suspend transmissions back to the host if the host is busy.

If the host does not observe the correct $\overline{\text{DRDY}}$ timing, random communication errors may result.

Initiating or Resetting Communications:

After a reset, or should communications be lost due to noise or out-of-sequence reception, the host should repeatedly wait for a period not less than the QT1481 communications time-out (110 ms ±5 ms), and send a `0x0F` (return last command) command until the complement of `0x0F`, which is `0xF0`, is received. Then, the host can resume normal run mode communications from a clean start.

Poll rate:

The typical poll rate in normal run operation should be no faster than once per 10 ms. Even 50 ms is more than fast enough to extract status data using the `0x06` command overview (see Section 4.7 on page 28) in most situations. Streaming commands like the `0x0D` command (dump setups (see Section 4.10 on page 31)) or multi-byte response commands like `0x07` can and should pace at the maximum possible rate.

Run Poll Sequence:

In normal run mode the host should limit traffic with a minimalist control structure (see Section 4.19 on page 32). The host should just send a `0x06` command until something requires a deeper state inspection. If there is more than one key in detect, the host should use `0x07` to find which additional keys are in detect. If there is an error, the host should ascertain the error type based on command `0x0B` and take appropriate action.

3.2 $\overline{\text{DRDY}}$ Pin

$\overline{\text{DRDY}}$ is an open-drain output (in SPI mode) or bidirectional pin (in UART mode) with an internal 20 kΩ – 50 kΩ pull-up resistor.

Most communications failures are the result of failure to properly observe the $\overline{\text{DRDY}}$ timing.

Serial communications pacing is controlled by this pin. Use of $\overline{\text{DRDY}}$ is critical to successful communications with the QT1481. In either UART or SPI mode, the host is permitted to perform a data transfer only when $\overline{\text{DRDY}}$ has returned high. Additionally, in UART mode, the QT1481 delays responses to the host if $\overline{\text{DRDY}}$ is being held low by the host.

After each byte transfer, $\overline{\text{DRDY}}$ goes low after a short delay and remains low until the QT1481 is ready for another transfer. A short delay occurs before $\overline{\text{DRDY}}$ is driven low because the QT1481 may otherwise be busy and requires a finite time to respond.

$\overline{\text{DRDY}}$ may go low for a microsecond only. During the period from the end of one transfer until $\overline{\text{DRDY}}$ goes low and back high again, the host should not perform another transfer. Therefore, before each byte transmission the host should first check that $\overline{\text{DRDY}}$ is high again.

If the host wants to perform a byte transfer with the QT1481 it should behave as follows:

1. Wait at least 100 µs after the previous transfer (time S5 in Figure 3-2 on page 23: $\overline{\text{DRDY}}$ is guaranteed to go low before this 100 µs expires).
2. Wait until $\overline{\text{DRDY}}$ is high (it may already be high).
3. Perform the next transfer with the QT1481.

In most cases it takes up to 3 ms for $\overline{\text{DRDY}}$ to return high again. However, this time is longer with some commands or if the STS_DEBUG setup is enabled, as follows:

`0x01` (Setups load): <20 ms
`0x02` (Low Level Cal and Offset): <20 ms
Add 15 ms to the above times if the STS_DEBUG setup is enabled.

Other $\overline{\text{DRDY}}$ specifications:

Min time $\overline{\text{DRDY}}$ is low: 1 µs
Max time $\overline{\text{DRDY}}$ is low after reset: 100 ms

3.3 SPI Communications

No special configuration is required to make the QT1481 operate in SPI mode. The QT1481 responds on the interface which is used to command it. SPI and UART interfaces cannot be used simultaneously.

SPI communications operate in slave mode only, and obey $\overline{\text{DRDY}}$ control signaling. The clocking is as follows:

Clock idle:	High
Clock shift out edge:	Falling
Clock data in edge:	Rising
Max clock rate:	4 MHz

SPI mode requires five signals to operate (see Figure 3-1 on page 22):

MOSI – Master out / Slave in data pin:

Used as an input for data from the host (master). This pin should be connected to the MOSI (DO) pin of the host device.

MISO – Master in / Slave out data pin:

Used as an output for data to the host. This pin should be connected to the MISO (DI) pin of the host. MISO floats in three-state mode between bytes when $\overline{\text{SS}}$ is high to facilitate multiple devices on one SPI bus.

SCK – SPI clock:

Input only clock from host. The host must shift out data on the falling SCK edge, the QT1481 clocks data in on the rising edge. The QT1481 likewise shifts data out on the falling edge of SCK back to the host so that the host can shift the data in on the rising edge.

Note: **Important:** SCK must idle high; it should never float.

$\overline{\text{SS}}$ – Slave select:

input only; acts as a framing signal to the sensor from the host. $\overline{\text{SS}}$must be low before and during reception of data from the host. It must not go high again until the SCK line has returned high; $\overline{\text{SS}}$ must idle high. This pin includes an internal pull-up resistor of 20 kΩ – 50 kΩ. When $\overline{\text{SS}}$ is high, MISO floats.

$\overline{\text{DRDY}}$ – Data Ready:

When high – indicates to the host that the QT1481 is ready to send or receive data. This pin idles high. This pin includes an internal pull-up resistor of 20 kΩ – 50 kΩ. In SPI mode this pin is an output only (that is, open drain with internal pull-up resistor).

Null Bytes:

When the QT1481 responds to a command with one or more response bytes, the host should issue null commands (0x00) to get the response bytes back. The host should not send new commands until all the responses are accepted back from the QT1481 from the prior command via nulls.

New commands attempted during intermediate byte transfers are ignored.

Wake operation:

The QT1481 can be configured to automatically sleep. The host must awaken the QT1481, when required, with a 8.5 µs minimum low level on the WS pin. With the $\overline{SS}$ line tied to WS, the host can simply toggle $\overline{SS}$ low for 8.5 µs minimum to wake the QT1481. The host should not send an actual SPI byte to prevent the QT1481 from seeing a byte it cannot properly interpret due to timing errors during wake-up. Alternatively, $\overline{SS}$ can be driven low 8.5 µs before the first SCK of each transfer.

There is an interval of approximately 1.5 ms from the pulse on WS before the QT1481 is able to resume processing. Transmissions to the QT1481 within this interval are discarded.

The recommended method to re-establish communications after Wake from Sleep is to send the QT1481 a 0x0F ("Get Last Command" command) repeatedly until the correct response comes back (the command's own complement, that is, 0xF0).

SPI Line Noise:

In some designs it is necessary to run SPI lines over ribbon cable across a lengthy distance on a PCB. This can introduce ringing, ground bounce, and other noise problems which can introduce false SPI clocking or false data. Simple RC networks and slower data rates are helpful to resolve these issues.

A CRC is appended to responses in order to detect transmission errors to a high level of certainty.

Figure 3-1. Communications Signals – SPI

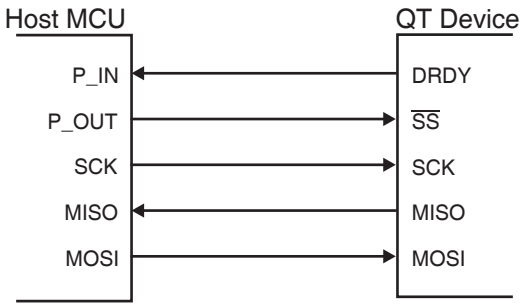
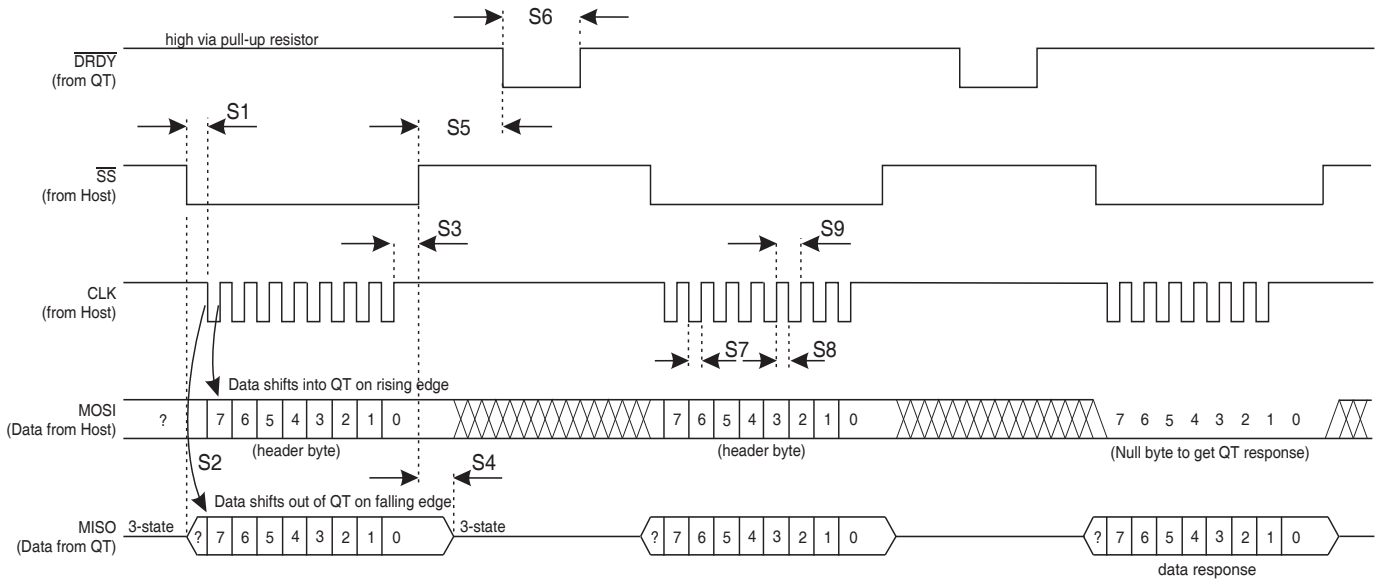


Figure 3-2. SPI Slave-only Mode Timing (Fosc = 16 MHz)



S1: > 125 ns	S2: < 20 ns	S3: > 25 ns	S4: < 20 ns	
S5: < 100 µs	S6: > 1 µs	S7: > 125 ns	S8: > 125 ns	S9: > 250 ns

3.4 UART Communications

See also Section 5.17 on page 46.

UART mode is selected as soon as the QT1481 receives any data on the UART Rx pin. There is no other configuration required to make the QT1481 operate in UART mode.

UART mode communications function in the same basic way as SPI communications. The baud rate is adjusted by means of setup parameter SR (Section 5.17 on page 46). Once a new baud rate has been set, the QT1481 must be reset for the new rate to take effect.

The major difference with SPI mode is that the UART mode is asynchronous and so the host does not clock the QT1481. No framing $\overline{SS}$ or clock signal is required, simplifying the interface greatly. Return data is sent from the QT1481 back to the host when the data is ready.

Multidrop capability:

Tx floats within 10 µs after each transmitted byte. This line can be shared with other UART based peripherals. Tx includes an internal 20 kΩ – 50 kΩ pull-up resistor to Vdd to prevent the line from floating down.

Wake operation:

The QT1481 can be configured to automatically sleep. The host must awaken the QT1481, when required, with a 8.5 µs (minimum) low level on the WS pin. With the Rx line tied to WS the QT1481 can be awaked with a dummy byte from the host. The first received UART byte from the host after a wake should be a `0xFF`; any other byte value could create a framing error. The start bit of the `0xFF` forms a convenient narrow wake pulse without being long enough to be interpreted as a byte during the wake operation.

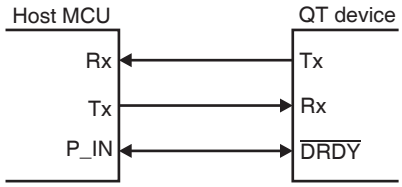
There is an interval of approximately 1.5 ms from the pulse on WS before the QT1481 is able to resume processing. Transmissions to the QT1481 within this interval are discarded.

The recommended method to re-establish communications after Wake from Sleep is to send a `0x0F` (“Get Last Command” command) repeatedly until the correct response comes back (the command's own complement, that is, `0xF0`).

Rx – Receive async data. This pin is an input only.

Tx – Transmit async data. Drives out when transmitting but floats within 10 µs of the end of the stop bit, to allow bussing with several similar devices. Tx should idle high, and it includes an internal 20 kΩ – 50 kΩ resistor to Vdd. Tx is push-pull when transmitting data for good drive characteristics.

Figure 3-3. Communications Signals – UART



UART transmission parameters are:

Baud rate:	9600 – 115,200
Start bits:	1
Data bits:	8
Parity:	None
Stop bits:	1

$\overline{\text{DRDY}}$ in UART mode: Section 3.2 on page 20 applies.

$\overline{\text{DRDY}}$ is bidirectional in UART mode and can be pulled down by either the QT1481 or the host (wire-AND), so that either device can be inhibited from sending data until the other is ready. The host should obey this control line or transmission errors can occur. The host should grant a 10 µs grace period after clamping $\overline{\text{DRDY}}$ low in which it can still accept the start bit of a transmission from the QT1481.

As explained in Section 3.2 on page 20, $\overline{\text{DRDY}}$ is not clamped low immediately after the QT1481 receives a byte; there can be up to a 100 µs delay from the end of the stop bit before $\overline{\text{DRDY}}$ goes low. Sampling of $\overline{\text{DRDY}}$ by the host should occur 100 µs after the byte has been fully sent. If $\overline{\text{DRDY}}$ is already high at this point, or becomes high, then it is clear to send.

Due to the asynchronous nature of UART timing, reception of a byte is considered complete when the receiver detects the stop bit, which is typically some considerable time before the transmitter actually terminates the stop bit. Depending on the baud rate, it is therefore possible for the QT1481 to assert the $\overline{\text{DRDY}}$ pulse and start transmitting a response during the stop bit of the command from the host.

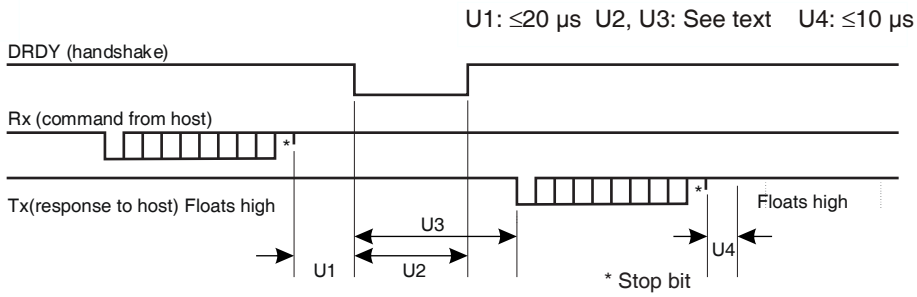
If the host needs to slow the pace of the QT1481 return data, it can assert $\overline{\text{DRDY}}$ before transmitting the stop bit of the command byte.

