



Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!



Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832

Email & Skype: info@chipsmall.com Web: www.chipsmall.com

Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China



1 Device Overview

1.1 Introduction

Freescale Semiconductor's MPR084 proximity capacitive touch sensor controller is one of a family of products designed to detect the state of capacitive touch pads. The MPR084 offers designers a cost-efficient alternative to mechanical keys for control panel applications.

The MPR084 uses an I²C interface to communicate with the host which configures the operation and an interrupt to advise the host of status changes. The MPR084 includes a piezo sounder drive which provides audible feedback to simulate mechanical key clicks. The MPR08X family has several implementations to use in your design including control panels and switch replacements. The MPR084 controls individual touch pads. Other members of the MPR08X family are well suited for other application interface situations such as individual touch pads or rotary/touch pad combinations.

Freescale offers a broad portfolio of proximity sensors for products ranging from appliance control panels to portable electronics. Target markets include consumer, appliance, industrial, medical and computer peripherals.

1.1.1 Devices in the MPR08X series

The MPR08X series of Proximity Capacitive Touch Sensor Controllers allows for a wide range of applications and implementations. Each of the products in [Table 1](#) perform a different application specific task and are optimized for this specific functionality.

Table 1. MPR08X Family Overview

Product	Bus	Sounder	Rotary/Slider	Touch Pad Array
MPR083	I ² C	Yes	8-pads	—
MPR084	I ² C	Yes	—	8 keys

1.1.2 Internal Block Diagram

The MPR084 consists of primary functional blocks; Interrupt Controller, I²C Serial Interface, Sounder Controller, Configuration and Status registers, Touch Pad Decoder, Magnitude Comparator and Recalibrator, EMI Burst/Noise Rejection Filter, Capacitance Measurement Analog Front End. Each of these blocks will be described in detail in their respective sections.

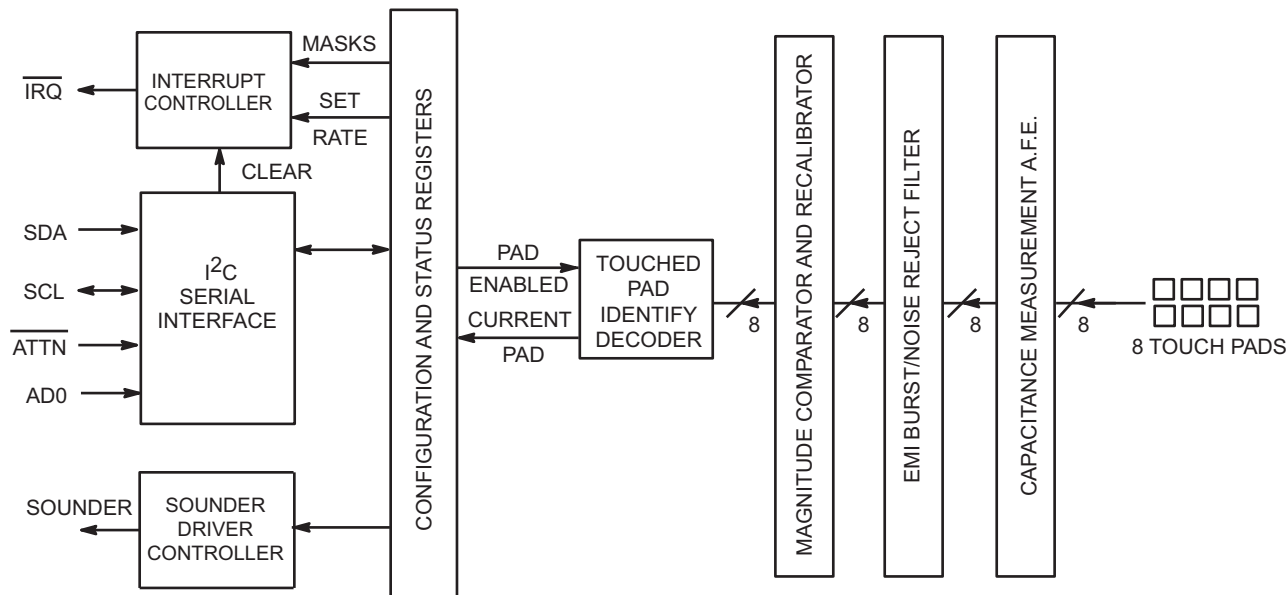


Figure 2. Functional Block Diagram

1.1.3 Terminology

The following terms are used to describe front panel interface and capacitive touch sensor technology throughout this document.

Table 2. Terminology

Term	Definition
Touch Sensor	A Touch Sensor is the combination of a Touch Sensor Controller and a connected conductive area referred to as an electrode.
Touch Sensor Controller	A Touch Sensor Controller is the intelligent part of a Touch Sensor which measures capacitance and differentiates between touched and untouched pads.
Key	A Key or Switch is a mechanical device that makes an electrical connection only when pressed.
Touch Pad	A Touch Pad is a type of capacitive sensor that is used for direct replacement of a Key. A capacitive touch sensor determines touch state by differentiating between high and low capacitances. When there is a change in the state this can be interpreted in the same way as a mechanical Key.
Solid Pad	A Solid or Full Pad is a type of touch pad where exactly one electrode is used
Split Pad	A Split Pad is a type of touch pad where more than one electrode is used. Split Pads are used to increase the total number of possible touch pads without increasing the electrical connections to the Touch Sensor Controller.
N-key Lockout	N-Key Lockout refers to the logic that determines how many keys can be simultaneously touched in a system. For example, 1-key lockout would only allow a single key to be touched before ignoring all future touches.
N-key rollover	N-Key Rollover refers to the logic that determines how many keys can be pressed in succession without releasing previous keys. For example, a system with 1-key lockout and 2-key rollover would allow 2-keys to be pressed in succession but would only report the second key once the first key was released.
I ² C	Inter-Integrated Circuit Communication

2 External Signal Description

2.1 Device Pin Assignment

Table 3 shows the pin assignment for the MPR084. For a more detailed description of the functionality of each pin, refer to the appropriate chapter.

Table 3. Device Pin Assignment

Pin	Name	Function
1	ATTN	Attention Pin. Input, active low, when asserted sets the Configuration Register's $\overline{\text{DCE}}$ bit high allowing communication with the part.
2	IRQ	Interrupt Request Pin. Output, active-low, open-drain interrupt request signaling new events.
3	VDD	Positive Supply Voltage
4	VSS	Ground
5	SCL	I ² C Serial Clock
6	SDA	I ² C Serial Data
7	AD0	Address input. Low = slave address 0x5C. High = slave address 0x5D.
8	SOUNDER	Sounder driver output. Connect a piezo sounder from this output to ground. Output is push-pull
9 - 16	E1, E2, E3, E4, E5, E6, E7, E8	Touch Pad Electrode connections.
PAD	Exposed pad	Exposed pad on package underside (QFN only). Connect to VSS.

