# CAP1188

# 8 Channel Capacitive Touch Sensor with 8 LED Drivers

## PRODUCT FEATURES

**Datasheet**

### General Description

The CAP1188, which incorporates SMSC's RightTouch® [1] technology, is a multiple channel Capacitive Touch sensor with multiple power LED drivers. It contains eight (8) individual capacitive touch sensor inputs with programmable sensitivity for use in touch sensor applications. Each sensor input automatically recalibrates to compensate for gradual environmental changes.

The CAP1188 also contains eight (8) LED drivers that offer full-on / off, variable rate blinking, dimness controls, and breathing. Each of the LED drivers may be linked to one of the sensor inputs to be actuated when a touch is detected. As well, each LED driver may be individually controlled via a host controller.

The CAP1188 includes Multiple Pattern Touch recognition that allows the user to select a specific set of buttons to be touched simultaneously. If this pattern is detected, then a status bit is set and an interrupt generated.

Additionally, the CAP1188 includes circuitry and support for enhanced sensor proximity detection.

The CAP1188 offers multiple power states operating at low quiescent currents. In the Standby state of operation, one or more capacitive touch sensor inputs are active and all LEDs may be used. If a touch is detected, it will wake the system using the WAKE/SPI_MOSI pin.

Deep Sleep is the lowest power state available, drawing 5uA (typical) of current. In this state, no sensor inputs are active. Driving the WAKE/SPI_MOSI pin or communications will wake the device.

1. SMSC, the SMSC logo and RightTouch are registered trademarks and the RightTouch logo is a trademark of Standard Microsystems Corporation ("SMSC").

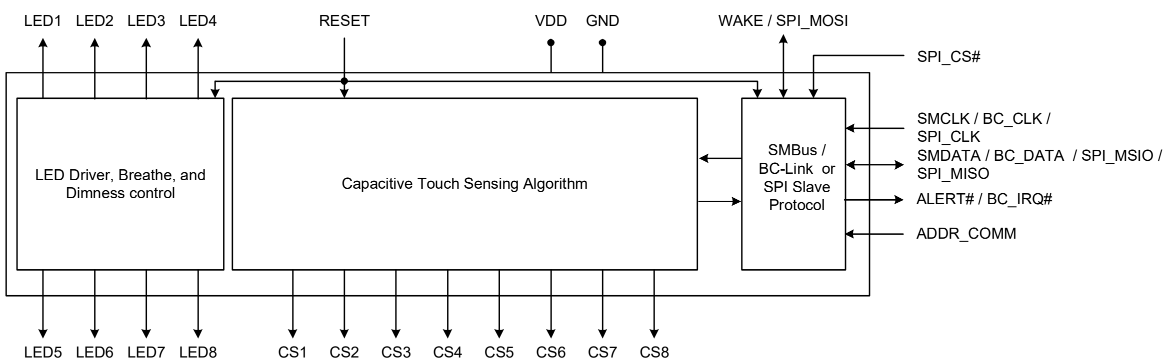
### Applications

- Desktop and Notebook PCs
- LCD Monitors
- Consumer Electronics
- Appliances

### Features

- Eight (8) Capacitive Touch Sensor Inputs
  - Programmable sensitivity
  - Automatic recalibration
  - Individual thresholds for each button
- Proximity Detection
- Multiple Button Pattern Detection
- Calibrates for Parasitic Capacitance
- Analog Filtering for System Noise Sources
- Press and Hold feature for Volume-like Applications
- Multiple Communication Interfaces
  - SMBus / I$^2$C compliant interface
  - SMSC BC-Link interface
  - SPI communications
  - Pin selectable communications protocol and multiple slave addresses (SMBus / I$^2$C only)
- Low Power Operation
  - 5uA quiescent current in Deep Sleep
  - 50uA quiescent current in Standby (1 sensor input monitored)
  - Samples one or more channels in Standby
- Eight (8) LED Driver Outputs
  - Open Drain or Push-Pull
  - Programmable blink, breathe, and dimness controls
  - Can be linked to Capacitive Touch Sensor inputs
- Dedicated Wake output flags touches in low power state
- System RESET pin
- Available in 24-pin 4mm x 4mm RoHS compliant QFN package

### Block Diagram



## Ordering Information:

| ORDERING NUMBER | PACKAGE | FEATURES |
|---|---|---|
| CAP1188-1-CP-TR | 24-pin QFN 4mm x 4mm (Lead-free RoHS compliant) | Eight capacitive touch sensor inputs, Eight LED drivers, Dedicated Wake, Reset, SMBus / BC-Link / SPI interfaces |

**Reel size is 4,000 pieces**

**This product meets the halogen maximum concentration values per IEC61249-2-21**

**For RoHS compliance and environmental information, please visit www.smsc.com/rohs**

*Please contact your SMSC sales representative for additional documentation related to this product such as application notes, anomaly sheets, and design guidelines.*

# Table of Contents

# List of Figures

# List of Tables

# Chapter 1 Pin Description

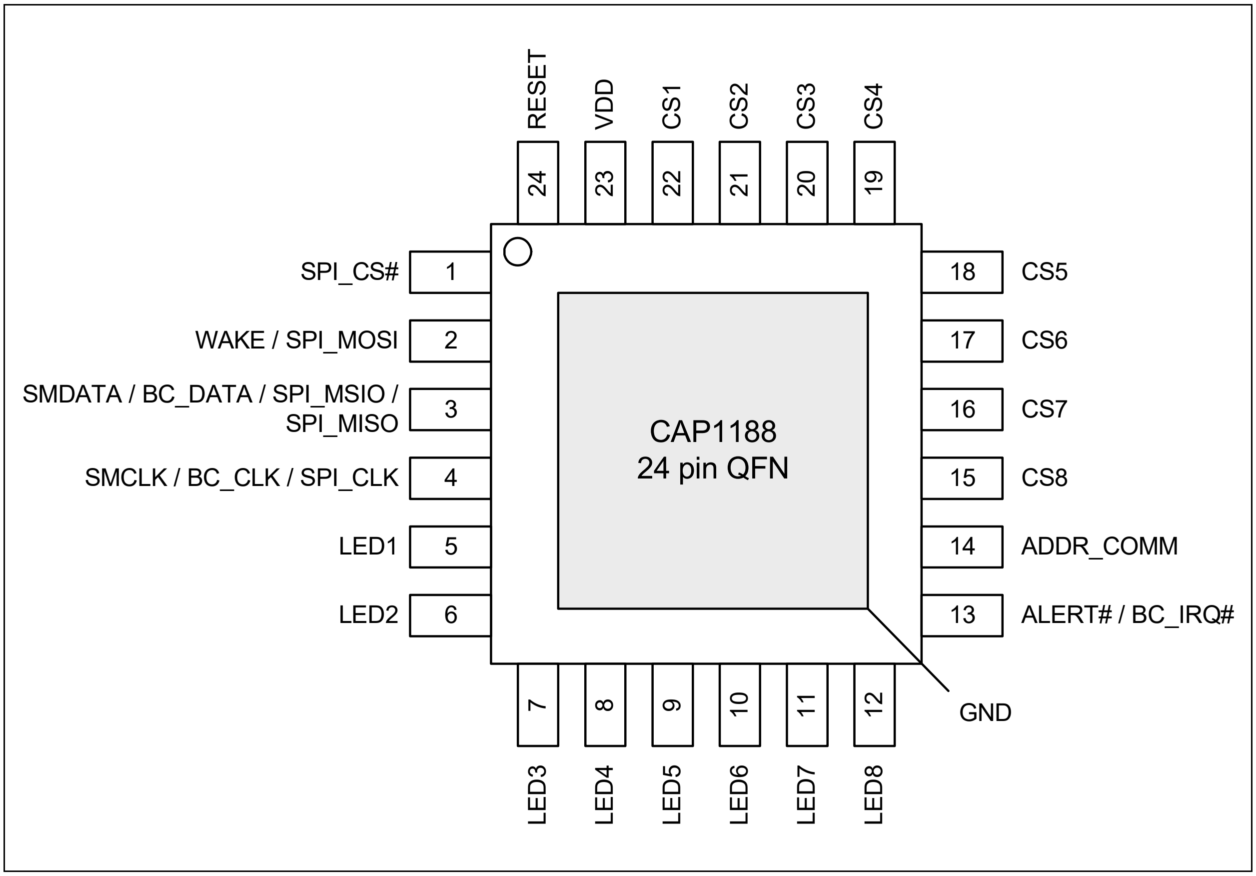



**Figure 1.1 CAP1188 Pin Diagram (24-Pin QFN)**

**Table 1.1 Pin Description for CAP1188**

| PIN # | PIN NAME | PIN FUNCTION | PIN TYPE | UNUSED CONNECTION |
|---|---|---|---|---|
| 1 | SPI_CS# | Active low chip-select for SPI bus | DI (5V) | Connect to Ground |
| 2 | WAKE / SPI_MOSI | WAKE - Active high wake / interrupt output - Standby power state - requires pull-down resistor | DO | Pull-down Resistor |
| | | WAKE - Active high wake input Deep Sleep power state - requires pull-down resistor | DI | |
| | | SPI_MOSI - SPI Master-Out-Slave-In port when used in normal mode | DI (5V) | Connect to Ground |

**Table 1.1 Pin Description for CAP1188 (continued)**

| PIN # | PIN NAME | PIN FUNCTION | PIN TYPE | UNUSED CONNECTION |
|---|---|---|---|---|
| 3 | SMDATA / BC_DATA / SPI_MSIO / SPI_MISO | SMDATA - Bi-directional, open-drain SMBus data - requires pull-up resistor | DIOD (5V) | n/a |
| | | BC_DATA - Bi-directional, open-drain BC-Link data - requires pull-up resistor | DIO | |
| | | SPI_MSIO - SPI Master-Slave-In-Out bidirectional port when used in bi-directional mode | DIO | |
| | | SPI_MISO - SPI Master-In-Slave-Out port when used in normal mode | DO | |
| 4 | SMCLK / BC_CLK / SPI_CLK | SMCLK - SMBus clock input - requires pull-up resistor | DI (5V) | n/a |
| | | BC_CLK - BC-Link clock input | DI (5V) | |
| | | SPI_CLK - SPI clock input | DI (5V) | |
| 5 | LED1 | Open drain LED 1 driver (default) | OD (5V) | Connect to Ground |
| | | Push-pull LED 1 driver | DO | leave open or connect to Ground |
| 6 | LED2 | Open drain LED 2 driver (default) | OD (5V) | Connect to Ground |
| | | Push-pull LED 2 driver | DO | leave open or connect to Ground |
| 7 | LED3 | Open drain LED 3 driver (default) | OD (5V) | Connect to Ground |
| | | Push-pull LED 3 driver | DO | leave open or connect to Ground |
| 8 | LED4 | Open drain LED 4 driver (default) | OD (5V) | Connect to Ground |
| | | Push-pull LED 4 driver | DO | leave open or connect to Ground |
| 9 | LED5 | Open drain LED 5 driver (default) | OD (5V) | Connect to Ground |
| | | Push-pull LED 5 driver | DO | leave open or connect to Ground |
| 10 | LED6 | Open drain LED 6 driver (default) | OD (5V) | Connect to Ground |
| | | Push-pull LED 6 driver | DO | leave open or connect to Ground |

**Table 1.1 Pin Description for CAP1188 (continued)**

| PIN # | PIN NAME | PIN FUNCTION | PIN TYPE | UNUSED CONNECTION |
|---|---|---|---|---|
| 11 | LED7 | Open drain LED 7 driver (default) | OD (5V) | Connect to Ground |
| | | Push-pull LED 7 driver | DO | leave open or connect to Ground |
| 12 | LED8 | Open drain LED 8 driver (default) | OD (5V) | Connect to Ground |
| | | Push-pull LED 8 driver | DO | leave open or connect to Ground |
| 13 | ALERT# / BC_IRQ# | ALERT# - Active low alert / interrupt output for SMBus alert or SPI interrupt - requires pull-up resistor (default) | OD (5V) | Connect to Ground |
| | | ALERT - Active high push-pull alert / interrupt output for SMBus alert or SPI interrupt | DO | leave open |
| | | BC_IRQ# - Active low interrupt / optional for BC-Link - requires pull-up resistor | OD (5V) | Connect to Ground |
| | | BC_IRQ - Active high push-pull interrupt / optional for BC-Link | DO | leave open |
| 14 | ADDR_COMM | Address / communications select pin - pull-down resistor determines address / communications mechanism | AI | n/a |
| 15 | CS8 | Capacitive Touch Sensor Input 8 | AIO | Connect to Ground |
| 16 | CS7 | Capacitive Touch Sensor Input 7 | AIO | Connect to GND |
| 17 | CS6 | Capacitive Touch Sensor Input 6 | AIO | Connect to GND |
| 18 | CS5 | Capacitive Touch Sensor Input 5 | AIO | Connect to GND |
| 19 | CS4 | Capacitive Touch Sensor Input 4 | AIO | Connect to GND |
| 20 | CS3 | Capacitive Touch Sensor Input 3 | AIO | Connect to GND |
| 21 | CS2 | Capacitive Touch Sensor Input 2 | AIO | Connect to GND |
| 22 | CS1 | Capacitive Touch Sensor Input 1 | AIO | Connect to GND |
| 23 | VDD | Positive Power supply | Power | n/a |
| 24 | RESET | Active high soft reset for system - resets all registers to default values. | DI (5V) | Connect to GND |
| Bottom Pad | GND | Ground | Power | n/a |

**APPLICATION NOTE:** When the ALERT# pin is configured as an active low output, it will be open drain. When it is configured as an active high output, it will be push-pull.

**APPLICATION NOTE:** For the 5V tolerant pins that have a pull-up resistor, the pull-up voltage must not exceed 3.6V when the CAP1188 is unpowered.

**APPLICATION NOTE:** The SPI_CS# pin should be grounded when SMBus, $I^2C$, or BC-Link communications are used.

The pin types are described in Table 1.2. All pins labeled with (5V) are 5V tolerant.

**Table 1.2 Pin Types**

| PIN TYPE | DESCRIPTION |
|---|---|
| Power | This pin is used to supply power or ground to the device. |
| DI | Digital Input - This pin is used as a digital input. This pin is 5V tolerant. |
| AIO | Analog Input / Output -This pin is used as an I/O for analog signals. |
| DIOD | Digital Input / Open Drain Output - This pin is used as a digital I/O. When it is used as an output, it is open drain and requires a pull-up resistor. This pin is 5V tolerant. |
| OD | Open Drain Digital Output - This pin is used as a digital output. It is open drain and requires a pull-up resistor. This pin is 5V tolerant. |
| DO | Push-pull Digital Output - This pin is used as a digital output and can sink and source current. |
| DIO | Push-pull Digital Input / Output - This pin is used as an I/O for digital signals. |

# Chapter 2 Electrical Specifications

**Table 2.1 Absolute Maximum Ratings**

| | | |
|---|---|---|
| Voltage on 5V tolerant pins ($V_{5VT\_PIN}$) | -0.3 to 5.5 | V |
| Voltage on 5V tolerant pins ($\|V_{5VT\_PIN} - V_{DD}\|$) Note 2.2 | 0 to 3.6 | V |
| Voltage on VDD pin | -0.3 to 4 | V |
| Voltage on any other pin to GND | -0.3 to VDD + 0.3 | V |
| Package Power Dissipation up to $T_A$ = 85°C for 24 pin QFN (see Note 2.3) | 0.9 | W |
| Junction to Ambient ($\theta_{JA}$) (see Note 2.4) | 58 | °C/W |
| Operating Ambient Temperature Range | -40 to 125 | °C |
| Storage Temperature Range | -55 to 150 | °C |
| ESD Rating, All Pins, HBM | 8000 | V |

**Note 2.1** Stresses above those listed could cause permanent damage to the device. This is a stress rating only and functional operation of the device at any other condition above those indicated in the operation sections of this specification is not implied.

**Note 2.2** For the 5V tolerant pins that have a pull-up resistor, the voltage difference between $V_{5VT\_PIN}$ and $V_{DD}$ must never exceed 3.6V.

**Note 2.3** The Package Power Dissipation specification assumes a recommended thermal via design consisting of a 3x3 matrix of 0.3mm (12mil) vias at 1.0mm pitch connected to the ground plane with a 2.5 x 2.5mm thermal landing.

**Note 2.4** Junction to Ambient ($\theta_{JA}$) is dependent on the design of the thermal vias. Without thermal vias and a thermal landing, the $\theta_{JA}$ is approximately 60°C/W including localized PCB temperature increase.