Null Bytes: Unlike SPI mode, there is no reason to send null bytes to the QT1481 in UART mode. The QT1481 responds to commands with data when ready, subject to the $\overline{\text{DRDY}}$ line being high.

UART Noise: In some designs it is necessary to run Tx and Rx over a lengthy distance. This can introduce ringing, ground bounce, and other noise problems which can corrupt data. Simple RC networks and slower data rates are helpful to resolve these issues. A CRC is appended to responses in order to detect transmission errors to a high level of certainty.

UART Timing Parameters: UART timings are as shown in Figure 3-4 on page 25. Delay timings for parameters U2 and U3 are dependent on the specific command. See Section 3.5.

Figure 3-4. UART Timing



3.5 Debug Output Interface

The QT1481 includes a debug interface which may be used for observing many internal operating variables, in real time, even while the part is actively communicating with a host over either the SPI or UART serial interfaces. The Debug interface provides a useful aid during product development (see Appendix B. on page 64).

4. Control Commands

4.1 Introduction

Refer to Table 4-5 on page 35 for further details.

Suggested command sequencing: See Section 4.19 on page 32.

The QT1481 features a set of commands which are used for control and status reporting. The host device has to send the command to the QT1481 and await a response.

SPI mode: While waiting the host should delay for 100 µs from the end of the command, then start to check if $\overline{\text{DRDY}}$ is, or goes, high. If it is high, then the host master can clock out the resulting byte(s).

Command timeouts in SPI mode: Where a command involves multiple byte transfers to the QT1481, each byte must be transmitted within 110 ms ±5 ms of the prior byte or the command times out. Where a command involves a multiple byte response from the QT1481, the entire command must be completed within 110 ms ±5 ms or the command times out. No error is reported for this condition. The command simply ceases.

UART mode: After the command is sent, the QT1481 sends back the response, usually starting within 1 ms. The host can clamp $\overline{\text{DRDY}}$ low (wire-AND logic) to inhibit a response (for up to 110 ms) if the host is not able to receive the transmission for a while.

Command timeouts in UART mode: Where a command involves multibyte transfers in either direction, each byte must be transmitted within 110 ms ±5 ms of the prior byte or the command times out. No error is reported for this condition. The command simply ceases.

Word return byte order: Where a word or long word is returned (16 or 24 bit number or bit pattern) the low order byte is sent or received first.

Response delays: The QT1481 requires processing time to complete command requests and respond with an answer to the host. These timings are the same for SPI and UART modes. Most commands respond with data in 1 ms maximum; timing parameters U2 and U3 in Figure 3-4 on page 25 are therefore 1 ms maximum for these commands. The exceptions are:

`0x01` (Setups upload):	<20 ms
`0x02` (Low Level Cal and Offset):	<20 ms

Add 15 ms to all stated response times if the STS_DEBUG Setup is enabled.

4.2 Null Command – `0x00`

This command is used to shift back data from the QT1481 in SPI mode. Since the host device is always the master in SPI mode, and data is clocked in both directions, the Null command is required frequently to act as a placeholder where the desire is to only get data back from the QT1481, not to send a command.

In SPI communications, when the QT1481 responds to a command with one or more response bytes, the host can issue a new command instead of a null on the last byte shift operation.

New commands during intermediate byte shift-out operations are ignored, and null bytes should always be used.

4.3 Enter Setups Mode – `0x01`

This command is used to initiate the Setups block transfer from host to QT1481.

The command must be repeated twice within 110 ms or the command fails; the repeating command must be sequential without any intervening command. After the second `0x01` from the host, the QT1481 stops scanning keys and replies with the character `0xFE`. This command suspends normal sensing starting from the receipt of the second `0x01`. A failure of the command causes a timeout.

Each byte in the block must arrive at the QT1481 no later than 110 ms after the previous one or a timeout occurs. Any timeout causes the QT1481 to cancel the block load and go back to normal operation.

If no response comes back, the command was not received and the QT1481 should preferably be reset by the host just in case there are any other problems.

If 0xFE is received by the host from the QT1481, then the host should begin to transmit the block of Setups to the QT1481. The $\overline{\text{DRDY}}$ line handshakes the data. The delay between bytes can be as short as 100 µs but the host can make it longer than this if required, but no more than 110 ms. The last two bytes the host should send is the CRC for the block of data only (the CRC should not include the command in its calculation).

After the block transfer the QT1481 checks the CRC and responds with 0x00 if there was an error. Regardless, it programs the internal EEPROM. If the CRC was correct it replies with a second 0xFE after the EEPROM was programmed.

If there was an error in the block transfer the QT1481 restores the last known good Setups from Flash memory the next time the QT1481 is reset. However until that point, the QT1481 attempts to operate using the new Setups block even if it is corrupt. Note that some Setups do not take effect until the part is reset (for example, baud rate).

At the end of the full block load sequence, the QT1481 restarts sensing without recalibration. Most of the setups in the block take effect immediately, but it is important to reset the QT1481 after a block load to make all the changes effective and permanent. See Section 4.6 on page 27.

Command response timing: Responses to the bytes in the setups block (both $\overline{\text{DRDY}}$ and return byte at the end) by the part can take as long as 20 ms each.

4.4 Low Level Cal and Offset – 0x02

This command must be repeated twice within 110 ms or the command fails. The repeating command must be sequential without any intervening command. After the second 0x02 from the host, the QT1481 replies with the character 0xFD. Shortly thereafter the QT1481 performs a calibration and offset procedure across all keys and restarts operation. If no 0xFD comes back, the command was not properly received or a previous command 0x02 is still being processed. This command takes up to 3 seconds to complete. The host can monitor the progress of the calibration by checking the QT1481 status byte, using command 0x06, during the course of the calibration. The calibration bit will be set throughout the process.

The low level calibration and offset procedure involves the device calibrating each key in turn at each of the operating frequencies selected with FREQ0, FREQ1 and FREQ2, calculating the difference between the signals at those frequencies and storing the results as offsets into CFO_1 and CFO_2 for each key. When the procedure is complete, the host can read back the setups and record CFO_1 and CFO_2 into its own copy of the setups block. The QT1481 does not change the Setups CRC, so there will be a mismatch in the Setups CRC after this command completes. The onus is on the host to compute the CRC and upload a definitive Setups block to the QT1481.

4.5 Cal All – 0x03

This command must be repeated twice within 110 ms or the command fails. The repeating command must be sequential without any intervening command.

After the second 0x03 from the host, the QT1481 replies with the character 0xFC. Shortly thereafter the QT1481 recalibrates all keys and restarts operation.

If no 0xFC comes back, the command was not properly received and the QT1481 should preferably be reset.

The host can monitor the progress of the recalibration by checking the QT1481 status byte, using command 0x06, during the course of the calibration.

A key shows an error flag (using command 0x8*k*) indicating the key calibration failed if the reference did not successfully converge with the signal average, or if its signal is below the low signal threshold.

4.6 Force Reset – 0x04

The command must be repeated twice within 110 ms or the command fails. The repeating command must be sequential without any intervening command. After the second 0x04, the QT1481 replies with the character 0xFB just prior to executing the reset operation.

After any reset, the QT1481 automatically performs a full key calibration on all keys.

The host can monitor the progress of the reset to see when the chip is operating again by checking the QT1481 status byte for recalibration by repeatedly issuing command `0x06` (see Section 4.7).

4.7 QT1481 Overview – `0x06`

Reports the IEC/EN60730 counters, a QT1481 status byte, and the first key declaring detect. The $\overline{\text{STATUS}}$ pin becomes inactive on processing this command, if it was made active by a key touch (or release).

This command returns five individual data bytes, plus two CRC bytes, in the sequence:

1. Device Status
2. Touch Overview
3. 100 ms Counter (for IEC/EN60730 compliance)
4. Matrix Scan Counter (for IEC/EN60730 compliance)
5. Signal Fail Counter (for IEC/EN60730 compliance)

A 16-bit CRC is appended to the response; this CRC folds in the command `0x06` itself initially.

Device Status: This byte is a collection of general status bits, see Table 4-1 on page 28.

Table 4-1. Device Status Bits

Bit	Description
7	Reserved
6	1 = Calibration has failed on an enabled key
5	1 = FMEA failure detected
4	1 = Setups CRC mismatch
3	1 = Mains sync error
2	1 = LSL failure
1	1 = Any key in calibration
0	Reserved

Bit 6: Set if calibration failed on any enabled key.

Bit 5: Set if an FMEA error was detected during operation. This flag will be set during calibration after reset, and should be ignored until calibration has completed. See Section 2.18 on page 16.

Bit 4: Set if a mismatch occurs with the setups CRC. If this happens, it is recommended that the QT1481 be reset, which forces the setups to be recovered from internal Flash. If a reset does not cure the problem, the setups block should be reloaded (see Section 5.31 on page 53) and the QT1481 reset again. The time required to recompute the Setups CRC and report an error is a maximum of 1150 ms.

Bit 3: Set if there was a mains sync error, for example there was no Sync signal detected within the allotted 100 ms amount of time. See Section 5.14 on page 45. This condition is not necessarily fatal to operation, however the QT1481 operates very slowly and may suffer from noise problems if the sync feature is required for noise reasons.

Bit 2: Reports that an enabled key has a low signal reference value, lower than the user-settable LSL value (see Section 5.18 on page 47).

Bit 1: Set if any key is in the process of calibrating.

Touch Overview: This byte gives an initial indication of the number of keys touched as well as the number of the first touched key, as follows:

Table 4-2. Key Touch Information

Bit	Description
7 – 6	0 = No touched keys 1 = 1 key in detect 2 = 2 keys in detect 3 = 3 or more keys in detect
5 – 0	Number of 1st key declaring detect (0 – 47)

Bits 7 – 6: A 2-bit unsigned value indicating the number of touched keys. A value of three indicates that three or more keys are touched.

Bits 5 – 0: A 6-bit unsigned value encoding the first or only key to be touched, in the range 0 – 47. This number is valid only if the value encoded in bits 7, 6 is non-zero, indicating a key is touched.

The IEC/EN60730 counters (100 ms, Matrix Scan, Signal Fail) can be used by the host to check the correct speed and operation of the QT1481. The host must check these values regularly to meet the requirements of IEC/EN60730.

IEC/EN60730 requires that each component of a system be checked for correct operation. Where correct speed of operation must be confirmed and the QT1481 has no way to perform such a cross-check internally, counters are exposed through this command to enable independent cross-checking by the host.

100 ms Counter:

This is an 8-bit unsigned counter that is incremented once every 100 ms, counting 256 steps repeatedly from 0 to 255. When the counter has reached 255 it wraps back to 0 at the next 100 ms interval. The counter should take between 25 and 26 seconds (256 x 100 ms = 25.6 s) to count up from zero to 255 and wrap back to zero again. The host must read this counter regularly and cross-check the counting rate against one of its own clock sources.

If the 100 ms counter is read once every second, for example, the host should find the counter has increased by 10 counts from the value returned at each previous read and should traverse one full count range (256 steps) when the host has read the counter 25 or 26 times. The host should verify that the 100 ms counter is incrementing at the expected rate. If the counter advances faster or slower than expected, there could be a fault with the QT1481 or the host, and the host should adopt an appropriate strategy to meet the required safety standard.

Signal Fail Counter:

This is an 8-bit unsigned counter that is incremented each time a signal capture failure occurs. Signal capture failure can occur where keys are enabled but do not physically exist. Only keys that exist should be enabled; all other keys should be disabled.

Signal capture failure can also occur where heavy noise spikes corrupt the signal; Occasional capture failure is to be expected and does not unduly affect the QT1481 performance. Regular capture failure would extend the key response time.

The host must check this counter regularly. Tests should be made with a heavy noise source during development to determine how the key response time is affect and determine a maximum acceptable count rate for this counter.

Matrix Scan Counter:

This is an 8-bit counter that is incremented before the start of each matrix scan, or keyscan cycle, counting 256 steps repeatedly from 0 to 255. When the counter has reached 255 it wraps back to 0 at the start of the next keyscan cycle. The keyscan cycle time should be measured with an oscilloscope during development.

The Matrix Scan count rate can be calculated directly from this. For example, if the keyscan cycle time is measured as 10 ms, the counter counts 256 steps in 2560 ms (256 × 10 ms). The host must read this counter regularly to check the matrix scan is operating at the expected rate.

If the Matrix Scan counter is read once every 100 ms, for example, the host should find the counter has increased by 10 counts from the value returned at each previous read and should traverse one full count range (256 steps) when the host has read the counter 25 or 26 times. The host should verify the counter is incrementing at the expected rate. If the counter advances faster or slower than expected, there could be a fault with the QT1481 or the host, and the host should adopt an appropriate strategy to meet the required safety standard.

Note: The $\overline{\text{STATUS}}$ pin becomes inactive on processing this command if it was made active by a key touch (or release). See Section 5.20 on page 47 for STS_TOUCH in the STS Setups byte.

4.8 Report Detections for All Keys – 0x07

Returns six bytes which indicate all keys which are in detection, if any, as a bit-field. Key 0 reports in bit 0 of byte 0, the first byte returned. Key 47 is reported in bit 7 of byte 5. See Table 4-3 on page 30.

A 16-bit CRC is appended to the response; this CRC folds in the command 0x07 itself initially.

Table 4-3. Bit Fields for Multiple Key Reporting and Key Numbering

		Bit Number (X Line Number)							
		7	6	5	4	3	2	1	0
Byte Number Returned (Y line #)	0	7	6	5	4	3	2	1	0
	1	15	14	13	12	11	10	9	8
	2	23	22	21	29	19	18	17	16
	3	31	30	29	28	27	26	25	24
	4	39	38	37	36	35	34	33	32
	5	47	46	45	44	43	42	41	40

4.9 Report Error Flags for All Keys – 0x0B

Returns six bytes which show error flags as a bit-field for all keys plus two CRC bytes. Key 0 reports in bit 0 of byte 0, the first byte returned; key 47 is reported in bit 7 of byte 5. See Table 4-3 and Table 5-5 on page 55.

This command reports an error flag for each enabled key if calibration has failed or if the key has a reference below the LSL (see Section 5.18 on page 47), or if the FMEA key gain test fails (see Section 5.19 on page 47).

A 16-bit CRC is appended to the response. This CRC folds in the command 0x0B itself initially.

4.10 Dump Setups Block – 0x0D

This command causes the QT1481 to dump the entire internal Setups block back to the host.