The two packages available for the MPR084 are a 5x5mm 16 pin QFN and a 4x5mm 16 pin TSSOP. Both of the packages and their respective pinouts are shown in Figure 3.

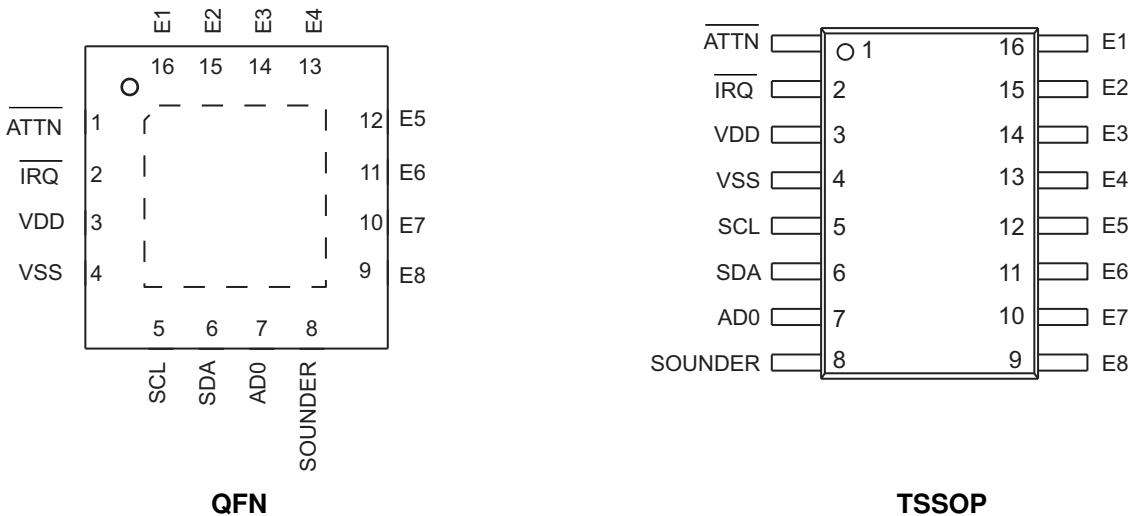


Figure 3. Package Pinouts

2.2 Recommended System Connections

The MPR084 Capacitive Touch Sensor Controller requires ten external passive components. When connecting the MPR084 in a touch sensor system, the electrode lines must have pull-up resistors. The recommended value for these pull-ups is 780k Ω . Some electrode arrays will require higher or lower values depending on the application.

In addition to the 8 resistors a bypass capacitor of 1 μ F should always be used between the VDD and VSS lines and a 4.7 k Ω pull-up resistor should be included on the IRQ.

The remaining 5 connections are SCL, SDA, $\overline{\text{IRQ}}$, $\overline{\text{ATTN}}$, and SOUNDER. Depending on the specific application, each of these control lines can be used by connecting them to a host system. In the most minimal system, the SCL and SDA must be connected to a master I²C interface to communicate with the MPR084. All of the connections for the MPR084 are shown by the schematic in [Figure 4](#).

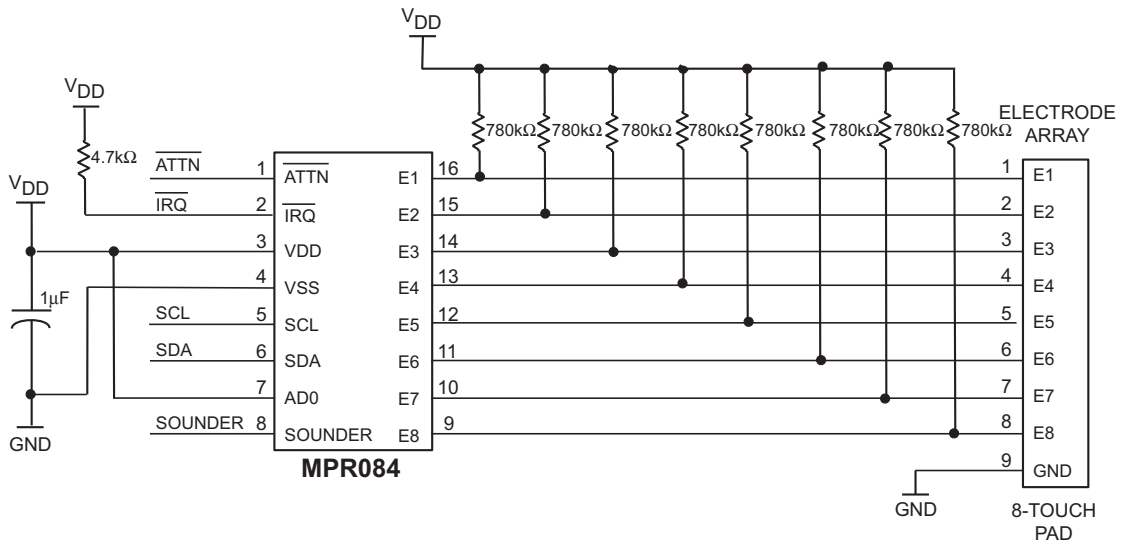


Figure 4. Recommended System Connections Schematic

Note that in this configuration the AD0 address line is tied high thus the slave address of the MPR084 0x5D. Alternatively the address line can be pulled low if the host system needs the MPR084 to be on address 0x5C. This functionality can also be used to incorporate two MPR084 devices in the same system.

2.3 Serial Interface

The MPR084 uses an I²C Serial Interface. The I²C protocol implementation and the specifics of communicating with the Touch Sensor Controller are detailed in the following sections.

2.3.1 Serial-Addressing

The MPR084 operates as a slave that sends and receives data through an I²C 2-wire interface. The interface uses a serial data line (SDA) and a serial clock line (SCL) to achieve bi-directional communication between master(s) and slave(s). A master (typically a microcontroller) initiates all data transfers to and from the MPR084, and generates the SCL clock that synchronizes the data transfer.

The MPR084 SDA line operates as both an input and an open-drain output. A pull-up resistor, typically 4.7kΩ, is required on SDA. The MPR084 SCL line operates only as an input. A pull-up resistor, typically 4.7kΩ, is required on SCL if there are multiple masters on the 2-wire interface, or if the master in a single-master system has an open-drain SCL output.

