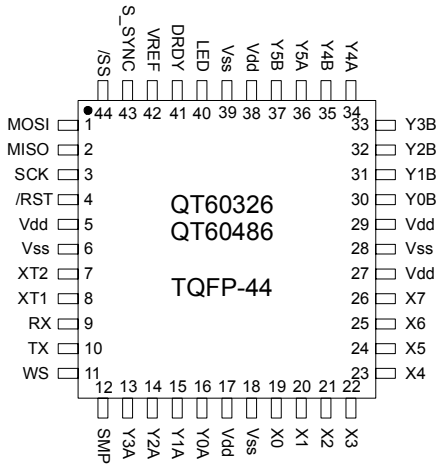
QT60326, QT60486
32 & 48 KEY QMATRIX™ ICs

- **Advanced second generation QMatrix™ controller**
- **Keys individually adjustable for sensitivity, response time, and many other critical parameters**
- **Panel thicknesses to 50mm through any dielectric**
- **32 and 48 key versions**
- **100% autocal for life - no in-field adjustments**
- **SPI Slave and UART interfaces**
- **Sleep mode with wake pin**
- **Adjacent key suppression feature**
- **Synchronous noise suppression pin**
- **Spread-spectrum modulation: high noise immunity**
- **Mix and match key sizes & shapes in one panel**
- **Low overhead communications protocol**
- **FMEA compliant design features**
- **Negligible external component count**
- **Extremely low cost per key**
- **44-pin Pb-free TQFP package**



APPLICATIONS -

- **Security keypanels**
- **Industrial keyboards**
- **Appliance controls**
- **Outdoor keypads**
- **ATM machines**
- **Touch-screens**
- **Automotive panels**
- **Machine tools**

These digital charge-transfer ("QT") QMatrix™ ICs are designed to detect human touch on up 48 keys when used with a scanned, passive X-Y matrix. They will project touch keys through almost any dielectric, e.g. glass, plastic, stone, ceramic, and even wood, up to thicknesses of 5 cm or more. The touch areas are defined as simple 2-part interdigitated electrodes of conductive material, like copper or screened silver or carbon deposited on the rear of a control panel. Key sizes, shapes and placement are almost entirely arbitrary; sizes and shapes of keys can be mixed within a single panel of keys and can vary by a factor of 20:1 in surface area. The sensitivity of each key can be set individually via simple functions over the SPI or UART port, for example via Quantum's QmBtn program, or from a host microcontroller. Key setups are stored in an onboard eeprom and do not need to be reloaded with each powerup.

These devices are designed specifically for appliances, electronic kiosks, security panels, portable instruments, machine tools, or similar products that are subject to environmental influences or even vandalism. It can permit the construction of 100% sealed, watertight control panels that are immune to humidity, temperature, dirt accumulation, or the physical deterioration of the panel surface from abrasion, chemicals, or abuse. To this end the device contains Quantum-pioneered adaptive auto self-calibration, drift compensation, and digital filtering algorithms that make the sensing function robust and survivable.

The parts can scan matrix touch keys over LCD panels or other displays when used with clear ITO electrodes arranged in a matrix. They do not require 'chip on glass' or other exotic fabrication techniques, thus allowing the OEM to source the matrix from multiple vendors. Materials such as such common PCB materials or flex circuits can be used.

External circuitry consists of a resonator and a few passive parts, all of which can fit into a 6.5 sq cm footprint (1 sq inch). Control and data transfer is via either an SPI or UART port.

These devices make use of an important new variant of charge-transfer sensing, transverse charge-transfer, in a matrix format that minimizes the number of required scan lines. Unlike older methods, it does not require one IC per key.

AVAILABLE OPTIONS

T_A	# Keys	Part Number
-40°C to +105°C	32	QT60326-AS-G
-40°C to +105°C	48	QT60486-AS-G

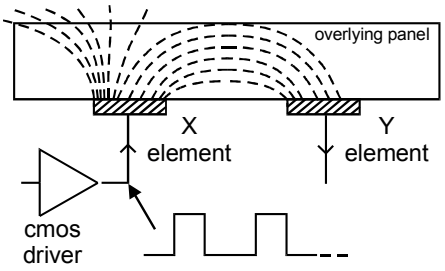
Contents

1 Overview

QMatrix devices are digital burst mode charge-transfer (QT) sensors designed specifically for matrix geometry touch controls; they include 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 possible lowest 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 2-sided adhesive acrylic film.

QMatrix parts employ transverse charge-transfer ('QT') sensing, a technology that senses changes in electrical charge forced across an electrode by a pulse edge (Figure 1-1).

Figure 1-1 Field flow between X and Y elements



QMatrix devices allow for a wide range of key sizes and shapes to be mixed together in a single touch panel. The approximate design rules for these keys can be seen in Figure 2-6.

The actual internal pattern style is not as important as is the need to achieve regular X and Y widths and spacings of sufficient size to cover the desired graphical key area or a little bit more; 2mm overhand is acceptable in most cases, since the fields drop off near the edges anyway. The overall key size can range from 10mm x 10mm up to 100mm x 100mm. The keys can be any shape including round, rectangular, square, etc. The internal pattern can be as simple as a single bar of Y or as complex as the interleaved structure shown in Figure 2-6.

For better surface moisture suppression, the outer perimeter of X should be as wide as possible, and there should be no ground planes near the keys. The variable 'T' in this drawing represents the total thickness of all materials that the keys must penetrate.

A picture of an actual board made using similar key geometries is shown on page 30.

The devices use both UART and SPI interfaces 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 device 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.

The parts are electrically identical with the exception of the number of keys which may be sensed.

1.1 Part differences

Versions of the device are capable of a maximum of 32 or 48 keys (QT60326, QT60486 respectively).

These devices are identical in all respects, except that each is capable of only the number of keys specified. These keys can be located anywhere within the electrical grid of 8 X and 6 Y scan lines.

Unused keys are always pared from the burst sequence in order to optimize speed. Similarly, in a given part a lesser number of enabled keys will cause any unused acquisition burst timeslots to be pared from the sampling sequence to optimize acquire speed. Thus, if only 40 keys are actually enabled, only 40 timeslots are used for scanning.

1.2 Enabling / Disabling Keys

The NDIL parameter is used to enable and disable keys in the matrix. Setting NDIL = 0 for a key disables it (Section 5.4). At no time can the number of enabled keys exceed the maximum specified for the device in the case of the QT60326.

On the QT60326, only the first 4 Y lines (Y0..Y3) are operational by default. On the QT60326, to use keys located on lines Y4 and Y5, one or more of the pre-enabled keys must be disabled simultaneously while enabling the desired new keys. This can be done in one Setups block load operation.

2 Hardware & Functional

2.1 Matrix Scan Sequence

The circuit operates by scanning each key sequentially, key by key. Key scanning begins with location X=0 / Y=0 (key #0). X axis keys are known as rows while Y axis keys are referred to as columns. Keys are scanned sequentially by row, for example the sequence X0Y0 X1Y0 .... X7Y0, X0Y1, X1Y1... etc. Keys are also numbered from 0..47. Key 0 is located at X0Y0. A table of key numbering is located on page 25.

Each key is sampled in a burst of acquisition pulses whose length is determined by the Setups parameter BL (page 22), which can be set on a per-key basis. A burst is completed entirely before the next key is sampled; at the end of each burst the resulting signal is converted to digital form and processed. The burst length directly impacts key gain; each key can have a unique burst length in order to allow tailoring of key sensitivity on a key by key basis.

2.2 Burst Paring

Keys that are disabled by setting NDIL =0 (page 21) have their bursts pared from the scan sequence to save time. 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.

2.3 Response Time

The response time of the device depends on the scan rate of the keys (Section 5.13), the number of keys enabled (Section 5.4), the detect integrator settings (Section 5.4), and the serial polling rate by the host microcontroller (or the use of the LED pin as an interrupt to the host; Sections 5.16, and Table 5.2 on page 25). An example timing:

Keys enabled (KE) = 20
Burst spacing (BS) = 1ms
NDIL = 3
FDIL = 5
Host polling rate (PR) = 10ms

The worst case response time is computed as:

((KE + FDIL) x NDIL x BS) + PR = Worst case response

((20 + 5) x 3 x 1ms) + 10ms = 85ms

The use of the LED pin to trigger host sampling can reduce this to ~75ms by saving the majority of the host polling time; see Section 5.16.

2.4 Oscillator

The oscillator can use either a quartz crystal or a ceramic resonator. In either case, the XT1 and XT2 must both be loaded with 22pF capacitors to ground. 3-terminal resonators having onboard ceramic capacitors are commonly available and are recommended. An external TTL-compatible frequency source can also be connected to XT1 in which case, XT2 should be left unconnected.

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

2.5 Sample Capacitors; Saturation Effects

The charge sampler capacitors on the Y pins should be the values shown. They should be X7R or NP0 ceramics or PPS film. The value of these capacitors is not critical but 4.7nF is recommended for most cases.

Cs voltage saturation is shown in Figure 2-1. This nonlinearity is caused by excessively negative voltage 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. The cause of this is usually from the burst length being too long, the Cs value being too small, or the X-Y coupling being too large. Solutions include loosening up the interdigitation of key structures, separating X and Y lines on the PCB more, increasing Cs, and decreasing the burst length.

Increasing Cs will make the part slower; decreasing burst length will make 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 more negative than -0.25 volts during any burst (not counting overshoot spikes which are probe artifacts), there is a potential saturation problem.

Figure 2-2 shows a defective waveform similar to that of 2-1, 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. 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 -0.25V. The waveform will appear deceptively straight, but it will start to flatten even before the -0.25V level is reached.

A correct waveform is shown in Figure 2-3. 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 QT60xx6 devices have no effect on conversion gain. However they do affect conversion time.

Unused Y lines should be left open.

2.6 Sample Resistors

There are 6 sample resistors (Rs) used to perform single-slope ADC conversion of the acquired charge on each Cs capacitor. These resistors directly control acquisition gain: larger values of Rs will proportionately increase signal gain. Values of Rs can range from 220KΩ to 1MΩ. 220KΩ is a reasonable value for most purposes.

Larger values for Rs will 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.