**Table 2.2 Electrical Specifications**

| $V_{DD}$ = 3V to 3.6V, $T_A$ = 0°C to 85°C, all Typical values at $T_A$ = 27°C unless otherwise noted. | | | | | | |
|---|---|---|---|---|---|---|
| **CHARACTERISTIC** | **SYMBOL** | **MIN** | **TYP** | **MAX** | **UNIT** | **CONDITIONS** |
| DC Power | | | | | | |
| Supply Voltage | $V_{DD}$ | 3.0 | 3.3 | 3.6 | V | |

**Table 2.2 Electrical Specifications (continued)**

| $V_{DD}$ = 3V to 3.6V, $T_A$ = 0°C to 85°C, all Typical values at $T_A$ = 27°C unless otherwise noted. | | | | | | |
|---|---|---|---|---|---|---|
| **CHARACTERISTIC** | **SYMBOL** | **MIN** | **TYP** | **MAX** | **UNIT** | **CONDITIONS** |
| Supply Current | $I_{STBY}$ | | 120 | 170 | uA | Standby state active<br>1 sensor input monitored<br>No LEDs active<br>Default conditions (8 avg, 70ms cycle time) |
| | $I_{STBY}$ | | 50 | | uA | Standby state active<br>1 sensor input monitored<br>No LEDs active<br>1 avg, 140ms cycle time, |
| | $I_{DSLEEP}$ | | 5 | 15 | uA | Deep Sleep state active<br>LEDs at 100% or 0% Duty Cycle<br>No communications<br>$T_A$ < 40°C<br>3.135 < $V_{DD}$ < 3.465V |
| | $I_{DD}$ | | 500 | 600 | uA | Capacitive Sensing Active<br>No LEDs active |
| Capacitive Touch Sensor Inputs | | | | | | |
| Maximum Base Capacitance | $C_{BASE}$ | | 50 | | pF | Pad untouched |
| Minimum Detectable Capacitive Shift | $\Delta C_{TOUCH}$ | 20 | | | fF | Pad touched - default conditions (1 avg, 35ms cycle time, 1x sensitivity) |
| Recommended Cap Shift | $\Delta C_{TOUCH}$ | 0.1 | | 2 | pF | Pad touched - Not tested |
| Power Supply Rejection | PSR | | ±3 | ±10 | counts / V | Untouched Current Counts<br>Base Capacitance 5pF - 50pF<br>Maximum sensitivity<br>Negative Delta Counts disabled<br>All other parameters default |
| Timing | | | | | | |
| RESET Pin Delay | $t_{RST\_DLY}$ | 10 | | | ms | |
| Time to communications ready | $t_{COMM\_DLY}$ | | | 15 | ms | |
| Time to first conversion ready | $t_{CONV\_DLY}$ | | 170 | 200 | ms | |
| LED Drivers | | | | | | |
| Duty Cycle | $DUTY_{LED}$ | 0 | | 100 | % | Programmable |
| Drive Frequency | $f_{LED}$ | | 2 | | kHz | |
| Sinking Current | $I_{SINK}$ | | | 24 | mA | $V_{OL}$ = 0.4 |
| Sourcing Current | $I_{SOURCE}$ | | | 24 | mA | $V_{OH}$ = $V_{DD}$ - 0.4 |

**Table 2.2 Electrical Specifications (continued)**

| $V_{DD}$ = 3V to 3.6V, $T_A$ = 0°C to 85°C, all Typical values at $T_A$ = 27°C unless otherwise noted. | | | | | | |
|---|---|---|---|---|---|---|
| **CHARACTERISTIC** | **SYMBOL** | **MIN** | **TYP** | **MAX** | **UNIT** | **CONDITIONS** |
| Leakage Current | $I_{LEAK}$ | | | ±5 | uA | powered or unpowered<br>TA < 85°C<br>pull-up voltage ≤ 3.6V if unpowered |
| I/O Pins | | | | | | |
| Output Low Voltage | $V_{OL}$ | | | 0.4 | V | $I_{SINK\_IO}$ = 8mA |
| Output High Voltage | $V_{OH}$ | $V_{DD}$ - 0.4 | | | V | $I_{SOURCE\_IO}$ = 8mA |
| Input High Voltage | $V_{IH}$ | 2.0 | | | V | |
| Input Low Voltage | $V_{IL}$ | | | 0.8 | V | |
| Leakage Current | $I_{LEAK}$ | | | ±5 | uA | powered or unpowered<br>$T_A$ < 85°C<br>pull-up voltage ≤ 3.6V if unpowered |
| RESET Pin Release to conversion ready | $t_{RESET}$ | | 170 | 200 | ms | |
| SMBus Timing | | | | | | |
| Input Capacitance | $C_{IN}$ | | 5 | | pF | |
| Clock Frequency | $f_{SMB}$ | 10 | | 400 | kHz | |
| Spike Suppression | $t_{SP}$ | | | 50 | ns | |
| Bus Free Time Stop to Start | $t_{BUF}$ | 1.3 | | | us | |
| Start Setup Time | $t_{SU:STA}$ | 0.6 | | | us | |
| Start Hold Time | $t_{HD:STA}$ | 0.6 | | | us | |
| Stop Setup Time | $t_{SU:STO}$ | 0.6 | | | us | |
| Data Hold Time | $t_{HD:DAT}$ | 0 | | | us | When transmitting to the master |
| Data Hold Time | $t_{HD:DAT}$ | 0.3 | | | us | When receiving from the master |
| Data Setup Time | $t_{SU:DAT}$ | 0.6 | | | us | |
| Clock Low Period | $t_{LOW}$ | 1.3 | | | us | |
| Clock High Period | $t_{HIGH}$ | 0.6 | | | us | |
| Clock / Data Fall Time | $t_{FALL}$ | | | 300 | ns | Min = 20+0.1$C_{LOAD}$ ns |
| Clock / Data Rise Time | $t_{RISE}$ | | | 300 | ns | Min = 20+0.1$C_{LOAD}$ ns |
| Capacitive Load | $C_{LOAD}$ | | | 400 | pF | per bus line |

**Table 2.2 Electrical Specifications (continued)**

| $V_{DD}$ = 3V to 3.6V, $T_A$ = 0°C to 85°C, all Typical values at $T_A$ = 27°C unless otherwise noted. | | | | | | |
|---|---|---|---|---|---|---|
| **CHARACTERISTIC** | **SYMBOL** | **MIN** | **TYP** | **MAX** | **UNIT** | **CONDITIONS** |
| BC-Link Timing | | | | | | |
| Clock Period | $t_{CLK}$ | 250 | | | ns | |
| Data Hold Time | $t_{HD:DAT}$ | 0 | | | ns | |
| Data Setup Time | $t_{SU:DAT}$ | 30 | | | ns | Data must be valid before clock |
| Clock Duty Cycle | Duty | 40 | 50 | 60 | % | |
| SPI Timing | | | | | | |
| Clock Period | $t_P$ | 250 | | | ns | |
| Clock Low Period | $t_{LOW}$ | 0.4 x $t_P$ | | 0.6 x $t_P$ | ns | |
| Clock High Period | $t_{HIGH}$ | 0.4 x $t_P$ | | 0.6 x $t_P$ | ns | |
| Clock Rise / Fall time | $t_{RISE}$ / $t_{FALL}$ | | | 0.1 x $t_P$ | ns | |
| Data Output Delay | $t_{D:CLK}$ | | | 10 | ns | |
| Data Setup Time | $t_{SU:DAT}$ | 20 | | | ns | |
| Data Hold Time | $t_{HD:DAT}$ | 20 | | | ns | |
| SPI_CS# to SPI_CLK setup time | $t_{SU:CS}$ | 0 | | | ns | |
| Wake Time | $t_{WAKE}$ | 10 | | 20 | us | SPI_CS# asserted to CLK assert |

**Note 2.5** The ALERT pin will not glitch high or low at power up if connected to VDD or another voltage.

**Note 2.6** The SMCLK and SMDATA pins will not glitch low at power up if connected to VDD or another voltage.

# Chapter 3 Communications

## 3.1 Communications

The CAP1188 communicates using the 2-wire SMBus or $I^2C$ bus, the 2-wire proprietary BC-Link, or the SPI bus. Regardless of communication mechanism, the device functionality remains unchanged. The communications mechanism as well as the SMBus (or $I^2C$) slave address is determined by the resistor connected between the ADDR_COMM pin and ground as shown in Table 3.1.

**Table 3.1 ADDR_COMM Pin Decode**

| PULL-DOWN RESISTOR (+/- 5%) | PROTOCOL USED | SMBUS ADDRESS |
|---|---|---|
| GND | SPI Communications using Normal 4-wire Protocol Used | n/a |
| 56k | SPI Communications using Bi-Directional 3-wire Protocol Used | n/a |
| 68k | BC-Link Communications | n/a |
| 82k | SMBus / $I^2C$ | 0101_100(r/w) |
| 100k | SMBus / $I^2C$ | 0101_011(r/w) |
| 120k | SMBus / $I^2C$ | 0101_010(r/w) |
| 150k | SMBus / $I^2C$ | 0101_001(r/w) |
| VDD | SMBus / $I^2C$ | 0101_000(r/w) |

### 3.1.1 SMBus ($I^2C$) Communications

When configured to communicate via the SMBus, the CAP1188 supports the following protocols: Send Byte, Receive Byte, Read Byte, Write Byte, Read Block, and Write Block. In addition, the device supports $I^2C$ formatting for block read and block write protocols.

**APPLICATION NOTE:** For SMBus/$I^2C$ communications, the SPI_CS# pin is not used and should be grounded; any data presented to this pin will be ignored.

See Section 3.2 and Section 3.3 for more information on the SMBus bus and protocols respectively.

### 3.1.2 SPI Communications

When configured to communicate via the SPI bus, the CAP1188 supports both bi-directional 3-wire and normal 4-wire protocols and uses the SPI_CS# pin to enable communications.

See Section 3.5 and Section 3.6 for more information on the SPI bus and protocols respectively.

### 3.1.3 BC-Link Communications

When BC-Link communications are used, the CAP1188 supports the read byte protocol and the write byte protocol.

**APPLICATION NOTE:** For BC-Link communications, the SPI_CS# pin is not used and should be grounded; any data presented to this pin will be ignored.

See Section 3.8 for more information on the BC-Link Bus and protocols respectively.

**APPLICATION NOTE:** Upon power up, the CAP1188 will not respond to any communications for up to 15ms. After this time, full functionality is available.

## 3.2 System Management Bus

The CAP1188 communicates with a host controller, such as an SMSC SIO, through the SMBus. The SMBus is a two-wire serial communication protocol between a computer host and its peripheral devices. A detailed timing diagram is shown in Figure 3.1. Stretching of the SMCLK signal is supported; however, the CAP1188 will not stretch the clock signal.

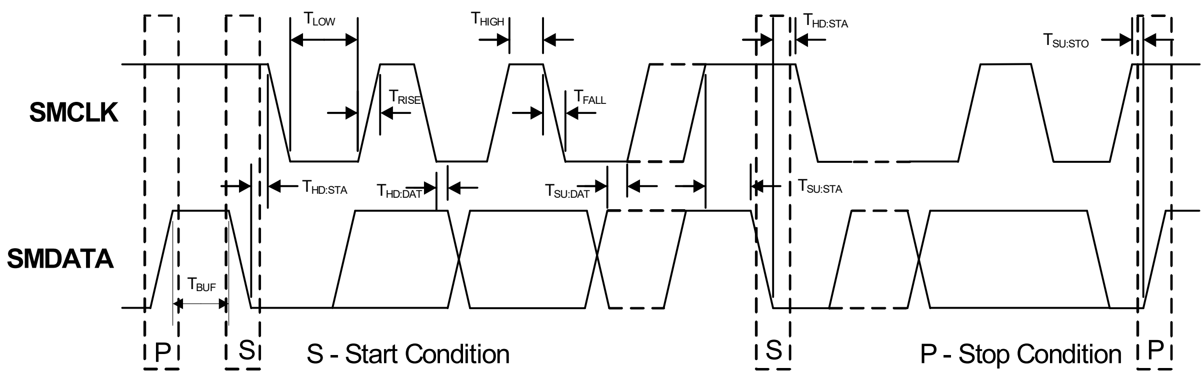



**Figure 3.1 SMBus Timing Diagram**

### 3.2.1 SMBus Start Bit

The SMBus Start bit is defined as a transition of the SMBus Data line from a logic ‘1’ state to a logic ‘0’ state while the SMBus Clock line is in a logic ‘1’ state.

### 3.2.2 SMBus Address and RD / $\overline{WR}$ Bit

The SMBus Address Byte consists of the 7-bit client address followed by the RD / $\overline{WR}$ indicator bit. If this RD / $\overline{WR}$ bit is a logic ‘0’, then the SMBus Host is writing data to the client device. If this RD / $\overline{WR}$ bit is a logic ‘1’, then the SMBus Host is reading data from the client device.

See Table 3.1 for available SMBus addresses.

### 3.2.3 SMBus Data Bytes

All SMBus Data bytes are sent most significant bit first and composed of 8-bits of information.

### 3.2.4 SMBus ACK and NACK Bits

The SMBus client will acknowledge all data bytes that it receives. This is done by the client device pulling the SMBus Data line low after the 8th bit of each byte that is transmitted. This applies to both the Write Byte and Block Write protocols.

The Host will NACK (not acknowledge) the last data byte to be received from the client by holding the SMBus data line high after the 8th data bit has been sent. For the Block Read protocol, the Host will ACK each data byte that it receives except the last data byte.

### 3.2.5 SMBus Stop Bit

The SMBus Stop bit is defined as a transition of the SMBus Data line from a logic ‘0’ state to a logic ‘1’ state while the SMBus clock line is in a logic ‘1’ state. When the CAP1188 detects an SMBus Stop bit and it has been communicating with the SMBus protocol, it will reset its client interface and prepare to receive further communications.

### 3.2.6 SMBus Timeout

The CAP1188 includes an SMBus timeout feature. Following a 30ms period of inactivity on the SMBus where the SMCLK pin is held low, the device will timeout and reset the SMBus interface.

The timeout function defaults to disabled. It can be enabled by setting the TIMEOUT bit in the Configuration register (see Section 5.6, "Configuration Registers").

### 3.2.7 SMBus and I$^2$C Compatibility

The major differences between SMBus and I$^2$C devices are highlighted here. For more information, refer to the SMBus 2.0 and I$^2$C specifications. For information on using the CAP1188 in an I$^2$C system, refer to SMSC AN 14.0 SMSC Dedicated Slave Devices in I$^2$C Systems.

1. CAP1188 supports I$^2$C fast mode at 400kHz. This covers the SMBus max time of 100kHz.
2. Minimum frequency for SMBus communications is 10kHz.
3. The SMBus client protocol will reset if the clock is held at a logic ‘0’ for longer than 30ms. This timeout functionality is disabled by default in the CAP1188 and can be enabled by writing to the TIMEOUT bit. I$^2$C does not have a timeout.
4. The SMBus client protocol will reset if both the clock and data lines are held at a logic ‘1’ for longer than 200µs (idle condition). This function is disabled by default in the CAP1188 and can be enabled by writing to the TIMEOUT bit. I$^2$C does not have an idle condition.
5. I$^2$C devices do not support the Alert Response Address functionality (which is optional for SMBus).
6. I$^2$C devices support block read and write differently. I$^2$C protocol allows for unlimited number of bytes to be sent in either direction. The SMBus protocol requires that an additional data byte indicating number of bytes to read / write is transmitted. The CAP1188 supports I$^2$C formatting only.

## 3.3 SMBus Protocols

The CAP1188 is SMBus 2.0 compatible and supports Write Byte, Read Byte, Send Byte, and Receive Byte as valid protocols as shown below.

All of the below protocols use the convention in Table 3.2.

**Table 3.2 Protocol Format**

| DATA SENT TO DEVICE | DATA SENT TO THE HOST |
|---|---|
| Data sent | Data sent |

### 3.3.1 SMBus Write Byte

The Write Byte is used to write one byte of data to a specific register as shown in Table 3.3.

**Table 3.3 Write Byte Protocol**

| START | SLAVE ADDRESS | WR | ACK | REGISTER ADDRESS | ACK | REGISTER DATA | ACK | STOP |
|---|---|---|---|---|---|---|---|---|
| 1 ->0 | YYYY_YYY | 0 | 0 | XXh | 0 | XXh | 0 | 0 -> 1 |

### 3.3.2 SMBus Read Byte

The Read Byte protocol is used to read one byte of data from the registers as shown in Table 3.4.

**Table 3.4 Read Byte Protocol**

| START | SLAVE ADDRESS | WR | ACK | REGISTER ADDRESS | ACK | START | CLIENT ADDRESS | RD | ACK | REGISTER DATA | NACK | STOP |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1->0 | YYYY_YYY | 0 | 0 | XXh | 0 | 1 ->0 | YYYY_YYY | 1 | 0 | XXh | 1 | 0 -> 1 |

### 3.3.3 SMBus Send Byte

The Send Byte protocol is used to set the internal address register pointer to the correct address location. No data is transferred during the Send Byte protocol as shown in Table 3.5.

**APPLICATION NOTE:** The Send Byte protocol is not functional in Deep Sleep (i.e., DSLEEP bit is set).

**Table 3.5 Send Byte Protocol**

| START | SLAVE ADDRESS | WR | ACK | REGISTER ADDRESS | ACK | STOP |
|---|---|---|---|---|---|---|
| 1 -> 0 | YYYY_YYY | 0 | 0 | XXh | 0 | 0 -> 1 |

### 3.3.4 SMBus Receive Byte

The Receive Byte protocol is used to read data from a register when the internal register address pointer is known to be at the right location (e.g., set via Send Byte). This is used for consecutive reads of the same register as shown in Table 3.6.

**APPLICATION NOTE:** The Receive Byte protocol is not functional in Deep Sleep (i.e., DSLEEP bit is set).

**Table 3.6 Receive Byte Protocol**

| START | SLAVE ADDRESS | RD | ACK | REGISTER DATA | NACK | STOP |
|---|---|---|---|---|---|---|
| 1 -> 0 | YYYY_YYY | 1 | 0 | XXh | 1 | 0 -> 1 |

# 3.4 I²C Protocols

The CAP1188 supports I²C Block Write and Block Read.

The protocols listed below use the convention in Table 3.2.

## 3.4.1 Block Write

The Block Write is used to write multiple data bytes to a group of contiguous registers as shown in Table 3.7.

**APPLICATION NOTE:** When using the Block Write protocol, the internal address pointer will be automatically incremented after every data byte is received. It will wrap from FFh to 00h.