Each transmission consists of a START condition ([Figure 5](#)) sent by a master, followed by the MPR084's 7-bit slave address plus R/W bit, a register address byte, one or more data bytes, and finally a STOP condition.

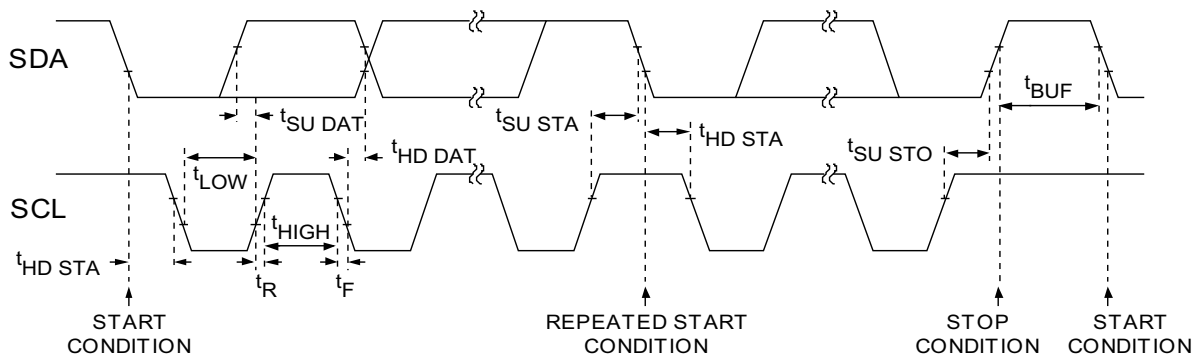


Figure 5. Wire Serial Interface Timing Details

2.3.2 Start and Stop Conditions

Both SCL and SDA remain high when the interface is not busy. A master signals the beginning of a transmission with a START (S) condition by transitioning SDA from high to low while SCL is high. When the master has finished communicating with the slave, it issues a STOP (P) condition by transitioning SDA from low to high while SCL is high. The bus is then free for another transmission.

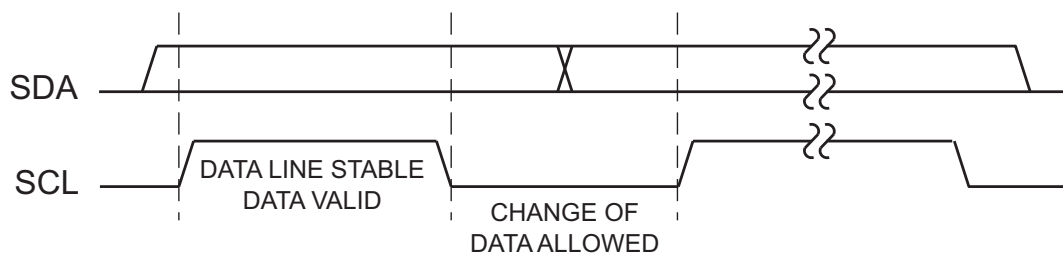


Figure 6. Start and Stop Conditions

2.3.3 Bit Transfer

One data bit is transferred during each clock pulse (Figure 7). The data on SDA must remain stable while SCL is high.

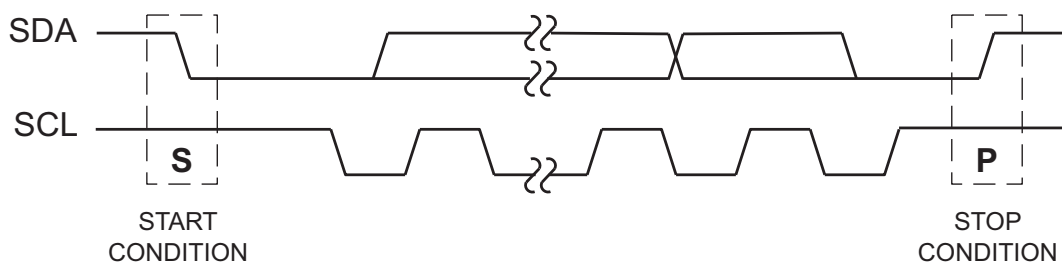


Figure 7. Bit Transfer

2.3.4 Acknowledge

The acknowledge bit is a clocked 9th bit (Figure 8) which the recipient uses to handshake receipt of each byte of data. Thus each byte transferred effectively requires 9 bits. The master generates the 9th clock pulse, and the recipient pulls down SDA during the acknowledge clock pulse, such that the SDA line is stable low during the high period of the clock pulse. When the master is transmitting to the MPR084, the MPR084 generates the acknowledge bit because the MPR084 is the recipient. When the MPR084 is transmitting to the master, the master generates the acknowledge bit because the master is the recipient.

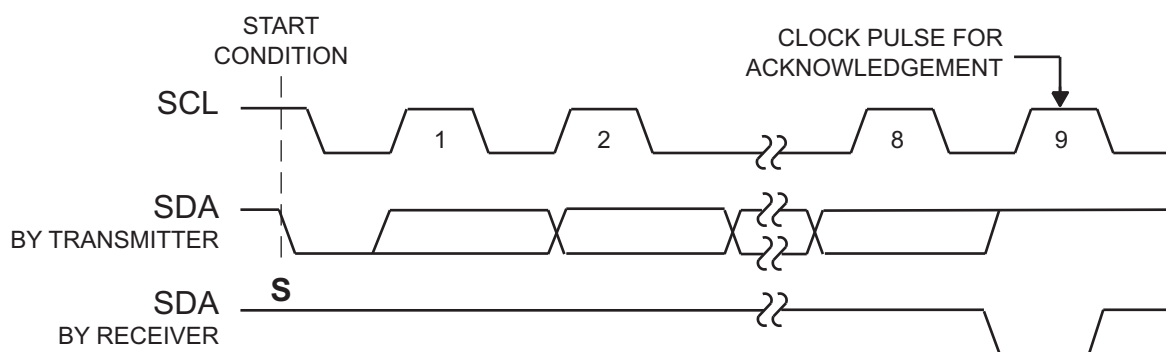


Figure 8. Acknowledge

2.3.5 The Slave Address

The MPR084 has a 7-bit long slave address (Figure 9). The bit following the 7-bit slave address (bit eight) is the R/W bit, which is low for a write command and high for a read command.