In UART mode, if the transfer is not paced faster than 110 ms ±5 ms per byte the transfer is aborted and the QT1481 times out. This could happen if the host were controlling DRDY. In SPI mode, the entire command must be competed within 110 ms ±5 ms or the transfer could be aborted.

During the transfer, sensing is halted. Sensing is resumed after the command has finished.

A 16-bit CRC is appended to the response; this CRC is the same as the Setups table CRC and is sent LSByte first.

4.11 EEPROM CRC – 0x0E

This command returns the 16-bit CRC calculated on the set-ups block and is sent back LSByte first. The CRC sent back is the same CRC that is appended to the end of the Setups block.

No CRC is appended to the response.

4.12 Return Last Command – 0x0F

This command returns the last received command character, in 1's complement (inverted). If the command is repeated twice or more, it returns the inversion of 0x0F, which is 0xF0.

If a prior command was not valid or was corrupted, it returns the bad command as well.

No CRC is appended to the response.

4.13 Internal Code – 0x10

This command returns an internal code, as a value from 0 – 255.

A 16-bit CRC is appended to the response. This CRC folds in the command 0x10 itself initially.

4.14 Internal Code – 0x13

This command returns an internal code as a value from 0 – 255.

A 16-bit CRC is appended to the response; this CRC folds in the command 0x13 itself initially.

4.15 Sleep – 0x16

The QT1481 replies with the character 0xE9 before setting an internal flag to indicate that low power sleep mode has been requested. The flag does not force the QT1481 to sleep immediately but requests the QT1481 to sleep at the end of each matrix scan whenever possible. Sleep must also be enabled in Setups (see Section 5. on page 38), otherwise this command has no effect.

The internal flag is cleared upon receipt of any valid command except command 0x16, but the QT1481 must already be awake in order to receive the command. The QT1481 is awake for at least one keyscan cycle time after a wake-up pulse at the WS pin. To cancel the internal sleep request flag, issue a wake-up pulse at the WS pin and then send any valid command other than 0x16 within one matrix scan cycle. The matrix scan cycle time is dependent on a number of Setups parameters and should be measured using an oscilloscope.

See Section 2.17 on page 16 for details of the sleep behavior.

4.16 Data Set for One Key – 0x4*k*

Returns the data set for key k, where k = 0 – 47. To form this command, the key number is logical-OR'd into the byte 0x40. This command returns 5 data bytes, plus two CRC bytes, in the sequence:

Signal	2 bytes
Reference	2 bytes
Normal Detect Integrator	1 byte

Signal and Reference are returned LSByte first.

A 16-bit CRC is appended to the response. This CRC folds in the command 0x4*k* itself initially.

4.17 Status for Key *k* – 0x8*k*

Returns a bit-field for key *k* where *k* is from 0 – 47. The bit-field is explained in Table 4-4:

Table 4-4. Status for Key *k*

Bit	Description
7	1 = Reserved
6	1 = Reserved
5	1 = Reserved
4	1 = FMEA KGTT test failed for this key
3	1 = Key is in detect
2	1 = Signal ref < LSL (low signal error). See Section 5.18 on page 47.
1	1 = Key is undergoing calibration
0	1 = Calibration on this key failed

Bit 4 (FMEA KGTT) will be set during calibration after reset, and should be ignored until calibration has completed. A 16-bit CRC is appended to the response. This CRC folds in the command 0x8*k* itself initially.

4.18 Cal Key *k* – 0xC*k*

This command must be repeated twice within 110 ms or the command fails. The repeating command must be sequential without any intervening command.

This command functions the same as the 0x03 Cal All command except that this command only affects one key *k* where *k* is from 0 to 47.

The chosen key *k* is recalibrated in its native timeslot. Normal running of the part is not interrupted and all other keys operate correctly throughout. This command is for use only during normal operation to try to recover a single key that has failed or is not calibrated correctly.

Returns the 1's complement of 0xC*k* just before the key is recalibrated.

4.19 Command Sequencing

To interface the QT1481 with a host, the flow diagram of Figure 4-1 on page 34 is suggested. The actual settings of the Setups block used should normally just be the default settings except where changes are specifically required, such as for sensitivity, timing, or AKS changes.

The circles in this drawing are communications interchanges between host and sensor. The rectangles are internal host states or processing events. If any communications exchange fails, either the QT1481 fails to respond within the allotted time, or the response CRC is incorrect, or the response is out of context (the response is clearly not for the intended command). In these cases the host should just repeat the command.

The control flow spends 99% of its time alternating between the two states within the dashed rectangle. If a key is detected, the control flow enters Key Detection Processing.

Stuck Key Detection processing (0xC*k*) is optional, since the QT1481 contains the max on-duration timeout function and can therefore recalibrate the stuck key automatically. However, the host can recalibrate stuck keys with greater flexibility if the recalibration timeouts are set to infinite and the host recalibrates them under specific conditions.

Error handling takes place whenever an error flag is detected, or the QT1481 stops communicating (not shown). The error handling procedure is up to the designer. However normally this would entail shutting down the product if the error is serious enough (for example, a key that will not calibrate, or a FMEA \ IEC/EN60730 class error).

Normally it is not required to reload the setups, since the QT1481 itself stores a backup copy of these in Flash memory should the EEPROM become corrupt. However the host should reset the QT1481 so that the QT1481 copies the Flash setups to EEPROM, and then the host should check that the EEPROM CRC is correct.

Only if this fails should the EEPROM be reloaded by the host. One exception to this rule is just after power-up, since a CRC error in the EEPROM setups at this point is clearly a critical error that would require reloading. This happens at the factory, during the very first power-up cycle.

The *Last Command* command can be used at any time to resynchronize failed communications, for example due to timing errors.

Figure 4-1. Suggested Communications Flow

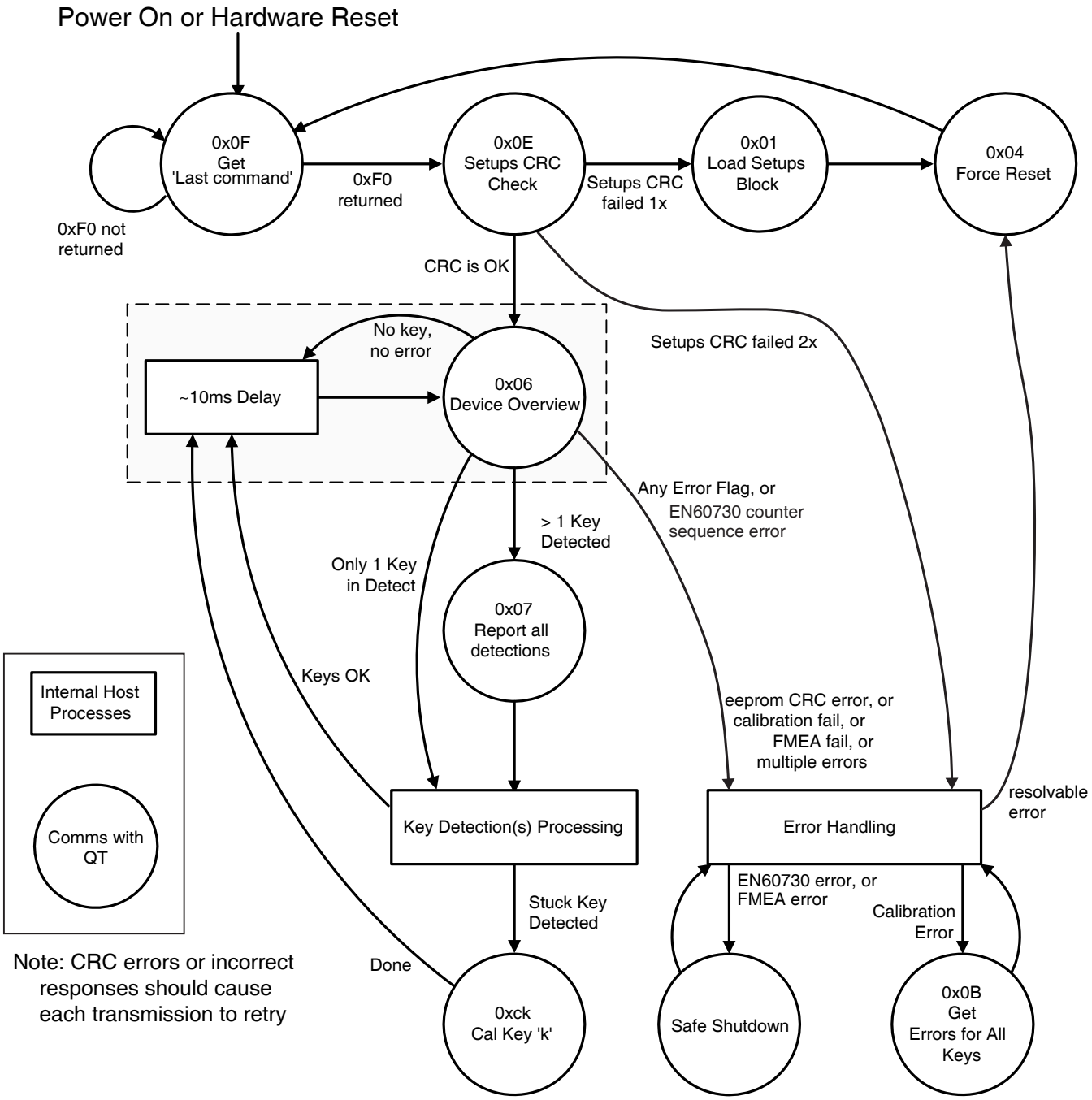


Table 4-5. Command Summary

Hex	Name	Description	#/Cmd	# Rtnd	Rtn Range	CRC	Description	Page
0x00	Null command	Used to get data back in SPI mode	1	1	0 – 0xFF	–	Flushes pending data from QT1481; one required to extract each response byte.	26
0x01	Enter Setups mode	Enter Setups, stop sensing; followed by block load of binary Setups of length nn. Command must be repeated twice consecutively without any intervening command in 110 ms to execute. Sensing autorestarts, however, the QT1481 should be reset after the block load to ensure all new setups take effect.	2 + nn + 2	2	0xFE + 0xFE OR 0xFE + 0x00 (err)	16	First 0xFE issued when ready to receive data, second 0xFE issued when all loaded and burned; else timeout. If two commands not received in 110 ms, times out and no response is issued. Part times out if each byte not received within 110 ms of previous byte. If CRC failure, returns 0x00 instead of 0xFE Data block length is nn + 2 (CRC-16). LSL and CRC should be sent low byte first. Internal EEPROM updates regardless of CRC health, but, if the CRC is bad, the EEPROM is declared invalid and thus on reset the EEPROM is restored from Flash backup, overwriting the desired (but corrupt) new setups.	26
0x02	Low Level Calibration and Offset	Forces QT1481 to calibrate at all three selected frequencies. Command must be repeated twice consecutively without any intervening command in 110 ms to execute. After completion, host should read back the setups block, and then upload the Setups block together with the correct CRC.	2	1	0xFD	–	0xFD returned if command successfully received and QT1481 is not processing a previous 0x02 command.	27
0x03	CAL all	Forces QT1481 to recalibrate all keys; re-enters RUN mode afterwards automatically; 0x03 must be repeated twice consecutively without any intervening command in 110 ms to execute	2	1	0xFC	–	Returns 1's complement of command to acknowledge command once the calibration has been initiated. If two commands not received in 110 ms, times out and no response is issued.	27
0x04	Force reset	Forces QT1481 to reset. Command must be repeated twice consecutively without any intervening command in 110 ms to execute	2	1	0xFB	–	Returns 1's complement of command to acknowledge command prior to reset. If two commands not received in 110 ms, times out and no response is issued.	27

Table 4-5. Command Summary (Continued)

Hex	Name	Description	#/Cmd	# Rtnd	Rtn Range	CRC	Description	Page
0x06	QT1481 Overview	Device status, indication of first key touched, IEC/EN60730 counters.	1	7	0 – 0xFF each byte	16	Returns 5 individual data bytes in following sequence: Device Status 1st key (range 0 – 47) 100 ms Counter (host must read for IEC/EN60730 compliance) Matrix Scan Counter (host must read for IEC/EN60730 compliance) Signal Fail Counter (host must read for IEC/EN60730 compliance) 16-bit CRC (of command + return data) is appended to return	28
0x07	Report all keys	Sends back all key detect status bits (bit-field)	1	8	0 – 0xFF 6 bytes	16	16-bit CRC (of command + return data) is appended to return	30
0x0B	Error flags for all	Error bit fields	1	8	0 – 0xFF 6 bytes	16	16-bit CRC (of command + return data) is appended to return	30
0x0D	Dump Setups	Returns Setups block area followed by CRC. Scanning is halted and then autorestarted after the command has completed.	1	nn+2	0 – 0xFF nn + 2 bytes	16	Dump of fixed length nn followed by CRC-16 CRC is same as CRC at end of Setups block load. LSL and CRC words are sent back to host low byte first. In UART mode, part times out if each byte not transmitted within 110 ms of previous byte. This can happen if $\overline{\text{DRDY}}$ is driven by the host. In SPI mode, the entire command must be completed within 110 ms.	31
0x0E	EEPROM CRC	Gets EEPROM CRC	1	2	0 – 0xFFFF	16	CRC-16 only on Setups block This CRC is the same as the CRC at the end of Setups block load. This word is returned low byte first.	31
0x0F	Return last command	Returns last command received	1	1	0 – 0xFF	–	Returns 1's complement of last command even if bad	31
0x10	Return internal code	Diagnostic code for factory use.	1	3	0 – 0xFF	16	Returns internal code. 16-bit CRC (of command + return data) is appended to return	31
0x13	Return internal code	Diagnostic code for factory use.	1	3	0 – 0xFF	16	Returns internal code. 16-bit CRC (of command + return data) is appended to return	31

Table 4-5. Command Summary (Continued)

Hex	Name	Description	#/Cmd	# Rtnd	Rtn Range	CRC	Description	Page
0x16	Sleep	Enter sleep at end of each matrix scan if no activity and if sleep is also enabled in Setups	1	1	0xE9	–	Returns 1's complement of command to acknowledge; wakes on INT to scan matrix before returning to sleep. Sleep disabled on receipt of any other valid command.	31
0x4*k*	Data for 1 key	Get signal, ref, Norm DI for key *k* {0 – 47} Signal: 2 bytes; Ref: 2 bytes; Norm DI: 1 byte	1	7	0 – 0xFF Each byte	16	Signal and ref are Tx as 2 bytes, LSByte first. 16-bit CRC (of command + return data) is appended to return	31
0x8*k*	Status for key k	Get status byte for key *k* {0 – 47}	1	3	0 – 0xFF	16	Bits 7 – 5: reserved Bit 4: 1 = FMEA KGTT test failed on this key Bit 3: 1 = key is in detect Bit 2: 1 = (Ref < LSL) Bit 1: 1 = key is in calibration Bit 0: 1 = calibration of this key failed 16-bit CRC (of command + return data) is appended to return	32
0xC*k*	CAL key k	Force calibration of key *k* where *k* = 0 – 47. Command must be repeated twice consecutively without any intervening command in 110 ms to execute	2	1	0x10 – 0x3F	–	Used in Run mode. Normal sensing of other keys not affected. CAL of k only takes place in the key's normal timeslot. Returns the ones complement of the command char, once the calibration is scheduled.	32

5. Setups

The QT1481 calibrates and processes all signals using a number of algorithms specifically designed to provide for high survivability in the face of adverse environmental challenges. It provides a large number of processing options which can be user-selected to implement very flexible, robust keypanel solutions.