**Table 3.7 Block Write Protocol**

| START | SLAVE ADDRESS | WR | ACK | REGISTER ADDRESS | ACK | REGISTER DATA | ACK |
|---|---|---|---|---|---|---|---|
| 1 ->0 | YYYY_YYY | 0 | 0 | XXh | 0 | XXh | 0 |
| **REGISTER DATA** | **ACK** | **REGISTER DATA** | **ACK** | **. . .** | **REGISTER DATA** | **ACK** | **STOP** |
| XXh | 0 | XXh | 0 | . . . | XXh | 0 | 0 -> 1 |

## 3.4.2 Block Read

The Block Read is used to read multiple data bytes from a group of contiguous registers as shown in Table 3.8.

**APPLICATION NOTE:** When using the Block Read protocol, the internal address pointer will be automatically incremented after every data byte is received. It will wrap from FFh to 00h.

**Table 3.8 Block Read Protocol**

| START | SLAVE ADDRESS | WR | ACK | REGISTER ADDRESS | ACK | START | SLAVE ADDRESS | RD | ACK | REGISTER DATA |
|---|---|---|---|---|---|---|---|---|---|---|
| 1->0 | YYYY_YYY | 0 | 0 | XXh | 0 | 1 ->0 | YYYY_YYY | 1 | 0 | XXh |
| **ACK** | **REGISTER DATA** | **ACK** | **REGISTER DATA** | **ACK** | **REGISTER DATA** | **ACK** | **. . .** | **REGISTER DATA** | **NACK** | **STOP** |
| 0 | XXh | 0 | XXh | 0 | XXh | 0 | . . . | XXh | 1 | 0 -> 1 |

# 3.5 SPI Interface

The SMBus has a predefined packet structure, the SPI does not. The SPI Bus can operate in two modes of operation, normal 4-wire mode and bi-directional 3-wire mode. All SPI commands consist of 8-bit packets sent to a specific slave device (identified by the CS pin).

The SPI bus will latch data on the rising edge of the clock and the clock and data both idle high.

All commands are supported via both operating modes. The supported commands are: Reset Serial interface, set address pointer, write command and read command. Note that all other codes received during the command phase are ignored and have no effect on the operation of the device.

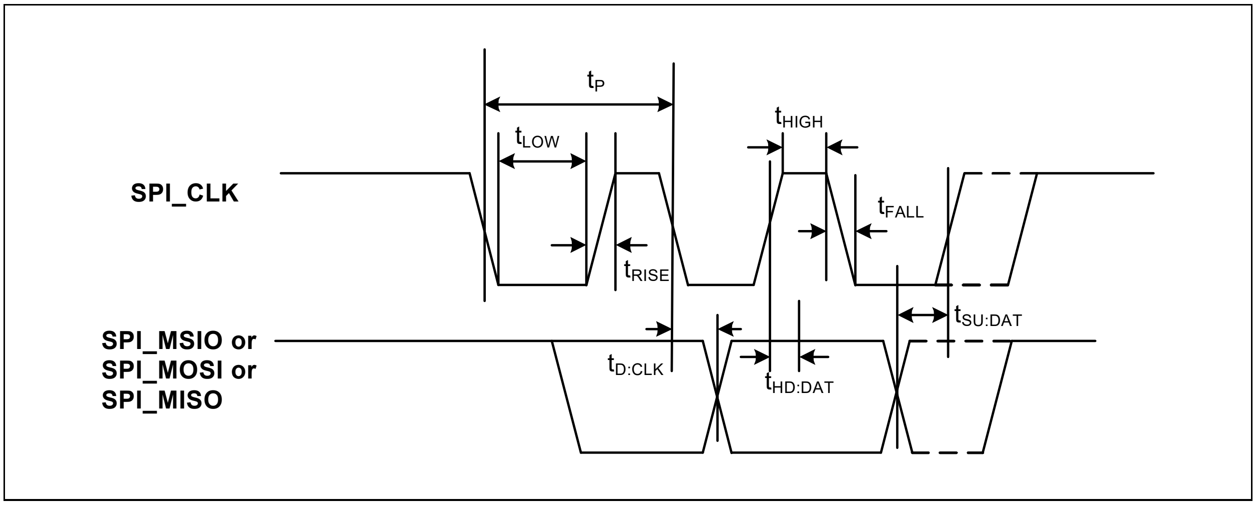



**Figure 3.2 SPI Timing**

## 3.5.1 SPI Normal Mode

The SPI Bus can operate in two modes of operation, normal and bi-directional mode. In the normal mode of operation, there are dedicated input and output data lines. The host communicates by sending a command along the CAP1188 SPI_MOSI data line and reading data on the SPI_MISO data line. Both communications occur simultaneously which allows for larger throughput of data transactions.

All basic transfers consist of two 8 bit transactions from the Master device while the slave device is simultaneously sending data at the current address pointer value.

Data writes consist of two or more 8-bit transactions. The host sends a specific write command followed by the data to write the address pointer. Data reads consist of one or more 8-bit transactions. The host sends the specific read data command and continues clocking for as many data bytes as it wishes to receive.

## 3.5.2 SPI Bi-Directional Mode

In the bi-directional mode of operation, the SPI data signals are combined into the SPI_MSIO line, which is shared for data received by the device and transmitted by the device. The protocol uses a simple handshake and turn around sequence for data communications based on the number of clocks transmitted during each phase.

All basic transfers consist of two 8 bit transactions. The first is an 8 bit command phase driven by the Master device. The second is by an 8 bit data phase driven by the Master for writes, and by the CAP1188 for read operations.

The auto increment feature of the address pointer allows for successive reads or writes. The address pointer will return to 00h after reaching FFh.

## 3.5.3 SPI_CS# Pin

The SPI Bus is a single master, multiple slave serial bus. Each slave has a dedicated CS pin (chip select) that the master asserts low to identify that the slave is being addressed. There are no formal addressing options.

### 3.5.4 Address Pointer

All data writes and reads are accessed from the current address pointer. In both Bi-directional mode and Full Duplex mode, the Address pointer is automatically incremented following every read command or every write command.

The address pointer will return to 00h after reaching FFh.

### 3.5.5 SPI Timeout

The CAP1188 does not detect any timeout conditions on the SPI bus.

## 3.6 Normal SPI Protocols

When operating in normal mode, the SPI bus internal address pointer is incremented depending upon which command has been transmitted. Multiple commands may be transmitted sequentually so long as the SPI_CS# pin is asserted low. Figure 3.3 shows an example of this operation.

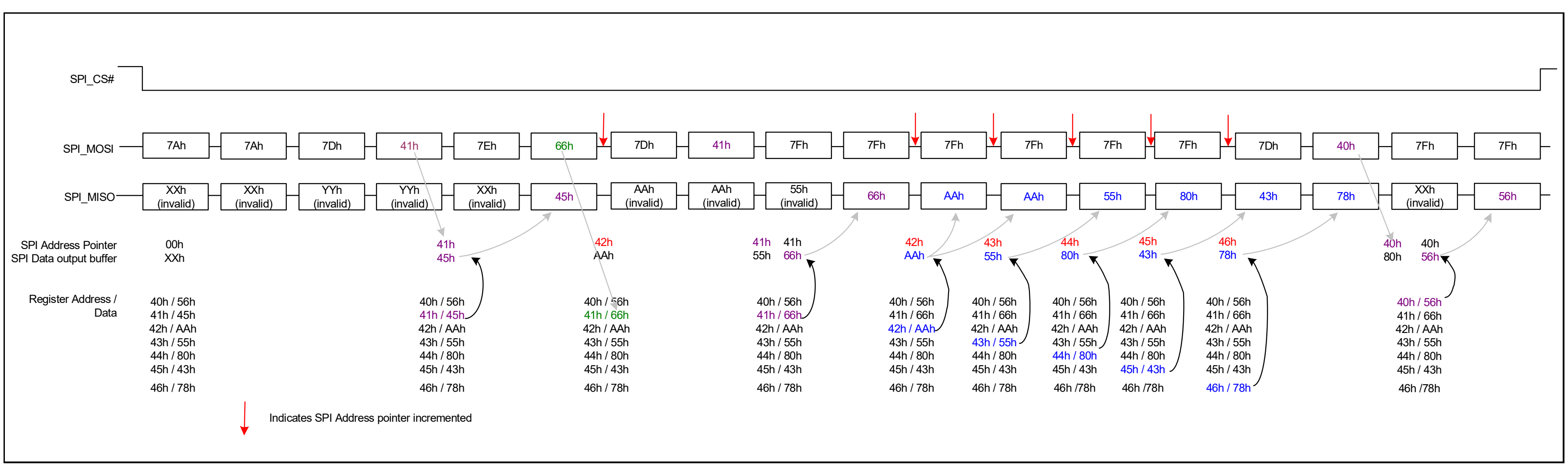



**Figure 3.3 Example SPI Bus Communication - Normal Mode**

### 3.6.1 Reset Interface

Resets the Serial interface whenever two successive 7Ah codes are received. Regardless of the current phase of the transaction - command or data, the receipt of the successive reset commands resets the Serial communication interface only. All other functions are not affected by the reset operation.

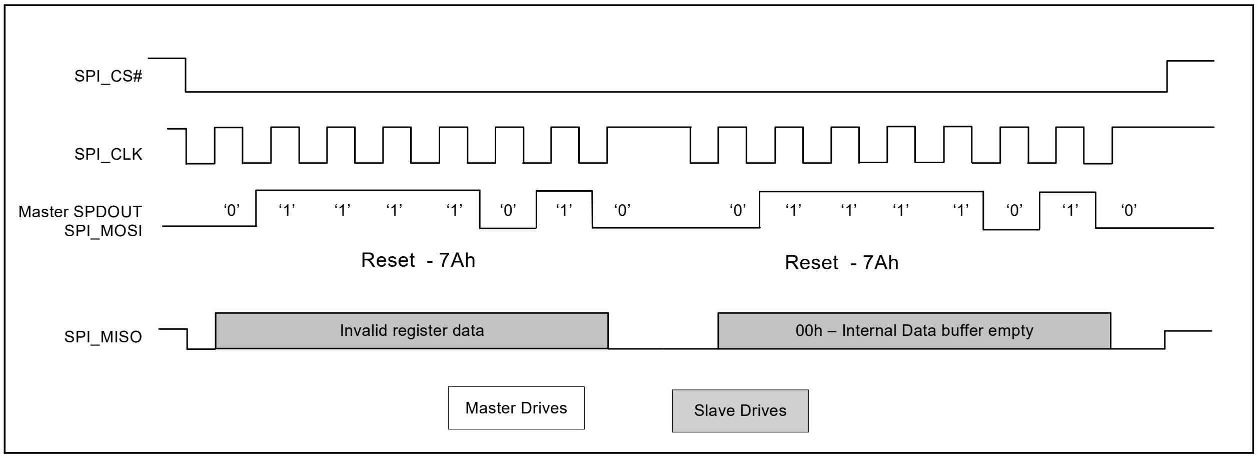



**Figure 3.4 SPI Reset Interface Command - Normal Mode**

## 3.6.2 Set Address Pointer

The Set Address Pointer command sets the Address pointer for subsequent reads and writes of data. The pointer is set on the rising edge of the final data bit. At the same time, the data that is to be read is fetched and loaded into the internal output buffer but is not transmitted.

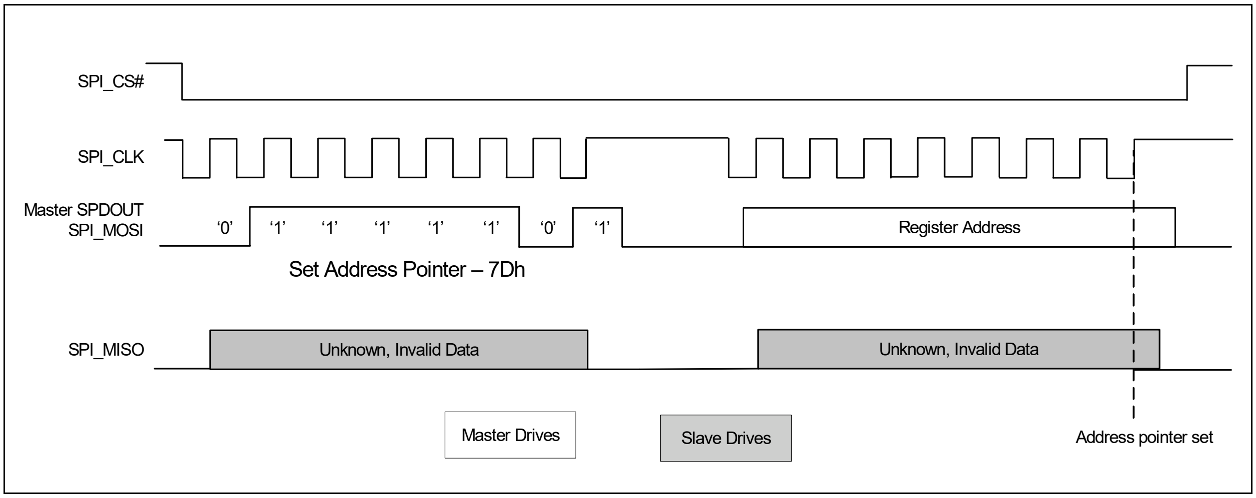



**Figure 3.5 SPI Set Address Pointer Command - Normal Mode**

## 3.6.3 Write Data

The Write Data protocol updates the contents of the register referenced by the address pointer. As the command is processed, the data to be read is fetched and loaded into the internal output buffer but not transmitted. Then, the register is updated with the data to be written. Finally, the address pointer is incremented.

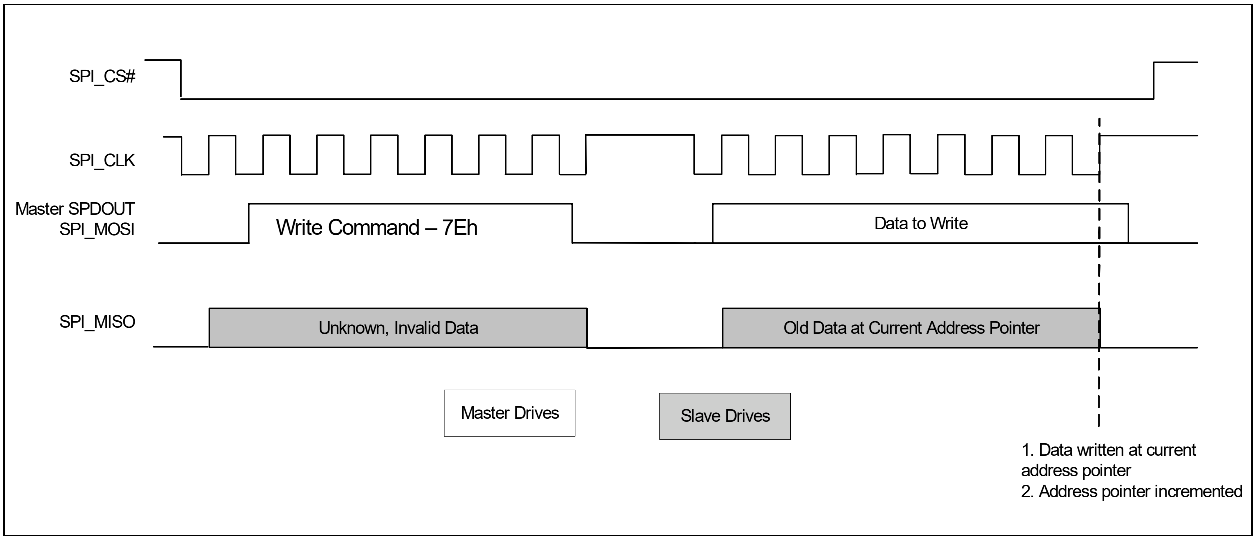



**Figure 3.6 SPI Write Command - Normal Mode**

## 3.6.4 Read Data

The Read Data protocol is used to read data from the device. During the normal mode of operation, while the device is receiving data, the CAP1188 is simultaneously transmitting data to the host. For the Set Address commands and the Write Data commands, this data may be invalid and it is recommended that the Read Data command is used.