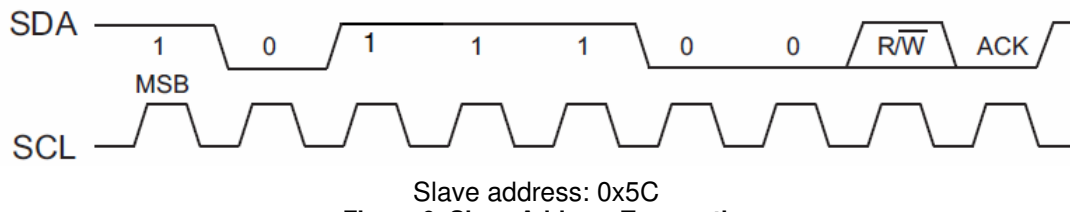


Figure 9. Slave Address Transaction

The MPR084 monitors the bus continuously, waiting for a START condition followed by its slave address. When a MPR084 recognizes its slave address, it acknowledges and is then ready for continued communication.

2.3.6 Message Format for Writing the MPR084

A write to the MPR084 comprises the transmission of the MPR084’s keyscan slave address with the R/W bit set to 0, followed by at least one byte of information. The first byte of information is the command byte. The command byte determines which register of the MPR084 is to be written by the next byte, if received. If a STOP condition is detected after the command byte is received, then the MPR084 takes no further action (Figure 10) beyond storing the command byte. Any bytes received after the command byte are data bytes.

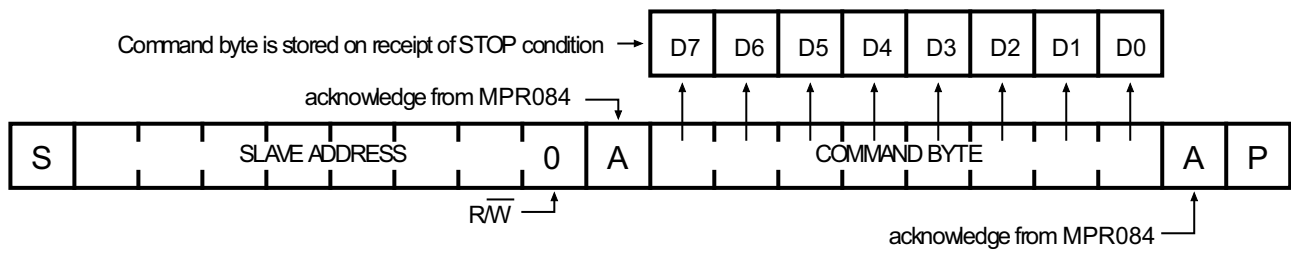


Figure 10. Command Byte Received

Any bytes received after the command byte are data bytes. The first data byte goes into the internal register of the MPR084 selected by the command byte (Figure 11).

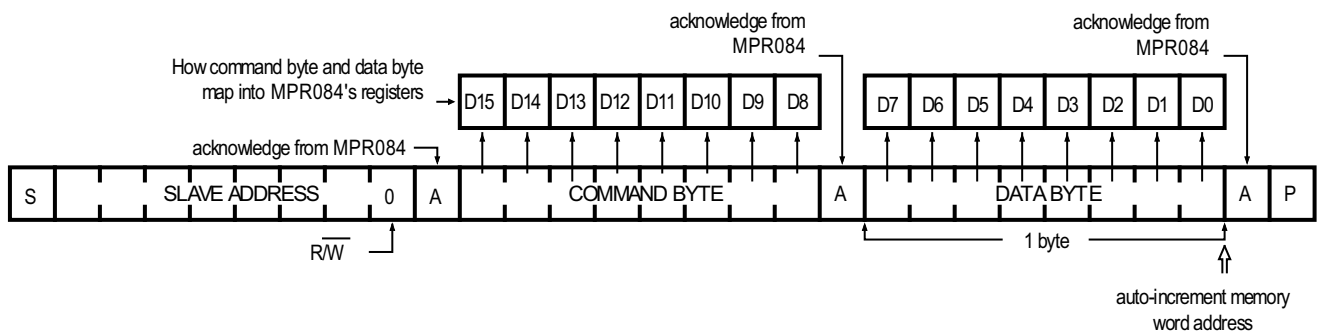


Figure 11. Command and Single Data Byte Received

If multiple data bytes are transmitted before a STOP condition is detected, these bytes are generally stored in subsequent MPR084 internal registers because the command byte address generally auto-increments (Section 2.4).

2.3.7 Message Format for Reading the MPR084

The MPR084 is read using the MPR084’s internally stored command byte as address pointer, the same way the stored command byte is used as address pointer for a write. The pointer generally auto-increments after each data byte is read using the same rules as for a write (Section 6.4.1). Thus, a read is initiated by first configuring the MPR084’s command byte by performing a write (Figure 12). The master can now read ‘n’ consecutive bytes from the MPR084, with the first data byte being read from the register addressed by the initialized command byte.

When performing read-after-write verification, remember to re-set the command byte's address because the stored command byte address will generally have been auto-incremented after the write (Section 2.4).

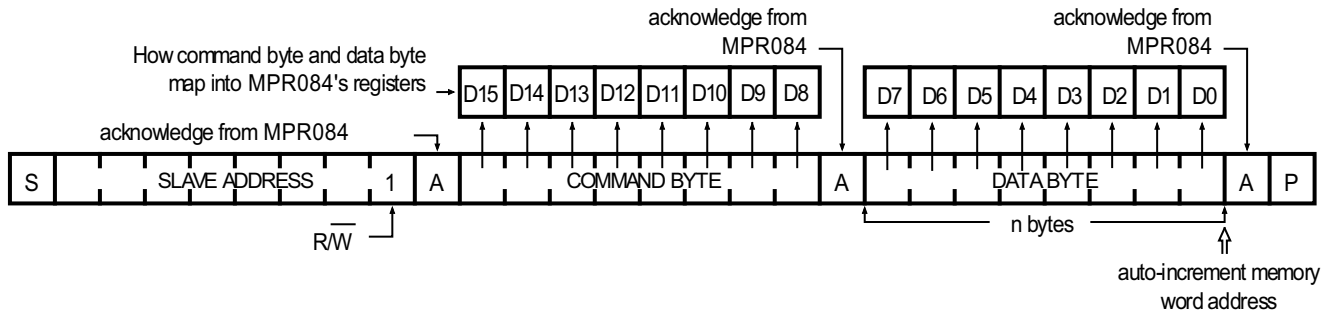


Figure 12. 'n' Data Bytes Received

2.3.8 Operation with Multiple Master

The application should use repeated starts to address the MPR084 to avoid bus confusion between I²C masters. On a I²C bus, once a master issues a start/repeated start condition, that master owns the bus until a stop condition occurs. If a master that does not own the bus attempts to take control of that bus, then improper addressing may occur. An address may always be rewritten to fix this problem. Follow I²C protocol for multiple master configurations.