User-defined Setups are employed to alter the algorithm to suit each application. The setups are loaded into the QT1481 in a block load over one of the serial interfaces and stored in an onboard EEPROM array. After a setups block load, the QT1481 should be reset to allow the new Setups block to be shadowed in internal Flash ROM and to allow all the new parameters to take effect. This reset can be either a hardware or software reset.

Refer to Table 5-4 on page 53 for a list of all Setups.

Block length issues: The setups block is 350 bytes long (including the two CRC bytes) to accommodate 48 keys. This can be a burden on smaller host controllers with limited memory. In larger quantities the QT1481 can be procured with the setups block preprogrammed from Atmel. If the application only requires a small number of keys (such as 16) then the setups table can be compressed in the host by filling large stretches of the Setups area with nulls.

Many setups employ lookup-table (LUT) value translation. The Setups Block Summary on page 56 shows all translation values.

Default values shown are factory defaults.

5.1 Negative Threshold – NTHR

The negative threshold value is established relative to a key signal reference value. The threshold is used to determine key touch when crossed by a negative-going signal swing after having been filtered by the detection integrator. Larger absolute values of threshold desensitize keys since the signal must travel farther in order to cross the threshold level. Conversely, lower thresholds make keys more sensitive.

As Cx and Cs drift, the reference point drift-compensates for these changes at a user-settable rate. The threshold level is recomputed whenever the reference point moves, and thus it is also drift compensated.

The amount of NTHR required depends on the amount of signal swing that occurs when a key is touched. Thicker panels or smaller key geometries reduce key gain (signal swing from touch), thus requiring smaller NTHR values to detect touch.

The negative threshold is programmed on a per-key basis using the Setup process. See Table 5-7 on page 56 and also Section 5.2 on page 39 (Threshold Multiplier – THRM)

Typical values: 3 to 8
(7 to 12 counts of threshold; 4 is internally added to NTHR to generate the threshold).

Default value: 6
(10 counts of threshold)

5.2 Threshold Multiplier – THRM

It is sometimes useful to be able to operate the QT1481 with much higher detect thresholds than can be set with NTHR alone. The Threshold Multiplier (THRM) is a multiplier which extends the range of NTHR and PTHR considerably. The operating detect threshold for a key is arrived at by multiplying NTHR or PTHR for that key by THRM. Note that the detect threshold range is extended at the expense of the step size. Table 5-1 shows the extended threshold range for each value of THRM.

Table 5-1. Extended Detect Threshold

THRM	Multiplier	Extended Detect Threshold
0	× 1	4 – 18
1	× 2	8 – 36
2	× 4	16 – 72
3	× 8	32 – 144

This function is programmed on a global basis. See Table 5-7 on page 56.

THRM Possible range: 0, 1, 2, 3 (× 1, × 2, × 4, × 8)

THRM Default value: 0 (× 1)

5.3 Positive Threshold – PTHR

The positive threshold is used to provide a mechanism for recalibration of the reference point when a key signal moves abruptly to the positive. This condition is not normal, and usually occurs only after a recalibration when an object is touching the key and is subsequently removed. The desire is normally to recover from these events quickly.

Positive hysteresis: PHYST is fixed at 12.5% of the positive threshold value and cannot be altered.

Positive threshold levels are programmed using the Setup process on a per-key basis. See also Section 5.2 on page 39 (Threshold Multiplier – THRM)

Typical values: 1 to 4
(5 to 8 counts of threshold; 4 is internally added to PTHR to generate the threshold)

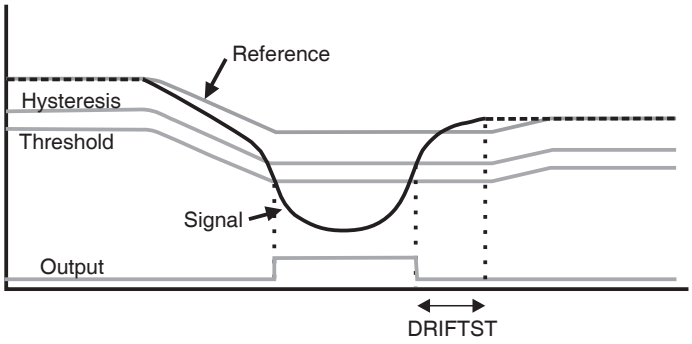
Default value: 2
(6 counts of threshold)

5.4 Drift Compensation – NDRIFT, PDRIFT

Signals can drift because of changes in Cx and Cs over time and temperature. It is crucial that such drift be compensated for, or false detections and sensitivity shifts can occur.

Drift compensation (Figure 5-1 on page 40) is performed by making the reference level track the raw signal at a slow rate, but only while there is no detection in effect. The rate of adjustment must be performed slowly, otherwise legitimate detections could be ignored. The QT1481 drift compensates using a slew-rate limited change to the reference level; the threshold and hysteresis values are slaved to this reference.

Figure 5-1. Thresholds and Drift Compensation



When a finger is sensed, the signal falls since the human body acts to absorb charge from the cross-coupling between X and Y lines. An isolated, untouched foreign object (a coin, or a water film) causes the signal to rise very slightly due to an enhancement of coupling. This is contrary to the way most capacitive sensors operate.

Once a finger is sensed, the drift compensation mechanism ceases since the signal is legitimately detecting an object. Drift compensation only works when the signal in question has not crossed the negative threshold level.

The drift compensation mechanism can be made asymmetric if desired; the drift-compensation can be made to occur in one direction faster than it does in the other simply by changing the NDRIFT and PDRIFT Setups parameters. This can be done on a per-key basis.

Specifically, drift compensation should be set to compensate faster for increasing signals than for decreasing signals. Decreasing signals should not be compensated quickly, since an approaching finger could be compensated for partially or entirely before even touching the touch pad. However, an obstruction over the sense pad, for which the sensor has already made full allowance, could suddenly be removed leaving the sensor with an artificially suppressed reference level and thus become insensitive to touch. In the latter case, the sensor should compensate for the object's removal by raising the reference level relatively quickly.

Drift compensation and the detection time-outs work together to provide for robust, adaptive sensing. The time-outs provide abrupt changes in reference calibration depending on the duration of the signal event.

NDRIFT and PDRIFT are effective while the part is awake. If sleep is enabled, SDC must also be configured so drift compensation operates at the desired rate.

NDRIFT Typical values: 9 to 11
(2 to 3.3 s / count of drift compensation translation via LUT, page 56)

NDRIFT Default value: 10
(2.5 s / count of drift compensation)

PDRIFT Typical values: 9 to 11
(2 to 3.3 s / count of drift compensation; translation via LUT, page 56)

PDRIFT Default value: 10
(2.5 s / count of drift compensation)

5.5 Detect Integrators – NDIL, FDIL

The NDIL parameter is used to enable and disable keys in the matrix and to provide signal filtering. To enable a key, its NDIL parameter should be nonzero (NDIL = 0 disables a key).

To suppress false detections caused by spurious events like electrical noise, the QT1481 incorporates a detection integrator or DI counter mechanism. A per-key counter is incremented each time the key has exceeded its threshold and is decremented each time the key does not exceed its threshold. When this counter reaches a preset limit the key is finally declared to be touched.

The DI mechanism uses two counters. The first is the fast DI counter FDIL. When a key signal is first noted to be below the negative threshold, the key enters 'fast burst' mode. In this mode the burst is rapidly repeated for up to the specified limit count of the fast DI counter. Each key has its own counter and its own specified fast-DI limit (FDIL), which can range from 1 to 15. (FDIL = 0 is invalid; do not use).

When fast-burst is entered the QT1481 locks onto the key and repeats the acquire burst until the fast-DI counter reaches FDIL or drops back to zero. After this the QT1481 resumes normal keyscanning and goes on to the next key.

The Normal DI counter counts the number of times the fast-DI counter reached its FDIL value. The Normal DI counter can only increment once per complete scan of all keys. Only when the Normal DI counter reaches NDIL does the key become formally active.

The net effect of this is that the sensor can rapidly lock onto and confirm a detection with many confirmations, while still scanning other keys. The ratio of fast to normal counts is completely user-settable via the Setups process. The total number of required confirmations is equal to FDIL times NDIL.

- If FDIL = 6 and NDIL = 2, the total detection confirmations required is 12, even though the QT1481 scanned through all keys only twice. The DI is extremely effective at reducing false detections at the expense of slower reaction times. In some applications a slow reaction time is desirable. TheDI can be used to intentionally slow down touch response in order to require the user to touch longer to operate the key.
- If FDIL = 1, the QT1481 functions conventionally. Each channel acquires only once in rotation, and the normal detect integrator counter (NDIL) operates to confirm a detection. Fast-DI is in essence not operational.
- If FDIL ≥2, then the fast-DI counter also operates in addition to the NDIL counter.
- If Signal <NTHR: The fast-DI counter is incremented towards FDIL due to touch.
- If Signal >NTHR then the fast-DI counter is decremented due to lack of touch.

Disabling a key: If NDIL = 0, the key becomes disabled. Keys disabled in this way are pared from the burst sequence in order to improve sampling rates and thus response time. See Section 2. on page 7.

Note: It is very important to disable keys that do not physically exist in the layout, otherwise real keys can incorrectly report their signal as zero.

This function is programmed on a per-key basis. Do not use FDIL = 0 because it is invalid.

NDIL Typical values: 2, 3

NDIL Default value: 2

FDIL Typical values: 4 to 6

FDIL Default value: 5

5.6 Detect Integrator Multiplier – DIM

It is sometimes useful to be able to operate the QT1481 with a higher detect integrator limit than can be set with NDIL alone. The Detect Integrator Multiplier (DIM) is a multiplier which extends the range of NDIL. The operating detect integrator limit for a key is arrived at by multiplying NDIL for that key by DIM. Note that the detect integrator range is extended at the expense of the step size. Table 5-2 shows the extended detect integrator range for each value of DIM.

This function is programmed on a global basis. See Table 5-7 on page 56.

Table 5-2. Extended Detect Integrator Limit

DIM	Multiplier	Extended Integrator Limit
0	× 1	1 – 15
1	× 2	2 – 30
2	× 4	4 – 60
3	× 8	8 – 120

If a key is disabled with NDIL = 0, DIM is ignored for that key.

DIM Default value: 0 (× 1)

DIM Possible range: 0, 1, 2, 3 (× 1, × 2, × 4, × 8)

5.7 Negative Recal Delay – NRD

If an object unintentionally contacts a key resulting in a detection for a prolonged interval it is usually desirable to recalibrate the key in order to restore its function, perhaps after a time delay of some seconds.

The Negative Recal Delay timer monitors such detections. If a detection event exceeds the timer's setting, the key is automatically recalibrated. After a recalibration has taken place, the affected key once again functions normally even if it is still being contacted by the foreign object. This feature is set on a per-key basis using the NRD setup parameter.

NRD can be disabled by setting it to zero (infinite timeout) in which case the key never autorecalibrates during a continuous detection (but the host could still command it).

NRD is set using one byte per key, which can range in value from 0 – 254. NRD above 0 is expressed in 0.5 s increments. Thus if NRD = 120, the timeout value is actually 60 seconds.

Note: 255 is an illegal number; do not use.

NRD Typical values: 20 to 60 (10 s to 30 s)

NRD Default value: 20 (10 s)

NRD Range: 0 – 254 (∞, 0.5 s – 127 s)

5.8 Positive Recalibration Delay – PRD

A recalibration can occur automatically if the signal swings more positive than the positive threshold level. This condition can occur if there is positive drift but insufficient positive drift compensation, or, if the reference moved negative due to a NRD autorecalibration, and thereafter the signal rapidly returned to normal (positive excursion).

As an example of the latter, if a foreign object or a finger contacts a key for period longer than the Negative Recal Delay (NRD), the key is by recalibrated to a new lower reference level.

Then, when the condition causing the negative swing ceases to exist (the object is removed), the signal can suddenly swing back positive to near its normal reference.

It is almost always desirable in these cases to cause the key to recalibrate quickly so as to restore normal touch operation. The time required to do this is governed by PRD. In order for this to work, the signal must rise through the positive threshold level PTHR continuously for the PRD period.

After the PRD interval has expired and the auto-recalibration has taken place, the affected key once again functions normally. PRD is set on a per-key basis.

The functioning of the PRD setting is determined by an index to a lookup table, found on page 56. The values of time can range from 0.1 s to 25 s. Setting the parameter to 0 disables the feature.

PRD Typical values: 5 to 8 (0.7 s to 2.0 s)
PRD Default value: 6 (1 s)
PRD Range: 0 – 15 (∞, 0.1 s – 25 s)

5.9 Burst Length – BL

The signal gain for each key is controlled by circuit parameters as well as the burst length.

The burst length is simply the number of times the charge-transfer (QT) process is performed on a given X line. Each QT process is simply the pulsing of an X line once, with corresponding Y lines enabled to capture the resulting charge passed through the keys capacitance Cx.

QT1481 uses a fixed number of QT cycles which are executed in burst mode. There can be up to 64 QT cycles in a burst, in accordance with the list of permitted values shown in Table 5-7 on page 56.

Increasing burst length directly affects key sensitivity. This occurs because the accumulation of charge in the charge integrator is directly linked to the burst length. The burst length can be set for each key individually; charge is selectively captured on all Y lines simultaneously during the burst on each X line.

Apparent touch sensitivity is also controlled by the Negative Threshold level (NTHR). Burst length and NTHR interact. Normally burst lengths should be kept as short as possible to limit RF emissions, but NTHR should be kept above 6 to reduce false detections due to external noise. The detection integrator mechanism also helps to prevent false detections.