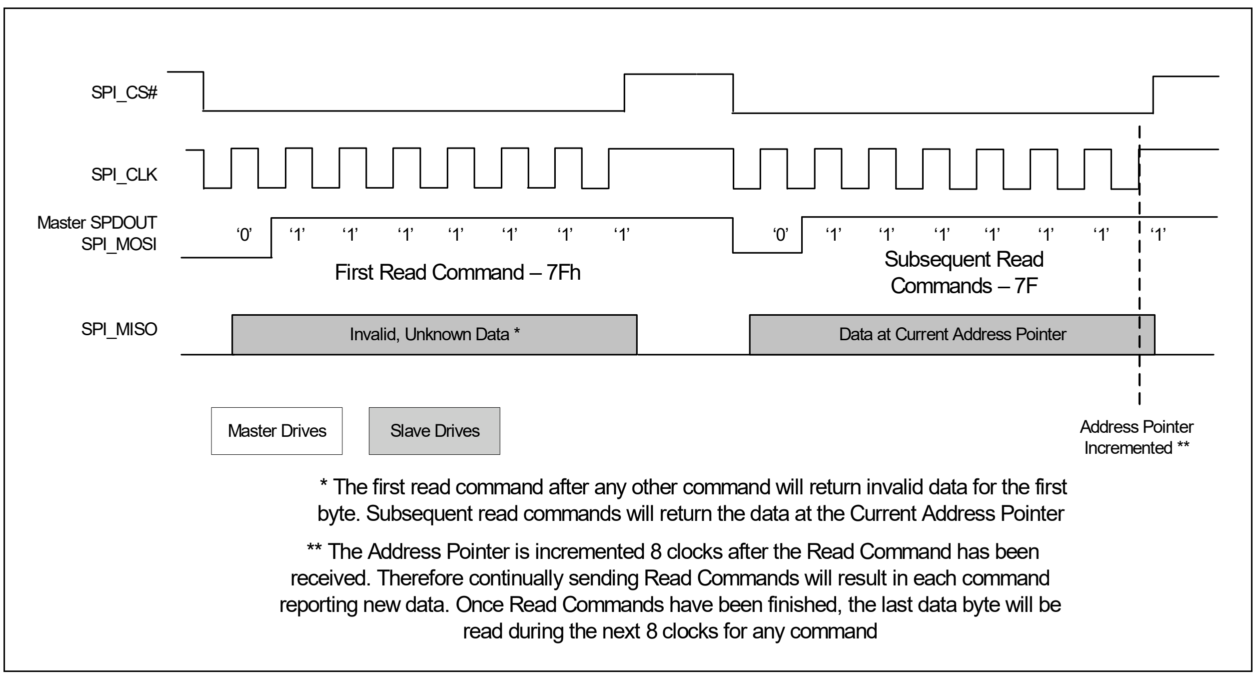



**Figure 3.7 SPI Read Command - Normal Mode**

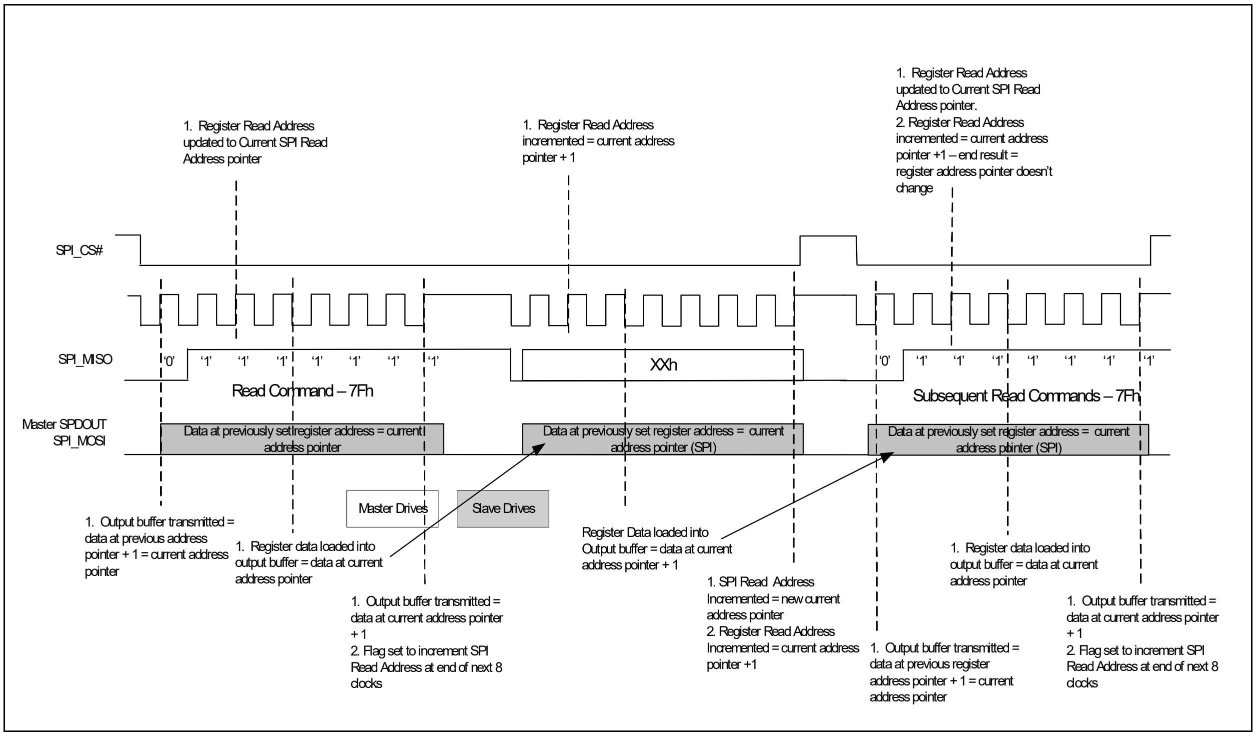



**Figure 3.8 SPI Read Command - Normal Mode - Full**

# 3.7 Bi-Directional SPI Protocols

## 3.7.1 Reset Interface

Resets the Serial interface whenever two successive 7Ah codes are received. Regardless of the current phase of the transaction - command or data, the receipt of the successive reset commands resets the Serial communication interface only. All other functions are not affected by the reset operation.

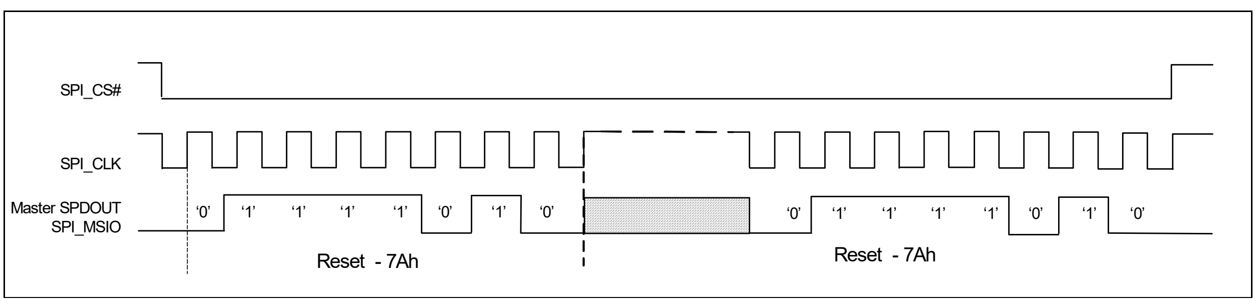



**Figure 3.9 SPI Reset Interface Command - Bi-directional Mode**

## 3.7.2 Set Address Pointer

Sets the address pointer to the register to be accessed by a read or write command. This command overrides the auto-incrementing of the address pointer.

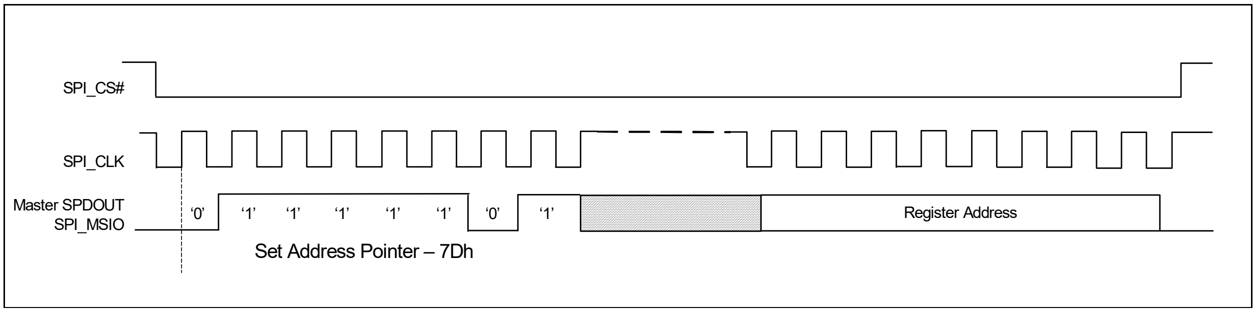



**Figure 3.10 SPI Set Address Pointer Command - Bi-directional Mode**

### 3.7.3 Write Data

Writes data value to the register address stored in the address pointer. Performs auto increment of address pointer after the data is loaded into the register.

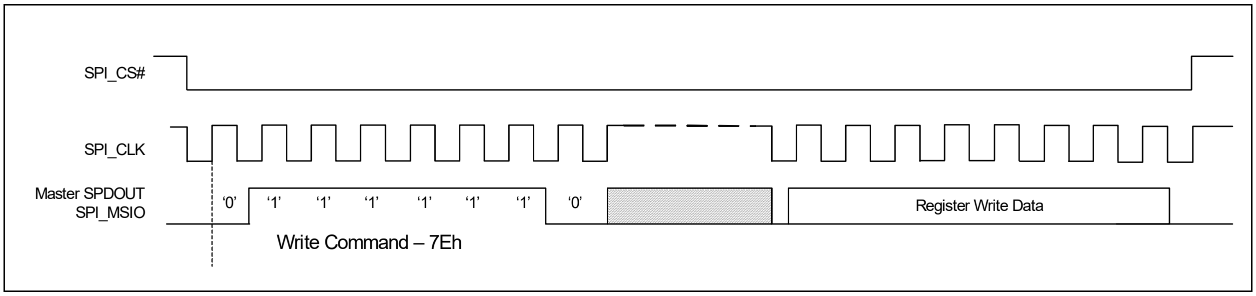



**Figure 3.11 SPI Write Data Command - Bi-directional Mode**

### 3.7.4 Read Data

Reads data referenced by the address pointer. Performs auto increment of address pointer after the data is transferred to the Master.

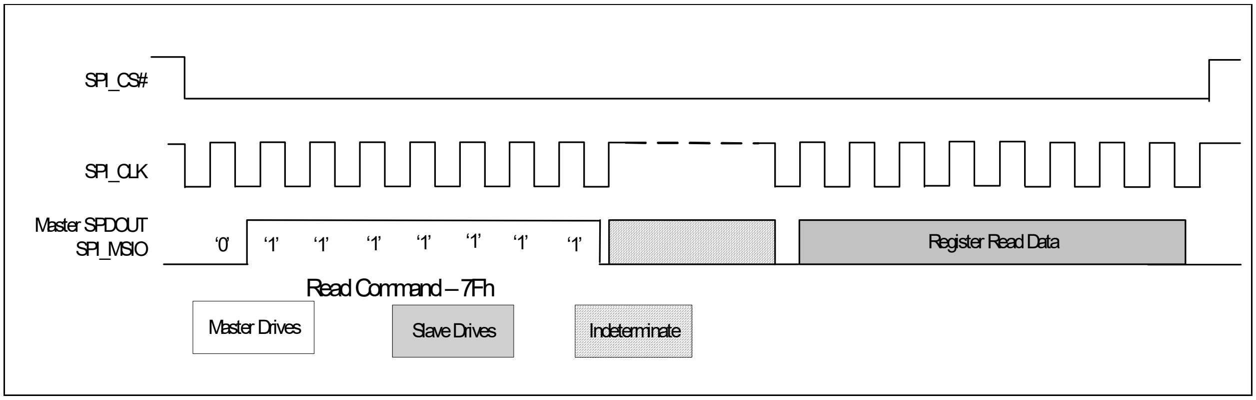



**Figure 3.12 SPI Read Data Command - Bi-directional Mode**

## 3.8 BC-Link Interface

The BC-Link is a proprietary bus developed to allow communication between a host controller device to a companion device. This device uses this serial bus to read and write registers and for interrupt processing. The interface uses a data port concept, where the base interface has an address register, data register and a control register, defined in the SMSC's 8051's SFR space.

Refer to documentation for the BC-Link compatible host controller for details on how to access the CAP1188 via the BC-Link Interface.

# Chapter 4 General Description

The CAP1188 is a multiple channel Capacitive Touch sensor with multiple power LED drivers. It contains eight (8) individual capacitive touch sensor inputs with programmable sensitivity for use in touch sensor applications. Each sensor input automatically recalibrates to compensate for gradual environmental changes.

The CAP1188 also contains eight (8) low side (or push-pull) LED drivers that offer full-on / off, variable rate blinking, dimness controls, and breathing. Each of the LED drivers may be linked to one of the sensor inputs to be actuated when a touch is detected. As well, each LED driver may be individually controlled via a host controller.

Finally, the device contains a dedicated RESET pin to act as a soft reset by the system.

The CAP1188 offers multiple power states. It operates at the lowest quiescent current during its Deep Sleep state. In the low power Standby state, it can monitor one or more channels and respond to communications normally. The device contains a wake pin (WAKE/SPI_MOSI) output to wake the system when a touch is detected in Standby and to wake the device from Deep Sleep.

The device communicates with a host controller using the SPI bus, SMSC BC-Link bus, or via SMBus / $I^2C$. The host controller may poll the device for updated information at any time or it may configure the device to flag an interrupt whenever a touch is detected on any sensor pad.

A typical system diagram is shown in Figure 4.1.

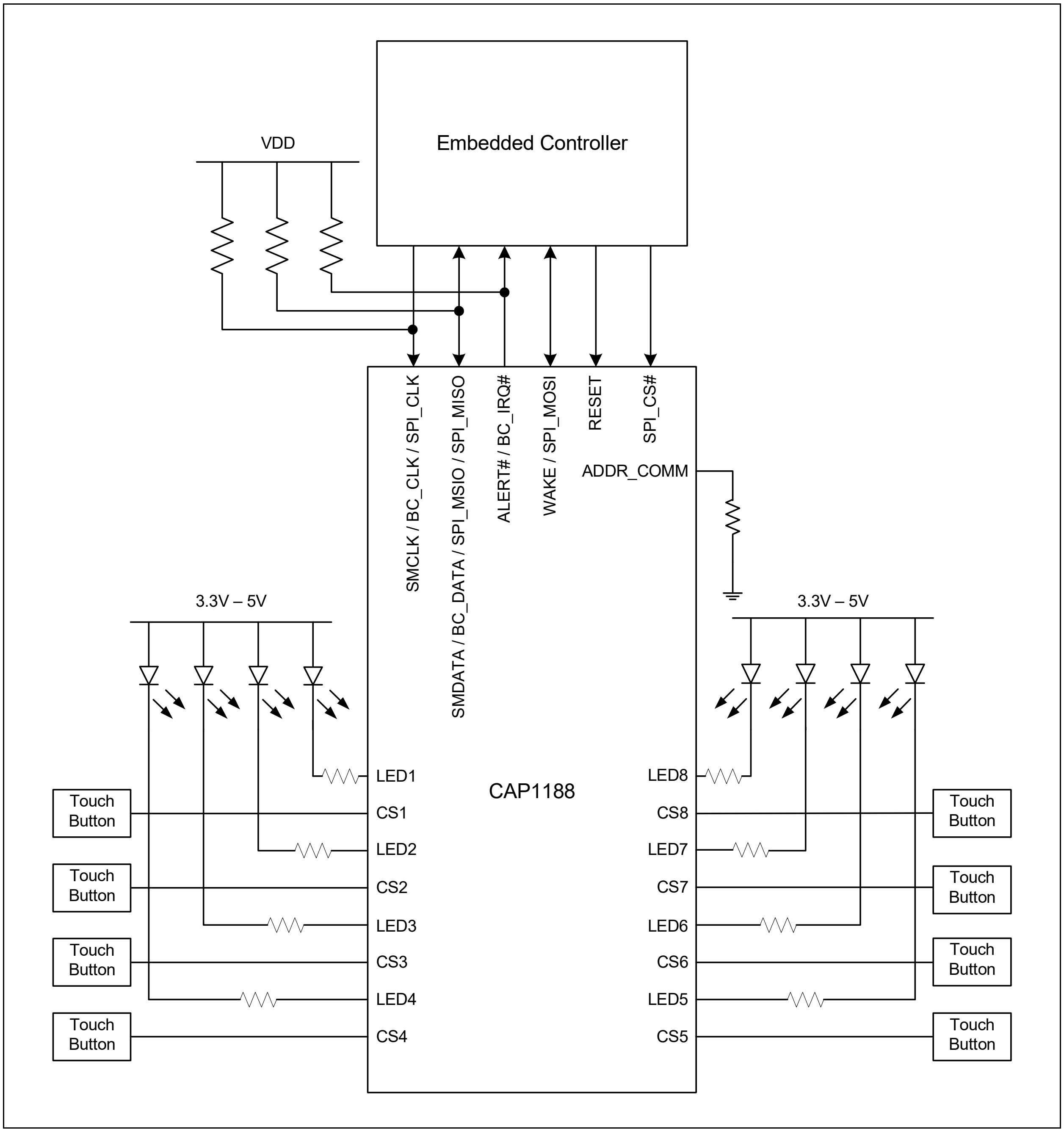



**Figure 4.1 System Diagram for CAP1188**

## 4.1 Power States

The CAP1188 has three operating states depending on the status of the STBY and DSLEEP bits. When the device transitions between power states, previously detected touches (for inactive channels) are cleared and the status bits reset.

1. Fully Active - The device is fully active. It is monitoring all active capacitive sensor inputs and driving all LED channels as defined.
2. Standby - The device is in a lower power state. It will measure a programmable number of channels using the Standby Configuration controls (see Section 5.20 through Section 5.22). Interrupts will still be generated based on the active channels. The device will still respond to communications normally and can be returned to the Fully Active state of operation by clearing the STBY bit.
3. Deep Sleep - The device is in its lowest power state. It is not monitoring any capacitive sensor inputs and not driving any LEDs. All LEDs will be driven to their programmed non-actuated state and no PWM operations will be done. While in Deep Sleep, the device can be awakened by SMBus or SPI communications targeting the device. This will not cause the DSLEEP to be cleared so the device will return to Deep Sleep once all communications have stopped.

   If the device is not communicating via the 4-wire SPI bus, then during this state of operation, if the WAKE/SPI_MOSI pin is driven high by an external source, the device will clear the DSLEEP bit and return to Fully Active.