Figure 2-1 VCs - Non-Linear During Burst
(Burst too long, or Cs too small, or X-Y capacitance too large)

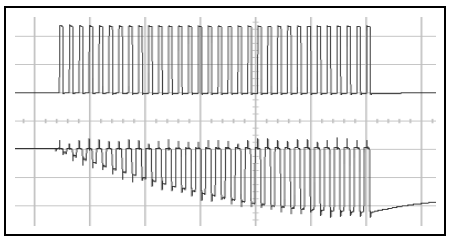


Figure 2-2 VCs - Poor Gain, Non-Linear During Burst
(Excess capacitance from Y line to Gnd)

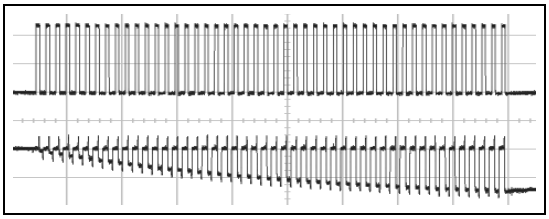


Figure 2-3 Vcs - Correct

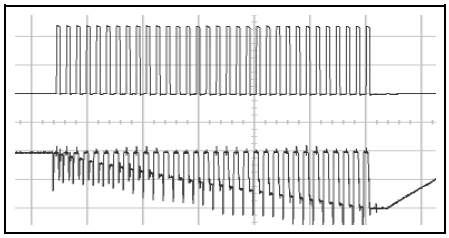


Figure 2-4 X-Drive Pulse Roll-off and Dwell Time

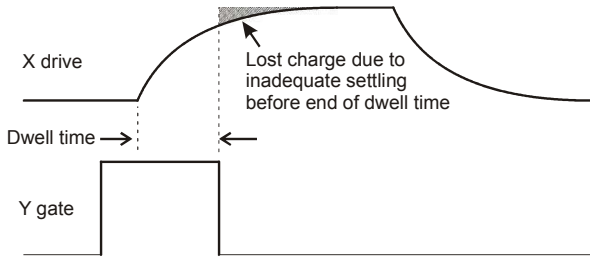
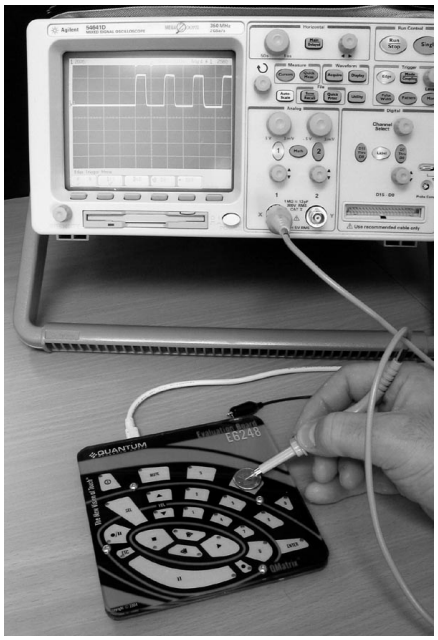


Figure 2-5 Probing X-Drive Waveforms with a Coin



2.7 Signal Levels

Using Quantum's 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 from 250 to 750 counts with properly designed key shapes (see appropriate Quantum app note on matrix key design). 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.

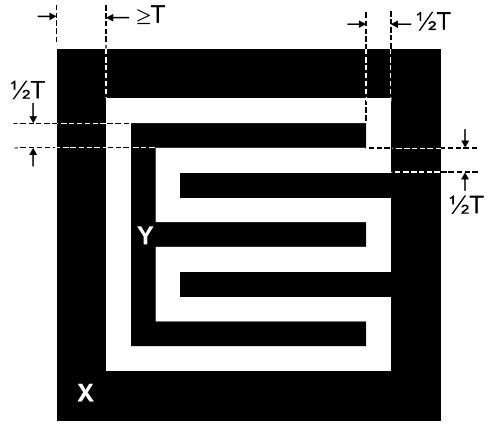
QmBtn software is available free of charge on Quantum's web site.

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 will increase the signal strengths as will increasing the sampling resistor (Rs) values.

Figure 2-6 Recommended Key Structure

'T' should ideally be similar to the complete thickness the fields need to penetrate to the touch surface. Smaller dimensions will also work but will give less signal strength. If in doubt, make the pattern coarser.



2.8 Matrix Series Resistors

The X and Y matrix scan lines should use series resistors (referred to as Rx and Ry respectively) for improved EMI performance.

X drive lines require them in most cases to reduce edge rates and thus reduce RF emissions. Typical values range from 1K to 20KΩ.

Y lines need them to reduce EMC susceptibility problems and in some extreme cases, ESD. Typical Y values range around 1KΩ. Y resistors act to reduce 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 for high frequency interference problems (above 20MHz) as the trace lengths between the components and the chip start to act as RF antennae.

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 will depending on key geometry and stray capacitance, and thus an oscilloscope is required to determine optimum values of both.

The upper limit of Rx can vary depending on key geometry and stray capacitance, and some experimentation and an oscilloscope are required to determine optimum values.

Dwell time (page 22) affects the duration in which charge coupled from X to Y can be captured. Increasing the dwell will increase the signal levels lost to higher values of Rx and Ry, as shown in Figure 2-4. Too short a dwell time will cause charge to be 'lost', if there is too much rising edge roll-off. Lengthening the dwell time will cause this lost charge to be recaptured, thereby restoring key sensitivity. In these devices, dwell time is adjustable (see Section 5.11) to one of 3 values.

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).

One way to determine X settling time is to monitor the fields using a patch of metal foil or a small coin over the key (Figure

2-5). Only one key along a particular X line needs to be observed, as each of the keys along that X line will be identical. The chosen dwell time should exceed the observed 95% settling of the X-pulse by 25% or more.

In almost all case, Ry should be set equal to Rx, which will ensure that the charge on the Y line is fully captured into the Cs capacitor.

2.9 Key Design

Circuits can be constructed out of a variety of materials including flex circuits, FR4, and even inexpensive single-sided CEM-1.

The actual internal pattern style is not as important as is the need to achieve regular X and Y widths and spacings of sufficient size to cover the desired graphical key area or a little bit more; ~3mm oversize is acceptable in most cases, since the key's electric fields drop off near the edges anyway. The overall key size can range from 10mm x 10mm up to 100mm x 100mm but these are not hard limits. The keys can be any shape including round, rectangular, square, etc. The internal pattern can be as simple as a single bar of Y within a solid perimeter of X, or (preferably) interdigitated as shown in Figure 2-6.

For better surface moisture suppression, the outer perimeter of X should be as wide as possible, and there should be no ground planes near the keys. The variable 'T' in this drawing represents the total thickness of all materials that the keys must penetrate.

See Figure 2-6 and page 30 for examples of key layouts.

2.10 PCB Layout, Construction

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 antennae. The Y (receive) lines are much more susceptible to noise pickup than the X (drive) lines.

Even more importantly, all signal related discrete parts (R's and C's) should be very close to the body of the chip. Wiring between the chip and the various R's and C's 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 and LED drive lines etc) will absorb the received key signals and reduce signal-to-noise ratio (SNR) and thus will be counterproductive. Ground planes around keys will also make water film effects worse.

Ground planes, if used, should be placed under or around the QT chip itself and the associated R's and C's in the circuit, under or around the power supply, and back to a connector, but nowhere else.

See page 30 for an example of a 1-sided PCB layout.

2.10.1 LED Traces and Other Switching Signals

Digital switching signals near the Y lines will induce transients into the acquired signals, deteriorating the SNR perfomance of the device. 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 10nF 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.

2.10.2 PCB Cleanliness

All capacitive sensors should be treated as highly sensitive circuits which can be influenced by stray conductive leakage paths. QT devices have a basic resolution in the femtofarad range; in this region, there is no such thing as 'no clean flux'. Flux absorbs moisture and becomes conductive between solder joints, causing signal drift and resultant false detections or transient losses of sensitivity or instability. Conformal coatings will trap in existing amounts of moisture which will then become highly temperature sensitive.

The designer should specify ultrasonic cleaning as part of the manufacturing process, and in cases where a high level of humidity is anticipated, the use of conformal coatings after cleaning to keep out moisture.

2.11 Power Supply Considerations

As these devices use the power supply itself as an analog reference, the power should be very clean and come from a separate regulator. A standard inexpensive LDO type regulator should be used that is not also used to power other loads such as LEDs, relays, or other high current devices. Load shifts on the output of the LDO can cause Vdd to fluctuate enough to cause false detection or sensitivity shifts.

Ceramic 0.1uF bypass capacitors should be placed very close and routed with short traces to all power pins of the IC. There should be at least 3 such capacitors around the part.

2.12 Startup / Calibration Times

The devices require initialization times as follows:

1. From very first powerup to ability to communicate:
   2,083ms (one time event to initialize all of eeprom, or to recover eeprom copy from FLASH in the event of eeprom corruption)
2. From powerup to ability to communicate:
   2,100 ms in the event the part is being forced to restore the factory defaults.
3. From powerup to ability to communicate:
   36 ms in the event the setups have been changed and the part needs to backup the EEPROM to flash.
4. Normal cold start to ability to communicate:
   3ms (Normal initialization from any reset)
5. Calibration time per key vs. burst spacings for 32 and 48 enabled keys (Table below):

Table 2-1 Calibration Timings

Burst Spacing, ms	Cal Time, ms, 32 keys	Cal Time, ms, 48 keys
*Auto	see text	see text
0.50	280	384
0.75	387	543
1.00	495	702
1.25	602	862
1.50	709	1,021
1.75	816	1,180
2.00	923	1,339
2.25	1031	1,500
2.50	1138	1,658
2.75	1245	1,817
3.00	1352	1,976

To the above, add 2,083ms, 36 ms, or 3ms from (1), (3), or (4) for the total elapsed time from reset to ability to report key detections.

***Auto mode determination time:** In Auto mode, burst spacings are assigned just after a reset event in a process that requires 30ms worst case per enabled key. Thus, if there are 32 keys enabled, the Auto mode calculation process requires 32 x 30ms = 960ms. Subsequent to the auto mode calculation time, the keys enter calibration mode. Thus, the startup time of the part is almost 1s longer than normal due to this 'determination time', and should be factored into the startup delay time.

Calibration time: The calibration time is shown in Table 2-1. Disabled keys are subtracted from the burst sequence and thus the cal time will be proportionately shorter than the numbers shown for lower key counts. In auto burst spacing mode, the burst spacing time should be measured on an oscilloscope and used to look up the calibration time value in the table.

Keys that cannot calibrate for some reason require 5 full cal cycles before they report as errors. However, the device can report back during this interval that the key(s) affected are still in calibration via status function bits. Keys in calibration also report back as being in error (Section 4.11) since keys in calibration are 'blind' to touch.

2.13 Reset Input

The /RST pin can be used to reset the device to simulate a power down cycle, in order to bring the part up into a known state should communications with the part be lost. The pin is active low, and a low pulse lasting at least 10µs must be applied to this pin to cause a reset.

To provide for proper operation during power transitions the devices have an internal brownout detector set to 4 volts.

The reset pin has an internal 30K ~ 60K resistor. A 2.2µF capacitor plus a diode to Vdd can be connected to this pin as a traditional reset circuit, but this is overkill.

A Force Reset command, 0x04 is also provided which generates an equivalent hardware reset.

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

2.14 Spread Spectrum Acquisitions

QT60xx6 devices use spread-spectrum acquisition modulation. This has the effect of drastically reducing EMI effects on the signals, while reducing the level of detectable RF emissions.

QT60xx6 spread spectrum operates by using a frequency chirp within each burst, in four different frequency bands.

This feature is hardwired into the device and cannot be disabled or modified.

2.15 Detection Integrators

See also Section 5.4, page 21.

The devices feature a detection integration mechanism, which acts to confirm a detection in a robust fashion. The basic idea is to increment a per-key counter each time the key has crossed its threshold. When this counter reaches a preset limit the key is finally declared to be touched. Example: If the limit value is 10, then the device has to detect a threshold crossing 10 times in succession without interruption, before the key is declared to be touched. If on any sample the signal is not seen to cross the threshold level, the counter is cleared and the process has to start over from the beginning.