BL Typical values: 2, 3 (48, 64 pulses / burst)

BL Default value: 2 (48 pulses / burst)

BL Possible values: 0, 1, 2, 3 (16, 32, 48, 64 pulses)

5.10 Adjacent Key Suppression Technology – AKS

The QT1481 incorporates Adjacent Key Suppression (AKS) technology that can be selected on a per-key basis. AKS technology permits the suppression of multiple key presses based on relative signal strength. This feature assists in solving the problem of surface moisture which can bridge a key touch to an adjacent key, causing multiple key presses. This feature is also useful for panels with tightly spaced keys, where a fingertip might inadvertently activate an adjacent key.

AKS technology works for keys that are AKS-enabled anywhere in the matrix and is not restricted to physically adjacent keys. The QT1481 has no knowledge of which keys are actually physically adjacent. When enabled for a key, Adjacent Key Suppression causes detections on that key to be suppressed if any other AKS-enabled key in the panel has a more negative signal deviation from its reference during the DI process. Once a key reaches detect it stays in detect as long as the touch remains, regardless of the signal strength on any other AKS-enabled keys.

This feature does not account for varying key gains (burst length) but ignores the actual negative detection threshold setting for the key. If AKS-enabled keys have different sizes, it may be necessary to reduce the gains of larger keys to equalize the effects of AKS technology. The signal threshold of the larger keys can be altered to compensate for this without causing problems with key suppression.

Adjacent Key Suppression works to augment the natural moisture suppression of narrow gated transfer switches creating a more robust sensing method.

AKS Default value: 0 (Off)

5.11 Oscilloscope Sync – SSYNC

The S_Sync pin can output a positive pulse oscilloscope sync that brackets the burst of a selected X line. More than one burst can output a sync pulse as determined by the Setups parameter SSYNC for each X line (see Table 5-3).

This feature is invaluable for diagnostics; without it, observing signals clearly on an oscilloscope for a particular burst is very difficult.

The bits of this Setup byte are allocated as shown in Table 5-3.

Table 5-3. SSYNC Bits

Bit	Scope Sync Output When Burst On...
7	X7
6	X6
5	X5
4	X4
3	X3
2	X2
1	X1
0	X0

Note: STS_DEBUG shares the use of Pin 43 with SSYNC, but only one feature should be enabled at a time. To prevent interference, all SSYNC bits should be set to zero (Off) if Debug output is desired.

This function is supported in Atmel QmBtn PC software.

SSYNC Default value: 0 (Off)

5.12 Negative Hysteresis – NHYST

The QT1481 employs programmable hysteresis levels of 6.25%, 12.5%, 25%, or 50%. The hysteresis is a percentage of the distance from the threshold level back towards the reference, and defines the point at which a touch detection drops out. A 12.5% hysteresis point is closer to the threshold level than to the signal reference level.

Hysteresis prevents chatter and works to make key detection more robust. Hysteresis is only used once the key has been declared to be in detection, in order to determined when the key should drop out.

Excessive amounts of hysteresis can result in stuck keys that do not release. Conversely, low amounts of hysteresis can cause key chatter due to noise or minor amounts of finger motion.

The hysteresis levels are set for all keys only; it is not possible to set the hysteresis differently from key to key.

NHYST Typical values: 0, 1 (6.25%, 12.5%)

NHYST Default value: 1 (12.5%)

NHYST Possible range: 0, 1, 2, 3 (6.25%, 12.5%, 25%, 50%)

5.13 Dwell Time – DWELL

The Dwell parameter in Setups causes the acquisition pulses to have differing charge capture durations. Generally, shorter durations provide for enhanced surface moisture suppression, while longer durations are required where the keypanel design includes higher-resistance tracks such as silver and ITO. Longer durations are also usually more compatible with EMC requirements.

Longer dwell times permit the use of larger series resistors in the X and Y lines to suppress RFI effects, without compromising key gain (Section 2.10 on page 12).

This setup lets the designer trade one requirement for another.

DWELL Typical value: 1 (188 ns)

DWELL Default value : 1 (188 ns)

DWELL Possible values: 0 – 15 (125 ns – 9.9 µs), accuracy is ±10%

5.14 Mains Sync – MSYNC

The MSync feature uses the WS pin. The Sleep and Sync features can be used simultaneously, in which case the QT1481 wakes on the mains sync signal, scans the matrix and then returns to sleep automatically.

External fields can cause interference leading to false detections or sensitivity shifts. Most fields come from AC power sources. RFI noise sources are heavily suppressed by the low impedance nature of the QT circuitry itself.

Noise such as from 50 Hz or 60 Hz fields becomes a problem if it is uncorrelated with acquisition signal sampling; uncorrelated noise can cause aliasing effects in the key signals. To suppress this problem the WS input allows bursts to synchronize to the noise source. This same input can also be used to wake the part from a low-power Sleep state.

The noise sync operating mode is set by parameter MSYNC in Setups.

The sync occurs only at the burst for the lowest numbered enabled key in the matrix. If it does not sleep at the end of the matrix scan, the QT1481 waits for the sync signal for up to 100 ms after the end of a preceding full matrix scan, then the matrix is scanned in its entirety again. If the QT1481 sleeps, it waits indefinitely for the mains sync.

The sync signal drive should be a buffered logic signal, or perhaps a diode-clamped signal, but never a raw AC signal from the mains. If mains sync is enabled and sleep is disabled, the QT1481 synchronizes to the falling sync edge. However, if both mains sync and sleep are enabled, the QT1481 is sensitive to a low level on WS. A first matrix scan occurs when the low level is first detected, and further matrix scans occur for as long as WS is held low. It is therefore recommended that WS is driven low for less than a single matrix scan time.

Since Noise sync is highly effective and inexpensive to implement, it is strongly advised to take advantage of it anywhere there is a possibility of encountering low frequency (50/60 Hz) electric fields. Atmel QmBtn software can show such noise effects on signals, and therefore assist in determining the need to make use of this feature.

If the sync feature is enabled but no sync signal exists, the sensor continues to operate but with a delay of 100 ms from the end of one scan to the start of the next, and hence has a slow response time. A failed Sync signal (one exceeding a 100 ms period) causes an error flag (see command `0x06`). From reset, the QT1481 first reports a mains sync error after initialisation followed by a delay of 100 ms waiting for the sync signal. This time interval may be determined by adding 100 ms to the initialisation times stated in Section 2.14 on page 14.

MSYNC Default value: 0 (Off)
MSYNC Possible range: 0, 1 (Off, On)

5.15 Restart Interrupted Burst – RIB

The RIB parameter in Setups allows a burst to be interrupted, and restarted, by host communications over the serial bus. The QT1481 has limited processing resources available such that a burst and host communication cannot both be serviced simultaneously. One must give way to the other. This setup lets the designer prioritize one over the other.

If RIB is configured on, a burst can be interrupted by a host communication, and is automatically restarted.

If RIB is configured off, bursts cannot be interrupted but, rather, the host communication is delayed until the burst has completed. The DRDY low period is stretched by the QT1481 during the burst.

This function is programmed on a global basis. See Table 5-8 on page 57.

RIB Default value: 0 (Off)

RIB Possible range: 0, 1 (off, On)

5.16 Sleep Drift Compensation – SDC

See also Section 5.4 on page 40 Sleep, NDRIFT and PDRIFT.

SDC allows the QT1481 to be configured for automatic sleep, and for modified drift compensation when sleep is enabled. Whenever the QT1481 goes to sleep, the whole device is shutdown, including the clock generator. All operations are stopped including matrix scanning and timers, which results in the internal time keeping running very slow and, in particular, drift compensation runs at a rate much slower than configured by NDRIFT and PDRIFT.

For example, with NDRIFT and PDRIFT configured such that drift compensation occurs once every second when the QT1481 is awake, and with the QT1481 being awake for 5 ms to scan the matrix before falling asleep for 495 ms, all internal timers are slowed by a factor of 100 (5/500), and drift compensation would occur at the much slower rate of just once every 100 s.

This would typically result in the key references not tracking signal variations adequately, and could result in false detections. To help resolve this, SDC can be configured so that the QT1481 performs drift compensation after a specific number of sleeps.

- With SDC = 0, sleep is disabled
- With SDC = 1, drift compensation occurs after every sleep
- With SDC = 7, drift compensation is applied after every sixty-four sleeps

This function is programmed on a global basis. See Table 5-8 on page 57.

SDC Default value: 0 (off, Sleep disabled)

SDC Possible range: 0 – 7 (off, 1 – 64)

5.17 Serial Rate – SR

The possible baud rates are shown in Table 5-7 on page 56. The rate chosen by this parameter only affects UART mode. SPI mode is slave-only and can clock at any rate from DC up to 4 MHz. The baud rate can be adjusted to one of five values from 9600 to 115.2 kbaud.

SR Default value: 0 (9600 baud)

5.18 Lower Signal Limit – LSL

This Setup determines the lowest acceptable value of signal level for all keys. If any key reference level falls below this value, the QT1481 declares an error condition in the key status bits (See Section 4.7 on page 28 and Section 4.9 on page 30).

Testing is required to ensure that there are adequate margins in this determination. Key size, shape, panel material, burst length, and dwell time all factor into the detected signal levels.

This parameter occupies 2 bytes (11 bits) of the setups.

LSL Default value: 100 (recommended value)
LSL Possible range: 0 – 2047

5.19 Key Gain Test Threshold – KGTT

The Key Gain test takes a special sample from each enabled key using half the usual burst length, and compares the resulting signal against each key normal signal. The test passes if the signal has decreased by the Key Gain Test Threshold (KGTT). The following equation must hold for the test to pass:

(Normal Signal – Test Signal) = KGTT

Disabled keys are not tested.

The Key Gain Test Threshold can be configured to a value between 4 and 64, via LUT (see Table 5-8 on page 57). KGTT occupies 4 bits only, sharing the same word as the Lower Signal Limit.

This function is programmed on a global basis.

KGTT Default value: 7 (32)

KGTT Possible range: 0 – 15 (4 – 64)

5.20 $\overline{\text{STATUS}}$ Output – STS

The $\overline{\text{STATUS}}$ pin is designed to be used as a status and error signaling mechanism for the host controller.

One use for this pin is to alert the host that there is key activity, in order to limit the amount of communication between the QT1481 and the host. The $\overline{\text{STATUS}}$ pin should ideally be connected to an interrupt pin on the host that can detect an edge, following which the host can proceed to poll the QT1481 for further information.

The $\overline{\text{STATUS}}$ pin is an open-drain output with an internal 20 kΩ – 50 kΩ pull-up resistor. This allows multiple devices to be connected together in a single wire-OR logic connection with the host. When the $\overline{\text{STATUS}}$ pin becomes active, the host can poll all devices to identify which one is reporting.

Table 5-5 on page 55 shows the possible internal conditions that can cause the $\overline{\text{STATUS}}$ pin to become active. Except for STS_DEBUG, the various items in the table are logical-OR'd together.

STS_TOUCH: When this option (STS, bit 5) is enabled, the $\overline{\text{STATUS}}$ output can be used to alert the host of touch changes. The $\overline{\text{STATUS}}$ output becomes active after reset and when there is a change in key state (either touch or touch release). It does not self-clear but becomes inactive again only after the host issues command `0x06`. After the host has issued command `0x06`, the $\overline{\text{STATUS}}$ output becomes inactive at the end of the matrix scan provided that there are no keys in detect and there are no other conditions demanding it be active. To avoid missing touches and future $\overline{\text{STATUS}}$ assert edges, the host should issue further `0x06` commands until $\overline{\text{STATUS}}$ becomes inactive.

STS_DEBUG: When this option is enabled (STS, bit 7, See Section B. on page 64), the QT1481 streams one frame of data out of the Debug port after each matrix scan. When STS_DEBUG is enabled it impacts the key response time because the next matrix scan is delayed until the debug frame has been fully transmitted. To prevent interference, this bit should be used exclusively, and not in conjunction with any other bit.

STS_RSTHOST: The $\overline{\text{STATUS}}$ pin can even be used as a watchdog for the host, to reset it should the host fail to send regular transmissions to the QT1481 (bit 0 of STS byte). The comms timeout required to generate the reset signal is about 2 seconds of inactivity. If this feature is enabled, it does not become effective until the first command is received from the host; therefore, it is assumed that there is at least some initial host activity for this feature to work.

Note: 1. The $\overline{\text{STATUS}}$ output is preserved during sleep.

2. To prevent interference, STS_DEBUG should not be enabled with any other item.

3. The reset pulse should be allowed to complete before the host sends commands to the QT1481. If commands are received from the host while $\overline{\text{STATUS}}$ is low, $\overline{\text{STATUS}}$ will remain low until the commands stop and the 2 s internal host reset timer is allowed to fully cycle.

STS Default value: `0x20` (STS_TOUCH)

5.21 Awake Time – AWAKE

The AWAKE feature is effective only if the part has been configured for automatic sleep, via SDC, and if the sleep command (`0x16`) has been issued. AWAKE determines the period of time that elapses from the last key release before the part tries to sleep.

An internal timer is restarted at each key release and runs for the time configured via AWAKE. The part will not enter sleep while this timer is active, or while any of the following conditions are present:

- $\overline{\text{DRDY}}$ asserted (low level)
- $\overline{\text{SS}}$ low (assume host trying to send a command)
- A command is being processed or response data is being returned or pending return to the host
- Any key calibrating
- Any key touch delta exceeds the threshold (positive or negative)

Note: If the sleep feature has been disabled, the QT1481 never sleeps and the AWAKE setup has no effect. The AWAKE period can be configured to one of 16 values between 100 ms and 25.4 s via LUT (see Table 5-8 on page 57).

This function is programmed on a global basis.

AWAKE default value: 15 (25.4 s)

AWAKE range: 0 – 15 (100 ms – 25.4 s)

AWAKE Timeout accuracy: to within ±50 ms

5.22 Drift Hold Time – DHT

Drift Hold Time (DHT) is used to suspend drift compensation from all keys while the keypad is being used. With the feature enabled, drift compensation is suspended while any key is touched and also for a period afterwards. DHT defines the length of time the drift compensation continues to be suspended after a key detection has finished.

This feature is particularly useful in cases of high-density keypads where touching a key or hovering a finger over the keypad would cause untouched keys to drift, and therefore create a sensitivity shift, and ultimately inhibit other touch detections.

DHT can be configured to one of 16 values between 100 ms and 25.4 s via LUT (see Table 5-8 on page 57).

This function is programmed on a global basis.

DHT default value: 11 (9 s)

DHT range: 0 – 15 (100 ms – 25.4 s)

5.23 Frequency Hopping Mode – FHM

Frequency hopping can be disabled altogether or enabled to one of three different active modes with FHM.

FHM = 0 – Frequency hopping disabled

If frequency hopping is disabled, the QT1481 always uses the same frequency, defined by FREQ0.