2.3.9 Device Reset

The $\overline{\text{RST}}$ is an active-low software reset. This is implemented in the Configuration Register by activating the $\overline{\text{RST}}$ bit. When asserted, the device clears any transaction to or from the MPR084 on the serial interface and configures the internal registers to the same state as a power-up reset (Table 4). The MPR084 then waits for a START condition on the serial interface.

The sensor controller is capable of operating down to 1.8 V, however, in order for the sensor controller to exit reset and startup correctly the host system must initially provide 2.0 V to 3.6 V input to V_{DD} and then follow the process in Figure 13. This process is required in applications that require regulated operation in the 1.8 V to 2.0 V range. In the case that the application uses an unregulated battery, then the battery must initially provide at least 2.0 V to correctly power-up the sensor controller which limits battery selection to the 2.0 V to 3.6 V range.

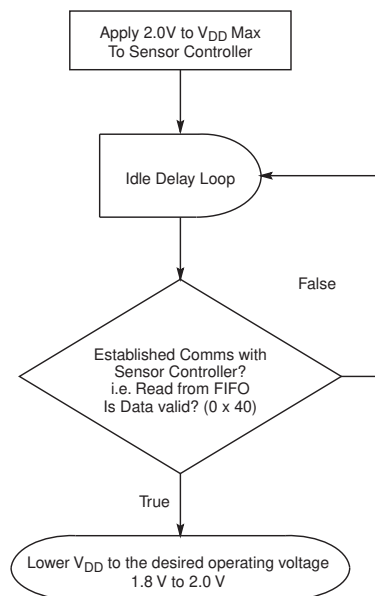


Figure 13. Low Voltage (1.8 V - 2.0 V) Power-up Sequence

3.6 Maximum Number of Touched Positions Register

The Maximum Number of Touched Positions Register adjusts the number of keys that can be concurrently reported as touched. The I²C slave address of the Maximum Number of Touched Positions Register is 0x0D.

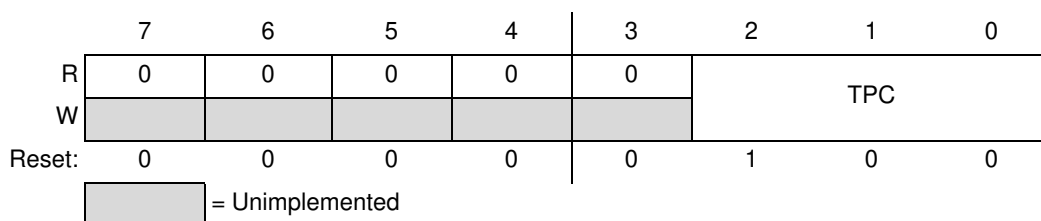


Figure 16. Maximum Number of Touched Positions Registers

Table 6. Maximum Number of Touched Positions Register Field Descriptions

Field	Description
2:0 TPC	Touched Positions Count – The Touched Positions Count selects or reports the number of simultaneously reported touches. 000 Encoding 0 – Sets the number of allowed touches to 0 ~ 111 Encoding 7 – Sets the number of allowed touches to 7

3.7 Electrode Channel Enable Mask Register

The Electrode Channel Mask Register adjusts to the number of keys that are scanned by the MPR084. The I²C slave address of the Electrode Channel Mask Register is 0x0C.

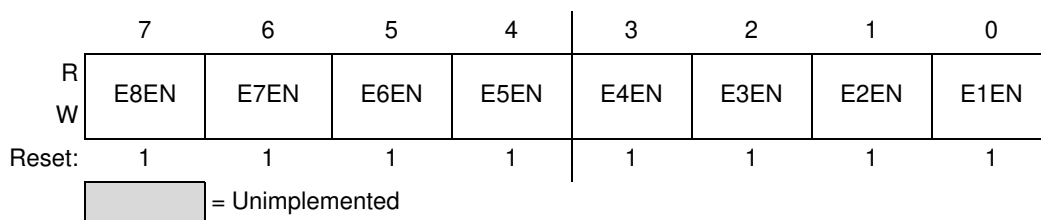


Figure 17. Electrode Channel Enable Mask Position Register

Table 7. Electrode Channel Enable Mask Register Field Descriptions

Field	Description
7 E8EN	Electrode 8 Enable – The Electrode 8 Enable bit enables or disables electrode number 8. 0 Electrode 8 Disable 1 Electrode 8 Enable
6 E7EN	Electrode 7 Enable – The Electrode 7 Enable bit enables or disables electrode number 7. 0 Electrode 7 Disable 1 Electrode 7 Enable
5 E6EN	Electrode 6 Enable – The Electrode 6 Enable bit enables or disables electrode number 6. 0 Electrode 6 Disable 1 Electrode 6 Enable
4 E5EN	Electrode 5 Enable – The Electrode 5 Enable bit enables or disables electrode number 5. 0 Electrode 5 Disable 1 Electrode 5 Enable

4.3 Run1 Mode

When in Run1 mode the sensor controller will run continuously. During Run1 all the modules are synchronized by the Master Tick Period. This value can be set by using the Master Tick Period Register as outlined in the following section.

While in this mode all functionality of the MPR084 is enabled; touch detection will occur, and I²C communication will be available. This mode is enabled by setting the Configuration Register’s RUNE and DCE bits high.

4.3.1 Master Tick Period Register

The Master Tick Period Register is used to set the master tick of this system. All parts of the system are synchronized to this counter. This register is overridden in all modes except for Run1. When not in Run1 mode, the value of this register is ignored and 8ms is used for the primary clock. The I²C slave address of the Master Tick Period Register is 0x0E.

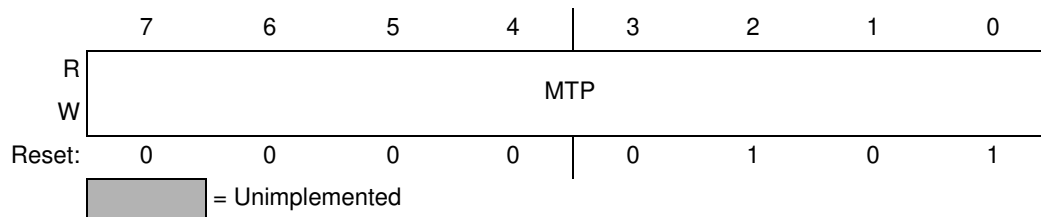


Figure 18. Master Tick Period Register

Table 10. Master Tick Period Register Field Descriptions

Field	Description
7:0 MTP	Master Tick Period – The Master Tick Period selects or reports the current value of the touch sensor controller’s primary clock. The resulting period must be in the range 5ms to 31ms. If the value is outside of this range the MTP will be set to 00011010. 00000000 Encoding 0 – Sets the primary clock multiplier to 5 ~ 00011010 Encoding 26 - Sets the primary clock multiplier to 31

4.4 Run2 Mode

When in Run2 mode the sensor controller will continue to scan the electrodes but a low power state will be enabled between each cycle. Because of this, any I²C communication that occurs, may or may not respond while the sensor is in this mode

If DCE is enabled the sensor controller transitions between low power and active states. During the active part of the cycle communication with the sensor controller is possible; however, Freescale always requires users to issue an ATTN signal prior to initiating communications. Accessing the I²C interface while DCE mode is enabled without sending an ATTN signal first is likely to produce invalid data.