The QT60xx6 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. If a fast-DI counter failed to reach its terminal count, the corresponding normal-DI counter is also reset. 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, i.e. 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 5x3=15 successive threshold crossings to declare a key as active.

2.16 FMEA Tests

FMEA (Failure Modes and Effects Analysis) 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 device failure, and the use of redundant signaling paths.

QT60xx6 devices incorporate special self-test features which allow products to pass such FMEA tests easily. These tests are performed during a dummy timeslot after the last enabled key.

The sequence of tests are performed repeatedly during normal running once all initialization, include the burst spacing optimization in auto mode, is complete. During initialization, all

FMEA error flags are cleared. Any FMEA errors will be 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 through a dedicated status command (page 15). Disabled keys are not tested. The existence of an error is also reported in normal key reporting commands such as Report 1st Key, page 15.

Assuming no detect events occur, the real time that elapses from the start of one sequence of FMEA tests to the start of the next, i.e. the FMEA sequence time, never exceeds 2s.

Also, since the devices only communicate in slave mode, the host can determine immediately if the QT has suffered a catastrophic failure. The QT 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 LED pin output, page 23).

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

Other tests incorporated into the devices include:

- A test for signal levels against a preset min value (LSL setup, see Section 5.15, page 23). If any signal level falls below this level, an error flag is generated.
- CRC communications checks on all critical command and data transmissions.
- 'Last-command' command to verify that an instruction was properly received.
- Loss of communications reset of the host controller.

For those applications requiring it, Quantum can supply sample FMEA test data on special request.

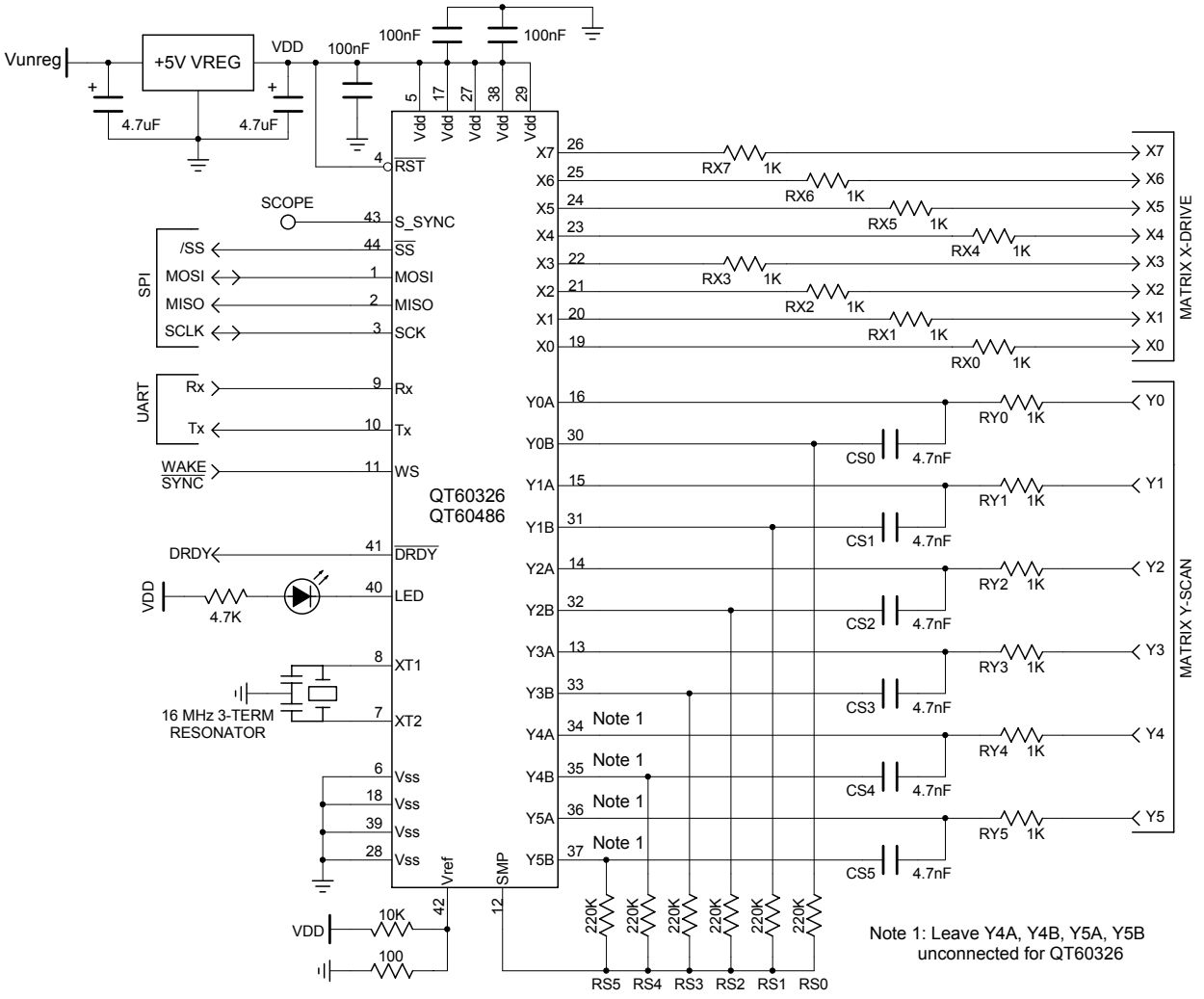
2.17 Wiring

Table 2.2 - Pin Listing

Applies to all devices

Pin	Function	I/O	Comments	If Unused, Connect To..
1	MOSI	I/O	SPI data input	Leave open
2	MISO	O	SPI data output	Leave open
3	SCK	I/O	SPI clock input	Leave open
4	/RST	I	Reset low; has internal 30K ~ 60K pull-up	Vdd
5	Vdd	P	Power, +5V	-
6	Vss	P	Supply ground	-
7	XT2	O	16 MHz 3-terminal resonator	Leave open
8	XT1	I		-
9	Rx	I	UART receive data input	Vdd
10	Tx	O	UART transmit data; has internal 20K ~ 50K pull-up	Leave open
11	WS	I	Wake-up from sleep input and/or sync input	Vdd
12	SMP	I/O	Sample output. Also - When forced high before reset, induces 'factory defaults' into all setups.	-
13	Y3A	I	Y line connection	Leave open
14	Y2A	I	Y line connection	Leave open
15	Y1A	I	Y line connection	Leave open
16	Y0A	I	Y line connection	Leave open
17	Vdd	P	Power, +5V	-
18	Vss	P	Supply 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, +5V	-
28	Vss	P	Supply ground	-
29	Vdd	P	Power, +5V	-
30	Y0B	I	Y line connection	Leave open
31	Y1B	I	Y line connection	Leave open
32	Y2B	I	Y line connection	Leave open
33	Y3B	I	Y line connection	Leave open
34	Y4A	I	Y line connection	Leave open
35	Y4B	I	Y line connection	
36	Y5A	I	Y line connection	Leave open
37	Y5B	I	Y line connection	
38	Vdd	P	Power, +5V	-
39	Vss	P	Supply ground	-
40	LED	O	Status output / LED indicator drive	Leave open
41	DRDY	O	1= Comms ready; has internal 20K ~ 50K pull-up	-
42	Vref	I	0.05V nominal +/-10% via external divider	-
43	S_Sync	O	Scope Sync: Synchronization test signal	Leave open
44	/SS	I	SPI slave select; has internal 20K ~ 50K pull-up	Leave open

Figure 2.7 Wiring Diagram



Note: Use either UART or SPI comm port but not both. Device defaults to SPI communications but if it receives a UART byte at any time, locks into UART mode instead. See Table 2.2 for connections when pins are unused.

3 Serial Communications

These devices can use either SPI or UART communications modes; it cannot use both at the same time. The part defaults to SPI mode unless it receives a byte over the UART interface. If a UART byte is received at any time, the UART interface is enabled and the SPI interface is totally disabled until after the next device reset.

The host device always initiates communications sequences; the QT is incapable of chattering data back to the host. This is intentional for FMEA purposes so that the host always has total control over the communications with the QT60xx6. In SPI mode the device is a slave, so that even return data following a command is controlled by the host. In UART mode, the device will still only respond back 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 QT. Some transmission types from the host or the QT employ a CRC check byte to provide for robust communications.

A DRDY line is provided that handshakes transmissions. Generally this is needed by the host from the QT to ensure that transmissions are not sent when the QT is busy or has not yet processed a prior command. In UART mode this line is bi-directional, and the QT can use it to suspend transmissions back to the host if the host is busy.

Initiating or Resetting Communications: After a reset, or, should communications be lost due to noise or out-of-sequence reception, the host should send a 0x0f (return last command) command repeatedly until the compliment of 0x0f, i.e. 0xf0, is received back. 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 10ms; even 50ms is more than fast enough to extract status data using the 0x06 command (report first key: see page 15) in most situations. Streaming commands like the 0x0d command (dump setups: see page 15) or multi-byte response commands like 0x07 or 0x08 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 also Section 4.23). 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 commands 0x0b and 0x0c and take appropriate action. Issuing a 0x07 command all the time is wasteful of bandwidth, requires more host processor time, and actually conveys less information (no error flags are sent via a 0x07 command).

3.1 DRDY Pin

DRDY is an open-drain output (in SPI mode) or bidirectional pin (in UART mode) with an internal 20K ~ 50K pull-up resistor.

Serial communications pacing is controlled by this pin. In either UART or SPI mode, the host is permitted to send data only when DRDY is high. In UART mode, the device additionally will hold up responses to the host if DRDY is being held low by the host. After a byte is received DRDY will always go low even if only for a few microseconds; during this period the host should not send data. Therefore, after each byte transmission the host should first check that DRDY is high again.

If the host desires to send a byte to the QT it should behave as follows:

1. If DRDY is low, wait
2. If DRDY is high: send a command to QT
3. Wait 20µs (time S5 in Figure 3-3: DRDY is guaranteed to go low before this 20µs expires)
4. Wait until DRDY is high (it may already be high again)
5. Send next command or a null byte 0x00 to QT

It takes up to 1ms for DRDY to go high again after a command, except for a few commands listed in Section 4:

0x01 (Setups load):	<20ms
0x0E (Get eeprom CRC):	<20ms
0x16 (Sleep):	<5ms

Other DRDY specs:

Min time DRDY is low:	1µs
Min time DRDY is low after reset:	1ms

3.2 SPI Communications

SPI mode is selected by default after reset. There is no other configuration required to make the device operate in SPI mode. If a UART byte occurs before or even after SPI transmissions have taken place, the device will switch to UART mode and remain in that mode until the device is reset.