FHM = 1 or 2

If frequency hopping is enabled with FHM = 1 or FHM = 2, the QT1481 continually monitors the noise across all keys and switches frequencies dynamically to try and find the frequency with the lower noise amplitude. Up to three different frequencies can be selected using FREQ0, FREQ1 and FREQ2. If frequency hopping between only two frequencies is desired, the same frequency should be selected for two of these functions.

The signal levels can vary slightly with frequency, and so the reference for each key might need a corresponding adjustment during frequency hopping to compensate for this and to prevent false detections and unresponsive keys. The mechanism used to adapt the references is different depending on the frequency hop mode set.

FHM = 1 – Calibrate after hop

If frequency hopping is enabled with FHM = 1, the QT1481 compensates for the variations in signal level by recalibrating each key immediately after each frequency hop. A negative aspect to this mode is the danger that a key is being touched at the time of the frequency hop. If this were to happen, the touched key would be recalibrated to the touching finger and the detect would be cancelled. However, on subsequent removal of the touch (for a time greater than or equal to the PRD function) the key would be recalibrated again and the detect flagged when the touch is re-established.

FHM = 2 – Adjust each key's reference during hop

If frequency hopping is enabled with FHM = 2, the QT1481 adjusts each key's reference at each frequency hop. The amount of the adjustment must first be configured using the setups CFO_1 and CFO_2.

The QT1481 will not switch the frequency during calibration if FHM is set to 1 or 2.

FHM = 3 – Frequency sweep

This mode offers high noise immunity and is recommended as the mode of choice.

If frequency hopping is enabled with FHM = 3, the QT1481 repeatedly sweeps the sampling frequency through a range bounded by two frequencies defined by FREQ0 and FREQ1. The frequency is changed after every matrix scan.

In this hop mode, the reference for each key is not adjusted as the frequency is changed, so the signal levels must not vary significantly over the selected frequency range otherwise false detects or unintentional touch dropouts could occur. Variations in signal levels can be limited by restricting the frequency sweep range. If the signal variation for any particular key is found to be too great, the range of frequencies should be narrowed by decreasing the difference in values set at FREQ0 and FREQ1.

This function is programmed on a global basis. See Table 5-7 on page 56.

FHM Default value: 1 = Calibrate all keys after hop

FHM Recommended value: 3 = Frequency sweep

FHM Possible range: 0 – 3 (0 = off
1 = Calibrate all keys after hop
2 = Adjust each key's reference during hop
3 = Frequency sweep)

5.24 Frequency 0 – FREQ0

FREQ0 is used in all frequency hopping modes and even if frequency hopping is disabled. In all modes, it defines the idle time between pulses in the burst. Larger values yield longer times between pulses and thus a lower fundamental frequency.

FHM = 0 – Frequency hopping disabled

If frequency hopping is disabled, the QT1481 always uses the same frequency, defined by FREQ0. Frequency hopping might not be desirable in all applications and it might be more appropriate to preselect a burst freq uency at the factory which is known not to coincide with other operating frequencies within the end product or other frequencies in the operating environment. In such cases, FHM can be set at zero, and the burst frequency set with FREQ0.

Together with DWELL, FREQ0 allows the fundamental frequency to be set in the range 31 kHz – 943 kHz. The frequency for a specific DWELL and FREQ0 combination should be measured using an oscilloscope (temporarily disable frequency hopping to make the measurement easier).

FHM = 1 or FHM = 2 – Frequency hopping between three frequencies

If frequency hopping is enabled with FHM = 1 or FHM = 2, the QT1481 can hop between three frequencies configured using FREQ0, FREQ1 and FREQ2.

FHM = 3 – Frequency sweep

With FHM = 3, the QT1481 sweeps a range of frequencies, with the upper frequency boundary (shortest idle time) defined by FREQ0. FREQ1 must be set to a value greater than FREQ0; the behavior is otherwise undefined.

This function is programmed on a global basis. See Table 5-7 on page 56.

FREQ0 Default value: 24 (delay cycles)

FREQ0 Possible range: 0 – 63 (Highest frequency to Lowest frequency)

5.25 Frequency1 – FREQ1

FREQ1 is used only if FHM is set to a non-zero value. It is not used if frequency hopping is disabled with FHM = 0.

With FHM = 1 or FHM = 2, FREQ1 allows configuration of one of three operating frequencies by defining the idle time between pulses in the burst. Larger values yield longer times between pulses and thus a lower fundamental frequency.

With FHM = 3, the QT1481 sweeps a range of frequencies, with the lower frequency boundary (longest idle time) defined by FREQ1. FREQ1 must be set to a value greater than FREQ0; the behavior is otherwise undefined.

This function is programmed on a global basis. See Table 5-7 on page 56.

FREQ1 Default value: 30 (delay cycles)

FREQ1 Possible range: 0 – 63 (Highest frequency to Lowest frequency)

5.26 Frequency2 – FREQ2

FREQ2 is used only if FHM = 1 or FHM = 2; it is not used if FHM is set to zero or 3.

With FHM = 1 or FHM = 2, FREQ2 allows configuration of one of three operating frequencies by defining the idle time between pulses in the burst. Larger values yield longer times between pulses and thus a lower fundamental frequency.

This function is programmed on a global basis. See Table 5-7 on page 56.

FREQ2 Default value: 36 (delay cycles)

FREQ2 Possible range: 0 – 63 (Highest frequency to Lowest frequency)

5.27 Noise Threshold – NSTHR

NSTHR is used only if FHM = 1 or FHM = 2; it is not used if FHM is set to 0 or 3.

When FHM = 1 or FHM = 2, the QT1481 considers a hop to one of the other frequencies when the noise at the current frequency consistently exceeds the threshold configured with NSTHR. NSTHR is used by the frequency hopping algorithms to determine if a signal delta should be considered as noise. A delta, of either polarity, greater than or equal to NSTHR, is considered as possible noise and forces the Noise Integrator counter to be incremented.

This function is programmed on a global basis. See Table 5-7 on page 56.

NSTHR Default value: 0 (5 counts)

NSTHR Possible range: 0 – 15 (5 – 50 counts)

5.28 Noise Integrator Limit – NIL

NIL is used only if FHM = 1 or FHM = 2; it is not used if FHM is set to zero or 3.

The QT1481 considers a hop to one of the other frequencies when the noise at the current frequency consistently exceeds the threshold configured with NSTHR. To prevent true touch events and other brief signal anomalies being considered as noise, the QT1481 uses counters, called noise integrators, to track the number of signal deltas that exceed the noise threshold. It maintains two counters, one for positive deltas and one for negative deltas.

The QT1481 considers a frequency hop only if both counters reach the noise integrator limit (NIL). The counters are reset to zero at the end of each matrix scan. This mechanism provides a robust way of detecting strong noise while suppressing unnecessary frequency hopping.

This function is programmed on a global basis. See Table 5-7 on page 56.

NIL Default value: 3

NIL Possible range: 0 – 15 (0 – 2 = factory use only)

5.29 Calibrated Frequency Offset – CFO_1 and CFO_2

The Calibrated Frequency Offset (CFO) values are used when the frequency hop mode is set to 2. They are used to adjust each key's reference whenever a frequency hop occurs, taking into account any differences in calibrated signal at the different frequencies. The device uses the highest selected frequency as a point of reference and calculates offsets in calibrated signals at the other two frequencies relative to the highest one. The following explanation assumes that FREQ0 defines the highest frequency, for convenience of this discussion, but that does not need to be the case.

Note: Configure FREQ0, FREQ1 and FREQ2 before trying to configure CFO_1 and CFO_2.

Command `0x02` can be invoked to automatically fill out the CFO_1 and CFO_2 values. Alternatively, CFO_1 and CFO_2 can be manually configured, or fine tuned.

Once FREQ0, FREQ1 and FREQ2 have been configured for the three chosen frequencies, CFO_1 and CFO_2 should be loaded with the signal offsets appropriate for the signal shifts observed when switching between the three different frequencies.

Follow these steps to determine the different signal levels for each key at each frequency and to determine appropriate values to load into CFO_1 and CFO_2.

1. Configure FHM = 0 to disable frequency hopping temporarily.
2. Configure FREQ0 to select the highest of the chosen frequencies. That is, FREQ0 should be set to a lower value than FREQ1 and FREQ2.
3. Recalibrate all keys.
4. Make a note of each key reference level, Ref(k, f0)
5. Configure FREQ0 to select the second of the chosen frequencies.
6. Recalibrate all keys.
7. Make a note of each key reference level, Ref(k, f1).

8. Configure FREQ0 to select the last of the chosen frequencies.
9. Recalibrate all keys.
10. Make a note of each key reference level, Ref(k, f2).
11. Determine the difference in signal for each key in turn when the frequency changes from FREQ0 to FREQ1, using the following equation:

    Diff(k, f0, f1) = Ref(k, f0) – Ref(k, f1)

    (this value will nearly always be positive. If it is negative, set Diff(k, f0, f1) to zero)
12. Store Diff(k, f0, f1) into the corresponding CFO_1 for the k.
13. Determine the difference in signal at each key when the frequency changes from FREQ0 to FREQ2, using the following equation:

    Diff(k, f0, f2) = Ref(k, f0) – Ref(k, f2)

    (this value will nearly always be positive. If it is negative, set Diff(k, f0, f1) to zero)
14. Store Diff(k, f0, f2) into the corresponding CFO_2 for key k.
15. Configure FHM = 2.

CFO_1/2 Default value: 0

CFO_1/2 Possible range: 0 – 255

5.30 Setups CRC – SCRC

The setups block terminates with a 16-bit CRC, SCRC, of the entire block. The formulae for calculating this CRC is shown in Appendix A.. The low order byte should be sent first.

5.31 Setups Block

Setups data is sent from the host to the QT1481 in a block of hex data. The block can only be loaded in Setups mode following two 0x01 commands (Section 4.3 on page 26). Refer to Table 5-7 on page 56 for further details.

Table 5-4. Setups Block

Byte	Parameter	Symbol	Bytes	Valid Range	Bits	Key Scope	Default Value	Description	Page
0	Neg threshold Pos Threshold	NTHR PTHR	48	NTHR = 0 – 15 PTHR = 0 – 15	4 4	1 1	6 2	Lower nibble = Neg Threshold – take operand and add 4 to get value Upper nibble = Pos Threshold – take operand and add 4 to get value	38 39
48	Neg Drift Comp Pos Drift Comp	NDRIFT PDRIFT	48	NDRIFT = 0 – 15 PDRIFT = 0 – 15	4 4	1 1	10 10	Lower nibble = Neg Drift comp – via LUT Upper nibble = Pos Drift comp – via LUT	40
96	Normal DI Limit Fast DI Limit	NDIL FDIL	48	NDIL = 0 – 15 FDIL = 0 – 15	4 4	1 1	2 5	Lower nibble = Normal DI Limit, values same as operand (0 = disabled burst) Upper nibble = Fast DI Limit, values same as operand (0 is invalid; do not use)	41
144	Neg recal delay	NRD	48	0 – 254	8	1	20	Range is in 0.5 sec increments; 0 = infinite; default = 10s (operand = 20) Range is {infinite, 0.5 – 127s}; 255 is illegal to use	42
192	Pos recal delay Burst Length AKS	PRD BL AKS	48	PRD = 0 – 15 BL = 0 – 3 AKS = 0, 1	4 2 1	1 1 1	6 2 0	Lower nibble = PRD, via LUT, default = 6 (1 second) Bits 5, 4: = BL, via LUT, default = 48 (setting = 2) Bit 6 = AKS, 1 – enabled	42 43 43
240	Cal.Freq.Offset 1	CFO_1	48	0 – 255	8	1	0	Value is the signal offset	51
288	Cal.Freq.Offset 2	CFO_2	48	0 – 255	8	1	0	Value is the signal offset	51
336	Neg Hysteresis Mains Sync Sleep Drift Comp Threshold Mult	NHYST MSYNC SDC THRM	1	NHYST = 0 – 3 MSYNC = 0, 1 SDC = 0 – 7 THRM = 0 – 3	2 1 3 2	48 48 48 48	1 0 0 0	Bits 1,0 = Neg hysteresis, all keys; default = 12.5% Bits 5 = Mains sync, negative edge, 1 = enabled; default = 0 (off) Bits 4 – 2 = Sleep Drift Compensation, 0 = sleep disabled (default) Bits 7, 6 = Threshold Multiplier; default = 0 (x1)	44 45 46 39
337	Dwell Time Serial rate	DWELL SR	1	DWELL = 0 – 15 SR = 0 – 4	4 4	48 device	1 0	Lower nibble = Dwell time, 15 values via LUT, default = 1 (188 ns) Upper nibble = serial rate via LUT – 9600, 19.200, 38.400, 57.600, 115.200 (UART)	45 46
338	Lower Signal Limit Restart Int. Burst FMEA KGTT	LSL RIB KGTT	2	LSL = 0 – 2047 RIB = 0, 1 KGTT = 0 – 15	11 1 4	48 48 48	100 0 7	Bits 10 – 0 = Lower limit of acceptable signal; below this value, QT1481 declares an error. The low order byte should be sent first. Bit 11 = Restart Interrupted Burst, 1 = enabled Bits 15 – 12 = KGTT Default = 7 (32 counts)	47 46 47
340	$\overline{\text{STATUS}}$ Output	STS	1	0 – 255	8	device	0x20	Defines the $\overline{\text{STATUS}}$ pin behavior; see Table 5-5 on page 55	47

Table 5-4. Setups Block

Byte	Parameter	Symbol	Bytes	Valid Range	Bits	Key Scope	Default Value	Description	Page
341	Awake Time Drift Hold Time	AWAKE DHT	1	0 – 15 0 – 15	4 4	48 48	15 11	Lower nibble = Awake Time, default = 15 (25.4s) Upper nibble = Drift Hold Time, default = 11 (9s)	48 48
342	Scope Sync	SSYNC	1	0 – 255	6	6 (X line)	0	Each bit enables the scope sync output for the burst on one X line. Bit n: 1 = Scope sync enabled for burst on Xn.	44
343	Frequency 0 Freq.Hop Mode	FREQ0 FHM	1	0 – 63 0 – 3	6 2	48	24 1	Bits 5 – 0 = Frequency 0. Default = 24 delay cycles Bits 7, 6 = Frequency hop mode. Default = 1 (recalibrate after each hop)	50 49
344	Detect Integrator Multiplier Frequency 1	DIM FREQ1	1	0 – 3 0 – 63	2 6	48 48	0 30	Bits 7 – 6 = DIM. Default = 0 (x1) Bits 5 – 0 = Frequency 1. Default = 30 delay cycles	42 50
345	Frequency 2	FREQ2	1	0 – 63	6	48	36	Bits 5 – 0 = Frequency 2. Default = 36 delay cycles	50
346	Noise Threshold Noise Integrator Limit	NSTHR NIL	1	0 – 15 0 – 15	4 4	48 48	0 3	Lower nibble = Noise Threshold. Default = 0 (5 counts) Upper nibble = Noise Integrator Limit. Default = 3	51 51
347	Reserved						0		
348	Setups CRC	SCRC	2	0 – 65k	16	device	–	CRC-16 of above setups, does NOT include CRC of command itself. The low order byte should be sent first. See also CRC notes, Appendix A..	52
	Block length		**350**						

Note: A CRC calculator for Microsoft Windows is available free of charge from Atmel, on request.