**APPLICATION NOTE:** In the Deep Sleep state, the LED output will be either high or low and will not be PWM'd at the min or max duty cycle.

**APPLICATION NOTE:** If the CAP1188 is configured to communicate using the BC-Link protocol, the device does not support Deep Sleep.

## 4.2 RESET Pin

The RESET pin is an active high reset that is driven from an external source. While it is asserted high, all the internal blocks will be held in reset including the communications protocol used. No capacitive touch sensor inputs will be sampled and the LEDs will not be driven. All configuration settings will be reset to default states and all readings will be cleared.

The device will be held in Deep Sleep that can only be removed by driving the RESET pin low. This will cause the RESET status bit to be set to a logic '1' and generate an interrupt.

## 4.3 WAKE/SPI_MOSI Pin Operation

The WAKE / SPI_MOSI pin is a multi-function pin depending on device operation. When the device is configured to communicate using the 4-wire SPI bus, this pin is an input.

However, when the CAP1188 is placed in Standby and is not communicating using the 4-wire SPI protocol, the WAKE pin is an active high output. In this condition, the device will assert the WAKE/SPI_MOSI pin when a touch is detected on one of its sampled sensor inputs. The pin will remain asserted until the INT bit has been cleared and then it will be de-asserted.

When the CAP1188 is placed in Deep Sleep and it is not communicating using the 4-wire SPI protocol, the WAKE/SPI_MOSI pin is monitored by the device as an input. If the WAKE/SPI_MOSI pin is driven high by an external source, the CAP1188 will clear the DSLEEP bit causing the device to return to Fully Active.

When the device is placed in Deep Sleep, this pin is a High-Z input and must have a pull-down resistor to GND for proper operation.

## 4.4 LED Drivers

The CAP1188 contains eight (8) LED drivers. Each LED driver can be linked to its respective capacitive touch sensor input or it can be controlled by the host. Each LED driver can be configured to operate in one of the following modes with either push-pull or open drain drive.

1. Direct - The LED is configured to be on or off when the corresponding input stimulus is on or off (or inverted). The brightness of the LED can be programmed from full off to full on (default). Additionally, the LED contains controls to individually configure ramping on, off, and turn-off delay.
2. Pulse 1 - The LED is configured to “Pulse” (transition ON-OFF-ON) a programmable number of times with programmable rate and min / max brightness. This behavior may be actuated when a press is detected or when a release is detected.
3. Pulse 2 - The LED is configured to “Pulse” while actuated and then “Pulse” a programmable number of times with programmable rate and min / max brightness when the sensor pad is released.
4. Breathe - The LED is configured to transition continuously ON-OFF-ON (i.e. to “Breathe”) with a programmable rate and min / max brightness.

When an LED is not linked to a sensor and is actuated by the host, there’s an option to assert the ALERT# pin when the initiated LED behavior has completed.

### 4.4.1 Linking LEDs to Capacitive Touch Sensor Inputs

All LEDs can be linked to the corresponding capacitive touch sensor input so that when the sensor input detects a touch, the corresponding LED will be actuated at one of the programmed responses.

## 4.5 Capacitive Touch Sensing

The CAP1188 contains eight (8) independent capacitive touch sensor inputs. Each sensor input has dynamic range to detect a change of capacitance due to a touch. Additionally, each sensor input can be configured to be automatically and routinely re-calibrated.

### 4.5.1 Sensing Cycle

Each capacitive touch sensor input has controls to be activated and included in the sensing cycle. When the device is active, it automatically initiates a sensing cycle and repeats the cycle every time it finishes. The cycle polls through each active sensor input starting with CS1 and extending through CS8. As each capacitive touch sensor input is polled, its measurement is compared against a baseline “Not Touched” measurement. If the delta measurement is large enough, a touch is detected and an interrupt is generated.

The sensing cycle time is programmable (see Section 5.10, "Averaging and Sampling Configuration Register").

### 4.5.2 Recalibrating Sensor Inputs

There are various options for recalibrating the capacitive touch sensor inputs. Recalibration re-sets the Base Count Registers (Section 5.24, "Sensor Input Base Count Registers") which contain the “not touched” values used for touch detection comparisons.

**APPLICATION NOTE:** The device will recalibrate all sensor inputs that were disabled when it transitions from Standby. Likewise, the device will recalibrate all sensor inputs when waking out of Deep Sleep.

#### 4.5.2.1 Manual Recalibration

The Calibration Activate Registers (Section 5.11, "Calibration Activate Register") force recalibration of selected sensor inputs. When a bit is set, the corresponding capacitive touch sensor input will be recalibrated (both analog and digital). The bit is automatically cleared once the recalibration routine has finished.

**Note:** During this recalibration routine, the sensor inputs will not detect a press for up to 200ms and the Sensor Base Count Register values will be invalid. In addition, any press on the corresponding sensor pads will invalidate the recalibration.

#### 4.5.2.2 Automatic Recalibration

Each sensor input is regularly recalibrated at a programmable rate (see Section 5.17, "Recalibration Configuration Register"). By default, the recalibration routine stores the average 64 previous measurements and periodically updates the base “not touched” setting for the capacitive touch sensor input.

**Note:** Automatic recalibration only works when the delta count is below the active sensor input threshold. It is disabled when a touch is detected.

#### 4.5.2.3 Negative Delta Count Recalibration

It is possible that the device loses sensitivity to a touch. This may happen as a result of a noisy environment, an accidental recalibration during a touch, or other environmental changes. When this occurs, the base untouched sensor input may generate negative delta count values. The NEG_DELTA_CNT bits (see Section 5.17, "Recalibration Configuration Register") can be set to force a recalibration after a specified number of consecutive negative delta readings.

**Note:** During this recalibration, the device will not respond to touches.

#### 4.5.2.4 Delayed Recalibration

It is possible that a “stuck button” occurs when something is placed on a button which causes a touch to be detected for a long period. By setting the MAX_DUR_EN bit (see Section 5.6, "Configuration Registers"), a recalibration can be forced when a touch is held on a button for longer than the duration specified in the MAX_DUR bits (see Section 5.8, "Sensor Input Configuration Register").

**Note:** Delayed recalibration only works when the delta count is above the active sensor input threshold. If enabled, it is invoked when a sensor pad touch is held longer than the MAX_DUR bit setting.

### 4.5.3 Proximity Detection

Each sensor input can be configured to detect changes in capacitance due to proximity of a touch. This circuitry detects the change of capacitance that is generated as an object approaches, but does not physically touch, the enabled sensor pad(s). When a sensor input is selected to perform proximity detection, it will be sampled from 1x to 128x per sampling cycle. The larger the number of samples that are taken, the greater the range of proximity detection is available at the cost of an increased overall sampling time.

### 4.5.4 Multiple Touch Pattern Detection

The multiple touch pattern (MTP) detection circuitry can be used to detect lid closure or other similar events. An event can be flagged based on either a minimum number of sensor inputs or on specific sensor inputs simultaneously exceeding an MTP threshold or having their Noise Flag Status Register bits set. An interrupt can also be generated. During an MTP event, all touches are blocked (see Section 5.15, "Multiple Touch Pattern Configuration Register").

### 4.5.5 Low Frequency Noise Detection

Each sensor input has an EMI noise detector that will sense if low frequency noise is injected onto the input with sufficient power to corrupt the readings. If this occurs, the device will reject the corrupted sample and set the corresponding bit in the Noise Status register to a logic ‘1’.

### 4.5.6 RF Noise Detection

Each sensor input contains an integrated RF noise detector. This block will detect injected RF noise on the CS pin. The detector threshold is dependent upon the noise frequency. If RF noise is detected on a CS line, that sample is removed and not compared against the threshold.

## 4.6 ALERT# Pin

The ALERT# pin is an active low (or active high when configured) output that is driven when an interrupt event is detected.

Whenever an interrupt is generated, the INT bit (see Section 5.1, "Main Control Register") is set. The ALERT# pin is cleared when the INT bit is cleared by the user. Additionally, when the INT bit is cleared by the user, status bits are only cleared if no touch is detected.

### 4.6.1 Sensor Interrupt Behavior

The sensor interrupts are generated in one of two ways:

1. An interrupt is generated when a touch is detected and, as a user selectable option, when a release is detected (by default - see Section 5.6). See Figure 4.3.
2. If the repeat rate is enabled then, so long as the touch is held, another interrupt will be generated based on the programmed repeat rate (see Figure 4.2).

When the repeat rate is enabled, the device uses an additional control called MPRESS that determines whether a touch is flagged as a simple “touch” or a “press and hold”. The MPRESS[3:0] bits set a minimum press timer. When the button is touched, the timer begins. If the sensor pad is released before the minimum press timer expires, it is flagged as a touch and an interrupt is generated upon release. If the sensor input detects a touch for longer than this timer value, it is flagged as a “press and hold” event. So long as the touch is held, interrupts will be generated at the programmed repeat rate and upon release (if enabled).

**APPLICATION NOTE:** Figure 4.2 and Figure 4.3 show default operation which is to generate an interrupt upon sensor pad release and an active-low ALERT# pin.

**APPLICATION NOTE:** The host may need to poll the device twice to determine that a release has been detected.

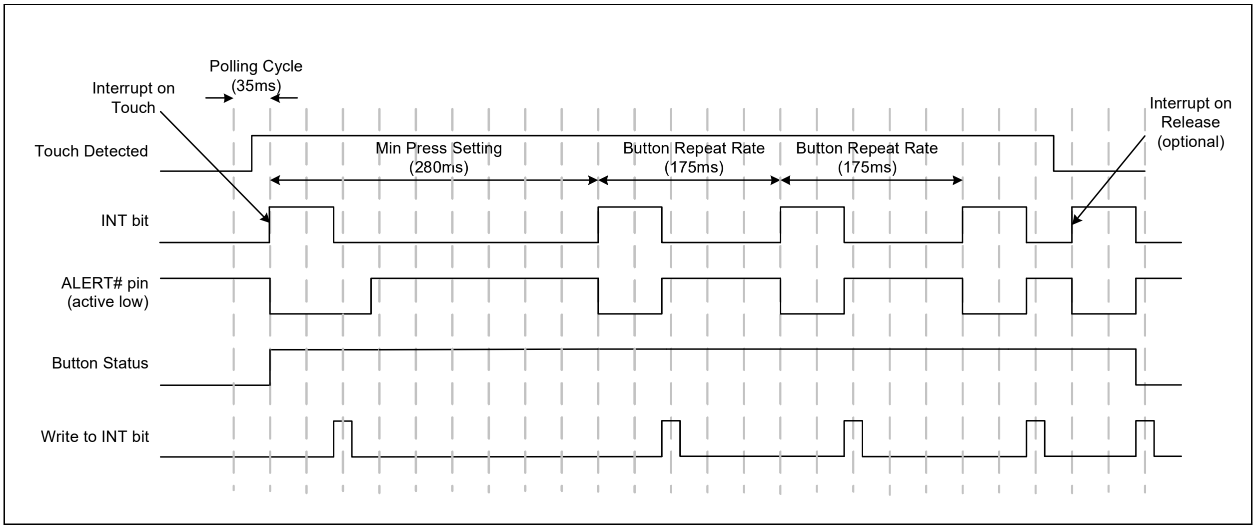



**Figure 4.2 Sensor Interrupt Behavior - Repeat Rate Enabled**

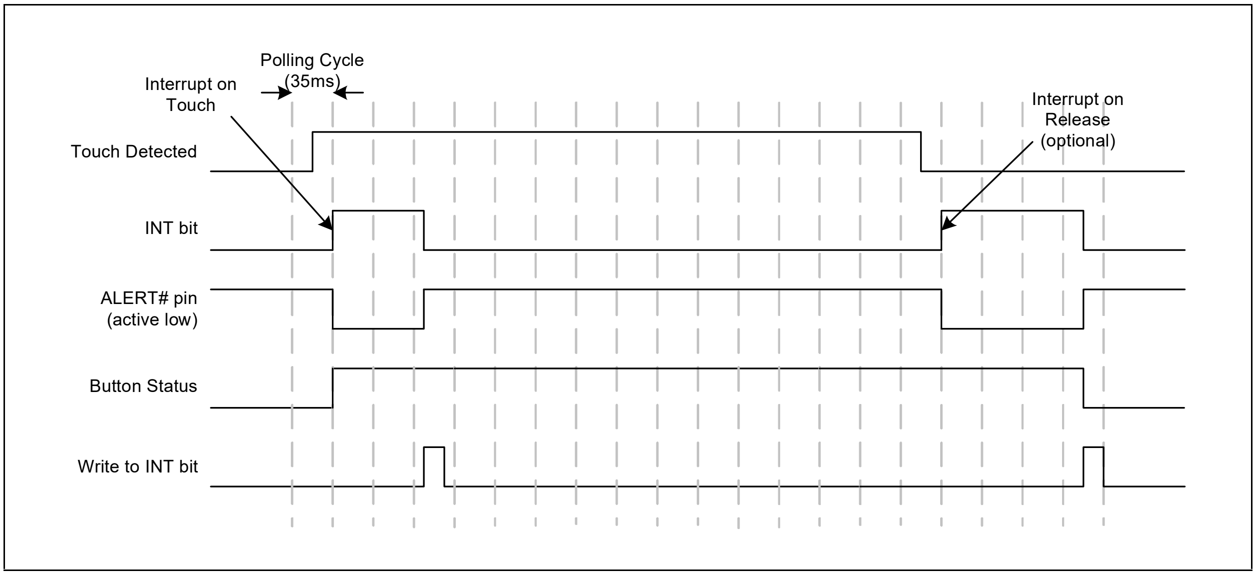



**Figure 4.3 Sensor Interrupt Behavior - No Repeat Rate Enabled**

# Chapter 5 Register Description

The registers shown in Table 5.1 are accessible through the communications protocol. An entry of '-' indicates that the bit is not used and will always read '0'.

**Table 5.1 Register Set in Hexadecimal Order**

| REGISTER ADDRESS | R/W | REGISTER NAME | FUNCTION | DEFAULT VALUE | PAGE |
|---|---|---|---|---|---|
| 00h | R/W | Main Control | Controls general power states and power dissipation | 00h | Page 41 |
| 02h | R | General Status | Stores general status bits | 00h | Page 42 |
| 03h | R | Sensor Input Status | Returns the state of the sampled capacitive touch sensor inputs | 00h | Page 42 |
| 04h | R | LED Status | Stores status bits for LEDs | 00h | Page 42 |
| 0Ah | R | Noise Flag Status | Stores the noise flags for sensor inputs | 00h | Page 44 |
| 10h | R | Sensor Input 1 Delta Count | Stores the delta count for CS1 | 00h | Page 44 |
| 11h | R | Sensor Input 2 Delta Count | Stores the delta count for CS2 | 00h | Page 44 |
| 12h | R | Sensor Input 3 Delta Count | Stores the delta count for CS3 | 00h | Page 44 |
| 13h | R | Sensor Input 4 Delta Count | Stores the delta count for CS4 | 00h | Page 44 |
| 14h | R | Sensor Input 5 Delta Count | Stores the delta count for CS5 | 00h | Page 44 |
| 15h | R | Sensor Input 6 Delta Count | Stores the delta count for CS6 | 00h | Page 44 |
| 16h | R | Sensor Input 7 Delta Count | Stores the delta count for CS7 | 00h | Page 44 |
| 17h | R | Sensor Input 8 Delta Count | Stores the delta count for CS8 | 00h | Page 44 |
| 1Fh | R/W | Sensitivity Control | Controls the sensitivity of the threshold and delta counts and data scaling of the base counts | 2Fh | Page 45 |
| 20h | R/W | Configuration | Controls general functionality | 20h | Page 47 |
| 21h | R/W | Sensor Input Enable | Controls whether the capacitive touch sensor inputs are sampled | FFh | Page 49 |
| 22h | R/W | Sensor Input Configuration | Controls max duration and auto-repeat delay for sensor inputs operating in the full power state | A4h | Page 49 |
| 23h | R/W | Sensor Input Configuration 2 | Controls the MPRESS controls for all sensor inputs | 07h | Page 51 |

**Table 5.1 Register Set in Hexadecimal Order (continued)**

| REGISTER ADDRESS | R/W | REGISTER NAME | FUNCTION | DEFAULT VALUE | PAGE |
|---|---|---|---|---|---|
| 24h | R/W | Averaging and Sampling Config | Controls averaging and sampling window | 39h | Page 52 |
| 26h | R/W | Calibration Activate | Forces re-calibration for capacitive touch sensor inputs | 00h | Page 54 |
| 27h | R/W | Interrupt Enable | Enables Interrupts associated with capacitive touch sensor inputs | FFh | Page 54 |
| 28h | R/W | Repeat Rate Enable | Enables repeat rate for all sensor inputs | FFh | Page 55 |
| 2Ah | R/W | Multiple Touch Configuration | Determines the number of simultaneous touches to flag a multiple touch condition | 80h | Page 56 |
| 2Bh | R/W | Multiple Touch Pattern Configuration | Determines the multiple touch pattern (MTP) configuration | 00h | Page 56 |
| 2Dh | R/W | Multiple Touch Pattern | Determines the pattern or number of sensor inputs used by the MTP circuitry | FFh | Page 58 |
| 2Fh | R/W | Recalibration Configuration | Determines re-calibration timing and sampling window | 8Ah | Page 58 |
| 30h | R/W | Sensor Input 1 Threshold | Stores the delta count threshold to determine a touch for Capacitive Touch Sensor Input 1 | 40h | Page 60 |
| 31h | R/W | Sensor Input 2 Threshold | Stores the delta count threshold to determine a touch for Capacitive Touch Sensor Input 2 | 40h | Page 60 |
| 32h | R/W | Sensor Input 3 Threshold | Stores the delta count threshold to determine a touch for Capacitive Touch Sensor Input 3 | 40h | Page 60 |
| 33h | R/W | Sensor Input 4 Threshold | Stores the delta count threshold to determine a touch for Capacitive Touch Sensor Input 4 | 40h | Page 60 |
| 34h | R/W | Sensor Input 5 Threshold | Stores the delta count threshold to determine a touch for Capacitive Touch Sensor Input 5 | 40h | Page 60 |
| 35h | R/W | Sensor Input 6 Threshold | Stores the delta count threshold to determine a touch for Capacitive Touch Sensor Input 6 | 40h | Page 60 |
| 36h | R/W | Sensor Input 7 Threshold | Stores the delta count threshold to determine a touch for Capacitive Touch Sensor Input 7 | 40h | Page 61 |
| 37h | R/W | Sensor Input 8 Threshold | Stores the delta count threshold to determine a touch for Capacitive Touch Sensor Input 8 | 40h | |
| 38h | R/W | Sensor Input Noise Threshold | Stores controls for selecting the noise threshold for all sensor inputs | 01h | Page 61 |