Figure 3-1 Basic SPI Connections

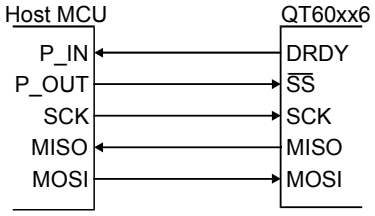
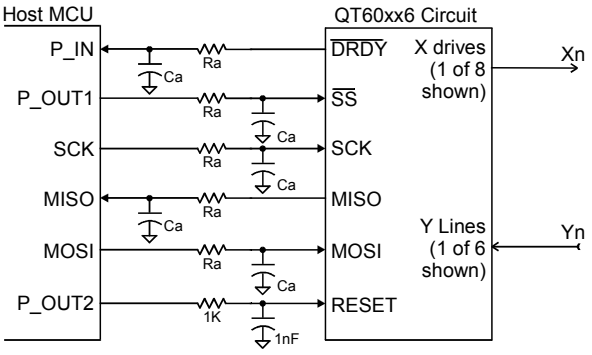


Figure 3-2 Filtered SPI Connections



Recommended Values of Ra & Ca

SPI Clock Rate	Ra	Ca
4MHz	680	33pF
400kHz	1,000	270pF
100kHz	2,200	470pF
50kHz	2,200	1nF

SPI communications operates in slave mode only, and obeys DRDY control signaling. The clocking is as follows:

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

SPI mode requires 5 signals to operate:

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 when /SS is high to allow multi-drop communications along with other slave parts.

SCK - SPI clock - input only clock from host. The host must shift out data on the falling SCK edge; the QT60xx6 clocks data in on the rising edge. The QT60xx6 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. **Important:** SCK must idle high; it should never float.

/SS - Slave select - input only; acts as a framing signal to the sensor from the host. /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; /SS must idle high. This pin includes an internal pull-up resistor of 20K ~ 50K. When /SS is high, MISO floats.

DRDY - Data Ready - active-high - indicates to the host that the QT is ready to send or receive data. This pin idles high. This pin includes an internal pull-up resistor of 20K ~ 50K. In SPI mode this pin is an output only (i.e. open drain with internal pull-up).

The MISO pin on the QT floats in 3-state mode between bytes when /SS is high. This facilitates multiple devices on one SPI bus.

Null Bytes: When the QT responds to a command with one or more response bytes, the host should issue a 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 QT from the prior command via nulls.

New commands attempted during intermediate byte transfers are ignored.

Wake operation: The device can be put into sleep mode with a serial command, 0x16 (page 16) and then be awakened later with a 10µs minimum low level on the WS pin. With the /SS line tied to WS, the host can simply toggle /SS low for 10µs minimum to wake the part; the host should not send an actual SPI byte to prevent the device from seeing a byte it cannot properly interpret due to timing errors during wakeup.

The recommended method to reestablish communications after Wake from Sleep is to send the QT device a 0x0F ('Get Last Command' command) repeatedly until the correct response comes back (the command's own compliment, i.e. 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 as shown in Figure 3-2.

CRC checks have also been added to critical commands in order to detect transmission errors to a high level of certainty.

3.3 UART Communications

See also SR setup parameter, page 23.

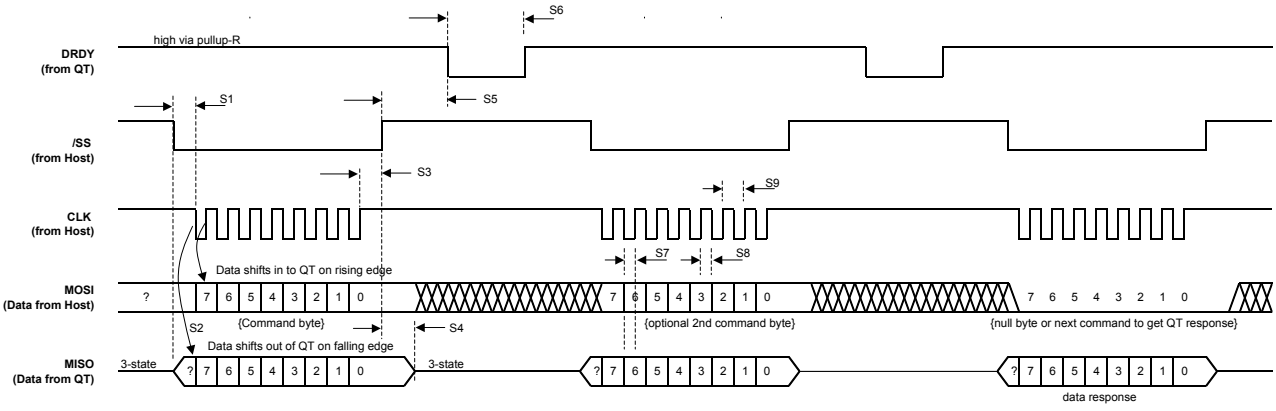
UART mode is selected as soon as the QT receives any data on the UART Rx pin. There is no other configuration required to make the device operate in UART mode. Once UART is selected after a power-up, the device cannot switch to SPI mode unless the device is reset.

UART mode communications functions in the same basic way as SPI communications. The Baud rate is adjusted by means of setup parameter 'SR' (page 23). Once a new Baud rate has been set, the device 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 QT. No framing /SS or clock signal is required, simplifying the interface greatly. Return data is sent from the QT back to the host when the data is ready.

Figure 3-3 SPI Slave-Only Mode Timing (Fosc = 16MHz)

S1: ≥125ns S2: ≤20ns S3: ≥25ns S4: ≤20ns
S5: ≤20µs S6: ≥1µs S7: ≥125ns S8: ≥125ns S9: ≥250ns



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

Wake operation: The device can be put into sleep mode with a serial command, 0x16 (page 16) and then be awaked with a dummy byte from the host, if the Rx and WS pins are connected together. 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.

The recommended method to reestablish 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 compliment, i.e. 0xF0).

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

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

UART transmission parameters are:

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

DRDY in UART mode: Section 3.1 applies.

DRDY is bi-directional in UART mode. DRDY can be pulled down by either the QT 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 10µs grace period after clamping DRDY low in which it can still accept the start bit of a transmission from the QT.

As explained in Section 3.1, DRDY is not clamped low immediately after the QT receives a byte; there can be up to a 20µs delay from the end of the stop bit before DRDY goes low. Sampling of DRDY by the host should occur 20µs after the byte has been fully sent; if DRDY is already high at this point, or becomes high, then it is clear to send.

Null Bytes: Unlike SPI mode, there is no reason to send null bytes to the QT in UART mode. The QT device will respond to commands with data when ready, subject to the 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.

Figure 3-4 UART Timing

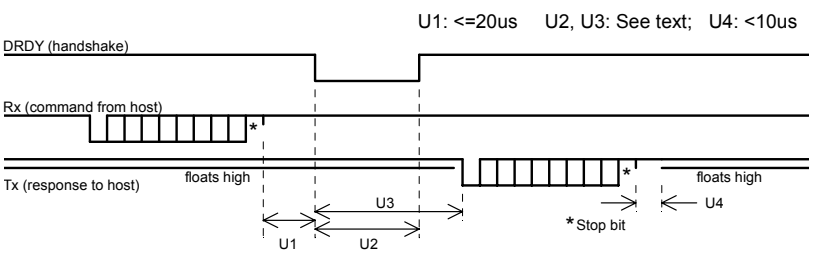
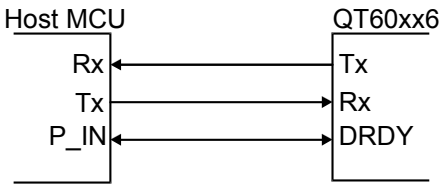


Figure 3-5 UART Connections



Simple RC networks and slower data rates are helpful to resolve these issues. CRC checks have been added to critical commands in order to detect transmission errors to a high level of certainty.

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

4 Control Commands

Refer to Table 4.2, page 18 for further details.

Suggested command sequencing: See Section 4.23, page 16.

The devices feature a set of commands which are used for control and status reporting. The host device has to send the command to the QT60xx6 and await a response.

SPI mode: While waiting the host should delay for 20 µs from the end of the command, then start to check if DRDY is or goes high. If it is high, then the host master can clock out the resulting byte(s).

UART mode: After the command is sent, the QT will send back the response usually starting within 1ms. The host can clamp DRDY low (wire-AND logic) to inhibit a response if the host is not able to receive the transmission for a while.

Command timeouts: Where a command involves multi-byte transfers in either direction, each byte must be transmitted within 100ms of the prior byte or the command will timeout. 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 device 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 1ms maximum; timing parameters U2 and U3 in Figure 3-4 will thus be 1ms maximum for these commands. The exceptions are:

0x01 (Setups upload):	<20ms
0x0E (Get eeprom CRC):	<20ms
0x16 (Sleep):	<5ms

4.1 Null Command - 0x00

Used to shift back data from the QT 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 QT, not to send a command.

In SPI communications, when the QT60xx6 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.

4.2 Enter Setups Mode - 0x01

This command is used to initiate the Setups block transfer from Host to QT.

The command must be repeated 2x within 100ms or the command will fail; the repeating command must be sequential without any intervening command. After the 2nd 0x01 from the host, the QT will stop scanning keys and reply with the character 0xFE. In SPI mode this character must be shifted out by sending a null (0x00) from the host. This command suspends normal sensing starting from the receipt of the second 0x01. A failure of the command will cause a timeout.

Each byte in the block must arrive at the QT no later than 100ms after the previous one or a timeout will occur. Any timeout will cause the device to cancel the block load and go back to normal operation.

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

If 0xFE is received by the host from the QT, then the host should begin to transmit the block of Setups to the QT. The DRDY line handshakes the data. The delay between bytes can be as short as 10µs but the host can make it longer than this if required, but no more than 100ms. The last two bytes the host should send is the CRC for the block of data only (ie the CRC should not include the command in its calculation).

After the block transfer the QT will check the CRC and respond with 0x00 if there was an error. Regardless, it will program the internal eeprom. If the CRC was correct it will reply with a second 0xFE after the eeprom was programmed.

If there was an error in the block transfer the device will restore the last known good Setups from Flash memory the next time the device is reset. However until that point, the device will attempt 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 (e.g. Baud rate).

At the end of the full block load sequence, the device restarts sensing without recalibration. Most of the setups in the block will take effect immediately, but it is important to reset the device after a block load to make all the changes effective and permanent. See Section 4.4.

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

4.3 Cal All - 0x03

This command must be repeated 2x within 100ms or the command will fail; the repeating command must be sequential without any intervening command.

After the 2nd 0x03 from the host, the QT will reply with the character 0xFC. Shortly thereafter the device will recalibrate all keys and restart operation.

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

The host can monitor the progress of the recalibration by checking the status byte, using command 0x05, during the course of the calibration.

A key will show an error flag (using command 0x8k) indicating the key has failed calibration if its signal is too noisy or if its signal is below the low signal threshold. A key is deemed too noisy if, at the end of calibration, the signal is no longer between its computed negative hysteresis level and positive thresholds.

4.4 Force Reset - 0x04

The command must be repeated 2x within 100ms or the command will fail; the repeating command must be sequential without any intervening command. After the 2nd 0x04, the device will reply with the character 0xFB just prior to executing the reset operation.

After any reset, the device 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 status byte for recalibration, by repeatedly issuing command 0x05 (see below).

4.5 General Status - 0x05

This command returns the general status bits. This command is not as useful as the 0x06 command for routine use. The bits returned from 0x05 are as follows:

BIT	Description
7	Reserved
6	1= communications failure
5	1= FMEA failure detected
4	1= eeprom corrupt
3	1= Mains sync error
2	1= calibration has failed on an enabled key or, an LSL failure
1	1= any key in calibration
0	1= any key in detect

Notes:

Bit 6: Set if a communications failure. This can be reset by sending command 0x0f (“last command command”) repeatedly until a response of 0xf0 is received.