This mode is enabled by setting the Configuration Register’s RUNE bit high and DCE bit low. The only way to exit this mode is to toggle the Attention Pin, refer to Section 4.7.

4.5 Stop1 Mode

When in Stop1 mode the sensor controller will not scan the electrodes. While capacitance sensing is disabled I²C communications will still be accepted and the sensor controller will maintain instantaneous response to all register requests. This is the only mode in which register values can be set.

This mode is enabled by setting the Configuration Register’s RUNE bit low and DCE bit high.

4.6 Stop2 Mode

When in Stop2 mode the sensor controller will not scan the electrodes or accept I²C communication. The MPR084 is off during this mode.

This mode is enabled by setting the Configuration Register’s RUNE bit low and DCE bit low. The only way to exit this mode is to toggle the Attention Pin, refer to Section 4.7.

5 Low Power Configuration

5.1 Introduction

The MPR084 features a Low Power mode that can reduce the power consumption into the microamps range. This feature can be used to both adjust the response time of the system, and change the conditions on which Low Power would be enabled.

5.2 Operation

This Low Power configuration is only active when the sensor controller is in Run2 mode. The Low Power mode decreases current consumption by increasing the response time of the MPR084. This increase is controlled through two factors.

During normal Run2 operation of the sensor controller the Max Response Time (MRT) is calculated by taking the product of the TASP and the primary clock. From Chapter 4 the primary clock is the (MTP + 5) ms. Since the sensor controller is in Run2, the primary clock is also multiplied by a factor of 8. The debounce rate of the MPR084 is 4 times the sample rate thus the MRT is represented by the following equation.

$$MRT_1 = \left(\frac{MTP + 5}{8} + 1 \right) \times TASP \times 4 \times 8ms \quad \text{Equation 1}$$

First, the Idle Interface Timeout (IIT) represents the total time the touch interface should remain idle before going into Low Power mode. This value can be calculated by taking the product of the ITP, TASP and primary clock (8ms) with a factor of 64. Thus the IIT is represented as follows:

$$MRT_2 = \left(\frac{MTP + 5}{8} + 1 \right) \times TASP \times SCD \times 4 \times 8ms \quad \text{Equation 2}$$

Second, the Max Response Time (MRT) represents the total time the touch interface should remain inactive before scanning the electrodes. This value can be calculated by taking the product of the SCD, TASP and primary clock (8ms) with a factor of 5. Thus the MRT is represented as follows:

$$ITT = \left(\frac{MTP + 5}{8} + 1 \right) \times TASP \times ITP \times 6 \times 8ms \quad \text{Equation 3}$$

When in Run2 mode, the sensor controller will initially scan the electrodes at the rate of MRT_1 . When scanning at MRT_1 and the touch interface remains idle for the IIT period then the scan period will change to MRT_2 . When scanning at MRT_2 and a touch is detected the scan rate will transition back to MRT_1 .

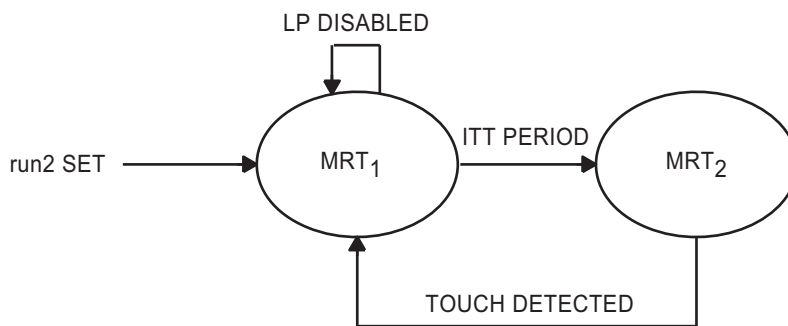


Figure 20. Low Power Scan Period Transition Diagram

5.3 Configuration

Low Power Configuration is achieved through setting two values; the Idle Timeout Period and the Sleep Cycle Duration. This functionality is described in the following section.

5.3.1 Low Power Configuration Register

The Low Power Configuration register is used to set both the Idle Timeout Period and Sleep Cycle Duration multiplication factors. The I²C slave address of the Low Power Configuration Register is 0x11.

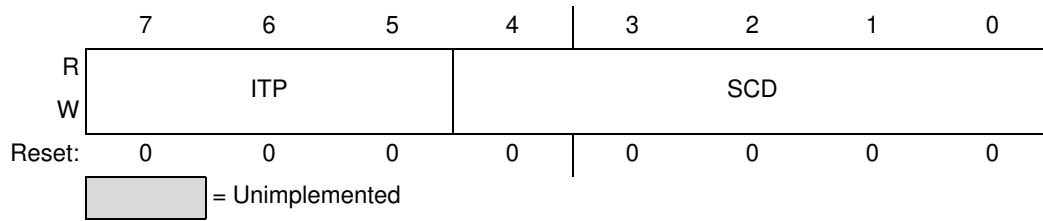


Figure 21. Low Power Configuration Register

Table 12. Low Power Configuration Register Field Descriptions

Field	Description
7:5 ITP	Idle Timeout Period – The Idle Timeout Period selects the amount to multiply the TASP (touch acquisition sample period) by to determine the idle interface timeout (IIT) period of the sensor controller. 000 Encoding 0 – Disables Low Power Mode 001 Encoding 1 – Sets the ITP multiplication factor to 1 ~ 111 Encoding 7 – Sets the ITP multiplication factor to 7
4:0 SCD	Sleep Cycle Duration – The Sleep Cycle Duration Field selects the amount to multiply the TASP (touch acquisition sample period) by to determine the Sleep period of the sensor controller. 00000 Encoding 0 – Disables Low Power Mode 00001 Encoding 1 – Sets the SCD multiplication factor to 1 ~ 11111 Encoding 31 – Sets the SCD multiplication factor to 31

6 Output Mechanisms

6.1 Introduction

The MPR084 has three primary methods for reporting data in addition to an \overline{IRQ} output that is described in Chapter 7. The three output systems are described in this section.