**Table 5.1 Register Set in Hexadecimal Order (continued)**

| REGISTER ADDRESS | R/W | REGISTER NAME | FUNCTION | DEFAULT VALUE | PAGE |
|---|---|---|---|---|---|
| Standby Configuration Registers | | | | | |
| 40h | R/W | Standby Channel | Controls which sensor inputs are enabled while in standby | 00h | Page 61 |
| 41h | R/W | Standby Configuration | Controls averaging and cycle time while in standby | 39h | Page 62 |
| 42h | R/W | Standby Sensitivity | Controls sensitivity settings used while in standby | 02h | Page 63 |
| 43h | R/W | Standby Threshold | Stores the touch detection threshold for active sensor inputs in standby | 40h | Page 64 |
| 44h | R/W | Configuration 2 | Stores additional configuration controls for the device | 40h | Page 47 |
| Base Count Registers | | | | | |
| 50h | R | Sensor Input 1 Base Count | Stores the reference count value for sensor input 1 | C8h | Page 65 |
| 51h | R | Sensor Input 2 Base Count | Stores the reference count value for sensor input 2 | C8h | Page 65 |
| 52h | R | Sensor Input 3 Base Count | Stores the reference count value for sensor input 3 | C8h | Page 65 |
| 53h | R | Sensor Input 4 Base Count | Stores the reference count value for sensor input 4 | C8h | Page 65 |
| 54h | R | Sensor Input 5 Base Count | Stores the reference count value for sensor input 5 | C8h | Page 65 |
| 55h | R | Sensor Input 6 Base Count | Stores the reference count value for sensor input 6 | C8h | Page 65 |
| 56h | R | Sensor Input 7 Base Count | Stores the reference count value for sensor input 7 | C8h | Page 65 |
| 57h | R | Sensor Input 8 Base Count | Stores the reference count value for sensor input 8 | C8h | Page 65 |
| LED Controls | | | | | |
| 71h | R/W | LED Output Type | Controls the output type for the LED outputs | 00h | Page 65 |
| 72h | R/W | Sensor Input LED Linking | Controls linking of sensor inputs to LED channels | 00h | Page 66 |
| 73h | R/W | LED Polarity | Controls the output polarity of LEDs | 00h | Page 67 |
| 74h | R/W | LED Output Control | Controls the output state of the LEDs | 00h | Page 68 |
| 77h | R/W | LED Linked Transition Control | Controls the transition when LEDs are linked to CS channels | 00h | Page 69 |
| 79h | R/W | LED Mirror Control | Controls the mirroring of duty cycles for the LEDs | 00h | Page 70 |

**Table 5.1 Register Set in Hexadecimal Order (continued)**

| REGISTER ADDRESS | R/W | REGISTER NAME | FUNCTION | DEFAULT VALUE | PAGE |
|---|---|---|---|---|---|
| 81h | R/W | LED Behavior 1 | Controls the behavior and response of LEDs 1 - 4 | 00h | Page 71 |
| 82h | R/W | LED Behavior 2 | Controls the behavior and response of LEDs 5 - 8 | 00h | Page 71 |
| 84h | R/W | LED Pulse 1 Period | Controls the period of each breathe during a pulse | 20h | Page 73 |
| 85h | R/W | LED Pulse 2 Period | Controls the period of the breathing during breathe and pulse operation | 14h | Page 74 |
| 86h | R/W | LED Breathe Period | Controls the period of an LED breathe operation | 5Dh | Page 75 |
| 88h | R/W | LED Config | Controls LED configuration | 04h | Page 76 |
| 90h | R/W | LED Pulse 1 Duty Cycle | Determines the min and max duty cycle for the pulse operation | F0h | Page 77 |
| 91h | R/W | LED Pulse 2 Duty Cycle | Determines the min and max duty cycle for breathe and pulse operation | F0h | Page 77 |
| 92h | R/W | LED Breathe Duty Cycle | Determines the min and max duty cycle for the breathe operation | F0h | Page 77 |
| 93h | R/W | LED Direct Duty Cycle | Determines the min and max duty cycle for Direct mode LED operation | F0h | Page 77 |
| 94h | R/W | LED Direct Ramp Rates | Determines the rising and falling edge ramp rates of the LEDs | 00h | Page 78 |
| 95h | R/W | LED Off Delay | Determines the off delay for all LED behaviors | 00h | Page 79 |
| B1h | R | Sensor Input 1 Calibration | Stores the upper 8-bit calibration value for sensor input 1 | 00h | Page 83 |
| B2h | R | Sensor Input 2 Calibration | Stores the upper 8-bit calibration value for sensor input 2 | 00h | Page 83 |
| B3h | R | Sensor Input 3 Calibration | Stores the upper 8-bit calibration value for sensor input 3 | 00h | Page 83 |
| B4h | R | Sensor Input 4 Calibration | Stores the upper 8-bit calibration value for sensor input 4 | 00h | Page 83 |
| B5h | R | Sensor Input 5 Calibration | Stores the upper 8-bit calibration value for sensor input 5 | 00h | Page 83 |
| B6h | R | Sensor Input 6 Calibration | Stores the upper 8-bit calibration value for sensor input 6 | 00h | Page 83 |
| B7h | R | Sensor Input 7 Calibration | Stores the upper 8-bit calibration value for sensor input 7 | 00h | Page 83 |
| B8h | R | Sensor Input 8 Calibration | Stores the upper 8-bit calibration value for sensor input 8 | 00h | Page 83 |

**Table 5.1 Register Set in Hexadecimal Order (continued)**

| REGISTER ADDRESS | R/W | REGISTER NAME | FUNCTION | DEFAULT VALUE | PAGE |
|---|---|---|---|---|---|
| B9h | R | Sensor Input Calibration LSB 1 | Stores the 2 LSBs of the calibration value for sensor inputs 1 - 4 | 00h | Page 83 |
| BAh | R | Sensor Input Calibration LSB 2 | Stores the 2 LSBs of the calibration value for sensor inputs 5 - 8 | 00h | Page 83 |
| FDh | R | Product ID | Stores a fixed value that identifies each product | 50h | Page 83 |
| FEh | R | Manufacturer ID | Stores a fixed value that identifies SMSC | 5Dh | Page 84 |
| FFh | R | Revision | Stores a fixed value that represents the revision number | 83h | Page 84 |

During Power-On-Reset (POR), the default values are stored in the registers. A POR is initiated when power is first applied to the part and the voltage on the VDD supply surpasses the POR level as specified in the electrical characteristics. Any reads to undefined registers will return 00h. Writes to undefined registers will not have an effect.

When a bit is “set”, this means that the user writes a logic ‘1’ to it. When a bit is “cleared”, this means that the user writes a logic ‘0’ to it.

## 5.1 Main Control Register

**Table 5.2 Main Control Register**

| ADDR | R/W | REGISTER | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 | DEFAULT |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 00h | R/W | Main Control | GAIN[1:0] | | STBY | DSLEEP | - | - | - | INT | 00h |

The Main Control register controls the primary power state of the device.

Bits 7 - 6 - GAIN[1:0] - Controls the gain used by the capacitive touch sensing circuitry. As the gain is increased, the effective sensitivity is likewise increased as a smaller delta capacitance is required to generate the same delta count values. The sensitivity settings may need to be adjusted along with the gain settings such that data overflow does not occur.

**APPLICATION NOTE:** The gain settings apply to both Standby and Active states.

**Table 5.3 GAIN Bit Decode**

| GAIN[1:0] | | CAPACITIVE TOUCH SENSOR GAIN |
|---|---|---|
| 1 | 0 | |
| 0 | 0 | 1 |
| 0 | 1 | 2 |
| 1 | 0 | 4 |
| 1 | 1 | 8 |

Bit 5 - STBY - Enables Standby.

- ‘0’ (default) - Sensor input scanning is active and LEDs are functional.
- ‘1’ - Capacitive touch sensor input scanning is limited to the sensor inputs set in the Standby Channel register (see Section 5.20). The status registers will not be cleared until read. LEDs that are linked to capacitive touch sensor inputs will remain linked and active. Sensor inputs that are no longer sampled will flag a release and then remain in a non-touched state. LEDs that are manually controlled will be unaffected.

Bit 4 - DSLEEP - Enables Deep Sleep by deactivating all functions. This bit will be cleared when the WAKE pin is driven high. If the CAP1188 is configured to communicate using the BC-Link protocol, this bit is ignored.

- ‘0’ (default) - Sensor input scanning is active and LEDs are functional.
- ‘1’ - All sensor input scanning is disabled. All LEDs are driven to their programmed non-actuated state and no PWM operations will be done. The status registers are automatically cleared and the INT bit is cleared.

Bit 0 - INT - Indicates that there is an interrupt. When this bit is set, it asserts the ALERT# pin. If a channel detects a touch and its associated interrupt enable bit is not set to a logic ‘1’, no action is taken.

This bit is cleared by writing a logic ‘0’ to it. When this bit is cleared, the ALERT# pin will be deasserted and all status registers will be cleared if the condition has been removed. If the WAKE/SPI_MOSI pin is asserted as a result of a touch detected while in Standby, it will likewise be deasserted when this bit is cleared.

Note that the WAKE / SPI_MOSI pin is not driven when communicating via the 4-wire SPI protocol.

- ‘0’ - No interrupt pending.
- ‘1’ - A touch has been detected on one or more channels and the interrupt has been asserted.

## 5.2 Status Registers

**Table 5.4 Status Registers**

| ADDR | R/W | REGISTER | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 | DEFAULT |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 02h | R | General Status | - | - | - | LED | RESET | MULT | MTP | TOUCH | 00h |
| 03h | R | Sensor Input Status | CS8 | CS7 | CS6 | CS5 | CS4 | CS3 | CS2 | CS1 | 00h |
| 04h | R | LED Status | LED8_ DN | LED7_ DN | LED6_ DN | LED5_ DN | LED4_ DN | LED3_ DN | LED2_ DN | LED1_ DN | 00h |

All status bits are cleared when the device enters the Deep Sleep (DSLEEP = ‘1’ - see Section 5.1).

### 5.2.1 General Status - 02h

Bit 4 - LED - Indicates that one or more LEDs have finished their programmed activity. This bit is set if any bit in the LED Status register is set.

Bit 3 - RESET - Indicates that the device has come out of reset. This bit is set when the device exits a POR state or when the RESET pin has been deasserted and qualified via the RESET pin filter (see Section 4.2). This bit will cause the INT bit to be set and is cleared when the INT bit is cleared.

Bit 2 - MULT - Indicates that the device is blocking detected touches due to the Multiple Touch detection circuitry (see Section 5.14). This bit will not cause the INT bit to be set and hence will not cause an interrupt.

Bit 1 - MTP - Indicates that the device has detected a number of sensor inputs that exceed the MTP threshold either via the pattern recognition or via the number of sensor inputs (see Section 5.15). This bit will cause the INT bit to be set if the MTP_ALERT bit is also set. This bit will not be cleared until the condition that caused it to be set has been removed.

Bit 0 - TOUCH - Indicates that a touch was detected. This bit is set if any bit in the Sensor Input Status register is set.

### 5.2.2 Sensor Input Status - 03h

The Sensor Input Status Register stores status bits that indicate a touch has been detected. A value of ‘0’ in any bit indicates that no touch has been detected. A value of ‘1’ in any bit indicates that a touch has been detected.

All bits are cleared when the INT bit is cleared and if a touch on the respective capacitive touch sensor input is no longer present. If a touch is still detected, the bits will not be cleared (but this will not cause the interrupt to be asserted - see Section 5.6).

Bit 7 - CS8 - Indicates that a touch was detected on Sensor Input 8. This sensor input can be linked to LED8.

Bit 6 - CS7 - Indicates that a touch was detected on Sensor Input 7. This sensor input can be linked to LED7.

Bit 5 - CS6 - Indicates that a touch was detected on Sensor Input 6. This sensor input can be linked to LED6.

Bit 4 - CS5 - Indicates that a touch was detected on Sensor Input 5. This sensor input can be linked to LED5.

Bit 3 - CS4 - Indicates that a touch was detected on Sensor Input 4. This sensor input can be linked to LED4.

Bit 2 - CS3 - Indicates that a touch was detected on Sensor Input 3. This sensor input can be linked to LED3.

Bit 1 - CS2 - Indicates that a touch was detected on Sensor Input 2. This sensor input can be linked to LED2.

Bit 0 - CS1 - Indicates that a touch was detected on Sensor Input 1. This sensor input can be linked to LED1.

### 5.2.3 LED Status - 04h

The LED Status Registers indicate when an LED has completed its configured behavior (see Section 5.31, "LED Behavior Registers") after being actuated by the host (see Section 5.28, "LED Output Control Register"). These bits are ignored when the LED is linked to a capacitive sensor input. All LED Status bits are cleared when the INT bit is cleared.

Bit 7 - LED8_DN - Indicates that LED8 has finished its behavior after being actuated by the host.

Bit 6 - LED7_DN - Indicates that LED7 has finished its behavior after being actuated by the host.

Bit 5 - LED6_DN - Indicates that LED6 has finished its behavior after being actuated by the host.

Bit 4 - LED5_DN - Indicates that LED5 has finished its behavior after being actuated by the host.

Bit 3 - LED4_DN - Indicates that LED4 has finished its behavior after being actuated by the host.

Bit 2 - LED3_DN - Indicates that LED3 has finished its behavior after being actuated by the host.

Bit 1 - LED2_DN - Indicates that LED2 has finished its behavior after being actuated by the host.

Bit 0 - LED1_DN - Indicates that LED1 has finished its behavior after being actuated by the host.

## 5.3 Noise Flag Status Registers

**Table 5.5 Noise Flag Status Registers**

| ADDR | R/W | REGISTER | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 | DEFAULT |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0Ah | R | Noise Flag Status | CS8_ NOISE | CS7_ NOISE | CS6_ NOISE | CS5_ NOISE | CS4_ NOISE | CS3_ NOISE | CS2_ NOISE | CS1_ NOISE | 00h |

The Noise Flag Status registers store status bits that are generated from the analog block if the detected noise is above the operating region of the analog detector or the RF noise detector. These bits indicate that the most recently received data from the sensor input is invalid and should not be used for touch detection. So long as the bit is set for a particular channel, the delta count value is reset to 00h and thus no touch is detected.

These bits are not sticky and will be cleared automatically if the analog block does not report a noise error.

**APPLICATION NOTE:** If the MTP detection circuitry is enabled, these bits count as sensor inputs above the MTP threshold (see Section 4.5.4, "Multiple Touch Pattern Detection") even if the corresponding delta count is not. If the corresponding delta count also exceeds the MTP threshold, it is not counted twice.

**APPLICATION NOTE:** Regardless of the state of the Noise Status bits, if low frequency noise is detected on a sensor input, that sample will be discarded unless the DIS_ANA_NOISE bit is set. As well, if RF noise is detected on a sensor input, that sample will be discarded unless the DIS_RF_NOISE bit is set.

## 5.4 Sensor Input Delta Count Registers

**Table 5.6 Sensor Input Delta Count Registers**

| ADDR | R/W | REGISTER | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 | DEFAULT |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 10h | R | Sensor Input 1 Delta Count | Sign | 64 | 32 | 16 | 8 | 4 | 2 | 1 | 00h |
| 11h | R | Sensor Input 2 Delta Count | Sign | 64 | 32 | 16 | 8 | 4 | 2 | 1 | 00h |
| 12h | R | Sensor Input 3 Delta Count | Sign | 64 | 32 | 16 | 8 | 4 | 2 | 1 | 00h |
| 13h | R | Sensor Input 4 Delta Count | Sign | 64 | 32 | 16 | 8 | 4 | 2 | 1 | 00h |
| 14h | R | Sensor Input 5 Delta Count | Sign | 64 | 32 | 16 | 8 | 4 | 2 | 1 | 00h |
| 15h | R | Sensor Input 6 Delta Count | Sign | 64 | 32 | 16 | 8 | 4 | 2 | 1 | 00h |

**Table 5.6 Sensor Input Delta Count Registers (continued)**

| ADDR | R/W | REGISTER | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 | DEFAULT |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 16h | R | Sensor Input 7 Delta Count | Sign | 64 | 32 | 16 | 8 | 4 | 2 | 1 | 00h |
| 17h | R | Sensor Input 8 Delta Count | Sign | 64 | 32 | 16 | 8 | 4 | 2 | 1 | 00h |

The Sensor Input Delta Count registers store the delta count that is compared against the threshold used to determine if a touch has been detected. The count value represents a change in input due to the capacitance associated with a touch on one of the sensor inputs and is referenced to a calibrated base “Not Touched” count value. The delta is an instantaneous change and is updated once per sensor input per sensing cycle (see Section 4.5.1, "Sensing Cycle").