5.32 STS Bits

The $\overline{\text{STATUS}}$ pin can be used to indicate a variety of things in combination. The STS parameter controls which states make the $\overline{\text{STATUS}}$ pin active.

Table 5-5. Control Byte Bits

Bit	Name	$\overline{\text{STATUS}}$ output (active low)	Default
7	STS_DEBUG	1= $\overline{\text{STATUS}}$ output used for Debug. This bit cannot be used with any other bit.	0
6	STS_KEYERROR	Active on any key error: (calibration failed, low signal)	0
5	STS_TOUCH	Active on any change in touch status	1
4	Reserved		–
3	STS_SETUPS	Active on Setups CRC mismatch	0
2	STS_MSYNC	Active on Mains sync error	0
1	Reserved		–
0	STS_RSTHOST	Host watchdog – active if no host communications within any 2 s period. Host reset pulse length is 150 ms. The host watchdog is not enabled until the first valid command is received from the host.	0

5.33 Key Mapping

Some commands return bit-fields related to keys. For example, command `0x07` (report all keys) returns 6 bytes containing flag bits, one per key, to indicate which keys are reporting touches. Table 5-6 shows the byte and bit order of the keys, and the key number reported in each bit.

The key number is related to the X and Y scan lines which address each particular key. Each byte in the return stream represents one set of keys along a Y line, that is, up to 8 keys. For example, key 0 is at location X0, Y0 and key 29 is at location X5, Y3

Note: Byte 0 is returned first.

Table 5-6. Key Mapping

Byte (Y Line)	Bit (X Line)							
	7	6	5	4	3	2	1	0
0	7	6	5	4	3	2	1	0
1	15	14	13	12	11	10	9	8
2	23	22	21	20	19	18	17	16
3	31	30	29	28	27	26	25	24
4	39	38	37	36	35	34	33	32
5	47	46	45	44	43	42	41	40

5.34 Setups Block Summary

Table 5-7. Setups Block Summary – Per Key Settings

	Parameter											
Index	**NTHR Counts**	**PTHR Counts**	**NDRIFT Secs**	**PDRIFT Secs**	**NDIL Counts**	**FDIL Counts**	**NRD Secs**	**PRD Secs**	**BL Pulses**	**AKS**	**CFO_1**	**CFO_2**
	Per Key											
0	4	4	0.1	0.1	Key off	Unused	0 (infinite)	0 (infinite)	16	**Off**	**0**	**0**
1	5	5	0.2	0.2	1	1	0.5 – 127 s	0.1	32	On	1 – 255	1 – 255
2	6	**6**	0.3	0.3	**2**	2	**10 s**	0.2	**48**			
3	7	7	0.4	0.4	3	3		0.3	64			
4	8	8	0.6	0.6	4	4		0.5				
5	9	9	0.8	0.8	5	**5**		0.7				
6	**10**	10	1	1	6	6		**1**				
7	11	11	1.2	1.2	7	7		1.5				
8	12	12	1.5	1.5	8	8		2				
9	13	13	2	2	9	9		3.2				
10	14	14	**2.5**	**2.5**	10	10		4.5				
11	15	15	3.3	3.3	11	11		6				
12	16	16	4.5	4.5	12	12		9				
13	17	17	6	6	13	13		12.3				
14	18	18	7.5	7.5	14	14		17.5				
15	19	19	10	10	15	15		25				

Typical values: For most touch applications, use the values shown in the shaded cells. Bold text items indicate default settings. The number to send to the QT1481 is the index number in the leftmost column (0 – 15), not numbers from the table. The QT1481 uses a lookup table internally to translate the indices 0 – 15 to the parameters for each function.

NRD is an exception: It can range from 0 – 254 which is translated from 1 = 0.5 s to 254 = 127 s with zero = infinity.

CFO_1 and **CFO_2** are exceptions. Their values, ranging from 0 to 255, are used directly and without any translation.

Table 5-8. Setups Block Summary – Global Settings

	Parameter																	
Index	RIB	NHYST	SDC	MSYNC	DWELL µs	SR baud	KGTT	AWAKE secs	DHT	SSYNC	THRM	FHM	FREQ0	FREQ1	FREQ2	NSTHR	NIL	LSL	DIM
	Global									X line	Global								
0	**Off**	6.25%	**Off**	**Off**	0.13	**9,600**	4	0.1	0.1	**Off**	**× 1**	Off	**24 cycles**	**30 cycles**	**36 cycles**	**5**	0 (factory only)	**100**	**× 1**
1	On	**12.5%**	1	On	**0.19**	19,200	8	0.2	0.2	On	× 2	**1**				8	1 (factory only)	0 – 2047	× 2
2		25%	2		0.4	38,400	12	0.3	0.3		× 4	2				11	2 (factory only)		× 4
3		50%	4		0.6	57,600	16	0.5	0.5		× 8	3				14	**3**		× 8
4			8		0.8	115,200	20	0.7	0.7							17	4		
5			16		0.9		24	1	1							20	5		
6			32		1.1		28	1.5	1.5							23	6		
7			64		1.3		**32**	2	2							26	7		
8					1.5		36	3.2	3.2							29	8		
9					1.7		40	4.5	4.5							32	9		
10					2.1		44	6	6							35	10		
11					2.6		48	9	**9**							38	11		
12					3.8		52	12.3	12.3							41	12		
13					5.1		56	15	15							44	13		
14					7.1		60	19	19							47	14		
15					9.9		64	**25.4**	25.4							50	15		

6. Specifications

6.1 Absolute Maximum Specifications

Parameter	Specification
Vdd	6.0 V
Max continuous pin current, any control or drive pin	±10 mA
Short circuit duration to ground, any pin	infinite
Short circuit duration to Vdd, any pin	infinite
Voltage forced onto any pin	0.5 V to (Vdd + 0.5 V)
Frequency of operation	17 MHz
EEPROM setups maximum writes	100,000 write cycles
CAUTION: Stresses beyond those listed 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.	

6.2 Recommended Operating Conditions

Parameter	Specification
Operating temp	–40°C to +105°C
Storage temp	–55°C to +125°C
Vdd	+4.75 V to 5.25 V
Supply ripple + noise	20 mV p-p max
Cx transverse load capacitance per key	0 to 20 pF
Fosc oscillator frequency	16 MHz ±2%

6.3 DC Specifications

Vdd = 5.0 V, Cs = 4.7 nF, Freq = 16 MHz, Ta (Ambient Temperature) = recommended range, unless otherwise noted

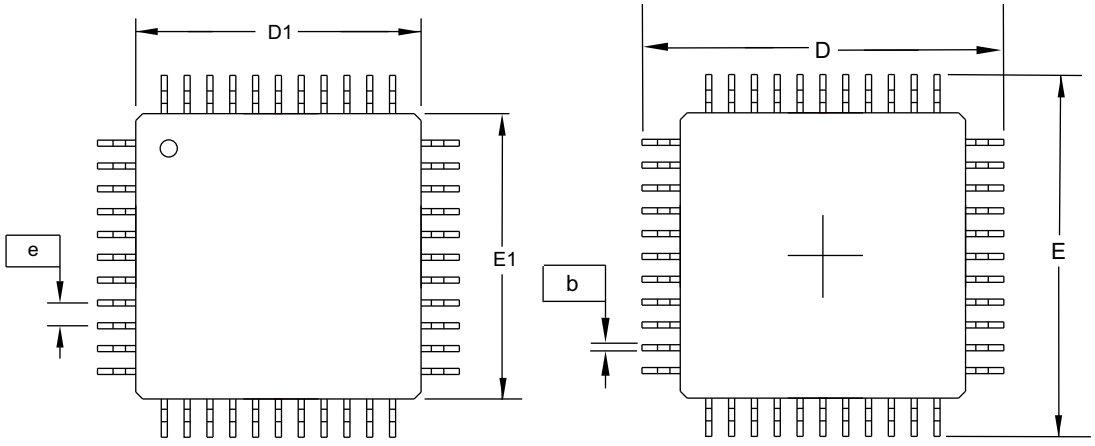
Parameter	Description	Min	Typ	Max	Units	Notes
Iddr	Supply current, running	–	13	–	mA	
Idds	Supply current, sleeping	–	15	–	µA	
Vr	Vdd internal reset voltage	–	4	–	V	
Vil	Low input logic level	–	–	0.2 × Vdd	V	
Vih	High input logic level	0.6 × Vdd	–	–	V	
Vol	Low output voltage	–	–	0.6	V	4mA sink
Voh	High output voltage	4.2	–	–	V	1 mA source
Iil	Input leakage current	–	–	±1	µA	
Ar	Acquisition resolution	–	9	11	bits	
Rp	Internal pull-up resistors	20	–	50	kΩ	$\overline{\text{DRDY}}$, $\overline{\text{SS}}$, TX, $\overline{\text{STATUS}}$ pins
Rrst	Internal $\overline{\text{RST}}$ pull-up resistor	30	–	60	kΩ	

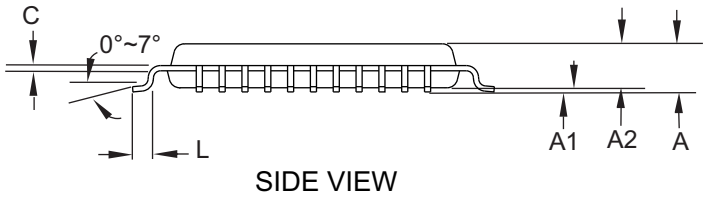
6.4 Timing Specifications

Vdd = 5.0 V, Cs = 4.7 nF, Freq = 16 MHz, Ta (Ambient Temperature) = recommended range, unless otherwise noted

Parameter	Description	Min	Typ	Max	Units	Notes
S1	↓$\overline{\text{SS}}$ to first ↓CLK edge	125	–	–	ns	SPI parameter controlled by host
S2	↓CLK to valid MISO	–	–	20	ns	SPI parameter controlled by QT1481
S3	Last ↑CLK to ↑$\overline{\text{SS}}$	25	–	–	ns	SPI parameter controlled by host
S4	↑$\overline{\text{SS}}$ to 3-state MISO	–	–	20	ns	SPI parameter controlled by QT1481
S5	↑$\overline{\text{SS}}$ to falling $\overline{\text{DRDY}}$	–	–	20	µs	SPI parameter controlled by QT1481
S6	$\overline{\text{DRDY}}$ low pulse width	1	–	–	µs	SPI parameter controlled by QT1481
S7	CLK low pulse width	125	–	–	ns	SPI parameter controlled by host
S8	CLK high pulse width	125	–	–	ns	SPI parameter controlled by host
S9	CLK period	250	–	–	ns	SPI parameter controlled by host
Fck	SPI Clock rate	–	–	4	MHz	SPI parameter controlled by host

6.5 Mechanical Dimensions





COMMON DIMENSIONS
(Unit of Measure = mm)

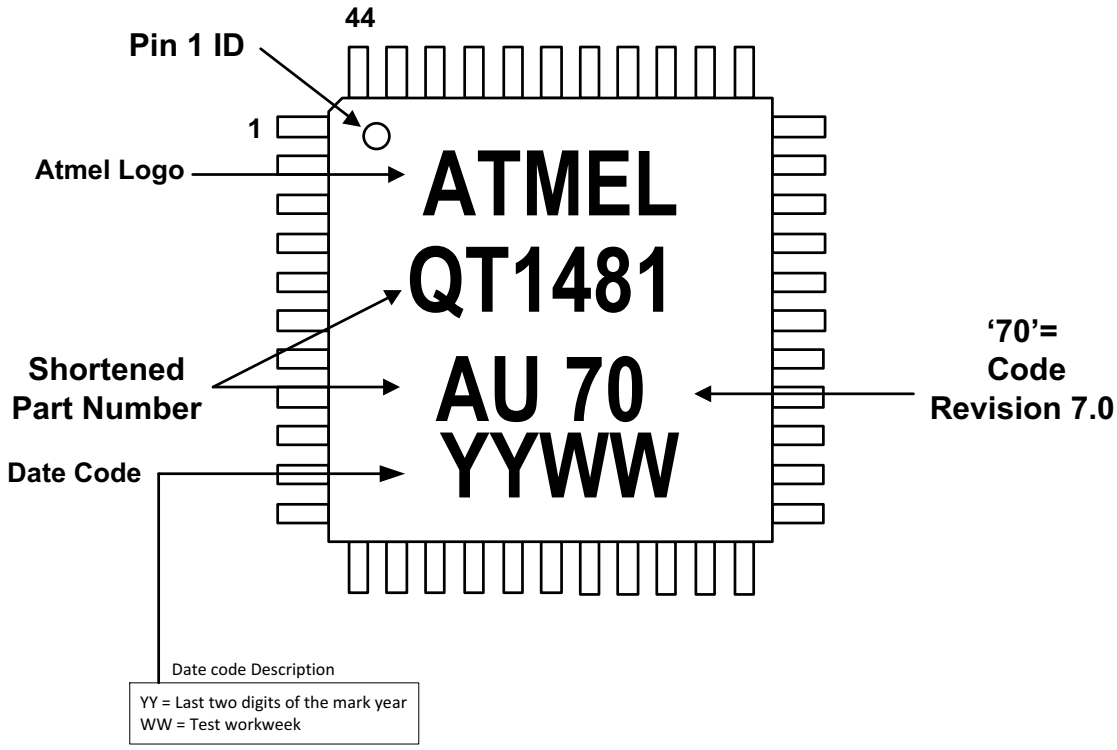
SYMBOL	MIN	NOM	MAX	NOTE
A	–	–	1.20	
A1	0.05	–	0.15	
A2	0.95	1.00	1.05	
D	11.75	12.00	12.25	
D1	9.90	10.00	10.10	Note 2
E	11.75	12.00	12.25	
E1	9.90	10.00	10.10	Note 2
B	0.30	–	0.45	
C	0.09	–	0.20	
L	0.45	–	0.75	
e	0.80 TYP			

Notes:
1. This package conforms to JEDEC reference MS-026, Variation ACB.
2. Dimensions D1 and E1 do not include mold protrusion. Allowable protrusion is 0.25 mm per side. Dimensions D1 and E1 are maximum plastic body size dimensions including mold mismatch.
3. Lead coplanarity is 0.10 mm maximum.