6.2 Instantaneous

The Instantaneous output shows the current status of the user interface. This information is displayed in terms of the current touched pad position that is touched. Only one touch can be shown at a time.

6.2.1 Touch Pad Status Register

The Touch Pad Status Register is a read only register for determining the current status of the touch pad. The I²C slave address of the Touch Pad Status Register is 0x02.

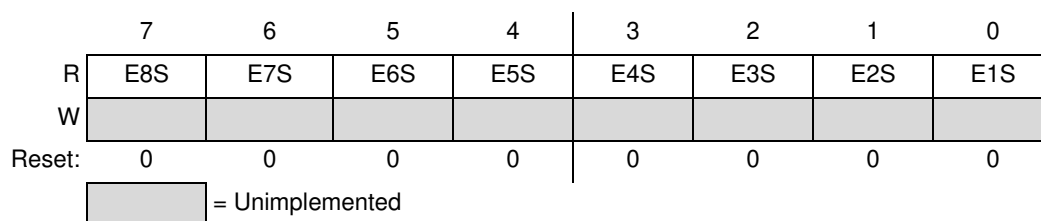


Figure 22. Touch Pad Status Register

Table 13. Touch Pad Status Register Field Descriptions

Field	Description
7 E8S	Electrode 8 Status – The Electrode 8 Status bit shows whether or not electrode 8 is touched. 0 Electrode 8 untouched 1 Electrode 8 touched
6 E7S	Electrode 7 Status – The Electrode 7 Status bit shows whether or not electrode 7 is touched. 0 Electrode 7 untouched 1 Electrode 7 touched
5 E6S	Electrode 6 Status – The Electrode 6 Status bit shows whether or not electrode 6 is touched. 0 Electrode 6 untouched 1 Electrode 6 touched
4 E5S	Electrode 5 Status – The Electrode 5 Status bit shows whether or not electrode 5 is touched. 0 Electrode 5 untouched 1 Electrode 5 touched
3 E4S	Electrode 4 Status – The Electrode 4 Status bit shows whether or not electrode 4 is touched. 0 Electrode 4 untouched 1 Electrode 4 touched
2 E3S	Electrode 3 Status – The Electrode 3 Status bit shows whether or not electrode 3 is touched. 0 Electrode 3 untouched 1 Electrode 3 touched
1 E2S	Electrode 2 Status – The Electrode 2 Status bit shows whether or not electrode 2 is touched. 0 Electrode 2 untouched 1 Electrode 2 touched
0 E1S	Electrode 1 Status – The Electrode 1 Status bit shows whether or not electrode 1 is touched. 0 Electrode 1 untouched 1 Electrode 1 touched

6.4.1 Fault Register

The Fault Register is a read only register that shows the fault number under the current sensor conditions. Any write to the Fault Register will clear the register, when in Stop mode. The Fault register cannot be cleared when the part is in a Run mode. The I²C slave address of the Fault Register is 0x01.

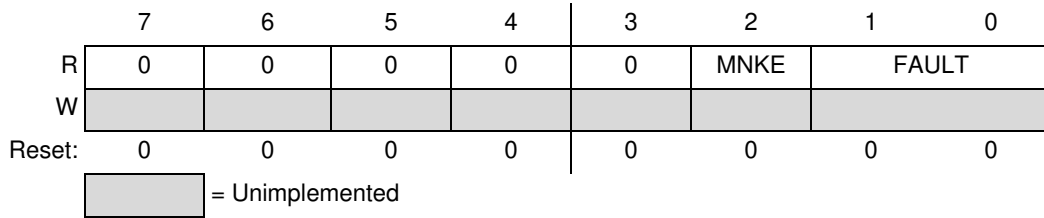


Figure 24. Fault Register

Table 15. Fault Register Field Descriptions

Field	Description
4 MNKE	Maximum Number of Keys Exceeded – The Maximum Number of Keys Exceeded status bit indicates whether or not more keys than allowed are currently being touched. 1 TPC Exceeded 0 TPC Not Exceeded
1:0 FAULT	Fault – The Fault code represents the currently asserted fault condition. 00 Encoding 0 – No fault detected 01 Encoding 1 – Short to VSS detected 10 Encoding 2 – Short to VDD detected

7 Interrupts

7.1 Introduction

The MPR084 has one interrupt output that is configured by registers and alerts the application when a touch or fault is detected. When running in Run2 or Stop2 mode where I²C communication is not available this feature alerts the user to sensor touches.

7.2 Condition for Interrupt

There are two cases that latch the Interrupt buffered data available or fault detected.

7.2.1 Buffered Data Available

The interrupt for Buffered Data Available will only trigger when the NDF (No Data Flag) transitions from high to low. This signifies that there is new data available in the buffer. The interrupt is deasserted on the first read/write of the FIFO Register and cannot be reasserted for buffered data until the FIFO is empty (either by reading all the data, or clearing the buffer).

7.2.2 Fault Detected

The interrupt for a Fault Detected condition is triggered any time the Fault condition in the Fault Register transitions from zero to non-zero. The interrupt is deasserted when the Fault Register is cleared (by writing to the Fault Register).

7.3 Settings

Interrupts are configured through I²C using the Configuration Register (Section 4.7). Two of the settings in this register will affect the interrupt functionality.

The Interrupt Enable (IRQEN) must be set high for the $\overline{\text{IRQ}}$ to be enabled. When low, all interrupts will be ignored, and the $\overline{\text{IRQ}}$ pin will never latch.

The Interrupt Rate (IRQR) sets the minimum delay between sequential triggered interrupts. The minimum interrupt period can be calculated by taking the product of the MCP (master clock period) and IRQR with a factor of 4. Thus, for the minimum setting an interrupt would be triggered no more often than 4 times the sensor scan rate.

$$\text{MinInterruptPeriod}(ms) = MCP \times IRQR \times 4 \quad \text{Equation 4}$$

If the MPR084 is using Run2, the minimum interrupt period would be represented by the following equation.

$$\text{MinInterruptPeriod}(ms) = \left(\frac{MTP + 5}{8} + 1 \right) \times 8 \times IRQR \times 4 \quad \text{Equation 5}$$

7.4 $\overline{\text{IRQ}}$ Pin

The $\overline{\text{IRQ}}$ pin is an open-drain, latching interrupt output which requires an external pull-up resistor. The pin will latch down based on the conditions in Section 6.2. The pin will reset when an I²C transmission reads/writes the appropriate register displaying information about the source of the interrupt. Thus if the source is buffered data available then a FIFO Buffer read/write will clear the $\overline{\text{IRQ}}$ pin. If the source is a fault detected then a write of the Fault Register will clear the pin.