The value presented is a standard 2’s complement number. In addition, the value is capped at a value of 7Fh. A reading of 7Fh indicates that the sensitivity settings are too high and should be adjusted accordingly (see Section 5.5).

The value is also capped at a negative value of 80h for negative delta counts which may result upon a release.

## 5.5 Sensitivity Control Register

**Table 5.7 Sensitivity Control Register**

| ADDR | R/W | REGISTER | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 | DEFAULT |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1Fh | R/W | Sensitivity Control | - | DELTA_SENSE[2:0] | | | BASE_SHIFT[3:0] | | | | 2Fh |

The Sensitivity Control register controls the sensitivity of a touch detection.

Bits 6-4 DELTA_SENSE[2:0] - Controls the sensitivity of a touch detection. The sensitivity settings act to scale the relative delta count value higher or lower based on the system parameters. A setting of 000b is the most sensitive while a setting of 111b is the least sensitive. At the more sensitive settings, touches are detected for a smaller delta capacitance corresponding to a “lighter” touch. These settings are more sensitive to noise, however, and a noisy environment may flag more false touches with higher sensitivity levels.

**APPLICATION NOTE:** A value of 128x is the most sensitive setting available. At the most sensitivity settings, the MSB of the Delta Count register represents 64 out of ~25,000 which corresponds to a touch of approximately 0.25% of the base capacitance (or a $\Delta$C of 25fF from a 10pF base capacitance). Conversely, a value of 1x is the least sensitive setting available. At these settings, the MSB of the Delta Count register corresponds to a delta count of 8192 counts out of ~25,000 which corresponds to a touch of approximately 33% of the base capacitance (or a $\Delta$C of 3.33pF from a 10pF base capacitance).

**Table 5.8 DELTA_SENSE Bit Decode**

| DELTA_SENSE[2:0] | | | SENSITIVITY MULTIPLIER |
|---|---|---|---|
| 2 | 1 | 0 | |
| 0 | 0 | 0 | 128x (most sensitive) |

**Table 5.8 DELTA_SENSE Bit Decode (continued)**

| DELTA_SENSE[2:0] | | | SENSITIVITY MULTIPLIER |
|---|---|---|---|
| **2** | **1** | **0** | |
| 0 | 0 | 1 | 64x |
| 0 | 1 | 0 | 32x (default) |
| 0 | 1 | 1 | 16x |
| 1 | 0 | 0 | 8x |
| 1 | 0 | 1 | 4x |
| 1 | 1 | 0 | 2x |
| 1 | 1 | 1 | 1x - (least sensitive) |

Bits 3 - 0 - BASE_SHIFT[3:0] - Controls the scaling and data presentation of the Base Count registers. The higher the value of these bits, the larger the range and the lower the resolution of the data presented. The scale factor represents the multiplier to the bit-weighting presented in these register descriptions.

**APPLICATION NOTE:** The BASE_SHIFT[3:0] bits normally do not need to be updated. These settings will not affect touch detection or sensitivity. These bits are sometimes helpful in analyzing the Cap Sensing board performance and stability.

**Table 5.9 BASE_SHIFT Bit Decode**

| BASE_SHIFT[3:0] | | | | DATA SCALING FACTOR |
|---|---|---|---|---|
| **3** | **2** | **1** | **0** | |
| 0 | 0 | 0 | 0 | 1x |
| 0 | 0 | 0 | 1 | 2x |
| 0 | 0 | 1 | 0 | 4x |
| 0 | 0 | 1 | 1 | 8x |
| 0 | 1 | 0 | 0 | 16x |
| 0 | 1 | 0 | 1 | 32x |
| 0 | 1 | 1 | 0 | 64x |
| 0 | 1 | 1 | 1 | 128x |
| 1 | 0 | 0 | 0 | 256x |
| All others | | | | 256x (default = 1111b) |

## 5.6 Configuration Registers

**Table 5.10 Configuration Registers**

| ADDR | R/W | REGISTER | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 | DEFAULT |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 20h | R/W | Configuration | TIMEOUT | WAKE_ CFG | DIS_ DIG_ NOISE | DIS_ ANA_ NOISE | MAX_ DUR_EN | - | - | - | A0h (rev B) 20h (rev C) |
| 44h | R/W | Configuration 2 | INV_LINK_ TRAN | ALT_ POL | BLK_PWR_ CTRL | BLK_POL_ MIR | SHOW_ RF_ NOISE | DIS_ RF_ NOISE | - | INT_ REL_n | 40h |

The Configuration registers control general global functionality that affects the entire device.

### 5.6.1 Configuration - 20h

Bit 7 - TIMEOUT - Enables the timeout and idle functionality of the SMBus protocol.

- ‘0’ (default for Functional Revision C) - The SMBus timeout and idle functionality are disabled. The SMBus interface will not time out if the clock line is held low. Likewise, it will not reset if both the data and clock lines are held high for longer than 200us. This is used for $I^2C$ compliance.
- ‘1’ (default for Functional Revision B) - The SMBus timeout and idle functionality are enabled. The SMBus interface will time out if the clock line is held low for longer than 30ms. Likewise, it will reset if both the data and clock lines are held high for longer than 200us.

Bit 6 - WAKE_CFG - Configures the operation of the WAKE pin.

- ‘0’ (default) - The WAKE pin is not asserted when a touch is detected while the device is in Standby. It will still be used to wake the device from Deep Sleep when driven high.
- ‘1’ - The WAKE pin will be asserted high when a touch is detected while the device is in Standby. It will also be used to wake the device from Deep Sleep when driven high.

Bit 5 - DIS_DIG_NOISE - Determines whether the digital noise threshold (see Section 5.19, "Sensor Input Noise Threshold Register") is used by the device. Setting this bit disables the feature.

- ‘0’ - The digital noise threshold is used. If a delta count value exceeds the noise threshold but does not exceed the touch threshold, the sample is discarded and not used for the automatic re-calibration routine.
- ‘1’ (default) - The noise threshold is disabled. Any delta count that is less than the touch threshold is used for the automatic re-calibration routine.

Bit 4 - DIS_ANA_NOISE - Determines whether the analog noise filter is enabled. Setting this bit disables the feature.

- ‘0’ (default) - If low frequency noise is detected by the analog block, the delta count on the corresponding channel is set to 0. Note that this does not require that Noise Status bits be set.
- ‘1’ - A touch is not blocked even if low frequency noise is detected.

Bit 3 - MAX_DUR_EN - Determines whether the maximum duration recalibration is enabled.

- ‘0’ (default) - The maximum duration recalibration functionality is disabled. A touch may be held indefinitely and no re-calibration will be performed on any sensor input.
- ‘1’ - The maximum duration recalibration functionality is enabled. If a touch is held for longer than the MAX_DUR bit settings, then the re-calibration routine will be restarted (see Section 5.8).

### 5.6.2 Configuration 2 - 44h

Bit 7 - INV_LINK_TRAN - Determines the behavior of the Linked LED Transition controls (see Section 5.29).

- ‘0’ (default) - The Linked LED Transition controls set the min duty cycle equal to the max duty cycle.
- ‘1’ - The Linked LED Transition controls will invert the touch signal. For example, a touch signal will be inverted to a non-touched signal.

Bit 6 - ALT_POL - Determines the ALERT# pin polarity and behavior.

- ‘0’ - The ALERT# pin is active high and push-pull.
- ‘1’ (default) - The ALERT# pin is active low and open drain.

Bit 5 - BLK_PWR_CTRL - Determines whether the device will reduce power consumption while waiting between conversion time completion and the end of the polling cycle.

- ‘0’ (default) - The device will always power down as much as possible during the time between the end of the last conversion and the end of the polling cycle.
- ‘1’ - The device will not power down the Cap Sensor during the time between the end of the last conversion and the end of the polling cycle.

Bit 4 - BLK_POL_MIR - Determines whether the LED Mirror Control register bits are linked to the LED Polarity bits. Setting this bit blocks the normal behavior which is to automatically set and clear the LED Mirror Control bits when the LED Polarity bits are set or cleared.

- ‘0’ (default) - When the LED Polarity controls are set, the corresponding LED Mirror control is automatically set. Likewise, when the LED Polarity controls are cleared, the corresponding LED Mirror control is also cleared.
- ‘1’ - When the LED Polarity controls are set, the corresponding LED Mirror control is not automatically set.

Bit 3 - SHOW_RF_NOISE - Determines whether the Noise Status bits will show RF Noise as the only input source.

- ‘0’ (default) - The Noise Status registers will show both RF noise and low frequency EMI noise if either is detected on a capacitive touch sensor input.
- ‘1’ - The Noise Status registers will only show RF noise if it is detected on a capacitive touch sensor input. EMI noise will still be detected and touches will be blocked normally; however, the status bits will not be updated.

Bit 2 - DIS_RF_NOISE - Determines whether the RF noise filter is enabled. Setting this bit disables the feature.

- ‘0’ (default) - If RF noise is detected by the analog block, the delta count on the corresponding channel is set to 0. Note that this does not require that Noise Status bits be set.
- ‘1’ - A touch is not blocked even if RF noise is detected.

Bit 0 - INT_REL_n - Controls the interrupt behavior when a release is detected on a button.

- ‘0’ (default) - An interrupt is generated when a press is detected and again when a release is detected and at the repeat rate (if enabled - see Section 5.13).
- ‘1’ - An interrupt is generated when a press is detected and at the repeat rate but not when a release is detected.

## 5.7 Sensor Input Enable Registers

**Table 5.11 Sensor Input Enable Registers**

| ADDR | R/W | REGISTER | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 | DEFAULT |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 21h | R/W | Sensor Input Enable | CS8_EN | CS7_EN | CS6_EN | CS5_EN | CS4_EN | CS3_EN | CS2_EN | CS1_EN | FFh |

The Sensor Input Enable registers determine whether a capacitive touch sensor input is included in the sampling cycle. The length of the sampling cycle is not affected by the number of sensor inputs measured.

Bit 7 - CS8_EN - Enables the CS8 input to be included during the sampling cycle.

- ‘0’ - The CS8 input is not included in the sampling cycle.
- ‘1’ (default) - The CS8 input is included in the sampling cycle.

Bit 6 - CS7_EN - Enables the CS7 input to be included during the sampling cycle.

Bit 5 - CS6_EN - Enables the CS6 input to be included during the sampling cycle.

Bit 4 - CS5_EN - Enables the CS5 input to be included during the sampling cycle.

Bit 3 - CS4_EN - Enables the CS4 input to be included during the sampling cycle.

Bit 2 - CS3_EN - Enables the CS3 input to be included during the sampling cycle.

Bit 1 - CS2_EN - Enables the CS2 input to be included during the sampling cycle.

Bit 0 - CS1_EN - Enables the CS1 input to be included during the sampling cycle.

## 5.8 Sensor Input Configuration Register

**Table 5.12 Sensor Input Configuration Register**

| ADDR | R/W | REGISTER | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 | DEFAULT |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 22h | R/W | Sensor Input Configuration | MAX_DUR[3:0] | | | | RPT_RATE[3:0] | | | | A4h |

The Sensor Input Configuration Register controls timings associated with the Capacitive sensor inputs 1 - 8.

Bits 7 - 4 - MAX_DUR[3:0] - (default 1010b) - Determines the maximum time that a sensor pad is allowed to be touched until the capacitive touch sensor input is recalibrated, as shown in Table 5.13.

**Table 5.13 MAX_DUR Bit Decode**

| MAX_DUR[3:0] | | | | TIME BEFORE RECALIBRATION |
|---|---|---|---|---|
| 3 | 2 | 1 | 0 | |
| 0 | 0 | 0 | 0 | 560ms |
| 0 | 0 | 0 | 1 | 840ms |

**Table 5.13 MAX_DUR Bit Decode (continued)**

| MAX_DUR[3:0] | | | | TIME BEFORE RECALIBRATION |
|---|---|---|---|---|
| 3 | 2 | 1 | 0 | |
| 0 | 0 | 1 | 0 | 1120ms |
| 0 | 0 | 1 | 1 | 1400ms |
| 0 | 1 | 0 | 0 | 1680ms |
| 0 | 1 | 0 | 1 | 2240ms |
| 0 | 1 | 1 | 0 | 2800ms |
| | 1 | 1 | 1 | 3360ms |
| 1 | 0 | 0 | 0 | 3920ms |
| 1 | 0 | 0 | 1 | 4480ms |
| 1 | 0 | 1 | 0 | 5600ms (default) |
| 1 | 0 | 1 | 1 | 6720ms |
| 1 | 1 | 0 | 0 | 7840ms |
| 1 | 1 | 0 | 1 | 8906ms |
| 1 | 1 | 1 | 0 | 10080ms |
| 1 | 1 | 1 | 1 | 11200ms |

Bits 3 - 0 - RPT_RATE[3:0] - (default 0100b) Determines the time duration between interrupt assertions when auto repeat is enabled. The resolution is 35ms the range is from 35ms to 560ms as shown in Table 5.14.

**Table 5.14 RPT_RATE Bit Decode**

| RPT_RATE[3:0] | | | | INTERRUPT REPEAT RATE |
|---|---|---|---|---|
| 3 | 2 | 1 | 0 | |
| 0 | 0 | 0 | 0 | 35ms |
| 0 | 0 | 0 | 1 | 70ms |
| 0 | 0 | 1 | 0 | 105ms |
| 0 | 0 | 1 | 1 | 140ms |
| 0 | 1 | 0 | 0 | 175ms (default) |
| 0 | 1 | 0 | 1 | 210ms |
| 0 | 1 | 1 | 0 | 245ms |
| 0 | 1 | 1 | 1 | 280ms |
| 1 | 0 | 0 | 0 | 315ms |
| 1 | 0 | 0 | 1 | 350ms |

**Table 5.14 RPT_RATE Bit Decode (continued)**

| RPT_RATE[3:0] | | | | INTERRUPT REPEAT RATE |
|---|---|---|---|---|
| **3** | **2** | **1** | **0** | |
| 1 | 0 | 1 | 0 | 385ms |
| 1 | 0 | 1 | 1 | 420ms |
| 1 | 1 | 0 | 0 | 455ms |
| 1 | 1 | 0 | 1 | 490ms |
| 1 | 1 | 1 | 0 | 525ms |
| 1 | 1 | 1 | 1 | 560ms |

## 5.9 Sensor Input Configuration 2 Register

**Table 5.15 Sensor Input Configuration 2 Register**

| ADDR | R/W | REGISTER | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 | DEFAULT |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 23h | R/W | Sensor Input Configuration 2 | - | - | - | - | M_PRESS[3:0] | | | | 07h |

Bits 3 - 0 - M_PRESS[3:0] - (default 0111b) - Determines the minimum amount of time that sensor inputs configured to use auto repeat must detect a sensor pad touch to detect a “press and hold” event. If the sensor input detects a touch for longer than the M_PRESS[3:0] settings, a “press and hold” event is detected. If a sensor input detects a touch for less than or equal to the M_PRESS[3:0] settings, a touch event is detected.

The resolution is 35ms the range is from 35ms to 560ms as shown in Table 5.16.

**Table 5.16 M_PRESS Bit Decode**

| M_PRESS[3:0] | | | | M_PRESS SETTINGS |
|---|---|---|---|---|
| **3** | **2** | **1** | **0** | |
| 0 | 0 | 0 | 0 | 35ms |
| 0 | 0 | 0 | 1 | 70ms |
| 0 | 0 | 1 | 0 | 105ms |
| 0 | 0 | 1 | 1 | 140ms |
| 0 | 1 | 0 | 0 | 175ms |
| 0 | 1 | 0 | 1 | 210ms |
| 0 | 1 | 1 | 0 | 245ms |
| 0 | 1 | 1 | 1 | 280ms (default) |
| 1 | 0 | 0 | 0 | 315ms |

**Table 5.16 M_PRESS Bit Decode (continued)**

| M_PRESS[3:0] | | | | M_PRESS SETTINGS |
|---|---|---|---|---|
| **3** | **2** | **1** | **0** | |
| 1 | 0 | 0 | 1 | 350ms |
| 1 | 0 | 1 | 0 | 385ms |
| 1 | 0 | 1 | 1 | 420ms |
| 1 | 1 | 0 | 0 | 455ms |
| 1 | 1 | 0 | 1 | 490ms |
| 1 | 1 | 1 | 0 | 525ms |
| 1 | 1 | 1 | 1 | 560ms |

## 5.10 Averaging and Sampling Configuration Register

**Table 5.17 Averaging and Sampling Configuration Register**

| ADDR | R/W | REGISTER | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 | DEFAULT |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 24h | R/W | Averaging and Sampling Config | | AVG[2:0] | | | SAMP_TIME[1:0] | | CYCLE_TIME [1:0] | | 39h |

The Averaging and Sampling Configuration register controls the number of samples taken and the total sensor input cycle time for all active sensor inputs while the device is functioning in Active state.

Bits 6 - 4 - AVG[2:0] - Determines the number of samples that are taken for all active channels during the sensor cycle as shown in Table 5.18. All samples are taken consecutively on the same channel before the next channel is sampled and the result is averaged over the number of samples measured before updating the measured results.