Atmel Package Drawing Contact: packagedrawings@atmel.com	TITLE **44A**, 44-lead 10.0 x 10.0x1.0 mm Body, 0.80 mm Lead Pitch, Thin Profile Plastic Quad Flat Package (TQFP)	GPC AIX	DRAWING NO. 44A	REV. D

6.6 Marking

The following part marking is used.



6.7 Part Number

Part Number	QS Number	Description
AT42QT1481-AU	QS738	44-pin 10 × 10 mm TQFP RoHS compliant IC
AT42QT1481-AUR	QS738	44-pin 10 × 10 mm TQFP RoHS compliant IC - Tape and reel

The part number comprises:

AT = Atmel

42 = Touch Business Unit

QT = Charge-transfer technology

1481 = (1) Keys, (48) number of channels, (1) variant number

AU = TQFP chip

R= Tape and reel

6.8 Moisture Sensitivity Level (MSL)

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

Appendix A. 16-bit CRC Algorithm

```
// 16 bits crc calculation. Initial crc entry value must be 0.
// The message is not augmented with 'zero' bits.
// polynomial = X16 + X12 + X5 + 1
// data is an 8 bit number, unsigned
// crc is a 16 bit number, unsigned
// repeat this function for each data block byte, folding the result
//   back into the call parameter crc

unsigned long sixteen_bit_crc(unsigned long crc, unsigned char data)
{     unsigned char index;// shift counter

      crc ^= (unsigned long)(data) << 8;
      index = 8;
      do                                          // loop 8 times
      {     if(crc & 0x8000)
            {     crc= (crc << 1) ^ 0x1021;
            }
            else
            {     crc= crc << 1;
            }
      } while(--index);
      return crc;
}
```

A CRC calculator for Microsoft Windows is available free of charge from Atmel.

Appendix B. DEBUG Output

The QT1481 includes a debug interface which may be used for observing many internal operating variables, in real time, even while the part is actively communicating with a host over either the SPI or UART serial interfaces. The Debug interface provides a useful aid during product development and uses two pins, one for clock and one for data, to stream data out of the part.

If STS_DEBUG is enabled in the STS Setups byte the QT1481 streams a 465-byte frame of data out of the two debug pins after each keyscan cycle. The transmission format is compatible with Atmel Plug-in USB card (Part Number 9206) and the data can be viewed using Atmel Hawkeye PC software (contact Atmel for information). Table B-1 shows the Debug interface details.

Table B-1. Debug Interface

Debug Clock output	Pin 43 (Dbg_Clk)
Debug Data output	Pin 40 (Dbg_Data)
Data valid	Clock high
Data changing	Clock low
Clock frequency	Approximately 500 kHz
Blank time between byte transmissions	5.5 µs
Frame transmission time	8.8 ms
Byte transmission order	Most significant bit (MSB) first

The meaning of each byte in the 465-byte frame is described in Table B-2.

Table B-2. Debug Output Data Frame

Frame Byte #	Description
0	Detect status for keys 0 (bit0) to 7 (bit7), one bit per key
1	Detect status for keys 8 (bit0) to 15 (bit7), one bit per key
2	Detect status for keys 16 (bit0) to 23 (bit7), one bit per key
3	Detect status for keys 24 (bit0) to 31 (bit7), one bit per key
4	Detect status for keys 32 (bit0) to 39 (bit7), one bit per key
5	Detect status for keys 40 (bit0) to 47 (bit7), one bit per key
6	Device status (identical to first byte returned from command 0x06)
7	Count of keys declaring detect
8	A 6-bit unsigned value encoding the first or only key to be touched, in the range 0 – 47
9	Time remaining before host reset pulse is issued at $\overline{\text{STATUS}}$ pin. Each count is 10 ms
10	Time remaining until normal drift compensation is resumed. Each count is 100 ms
11	Time remaining before QT1481 tries to sleep (if enabled). Each count is 100 ms
12	Time remaining before the current SPI or UART command times out. Each count is 10 ms
13	Time remaining before the wait for the next MSYNC signal times out. Each count is 10 ms

Table B-2. Debug Output Data Frame (Continued)

Frame Byte #	Description
14	100 ms counter (IEC/EN60730)
15	Signal Fail counter (IEC/EN60730)
16	Matrix Scan counter (IEC/EN60730)
17	Index of current frequency
18 – 26	Data for key 0. See Table B-3 on page 67 for details of the data set for one key.
27 – 35	Data for key 8
36 – 44	Data for key 16
45 – 53	Data for key 24
54 – 62	Data for key 32
63 – 71	Data for key 40
72 – 80	Data for key 1
81 – 89	Data for key 9
90 – 98	Data for key 17
99 – 107	Data for key 25
108 – 116	Data for key 33
117 – 125	Data for key 41
126 – 134	Data for key 2
135 – 143	Data for key 10
144 – 152	Data for key 18
153 – 161	Data for key 26
162 – 170	Data for key 34
171 – 179	Data for key 42
180 – 188	Data for key 3
189 – 197	Data for key 11
198 – 206	Data for key 19
207 – 215	Data for key 27
216 – 224	Data for key 35
225 – 233	Data for key 43
234 – 242	Data for key 4
243 – 251	Data for key 12
252 – 260	Data for key 20
261 – 269	Data for key 28
270 – 278	Data for key 36

Table B-2. Debug Output Data Frame (Continued)

Frame Byte #	Description
279 – 287	Data for key 44
288 – 296	Data for key 5
297 – 305	Data for key 13
306 – 314	Data for key 21
315 – 323	Data for key 29
324 – 332	Data for key 37
333 – 341	Data for key 45
342 – 350	Data for key 6
351 – 359	Data for key 14
360 – 368	Data for key 22
369 – 377	Data for key 30
378 – 386	Data for key 38
387 – 395	Data for key 46
396 – 404	Data for key 7
405 – 413	Data for key 15
414 – 422	Data for key 23
423 – 431	Data for key 31
432 – 440	Data for key 39
441 – 449	Data for key 47. See Table B-3 for details of the data set for one key.
450 – 451	Setups CRC (SCRC), value most recently computed by the QT1481
452	Test pattern currently being used for various internal IEC/EN60730 tests
453 – 454	Key gain test sample taken on Y0
455 – 456	Key gain test sample taken on Y1
457 – 458	Key gain test sample taken on Y2
459 – 460	Key gain test sample taken on Y3
461 – 462	Key gain test sample taken on Y4
463 – 464	Key gain test sample taken on Y5

Table B-3. Format of Data Set for One Key

Offset Within Data Set	Description
1 – 0	Bits 12 – 0: Signal Bits 15 – 13: Calibration: 0 = Pending 1 – 4 = In progress 5 = Success 6 = Failed
3 – 2	Bits 12 – 0: Reference Bit 13: 1= LSL fail Bit 14: 1= Detect Bit 15: 1= KGTT FMEA fail
4	Normal DI
5	Bits 3 – 0: Fast DI
6	Negative Detect Time. Time remaining before key is recalibrated. Each count = 500 ms
7	Positive Detect Time. Time remaining before key is recalibrated. Each count = 500 ms
8	Drift compensation count, range –127 to +127. Each count = 100 ms. Recalibration occurs when the drift counter reaches the value set by NDRIFT (negative count) or PDRIFT (positive count)

Appendix C. Conducted Noise Immunity

Electrically conducted noise can increase the noise on the touch signals considerably and can lead to both false detects and missed touches. There is a specific test for conducted immunity, as part of typical EMC testing, which injects noise into the device under test across a broad range of frequencies and with significant amplitude. This test is designed to test the immunity of devices and products against environmental noise that is generated by commercial radio transmitters and other sources. Passing this test can be a challenge and might appear daunting, but with good design practise from the outset coupled with fine tuning of the QT1481 frequency hopping Setups and, designs based on the QT1481 show excellent noise immunity and can pass the conducted noise test by some margin.

From the outset, the design must target achieving the best possible touch sensitivity while minimizing noise. A clean design can achieve excellent signal delta on touch, but its easy to destroy this quality by pushing the overall product requirements too far and poor attention to detail during the design. The more imperfections that exist in a design, the more the sensitivity will be eroded and the conducted immunity performance with it.

A good target for signal delta on touch is 100 or even 150 counts.

Best touch performance is achieved with thinner overlay panels, optimum key electrode design, clean tracking between the QT1481 and the keys with low stray capacitance between X and Y traces and low stray capacitance from X and Y traces to ground, coupled with appropriate selection of the external matrix components. See the documents referred to in “Associated Documents” on page 70 for further details on best practice for designing a touch sensor interface.

The following paragraphs make some recommendations for initial values or type for the additional matrix components to accompany a QT1481 design together with some Setups the have been found to produce good noise immunity results.

Cs - NPO/COG

Use COG type for the charge sample capacitors (Cs0 - Cs7). This type of capacitor exhibits excellent stability, albeit at a higher cost, and is preferred over X7R and other types. Lower values can be used to reduce the cost or help with availability provided the charge transfer is not too high such as to saturate Cs. An excellent layout with low stray capacitance will typically allow Cs to be reduced as low as 1 nF.

Rs - 1 MΩ

Increasing the digital conversion ramp resistors (Rs0 – Rs7) increases sensitivity. The optimum selection for Rs is one that balances highest achievable sensitivity against conversion time and other undesirable side effects such as increased noise and possibly some reduction in temperature stability. Any increase in noise is typically insignificant compared to conducted noise observed during conducted immunity EMC testing.

Rs can be increased as high as 5 MΩ, although this is probably extreme in most cases. A good initial value is 1 MΩ. This value increases sensitivity to a good level with acceptable additional signal noise and often achieves the best compromise in Signal-to-Noise Ratio (SNR). The increased sensitivity allows a higher detect threshold to be employed, preventing noise spikes exceeding the threshold and thus preventing false detects.

Ry - 47 kΩ

Immunity to conducted electrical noise can be increased considerably by increasing the series resistors in the matrix Y lines (Ry0 - Ry7) at the expense of moisture tolerance. There is a practical limit to how far their value can be increased because higher values must be accompanied by longer dwell times, possibly in excess of the longest settings available in the device. In any case excessively long dwell times would slow the response time intolerably.

DWELL

Increasing the Ry values alone results in a reduction in charge transfer and sensitivity. This can be recovered by increasing the charge transfer time, or dwell time, through higher DWELL settings. To achieve the optimum DWELL setting, start with the maximum value and observe the reference values. Then reduce the dwell, con-

tinue to observe the reference values, and choose the lowest DWELL setting where the reference values are not significantly reduced from their maximum.

Longer dwell times also result in reduced moisture tolerance, and so a careful balance might be necessary between these conflicting requirements.

NTHR & THRM

Together NTHR(and PTHR) and THRM allow the threshold to be set. The threshold should be set as high as possible to prevent false detects but should not be set so high that true touches are not detected. The Atmel QmBtn or Hawkeye software can be used to observe the signals and signal delta on touch, and used to determine the optimum threshold settings for a specific design.

The threshold will typically need setting at a lower value to accommodate noise than is evident when testing in the absence of noise. This may initially seem counter intuitive until consideration is given to the fact that heavy noise can cause undesirable detect dropouts as well as missed touch events. The threshold should be set as high as possible, but not so high that the noise causes detect dropouts while the key is still touched.

Frequency Hopping - FHM = 3 (sweep)

The immunity of QT1481 based designs to higher noise amplitude can be increased by configuring the frequency hopping with 3 different frequencies and using FHM = 2. Choosing the 3 frequencies is a process of trial and error and varies from design to design, but a considerable increase in tolerable noise amplitude can be achieved once 3 appropriate frequencies have been identified.

However, frequency hopping using sweep mode (FHM = 3) is much easier to configure and may provide sufficient immunity for many applications and products.

Associated Documents

The following documents are a good source of general information regarding design and development of Atmel touch sensors and should be studied before starting the development of a new touch interface.

- QTAN0079 – *Buttons, Sliders and Wheels Sensor Design Guide*

  Refer to this application note for details of different possible X/Y electrode designs and patterns, and the optimum geometry to match the panel thickness.
- QTAN0062 – *Qtouch and Qmatrix Sensitivity Tuning for Keys, Sliders and Wheels*
- AVR3000 *Qtouch Conducted Immunity*

  Refer to this guide for further information on immunity from conducted noise.

These documents are available on the Atmel website (www.atmel.com).

Revision History

Revision No.	History
Revision AX – December 2010	•Initial release of datasheet for chip revision 4.0.
Revision BX – June 2011	• Datasheet updated for chip revision 6.0. • Command 0x02 added. • Frequency hopping updated. • NVT removed. • KHOPOF1, KHOPOF2, HOPOF removed and replaced by CFO_1 and CFO_2. • Setups Block – some setup values changed. • Debug output data frame values changed. • Some timings have changed. • Some text has been edited for clarification.
Revision CX – March 2013	• Datasheet updated for code revision 7.0 – Preliminary. • Front page – key outline size changed from 10 x 10 mm to 6 x 6 mm. • Front page – FMEA and IEC/EN60730 compliance reworded. • Front page – SPI text added to. • Section 2.5 – note added to end • Section 2.16 – text amended. • Updated text in Section 2.17 on page 16 • Removed incorrect text from Section 2.18 on page 16 • Section 2.20 – frequency hopping text amended. • Section 4.7 – in description of individual data bytes, the content of the bytes has been indented to make a clearer distinction between each one. • Table 4-5 – the CRC of 0x13 has changed to 16. • Table 4-5 – the return range of 0xC*k* is now 0x10 – 0x3F. • Table 4-5 – the return range of 0x4*k* is now 0 – 0xFF. • Section 5.19 – ‘reference’ has changed to ‘normal signal’. • Table 5-4 – the number of bits for address 342 ‘scope sync’ has changed to 6. • Table 5-4 – address 347 has been added. It is reserved. • Table 5-7 – text added at bottom of table to say that CFO_1 and CFO_2 are exceptions. • Table 5-8 – the values for FHM have changed. • Section 5.23 to Section 5.28 – text amended and added. • Section 6.1 – ‘Vdd’ and ‘voltage forced onto any pin’ values changed. • Section 6.1 – various values changed. • Added “Conducted Noise Immunity” on page 68 • Added documents to list in “Associated Documents” on page 70

Revision No.	History
Revision DX - August 2013	• Section 6.3 – Expanded Ta (Ambient Temperature) • Section 6.6 – Updated the Part marking diagram • Section 6.7 – Updated the Part number details
Revision EX - July 2014	Non-technical updates

Atmel | Enabling Unlimited Possibilities®

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

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

Atmel 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