Bit 5: Set if an FMEA error was detected during operation. See Section 2.16. A further amplification of what the FMEA error consisted of is described in Section 4.12.

Bit 4: Set if eeprom corruption was detected. If this happens, it is recommended that the device be reset. If a reset does not cure the problem, the Setups block should be reloaded (Section 4.2) and the device reset again.

Bit 3: Set if there was a mains sync error, for example there was no Sync signal detected within the allotted 100ms amount of time. See Section 5.12. This condition is not necessarily fatal to operation, however the device will operate very slowly and may suffer from noise problems if the sync feature was required for noise reasons.

Bit 2: Reports either a cal failure (failed in 5 sequential attempts) on any enabled key or, that an enabled key has a very low signal reference value, lower than the user-settable LSL value (Section 5.15).

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

Bit 0: Set if any key is in detection (touched).

A CRC byte is appended to the response to the 0x05 command; this CRC folds in the command value 0x05 itself initially.

4.6 Report 1st Key - 0x06

Reports the first or only key to be touched, plus indicates if there are yet other keys that are also touched.

The return bits are as follows:

BIT	Description
7	1= more than 1 key is active
6	1= any error condition is present
5	Key bit 5
4	Key bit 4
3	Key bit 3
2	Key bit 2
1	Key bit 1
0	Key bit 0

Bits 5..0 encode for the first detected key in range 0..47. If no keys are active, these 6 bits are all 1's (0x3F, 63 decimal when bits 6, 7 are masked off).

If 2 or more keys in detection, bit 7 is set and the host should interrogate the part via the 0x07 command to read out all the key detections. This one command should be the dominant interrogation command in the host interface; further commands can be issued if the response to 0x06 warrants it. For example, if there is an error flag, command 0x05 can be sent to find the cause. If bit 7 is set, the command 0x07 can be sent to find further keys in detection.

A CRC byte is appended to the response; this CRC folds in the command 0x06 itself initially.

4.7 Report Detections for All Keys - 0x07

Returns six bytes which indicate all keys in detection if any, as a bitfield. Key 0 reports in bit 0 of the first byte returned; key 47 is reported in bit 7 of the last byte returned. See Table 4.1 and Table 5.3, page 25.

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

Table 4.1 Bit fields for multiple key reporting and key numbering

Key #		Bit Number (X line #)							
		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	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

4.8 Report Signals for All Keys - 0x08

Returns the raw signal values for all keys. Each value is a 16-bit number, and there are 48 words returned. No CRC is appended to the return, so the data should not be considered secure. The low byte of key 0 is returned first.

4.9 Report References for All Keys - 0x09

Returns the reference values for all keys. Each value is a 16-bit number, and there are 48 words returned. No CRC is appended to the return, so the data should not be considered secure. The low byte of key 0 is returned first.

4.10 Report Deltas for All Keys - 0x0a

Returns the delta signal values with respect to the reference levels for all keys. Each value is an 8-bit unsigned number representing {reference - signal}; negative results are truncated to zero, i.e. those where the signal rises above the reference value. This command returns 48 bytes. No CRC is appended to the return, so the data should not be considered secure. The byte for key 0 is returned first.

If the delta value attempts to exceed 255, the result is limited to 255.

4.11 Report Error Flags for All Keys - 0x0b

Returns six bytes which show error flags as a bitfield for all keys. Key 0 reports in bit 0 of the first byte returned; key 47 is reported in bit 7 of the last byte returned. See Table 4.1 and Table 5.3, page 25.

One type of error reported with this command is the signal being below its limit point (Section 5.15).

A key that is in calibration also is reported as an error in the response since it cannot operate as a key while it is calibrating. This type of error flag is self-cleared once the key exits from calibration. A calibration error flag due to an actual cal error can be found thereafter using command 0x8k (Section 4.21).

A CRC byte is appended to the response; this CRC folds in the command 0x0b itself initially.

Note that these error bits exclude FMEA failure modes.

4.12 Report FMEA Status - 0x0c

Returns one byte which shows the FMEA error status of the X and/or Y matrix scan lines. If an X line is in error, the corresponding bit (below) is set. If a Y line has an FMEA error, the entire field is set to ones (0xFF).

Due to the physics of matrix wiring, a fault on any Y line will cause faults to be reported on all X lines as well. It is not possible to separate out these faults for reporting purposes.

b7	b6	b5	b4	b3	b2	b1	b0
X7	X6	X5	X4	X3	X2	X1	X0

A CRC byte is appended to the response; this CRC folds in the command 0x0c itself initially.

For more information see Section 2.16.

4.13 Dump Setups Block - 0x0d

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

If the transfer is not paced faster than 100ms per byte the transfer will be aborted and the device will time out. This can happen if the host is also controlling DRDY.

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.

4.14 Eeprom CRC - 0x0e

This command returns the 16-bit CRC calculated from the eeprom contents. The CRC is sent back LSByte first. The CRC sent back is the same CRC that is appended to the end of the Setups block.

This command requires substantial amounts of time to process and return a result; it is not recommended to use this command except perhaps on startup or very infrequently.

Command response timing: The response to this command can take as long as 20ms.

No CRC is appended to the response.

4.15 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 will return the inversion of 0x0f, 0xf0.

If a prior command was not valid or was corrupted, it will return the bad command as well. This command also will reset the communications error flag (Section 4.5).

No CRC is appended to the response.

4.16 Internal Code - 0x10

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

A CRC byte is appended to the response; this CRC folds in the command 0x10 itself initially.

4.17 Internal Code - 0x11

This command returns an internal code word (3 bytes) of the part for factory diagnostic purposes.

A CRC byte is appended to the response; this CRC folds in the command 0x11 itself initially.

4.18 Internal Code - 0x12

This command returns an internal code word (2 bytes) of the part for factory diagnostic purposes.

No CRC is appended to the response.

4.19 Sleep - 0x16

The command must be repeated 2x within 100ms or the command will fail. After the 2nd 0x16 from the host, the device will reply with the character 0xE9, then sleep. On wake from Sleep, the device will issue a 0x01 character back to the host.

Communications response timing: Responses to this command may take from a few microseconds up to 5ms, depending on what the device was doing at the moment the command arrived.

After the response, the device will enter low power sleep mode until awakened by a >10µs low level on the WS pin. When it wakes, it will resume current operation in the state from which it exited and attempt to send a 0x01 code back to the host to signal that it is ready to communicate again.

During Sleep the DRDY pin is held low, and released once the device awakes and is ready to return the 0x01 code.

The WS pin can be connected to Rx or /SS to provide a 'free' wakeup connection from the host controller. In SPI mode with /SS tied to WS, a /SS toggle (any low pulse of at least 10µs) under software control from the host controller without an actual SPI transmission will wake the device.

In UART mode, with Rx tied to WS a 0xFF byte should be sent to provide a pulse on WS. 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.

A recommended method to reestablish communications after Wake from Sleep is to send the QT device a 0x0F ('Last Command' command) repeatedly until the correct response comes back (the command's own compliment, i.e. 0xF0).

If Sync mode is also enabled, the part will assume the wakeup pulse is also a sync signal, and resume scanning starting with Key 0 (which is not necessarily where it left off scanning when it went to sleep).

WS Note: The device checks the WS pin and waits for it to return high before the part actually begins sleep. There is a timeout limit on this wait of ~2s; if the WS pin has not gone high after this time, the part will reset itself.

4.20 Data Set for One Key - 0x4k

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 bytes, in the sequence:

Signal (2 bytes)
Reference (2 bytes)
Normal Detect Integrator (1 byte)

Signal and Reference are returned LSByte first.

No CRC is appended.

4.21 Status for Key 'k' - 0x8k

Returns a bitfield for key 'k' where k is from {0..47}. The bitfield indicates as follows:

BIT	Description
7	1= reserved
6	1= reserved
5	1= reserved
4	1= key is enabled
3	1= key is in detect
2	1= signal ref < LSL (low signal error)
1	1= key is undergoing calibration
0	1= cal on this key failed 5 times

Bit 2 - LSL notes: See page 23.

A CRC byte is appended to the response; this CRC folds in the command 0x8k itself initially.

4.22 Cal Key 'k' - 0xck

This command must be repeated 2x within 100ms or the command will fail; the repeating command must be sequential without any intervening command.

This command functions the same as 0x03 CAL command except this command only affects one key 'k' where 'k' is from 0 to 47.

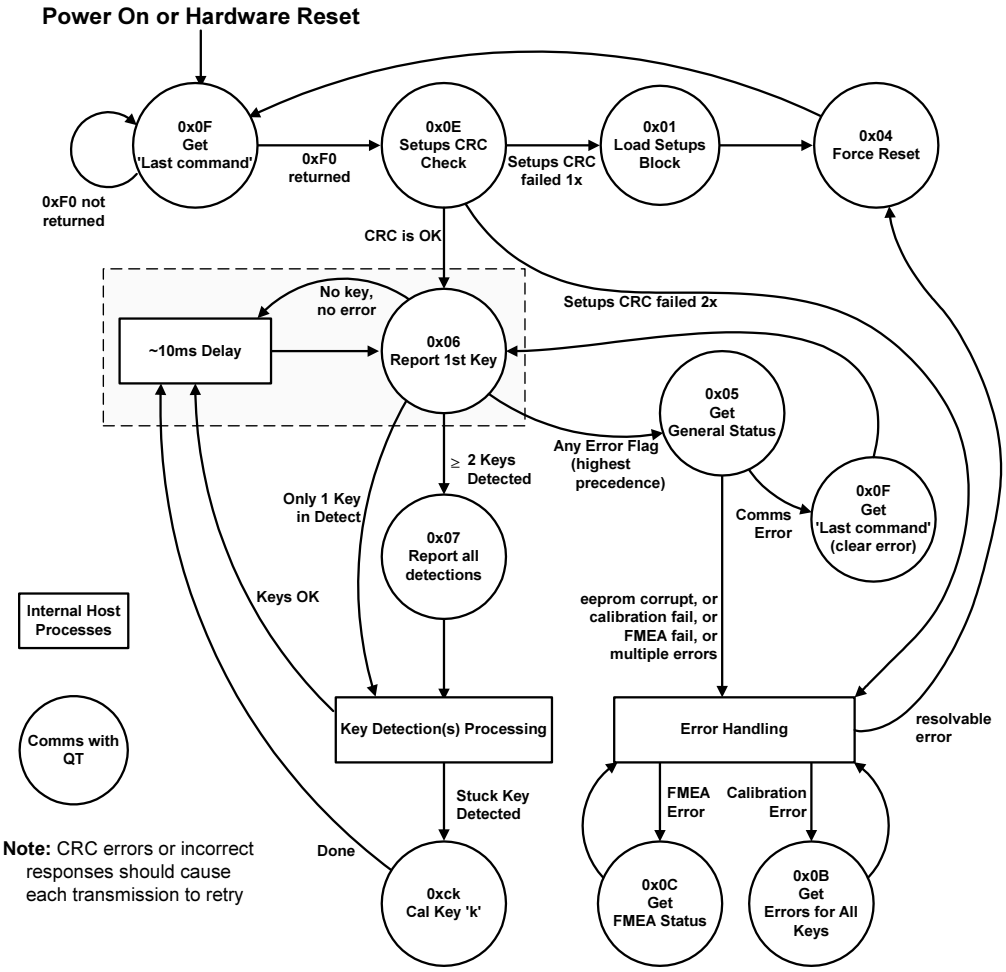
The chosen key 'k' is recalibrated in its native timeslot; normal running of the part is not interrupted and all other keys operate correctly throughout. This command is for use only during normal operation to try to recover a single key that has failed or is not calibrated correctly.