7.4.1 $\overline{\text{IRQ}}$ Pin Timing

The `MinInterruptPeriod` is implemented as a hold off of $\overline{\text{IRQ}}$ latching per [Figure 25](#) and [Figure 26](#). In the first case the `MinInterruptPeriod` is longer than the interval between sequential interrupt source events, thus it delays the $\overline{\text{IRQ}}$ from latching until the `MinInterruptPeriod` has elapsed. In the second case the `MinInterruptPeriod` is shorter than the interval between sequential interrupt source events, thus the $\overline{\text{IRQ}}$ latches as it normally would without additional delay.

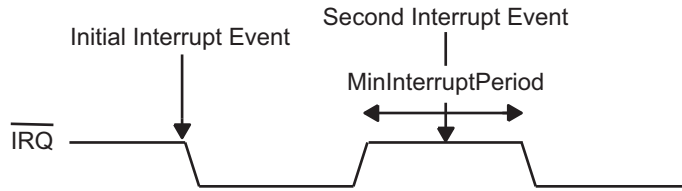


Figure 25. $\overline{\text{IRQ}}$ Timing Diagram - Case 1

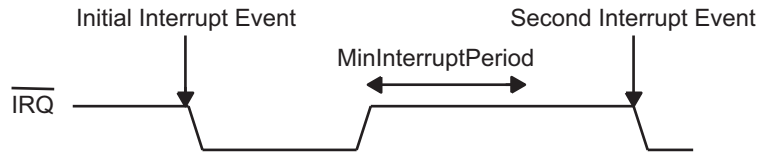


Figure 26. $\overline{\text{IRQ}}$ Timing Diagram - Case 2

8 Calibration

8.1 Introduction

The MPR084 is self-calibrating. This is done both at initial start-up of the device and during run time.

8.2 Initial Start-up Conditions

Initial calibration of the MPR084 occurs every time the device resets. The first key detection cycle is used as a baseline capacitance value for all remaining calculations. Thus, a touch is detected by taking the difference between this baseline value and the current capacitance on the electrode.

8.3 Auto-Calibration

The MPR084 has an auto-calibration feature. This is enabled through the Touch Pad Configuration Register (Section 3.4), by setting the ACE bit high. Auto calibration is done by two mechanisms. The basic auto-calibration will recalculate the baseline value after 6 sample periods. Thus the auto calibrate period can be calculate by multiplying the master clock period (in milliseconds) and the touch acquisition sample period with a factor of 64.

$$AutoCalibrationPeriod(ms) = MCP \times TASP \times 64 \quad \text{Equation 6}$$

If a touch is currently being detected the auto-calibration will not engage and calibration will be ignored. The device can also be calibrated when a key is being touched, this is controlled by stuck key detection.

8.4 Stuck Key Detection

The Stuck Key Detection system allows the application to specify the maximum amount of time a touch should be detected before it is calibrated into the baseline and the touch is ignored. This is controlled by setting the Stuck Key Timeout multiplication factor (SKT). The timeout period can be calculated by multiplying the SKT, master clock period (in ms) and touch acquisition sample period with a factor of 64.

$$AutoCalibrationPeriod(ms) = MCP \times TASP \times SKT \times 64 \quad \text{Equation 7}$$

When Stuck Key Detection is off a touched key will remain touched indefinitely and never be calibrated into the baseline value.

8.4.1 Stuck Key Timeout Register

The Stuck Key Timeout Register is used to determine the electrode scan period of the system. The I²C slave address of the Stuck Key Timeout Register is 0x12.

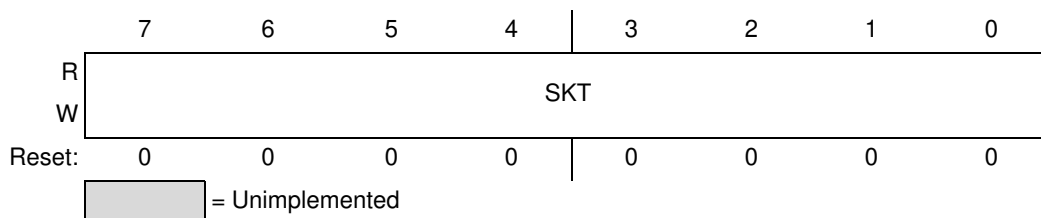


Figure 27. Stuck Key Timeout Register

Table 16. Stuck Key Timeout Register Field Descriptions

Field	Description
7:0 SKT	<p>Stuck Key Timeout – The Stuck Key Timeout field selects or reports the multiplication factor that is used to determine how often electrodes are calibrated while a touch is being detected.</p> <p>00000000 Encoding 0 – Turns off Stuck Key Detection</p> <p>00000001 Encoding 1 – Sets the SKT multiplication factor to 2</p> <p>~</p> <p>11111111 Encoding 255 – Sets the SKT multiplication factor to 256</p>

9 Sensitivity

9.1 Introduction

The MPR084 can operate in a variety of environments with a variety of different electrode patterns. Because of this it is necessary to adjust the relative sensitivity of the sensor controller. Usually this requires fine tuning in any final application.

There are many factors that must be taken into account, but much of the time this value is relative to the capacitance changes generated by a touch. Since capacitance is directly proportional to the dielectric constant of the material and the area of the pad, while inversely proportional to the distance between pads these are the primary factors.

$$C = \frac{ke_0A}{d} \quad \text{Equation 8}$$

As the relative capacitance rises the sensitivity setting of the MPR084 should be adjusted accordingly. Thus a very high sensitivity value represents a large A and a small d.

9.2 Adjusting the Sensitivity

The sensitivity of the MPR084 is adjusted by varying the Sensitivity Threshold Registers.

9.2.1 Sensitivity Threshold Registers

The Sensitivity Threshold registers all sensitivity of the MPR084 to be adjusted for any situation. The I²C slave address of the Sensitivity Threshold Registers is 0x04 - 0x0B.



Figure 28. Sensitivity Threshold Register Format

Table 17. Sensitivity Threshold Register Format Descriptions

Field	Description
7:0 ST	Sensitivity Threshold – The Sensitivity Threshold selects or reports the sensitivity setting of the Sensor Controller. The resulting value must be in the range 1 to 64 units. If the value is outside of this range the ST will be set to 00111111. 00000000 Encoding 0 – Sets the sensitivity to level 1 ~ 00111111 Encoding 63 – Sets the sensitivity to level 64