Returns the 1's compliment of 0xck just before the key is recalibrated.

4.23 Command Sequencing

To interface the device with a host, the flow diagram of Figure 4-1, page 17, is suggested. The actual settings of the Setups block used should normally just be the default settings except

Figure 4-1 Suggested Communications Flow



where changes are specifically required, such as for sensiti vity, 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 device will fail to respond within the allotted time, or the response CRC will be incorrect, or the response will be 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 will spend 99% of its time alternating between the two states within the dashed rectangle. If a key is detected, the control flow will enter ‘Key Detection Processing’.

Stuck Key Detection processing (0xck) is optional, since the device 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 device 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 class error).

Normally it is not required to reload the setups, since the device itself stores a backup copy of these in Flash memory should the eeprom become corrupt. However the host should reset the QT so that the device will copy the Flash setups to eeprom, and then the QT should check that the eeprom CRC code is correct. Only if this fails should the eeprom be reloaded by the host. One exception to this rule is just after powerup, 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 powerup cycle.

The ‘Last Command’ command can be used at any time to clear comms error flags and to resynchronize failed communications, for example due to timing errors etc.

Table 4.2 Command Summary

Hex	Name	Description	#/Cmd	# rtnd	Rtn range	CRC	Notes	Page
0x00	Null command	Used to get data back in SPI mode	1	1	0..0xFF	-	Flushes pending data from QT; one required to extract each response byte.	13
0x01	Enter Setups mode	Enter Setups, stop sensing; followed by block load of binary Setups of length 'nn'. Command must be repeated 2x consecutively without any intervening command in 100ms to execute. Sensing auto-restarts, however, the device should be reset after the block load to ensure all new setups will take effect.	2 + nn + 2	2	0xFE + 0xFE OR 0xFE + 0x00 (err)	16	First 0xFE issued when ready to get data, second 0xFE issued when all loaded and burned; else timeout. If 2 commands not received in 100ms, times out and no response is issued. Part will timeout if each byte not received within 100ms 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. A CRC of 0x0000 is also acceptable in which case the CRC is not checked. Internal EEPROM will update regardless of CRC health, but, if the CRC is bad, the EEPROM will be declared invalid and thus on reset the EEPROM will be restored from flash backup, overwriting the desired (but corrupt) new setups.	14
0x03	CAL all	Force device to recalibrate all keys; reenters RUN mode afterwards automatically; 0x03 must be repeated 2x consecutively without any intervening command in 100ms to execute	2	1	0xFC	-	Returns 1's complement of command to acknowledge cmd once the cal has been initiated. If 2 commands not received in 100ms, times out and no response is issued.	14
0x04	Force reset	Force device to reset. Command must be repeated 2x consecutively without any intervening command in 100ms to execute	2	1	0xFB	-	Returns 1's complement of command to acknowledge command prior to reset. If 2 commands not received in 100ms, times out and no response is issued.	14
0x05	General status	Get general part status.	1	2	0..0xFF	8	Bit 7: reserved Bit 6: 1= comms error: unrecognized command received This bit can be reset by 0x0F cmmd Bit 5: 1= FMEA failure Bit 4: 1= eeprom is corrupt Bit 3: 1= line sync failure Bit 2: 1= cal failed 5 times on an enabled key, or, an enabled key has a low reference (Ref < lower sig lim) Bit 1: 1= any key in calibration Bit 0: 1= any key is in detect Last return byte is CRC-8 of cmmd + return data	14
0x06	Report 1st key	Get indication of first touched key + others	1	2	0..0xFF	8	Bit 7: 1= indicates 2 or more touches if set. Bit 6: 1= any of the following prevail: calibrating, key(s) failed cal 5 times, sync fail, comms error, FMEA failure, EEPROM corrupt. Bits 5..0: indicates key number (0..47) of first key touched; reads 0x3F (63 decimal) if no touch. 2nd return byte is CRC-8 of cmmd + return data	15
0x07	Report all keys	Sends back all key detect status bits (bitfield)	1	7	0..0xFF 6 bytes	8	Last return byte is CRC-8 of cmmd + return data	15
0x08	Signals for all	Sends back all key signal levels	1	96	0..0xFFFF 48 words	-	Returns block data for all keys' signals The low order byte is returned first.	15

Hex	Name	Description	#/Cmd	# rtnd	Rtn range	CRC	Notes	Page
0x09	References for all	Sends back all key reference levels	1	96	0..0xFFFF 48 words	-	Returns block data for all keys' references. The low order byte is returned first.	15
0x0a	Deltas for all	Sends back all key delta signals from ref	1	48	0..0xFF 48 bytes	-	Returns block data for all keys' signal deltas from refs, {reference - signal}; Unsigned binary bytes, in range 0..255.	15
0x0b	Error flags for all	Error bit fields	1	7	0..0xFF 6 bytes	8	Last return byte is CRC-8 of cmmd + return data	15
0x0c	FMEA status	FMEA bitfield on X, Y lines	1	2	0..0xFF	8	Last return byte is CRC-8 of cmmd + return data	15
0x0d	Dump Setups	Returns Setups block area followed by CRC. Scanning is halted and then auto-restarted after the cmd 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. Part will timeout if each byte not transmitted within 100ms of previous byte. (This can happen if DRDY is driven by the host).	15
0x0e	Eeprom CRC	Get eeprom CRC	1	2	0..0xFFFF	16	CRC-16 only on Setups array section of eeprom This CRC is the same as the CRC at the end of Setups block load. This word is returned low byte first.	15
0x0F	Return last cmmd	Returns last command received	1	1	0..0xFF	-	Returns 1's compliment of last command even if bad. Resets the communications error flag.	16
0x10	Return internal code	Diagnostic code for factory use.	1	2	0..0xFF	8	Returns internal code. 2nd byte is CRC-8 of cmmd + return data	16
0x11	Return internal code	Diagnostic code for factory use.	1	4	0..0xFFFFFF	8	Returns internal code. Last byte is CRC-8 of cmmd + return data. The low order byte is returned first.	16
0x12	Return internal code	Diagnostic code for factory use.	1	2	0.0xFFFF		Diagnostic code for factory use. The low order byte is returned first.	16
0x16	Sleep	Enter sleep; Command must be repeated 2x consecutively without any intervening command in 100ms to execute.	2	1 + 1	0xE9 + 0x01	-	Returns 1's complement of command to acknowledge; wakes on INT, meanwhile sleeps in low power mode; 0x01 when restarted. If 2 commands not received in 100ms, times out and no response is issued. DRDY is held low while the part is asleep. DRDY is released high once awake and ready to return the 0x01.	16
0x4k	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	5	0..FF Each byte	-	Diagnostic use only, not to be relied upon (no CRC). Signal and ref are Tx as 2 bytes, LSB first.	16
0x8k	Status for key 'k'	Get status byte for key 'k' {0..47}	1	2	0..FF	8	Bits 7..5: reserved Bit 4: 1= key is enabled 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 5 times Second return byte is CRC-8 of cmmd + return data	16
0xck	CAL key 'k'	Force calibration of key # k where k= 0..47. Command must be repeated 2x consecutively without any intervening command in 100ms to execute	2	1	~0xck	-	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 compliment of the cmd char, once the cal is scheduled.	16

5 Setups

The devices calibrate and process all signals using a number of algorithms specifically designed to provide for high survivability in the face of adverse environmental challenges. They provide 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 these algorithms to suit each application. These setups are loaded into the device in a block load over one of the serial interfaces. The Setups are stored in an onboard eeprom array. After a setups block load, the device 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 5.1, page 24 for a list of all Setups.

Block length issues: The setups block is 247 bytes long to accommodate 48 keys. This can be a burden on smaller host controllers with limited memory. In larger quantities the devices can be procured with the setups block preprogrammed from Quantum. 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 value translation. The Setups Block Summary on page 26 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's 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 also is 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', i.e. 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, page 26.

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 Positive Threshold - PTHR

The positive threshold is used to provide a mechanism for recalibration of the reference point when a key's 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 in using the Setup process on a per-key basis.

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)

5.3 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, else false detections and sensitivity shifts can occur.

Drift compensation (Figure 5-1) 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 devices drift compensate using a slew-rate limited change to the reference level; the threshold and hysteresis values are slaved to this reference.

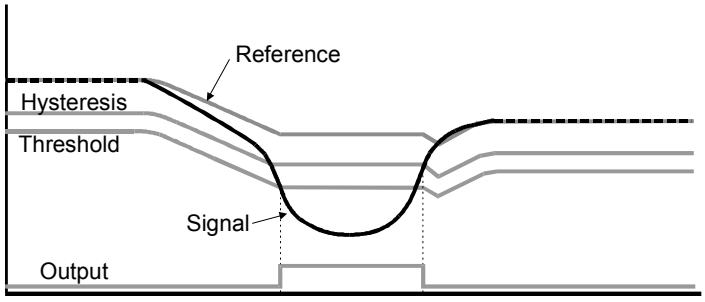
When a finger is sensed, the signal falls since the human body acts to absorb charge from the cross-coupling between X and Y lines. An isolated, untouched foreign object (a coin, or a water film) will cause the signal to rise very slightly due to an enhancement of coupling. This is contrary to the way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

Figure 5-1 Thresholds and Drift Compensation



sensor has already made full allowance for, could suddenly be removed leaving the sensor with an artificially suppressed reference level and thus become insensitive to touch. In this 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 Typical values: 9 to 11
(2 to 3.3 seconds per count of drift compensation)
NDRIFT Default value: 10
(2.5s / count of drift compensation)
PDRIFT Typical values: 3 to 5
(0.4 to 0.8 seconds per count of drift compensation; translation via LUT, page 26)
PDRIFT Default value: 4
(0.6s / count of drift compensation)

5.4 Detect Integrators - NDIL, FDIL

NDIL is used to enable keys and to provide signal filtering. To enable a key, its NDIL parameter should be non-zero (ie NDIL=0 disables a key).

To suppress false detections caused by spurious events like electrical noise, the device incorporates a 'detection integrator' or DI counter mechanism that acts to confirm a detection by consensus (all detections in sequence must agree). The DI mechanism counts sequential detections of a key that appears to be touched, after each burst for the key. For a key to be declared touched, the DI mechanism must count to completion without even one detection failure.

The DI mechanism uses two counters. The first is the 'fast DI' counter FDIL. When a key's 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. When fast-burst is entered the QT device locks onto the key and repeats the acquire burst until the fast-DI counter reaches FDIL, or, the detection fails beforehand. After this the device resumes normal key scanning 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 = 5 and NDIL = 2, the total detection confirmations required is 10, even though the device only 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; the DI 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 device 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 $\geq$ 2, then the fast-DI counter also operates in addition to the NDIL counter.

If Signal $\leq$ NThr: The fast-DI counter is incremented towards FDIL due to touch.

If Signal >NThr then the fast-DI counter is cleared 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.

NDIL Typical values: 2, 3
NDIL Default value: 2
FDIL Typical values: 4 to 6
FDIL Default value: 5

5.5 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 will be automatically recalibrated. After a recalibration has taken place, the affected key will once again function 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 will never auto-recalibrate 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 0.5s increments. Thus if NRD =120, the timeout value will actually be 60 seconds. 255 is not a legal number to use.

NRD Typical values: 20 to 60 (10s to 30s)
NRD Default value: 20 (10s)
NRD Range: 0..254 (∞, 0.5s .. 127s)

5.6 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 auto-recalibration, 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 (e.g. 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 will once again function normally. PRD is set on a per-key basis.

The functioning of the PRD setting is determined by an offset to a lookup table, found on page 26. The values of time can range from 0.1s to 25s. Setting the parameter to 0 will disable the feature.

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

5.7 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 key. Each QT process is simply the pulsing of an X line once, with a corresponding Y line enabled to capture the resulting charge passed through the key's capacitance Cx.

QT60xx6 devices use 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 5.4.

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 of each key can be set individually, allowing for direct digital control over the signal gains of each key individually.

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: 16, 32, 48, 64

5.8 Adjacent Key Suppression - AKS

These devices incorporate adjacent key suppression ('AKS' - patent pending) that can be selected on a per-key basis. AKS 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 works for keys that are AKS-enabled anywhere in the matrix and is not restricted to physically adjacent keys; the device 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.

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. 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)

5.9 Oscilloscope Sync - SSYNC

Pin 43 (S_Sync) can output a positive pulse oscilloscope sync that brackets the burst of a selected key. More than one burst can output a sync pulse as determined by the Setups parameter SSYNC for each key.

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

This function is supported in Quantum's QmBtn PC software via a checkbox.

SSYNC Default value: 0 (Off)

5.10 Negative Hysteresis - NHYST

The devices employ 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 will drop 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 used only 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 'sticking 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%)

5.11 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 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 2.8).

This setup lets the designer trade off one requirement for with the other.

DWELL typical value: 1 (187.5ns)
DWELL default value: 1 (187.5ns)
DWELL possible values: 0, 1, 2 (125, 187.5, 312.5ns)

5.12 Mains Sync - MSYNC

The MSync feature uses the WS pin. The Sleep and Sync features can be used simultaneously; the part can be put into Sleep mode, but awakened by a mains sync signal at the desired time.

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 50Hz or 60Hz 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; the device waits for the sync signal for up to 100ms after the end of a preceding full matrix scan, then when a negative sync edge is received, the matrix is scanned in its entirety again.

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. The device will sync to the falling edge.

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 (i.e. 50/60Hz) electric fields. Quantum's QmBtn software can show such noise effects on signals, and will hence assist in determining the need to make use of this feature.

If the sync feature is enabled but no sync signal exists, the sensor will continue to operate but with a delay of 100ms from the end of one scan to the start of the next, and hence will have a slow response time. A failed Sync signal (one exceeding a 100ms period) will cause an error flag (see commands 0x05, 0x06).

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

5.13 Burst Spacing - BS

The interval of time from the start of one burst to the start of the next is known as the *burst spacing*. This is an alterable parameter which affects all keys. The burst spacing can be viewed as a scheduled timeslot in which a burst occurs. This approach results in an orderly and predictable sequencing of key scanning with predictable response times.

Shorter spacings result in a faster response time to touch; longer spacings permit higher burst lengths and longer conversion times but slow down response time.

An automatic setting is also available that performs a 'best fit' timeslot determination for each key's acquisition burst. The fit is determined on power-up each time and is fixed thereafter until reset again.

See Table 5.4 (page 26) for possible values.

BS Default value: 0 (Automatic)
BS Possible range: 0..11 (Auto, 500µs .. 3ms)

5.14 Serial Rate - SR

The possible Baud rates are shown in Table 5.4. 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 4Mhz.

The Baud rate can be adjusted to one of 5 values from 9600 to 115.2K baud.

SR Default value: 0 (9600 Baud)

5.15 Lower Signal Limit - LSL

This Setup determines the lowest acceptable value of signal level for all keys. If any key's reference level falls below this value, the device declares an error condition in the key status bits (Sections 4.5, 4.6, 4.8, 4.11).

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 of the setups table. The low order byte should be sent first.

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

5.16 LED / Alert Output - LED

Refer also to Table 5.2 page 25 for details.

Pin 40 is designed to drive a low-current LED or to be used as a status and error signaling mechanism for the host controller, primarily for FMEA purposes.

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 device and the host. The LED 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 device for key activations.

Note that LED polarity can be reversed in the setups byte.

Table 5.2, on page 25 shows the possible internal conditions that can cause the LED pin to go active. The various items in the table are logical-OR'd together.

The LED pin can even be used as a watchdog for the host, to reset it should the host fail to send regular transmissions to the QT (bit 0 of LED 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 that the LED state will be preserved during sleep.

LED Default value: 0x6c
(see Table 5.2, page 25 for details)

5.17 Host CRC - HCRC

The setups block terminates with a 16-bit CRC, HCRC, of the entire block. The formulae for calculating this CRC and the 8-bit CRC also used in the device are shown in Section 7. The low order byte should be sent first.

Table 5.1 Setups Block

Setups data is sent from the host to the QT in a block of hex data. The block can only be loaded in Setups mode following two 0x 01 commands (page 14). The devices this datasheet pertain to have the same block length. Refer also to Table 5.4, page 26 for further details, and all of Section 5.

Item #	Byte	Parameter	Symbol	Bytes	Valid range	Bits	Key Scope	Default Value	Description	Page
1	0	Neg thresh Pos Thresh	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	20 20
2	48	Neg Drift Comp Pos Drift Comp	NDRIFT PDRIFT	48	NDRIFT = 0..15 PDRIFT = 0..15	4 4	1 1	10 4	Lower nibble = Neg Drift comp - Via LUT Upper nibble = Pos Drift comp - Via LUT	20
3	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 does not work)	21
4	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	21
5	192	Pos recal delay Burst Length AKS Scope Sync	PRD BL AKS SSYNC	48	PRD = 0..15 BL = 0..3 AKS = 0, 1 SSYNC = 0, 1	4 2 1 1	1 1 1 1	6 2 0 0	Lower nibble = PRD, via LUT, default = 6 (1 sec) Bits 5, 4: = BL, via LUT, default = 48 (setting =2) Bit 6 = AKS, 1 - enabled Bit 7 = Scope sync, 1 = enabled	21 22 22 22
6	240	Neg Hysteresis Dwell Time Mains Sync	NHYST DWELL MSYNC	1	NHYST = 0..3 DWELL = 0..2 MSYNC = 0, 1	4 2 1	48 48 48	1 1 0	Lower nibble = Neg hysteresis, all keys; default = 12.5% Bits 5, 4 = Dwell time, 3 values via LUT, default = 187.5ns Bits 6 = Mains sync, negative edge, 1 = enabled; default = 0 (off)	22 22 22
7	241	Burst spacing Serial rate	BS SR	1	BS = 0..11 SR = 0..4	4 4	48 device	0 0	Lower nibble = burst spacing; default = 0 (automatic) Upper nibble = serial rate via LUT - 9600, 19.2K, 38.4K, 57.6K, 115.2K (UART)	23 23
8	242	Lower signal Limit	LSL	2	0..2047	16	48	100	Lower limit of acceptable signal; below this value, device declares an error. The low order byte should be sent first.	23
9	244	LED Function	LED	1	0..255	8	device	0x6c	Controls what the LED does; see table, below.	23
10	245	Host CRC	HCRC	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, page 29.	23
		Block length		**247**						

CRC Note: *A CRC calculator for Windows is available free of charge from Quantum Research on request.*

Table 5.2 LED Function Control Byte Bits

See also page 23. The LED pin can be used to indicate a variety of things in combination. The LED control byte controls which states make the LED pin active. The active state can also be set either high or low by changing bit 7 in this byte. One purpose for these functions is to provide an FMEA-compliant mechanism for fault detection via an alternative path to the serial comms path. Another is to provide an interrupt signal to a host controller to reduce the amount of required comms traffic. Another reason is to simply light an LED on a sensing fault, a keypress, or a comms failure for diagnostic purposes.

Bit	1 =	0 =	Default
7	LED pin is active high polarity	LED pin is active low polarity	0
6	Active on any key error: (cal, cal failed, low sig)	Key errors have no effect	1
5	Active on any keypress	Not active on any keypress	1
4	Active while in sleep	Inactive on Sleep	0
3	Active on eeprom error	Inactive on eeprom error	1
2	Active on Mains sync error	Inactive on Mains sync error	1
1	Active on comms error (unrecognized command): LED pin is set active on error, inactive again when 'get last cmd' is called, or part is reset.	Inactive on comms error	0
0	Host watchdog. Active if no host comms within any 2s period. Host reset pulse length is 150ms. The host watchdog is not enabled until the first valid cmd is received from the host.	Communications unmonitored	0

Table 5.3 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. The following table shows the byte and bit order of the keys. The table contains 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, i.e. up to 8 keys. Thus, key 0 is at location X0,Y0 and key 29 is at location X5,Y3. .

		Bit (X line)							
		7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Byte (Y line)	**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

Note: Byte 0 is returned first.

Table 5.4 Setups Block Summary

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

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

Index Number	Parameter															
	NTHR counts	**PTHR counts**	**NDRIFT secs**	**PDRIFT secs**	**NDIL counts**	**FDIL counts**	**NRD secs**	**PRD secs**	**BL pulses**	**AKS**	**Scope Sync**	**NHYST**	**DWELL**	**MSYNC**	**BS**	**UART**
	Per key	Per key	Per key	Per key	Per key	Per key	Per key	Per key	Per key	Per key	Per key	Global	Global	Global	Global	Global
0	4	4	0.1	0.1	Key off	unused	0 (Infinite)	0 (Infinite)	16	**- Off -**	**- Off -**	6.25%	125ns	**- Off -**	**- Auto -**	**-9,600-**
1	5	5	0.2	0.2	1	1	0.5 .. 127s	0.1	32	On	On	**-12.5%-**	**-187.5ns-**	On	500µs	19,200
2	6	**- 6 -**	0.3	0.3	**- 2 -**	2	Default= 10s	0.2	**- 48 -**			25%	312.5ns		750µs	38,400
3	7	7	0.4	0.4	3	3		0.3	64			50%			1,000µs	57,600
4	8	8	0.6	**- 0.6 -**	4	4		0.5							1,250µs	115,200
5	9	9	0.8	0.8	5	**- 5 -**		0.7							1,500µs	
6	**- 10 -**	10	1	1	6	6		**- 1 -**							1,750µs	
7	11	11	1.2	1.2	7	7		1.5							2,000µs	
8	12	12	1.5	1.5	8	8		2							2,250µs	
9	13	13	2	2	9	9		3.2							2,500µs	
10	14	14	**- 2.5 -**	2.5	10	10		4.5							2,750µs	
11	15	15	3.3	3.3	11	11		6							3,000µs	
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								

6 Specifications

6.1 Absolute Maximum Electrical Specifications

Operating temp. . . . . -40°C ~ +105°C
Storage temp. . . . . -55°C ~ +125°C
V_{DD}. . . . . -0.5 ~ +5.5V
Max continuous pin current, any control or drive pin . . . . . ±10mA
Short circuit duration to ground, any pin . . . . . infinite
Short circuit duration to V_{DD}, any pin. . . . . infinite
Voltage forced onto any pin . . . . . -0.6V ~ (Vdd + 0.6) Volts
Frequency of operation. 17MHz
Eeprom setups maximum writes. . . . . 100,000 write cycles

6.2 Recommended operating conditions

V_{DD}. . . . . +4.75 ~ 5.25V
Supply ripple+noise. . . . . 5mV p-p max
Cx transverse load capacitance per key. . . . . 0 ~ 20pF
Fosc oscillator frequency. . . . . 16MHz +/-2%

6.3 DC Specifications

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

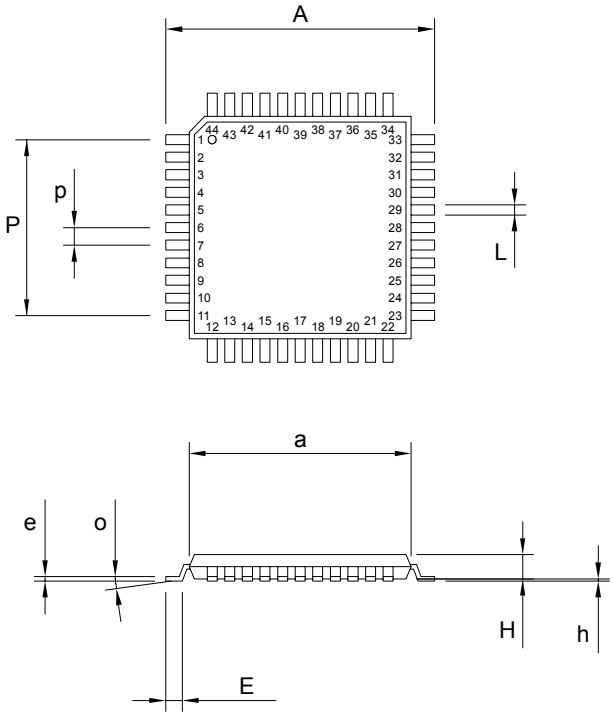
Parameter	Description	Min	Typ	Max	Units	Notes
Iddr	Supply current, running			25	mA	
Idds	Supply current, sleep		20		µA	
Vr	Vdd internal reset voltage		4		V	
Vil	Low input logic level			0.8	V	
Vhl	High input logic level	2.2			V	
Vol	Low output voltage			0.6	V	4mA sink
Voh	High output voltage	Vdd-0.7			V	1mA source
Iil	Input leakage current			±1	µA	
Ar	Acquisition resolution		9	11	bits	
Rp	Internal pullup resistors	20		50	kW	DRDY, /SS, TX pins
Rrst	Internal /RST pullup resistor	30		60	kW	

6.4 Timing Specifications

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

Parameter	Description	Min	Typ	Max	Units	Notes
T_{BS}	Burst spacing	500		3,000	µs	Adjustable parameter via Setups
Tpc	Pulse spacing, average @ dwell = 125ns @ dwell = 187.5ns @ dwell = 312.5ns	 1.80 1.90 2.00		 2.75 2.80 2.92	µs	
Fqt	Burst frequency range @ dwell = 125ns @ dwell = 187.5ns @ dwell = 312.5ns	 364 357 342		 556 526 500	kHz	Spread spectrum range
S1	↓ /SS to first ↓ CLK edge	125			ns	SPI parameter controlled by host
S2	↓ CLK to valid MISO			20	ns	SPI parameter controlled by QT
S3	Last ↑ CLK to ↑ /SS	25			ns	SPI parameter controlled by host
S4	↑ /SS to 3-state MISO			20	ns	SPI parameter controlled by QT
S5	↑ /SS to falling DRDY			20	µs	SPI parameter controlled by QT
S6	DRDY low pulse width	1			µs	SPI parameter controlled by QT
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



Package Type: 44 Pin TQFP						
SYMBOL	Millimeters			Inches		
	Min	Max	Notes	Min	Max	Notes
a	9.90	10.10	SQ	0.386	0.394	SQ
A	11.75	12.21	SQ	0.458	0.478	SQ
e	0.09	0.20		0.003	0.008	
E	0.45	0.75		0.018	0.030	
h	0.05	0.15		0.002	0.006	
H	-	1.20		-	0.047	
L	0.30	0.45		0.012	0.018	
p	0.80	0.80	BSC	0.031	0.031	BSC
P	8.00	8.00	BSC	0.315	0.315	BSC
o	0	7		0	7	

6.6 Marking

T_A	TQFP Part Number	Keys	Marking
-40°C to +105°C	QT60326-AS-G	32	QT60326-AG
-40°C to +105°C	QT60486-AS-G	48	QT60486-AG

7 Appendix

7.1 8-Bit CRC Software C Algorithm

```
// 8 bits crc calculation. Initial crc entry value must be 0.
// polynomial = X8 + X5 + X4 + 1
// data is an 8 bit number;  crc is an unsigned 8 bit number
// repeat this function for each data block byte, folding the result
//   back into the call parameter crc

unsigned char eight_bit_crc(unsigned char crc, unsigned char data)
{  unsigned char index;  // shift counter
   unsigned char fb;     // intermediate test bit

   index = 8;            // initialize the shift counter
   do                    // loop 8 times
   {  fb = (crc ^ data) & 0x01;
      data >>= 1;
      crc >>= 1;

      if(fb)
      {  crc ^= 0x8c;
      }
   } while(--index);
return crc;
}
```

7.2 16-Bit CRC Software C Algorithm

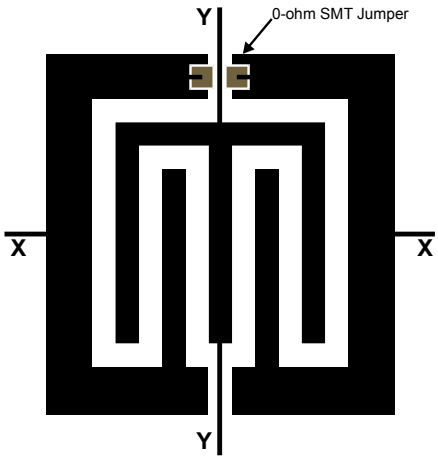
```
// 16 bits crc calculation. Initial crc entry value must be 0.
// The message is not augmented with 'zero' bits.
// polynomial = X16 + X12 + X5 + 1
// data is an 8 bit number, unsigned
// crc is a 16 bit number, unsigned
// repeat this function for each data block byte, folding the result
//   back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

7.3 1-Sided Key Layout

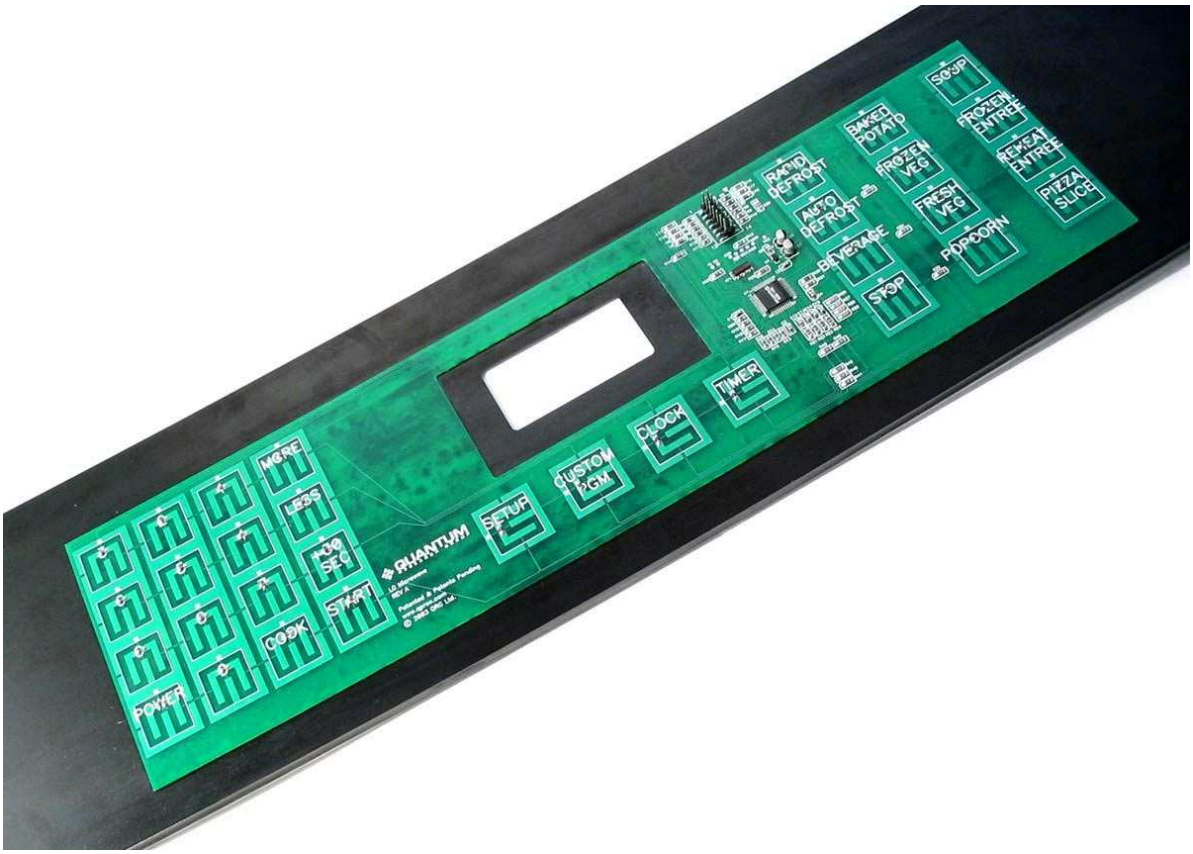
This key design can be made on a 1-sided SMT PCB. A single 0-ohm jumper allows the wiring to be done on a single side with full pass-through of X and Y traces to allow matrix connections to be made across a large number of keys. Key size, shape, and number of interleavings can be varied substantially from this drawing. The below drawing shows 6 interleave white spaces; only a double interleave is required in the case of smaller keys.

The PCB is bonded to a panel on its underside, and the fields fire through the PCB, adhesive, and panel in that sequence. This results in a very low cost design.



7.4 PCB Layout

Shown is an example 32-key PCB layout using inexpensive 1-sided CEM-1 or FR-1 PCB laminate. The key layouts follow the design rules of Figure 2-6 (page 5) and as shown above.



NOTES

Corporate Headquarters

1 Mitchell Point
Ensign Way, Hamble SO31 4RF
Great Britain
Tel: +44 (0)23 8056 5600 Fax: +44 (0)23 8045 3939

admin@qprox.com
www.qprox.com

North America

651 Holiday Drive Bldg. 5 / 300
Pittsburgh, PA 15220 USA
Tel: 412-391-7367 Fax: 412-291-1015

Development Team: Dr. Tim Ingersoll, Samuel Brunet, Hal Philipp