For example, if CS1, CS2, and CS3 are sampled during the sensor cycle, and the AVG[2:0] bits are set to take 4 samples per channel, then the full sensor cycle will be: CS1, CS1, CS1, CS1, CS2, CS2, CS2, CS2, CS3, CS3, CS3, CS3.

**Table 5.18 AVG Bit Decode**

| AVG[2:0] | | | NUMBER OF SAMPLES TAKEN PER MEASUREMENT |
|---|---|---|---|
| **2** | **1** | **0** | |
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 2 |
| 0 | 1 | 0 | 4 |
| 0 | 1 | 1 | 8 (default) |
| 1 | 0 | 0 | 16 |

**Table 5.18 AVG Bit Decode (continued)**

| AVG[2:0] | | | NUMBER OF SAMPLES TAKEN PER MEASUREMENT |
|---|---|---|---|
| **2** | **1** | **0** | |
| 1 | 0 | 1 | 32 |
| 1 | 1 | 0 | 64 |
| 1 | 1 | 1 | 128 |

Bits 3 - 2 - SAMP_TIME[1:0] - Determines the time to take a single sample as shown in Table 5.19.

**Table 5.19 SAMP_TIME Bit Decode**

| SAMP_TIME[1:0] | | SAMPLE TIME |
|---|---|---|
| **1** | **0** | |
| 0 | 0 | 320us |
| 0 | 1 | 640us |
| 1 | 0 | 1.28ms (default) |
| 1 | 1 | 2.56ms |

Bits 1 - 0 - CYCLE_TIME[1:0] - Determines the overall cycle time for all measured channels during normal operation as shown in Table 5.20. All measured channels are sampled at the beginning of the cycle time. If additional time is remaining, then the device is placed into a lower power state for the remaining duration of the cycle.

**Table 5.20 CYCLE_TIME Bit Decode**

| CYCLE_TIME[1:0] | | OVERALL CYCLE TIME |
|---|---|---|
| **1** | **0** | |
| 0 | 0 | 35ms |
| 0 | 1 | 70ms (default) |
| 1 | 0 | 105ms |
| 1 | 1 | 140ms |

**APPLICATION NOTE:** The programmed cycle time is only maintained if the total averaging time for all samples is less than the programmed cycle. The AVG[2:0] bits will take priority so that if more samples are required than would normally be allowed during the cycle time, the cycle time will be extended as necessary to accommodate the number of samples to be measured.

## 5.11 Calibration Activate Register

**Table 5.21 Calibration Activate Register**

| ADDR | R/W | REGISTER | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 | DEFAULT |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 26h | R/W | Calibration Activate | CS8_ CAL | CS7_ CAL | CS6_ CAL | CS5_ CAL | CS4_ CAL | CS3_ CAL | CS2_ CAL | CS1_ CAL | 00h |

The Calibration Activate register forces the respective sensor inputs to be re-calibrated affecting both the analog and digital blocks. During the re-calibration routine, the sensor inputs will not detect a press for up to 600ms and the Sensor Input Base Count register values will be invalid. During this time, any press on the corresponding sensor pads will invalidate the re-calibration. When finished, the CALX[9:0] bits will be updated (see Section 5.39).

When the corresponding bit is set, the device will perform the calibration and the bit will be automatically cleared once the re-calibration routine has finished.

Bit 7 - CS8_CAL - When set, the CS8 input is re-calibrated. This bit is automatically cleared once the sensor input has been re-calibrated successfully.

Bit 6 - CS7_CAL - When set, the CS7 input is re-calibrated. This bit is automatically cleared once the sensor input has been re-calibrated successfully.

Bit 5 - CS6_CAL - When set, the CS6 input is re-calibrated. This bit is automatically cleared once the sensor input has been re-calibrated successfully.

Bit 4 - CS5_CAL - When set, the CS5 input is re-calibrated. This bit is automatically cleared once the sensor input has been re-calibrated successfully.

Bit 3 - CS4_CAL - When set, the CS4 input is re-calibrated. This bit is automatically cleared once the sensor input has been re-calibrated successfully.

Bit 2 - CS3_CAL - When set, the CS3 input is re-calibrated. This bit is automatically cleared once the sensor input has been re-calibrated successfully.

Bit 1 - CS2_CAL - When set, the CS2 input is re-calibrated. This bit is automatically cleared once the sensor input has been re-calibrated successfully.

Bit 0 - CS1_CAL - When set, the CS1 input is re-calibrated. This bit is automatically cleared once the sensor input has been re-calibrated successfully.

## 5.12 Interrupt Enable Register

**Table 5.22 Interrupt Enable Register**

| ADDR | R/W | REGISTER | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 | DEFAULT |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 27h | R/W | Interrupt Enable | CS8_ INT_EN | CS7_ INT_EN | CS6_ INT_EN | CS5_ INT_EN | CS4_ INT_EN | CS3_ INT_EN | CS2_ INT_EN | CS1_ INT_EN | FFh |

The Interrupt Enable register determines whether a sensor pad touch or release (if enabled) causes the interrupt pin to be asserted.

Bit 7 - CS8_INT_EN - Enables the interrupt pin to be asserted if a touch is detected on CS8 (associated with the CS8 status bit).

- ‘0’ - The interrupt pin will not be asserted if a touch is detected on CS8 (associated with the CS8 status bit).
- ‘1’ (default) - The interrupt pin will be asserted if a touch is detected on CS8 (associated with the CS8 status bit).

Bit 6 - CS7_INT_EN - Enables the interrupt pin to be asserted if a touch is detected on CS7 (associated with the CS7 status bit).

Bit 5 - CS6_INT_EN - Enables the interrupt pin to be asserted if a touch is detected on CS6 (associated with the CS6 status bit).

Bit 4 - CS5_INT_EN - Enables the interrupt pin to be asserted if a touch is detected on CS5 (associated with the CS5 status bit).

Bit 3 - CS4_INT_EN - Enables the interrupt pin to be asserted if a touch is detected on CS4 (associated with the CS4 status bit).

Bit 2 - CS3_INT_EN - Enables the interrupt pin to be asserted if a touch is detected on CS3 (associated with the CS3 status bit).

Bit 1 - CS2_INT_EN - Enables the interrupt pin to be asserted if a touch is detected on CS2 (associated with the CS2 status bit).

Bit 0 - CS1_INT_EN - Enables the interrupt pin to be asserted if a touch is detected on CS1 (associated with the CS1 status bit).

## 5.13 Repeat Rate Enable Register

**Table 5.23 Repeat Rate Enable Register**

| ADDR | R/W | REGISTER | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 | DEFAULT |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 28h | R/W | Repeat Rate Enable | CS8_ RPT_EN | CS7_ RPT_EN | CS6_ RPT_EN | CS5_ RPT_EN | CS4_ RPT_EN | CS3_ RPT_EN | CS2_ RPT_EN | CS1_ RPT_EN | FFh |

The Repeat Rate Enable register enables the repeat rate of the sensor inputs as described in Section 4.6.1.

Bit 7 - CS8_RPT_EN - Enables the repeat rate for capacitive touch sensor input 8.

- ‘0’ - The repeat rate for CS8 is disabled. It will only generate an interrupt when a touch is detected and when a release is detected (if enabled) no matter how long the touch is held for.
- ‘1’ (default) - The repeat rate for CS8 is enabled. In the case of a “touch” event, it will generate an interrupt when a touch is detected and a release is detected (as determined by the INT_REL_n bit - see Section 5.6). In the case of a “press and hold” event, it will generate an interrupt when a touch is detected and at the repeat rate so long as the touch is held.

Bit 6 - CS7_RPT_EN - Enables the repeat rate for capacitive touch sensor input 7.

Bit 5 - CS6_RPT_EN - Enables the repeat rate for capacitive touch sensor input 6.

Bit 4 - CS5_RPT_EN - Enables the repeat rate for capacitive touch sensor input 5.

Bit 3 - CS4_RPT_EN - Enables the repeat rate for capacitive touch sensor input 4.

Bit 2 - CS3_RPT_EN - Enables the repeat rate for capacitive touch sensor input 3.

Bit 1 - CS2_RPT_EN - Enables the repeat rate for capacitive touch sensor input 2.

Bit 0 - CS1_RPT_EN - Enables the repeat rate for capacitive touch sensor input 1.

## 5.14 Multiple Touch Configuration Register

**Table 5.24 Multiple Touch Configuration**

| ADDR | R/W | REGISTER | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 | DEFAULT |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 2Ah | R/W | Multiple Touch Config | MULT_ BLK_ EN | - | - | - | B_MULT_T[1:0] | | - | - | 80h |

The Multiple Touch Configuration register controls the settings for the multiple touch detection circuitry. These settings determine the number of simultaneous buttons that may be pressed before additional buttons are blocked and the MULT status bit is set.

Bit 7 - MULT_BLK_EN - Enables the multiple button blocking circuitry.

- ‘0’ - The multiple touch circuitry is disabled. The device will not block multiple touches.
- ‘1’ (default) - The multiple touch circuitry is enabled. The device will flag the number of touches equal to programmed multiple touch threshold and block all others. It will remember which sensor inputs are valid and block all others until that sensor pad has been released. Once a sensor pad has been released, the N detected touches (determined via the cycle order of CS1 - CS8) will be flagged and all others blocked.

Bits 3 - 2 - B_MULT_T[1:0] - Determines the number of simultaneous touches on all sensor pads before a Multiple Touch Event is detected and sensor inputs are blocked. The bit decode is given by Table 5.25.

**Table 5.25 B_MULT_T Bit Decode**

| B_MULT_T[1:0] | | NUMBER OF SIMULTANEOUS TOUCHES |
|---|---|---|
| 1 | 0 | |
| 0 | 0 | 1 (default) |
| 0 | 1 | 2 |
| 1 | 0 | 3 |
| 1 | 1 | 4 |

## 5.15 Multiple Touch Pattern Configuration Register

**Table 5.26 Multiple Touch Pattern Configuration**

| ADDR | R/W | REGISTER | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 | DEFAULT |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 2Bh | R/W | Multiple Touch Pattern Config | MTP_ EN | - | - | | MTP_TH[1:0] | | COMP_ PTRN | MTP_ ALERT | 00h |

The Multiple Touch Pattern Configuration register controls the settings for the multiple touch pattern detection circuitry. This circuitry works like the multiple touch detection circuitry with the following differences:

1. The detection threshold is a percentage of the touch detection threshold as defined by the MTP_TH[1:0] bits whereas the multiple touch circuitry uses the touch detection threshold.
2. The MTP detection circuitry either will detect a specific pattern of sensor inputs as determined by the Multiple Touch Pattern register settings or it will use the Multiple Touch Pattern register settings to determine a minimum number of sensor inputs that will cause the MTP circuitry to flag an event. When using pattern recognition mode, if all of the sensor inputs set by the Multiple Touch Pattern register have a delta count greater than the MTP threshold or have their corresponding Noise Flag Status bits set, the MTP bit will be set. When using the absolute number mode, if the number of sensor inputs with thresholds above the MTP threshold or with Noise Flag Status bits set is equal to or greater than this number, the MTP bit will be set.
3. When an MTP event occurs, all touches are blocked and an interrupt is generated.
4. All sensor inputs will remain blocked so long as the requisite number of sensor inputs are above the MTP threshold or have Noise Flag Status bits set. Once this condition is removed, touch detection will be restored. Note that the MTP status bit is only cleared by writing a ‘0’ to the INT bit once the condition has been removed.

Bit 7 - MTP_EN - Enables the multiple touch pattern detection circuitry.

- ‘0’ (default) - The MTP detection circuitry is disabled.
- ‘1’ - The MTP detection circuitry is enabled.

Bits 3-2 - MTP_TH[1:0] - Determine the MTP threshold, as shown in Table 5.27. This threshold is a percentage of sensor input threshold (see Section 5.18, "Sensor Input Threshold Registers") when the device is in the Fully Active state or of the standby threshold (see Section 5.23, "Standby Threshold Register") when the device is in the Standby state.

**Table 5.27 MTP_TH Bit Decode**

| MTP_TH[1:0] | | |
|---|---|---|
| **1** | **0** | **THRESHOLD DIVIDE SETTING** |
| 0 | 0 | 12.5% (default) |
| 0 | 1 | 25% |
| 1 | 0 | 37.5% |
| 1 | 1 | 100% |

Bit 1 - COMP_PTRN - Determines whether the MTP detection circuitry will use the Multiple Touch Pattern register as a specific pattern of sensor inputs or as an absolute number of sensor inputs.

- ‘0’ (default) - The MTP detection circuitry will use the Multiple Touch Pattern register bit settings as an absolute minimum number of sensor inputs that must be above the threshold or have Noise Flag Status bits set. The number will be equal to the number of bits set in the register.
- ‘1’ - The MTP detection circuitry will use pattern recognition. Each bit set in the Multiple Touch Pattern register indicates a specific sensor input that must have a delta count greater than the MTP threshold or have a Noise Flag Status bit set. If the criteria are met, the MTP status bit will be set.

Bit 0 - MTP_ALERT - Enables an interrupt if an MTP event occurs. In either condition, the MTP status bit will be set.

- ‘0’ (default) - If an MTP event occurs, the ALERT# pin is not asserted.
- ‘1’ - If an MTP event occurs, the ALERT# pin will be asserted.

## 5.16 Multiple Touch Pattern Register

**Table 5.28 Multiple Touch Pattern Register**

| ADDR | R/W | REGISTER | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 | DEFAULT |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 2Dh | R/W | Multiple Touch Pattern | CS8_ PTRN | CS7_ PTRN | CS6_ PTRN | CS5_ PTRN | CS4_ PTRN | CS3_ PTRN | CS2_ PTRN | CS1_ PTRN | FFh |

The Multiple Touch Pattern register acts as a pattern to identify an expected sensor input profile for diagnostics or other significant events. There are two methods for how the Multiple Touch Pattern register is used: as specific sensor inputs or number of sensor input that must exceed the MTP threshold or have Noise Flag Status bits set. Which method is used is based on the COMP_PTRN bit (see Section 5.15). The methods are described below.

1. Specific Sensor Inputs: If, during a single polling cycle, the specific sensor inputs above the MTP threshold or with Noise Flag Status bits set match those bits set in the Multiple Touch Pattern register, an MTP event is flagged.
2. Number of Sensor Inputs: If, during a single polling cycle, the number of sensor inputs with a delta count above the MTP threshold or with Noise Flag Status bits set is equal to or greater than the number of pattern bits set, an MTP event is flagged.

Bit 7 - CS8_PTRN - Determines whether CS8 is considered as part of the Multiple Touch Pattern.

- ‘0’ - CS8 is not considered a part of the pattern.
- ‘1’ - CS8 is considered a part of the pattern, or the absolute number of sensor inputs that must have a delta count greater than the MTP threshold or have the Noise Flag Status bit set is increased by 1.

Bit 6 - CS7_PTRN - Determines whether CS7 is considered as part of the Multiple Touch Pattern.

Bit 5 - CS6_PTRN - Determines whether CS6 is considered as part of the Multiple Touch Pattern.

Bit 4 - CS5_PTRN - Determines whether CS5 is considered as part of the Multiple Touch Pattern.

Bit 3 - CS4_PTRN - Determines whether CS4 is considered as part of the Multiple Touch Pattern.

Bit 2 - CS3_PTRN - Determines whether CS3 is considered as part of the Multiple Touch Pattern.

Bit 1 - CS2_PTRN - Determines whether CS2 is considered as part of the Multiple Touch Pattern.

Bit 0 - CS1_PTRN - Determines whether CS1 is considered as part of the Multiple Touch Pattern.

## 5.17 Recalibration Configuration Register

**Table 5.29 Recalibration Configuration Registers**

| ADDR | R/W | REGISTER | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 | DEFAULT |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 2Fh | R/W | Recalibration Configuration | BUT_ LD_TH | NO_ CLR_ INTD | NO_ CLR_ NEG | NEG_DELTA_ CNT[1:0] | | CAL_CFG[2:0] | | | 8Ah |

The Recalibration Configuration register controls the automatic re-calibration routine settings as well as advanced controls to program the Sensor Input Threshold register settings.

Bit 7 - BUT_LD_TH - Enables setting all Sensor Input Threshold registers by writing to the Sensor Input 1 Threshold register.

- ‘0’ - Each Sensor Input X Threshold register is updated individually.
- ‘1’ (default) - Writing the Sensor Input 1 Threshold register will automatically overwrite the Sensor Input Threshold registers for all sensor inputs (Sensor Input Threshold 1 through Sensor Input Threshold 8). The individual Sensor Input X Threshold registers (Sensor Input 2 Threshold through Sensor Input 8 Threshold) can be individually updated at any time.

Bit 6 - NO_CLR_INTD - Controls whether the accumulation of intermediate data is cleared if the noise status bit is set.

- ‘0’ (default) - The accumulation of intermediate data is cleared if the noise status bit is set.
- ‘1’ - The accumulation of intermediate data is not cleared if the noise status bit is set.

**APPLICATION NOTE:** Bits 5 and 6 should both be set to the same value. Either both should be set to ‘0’ or both should be set to ‘1’.

Bit 5 - NO_CLR_NEG - Controls whether the consecutive negative delta counts counter is cleared if the noise status bit is set.

- ‘0’ (default) - The consecutive negative delta counts counter is cleared if the noise status bit is set.
- ‘1’ - The consecutive negative delta counts counter is not cleared if the noise status bit is set.

Bits 4 - 3 - NEG_DELTA_CNT[1:0] - Determines the number of negative delta counts necessary to trigger a digital re-calibration as shown in Table 5.30.

**Table 5.30 NEG_DELTA_CNT Bit Decode**

| NEG_DELTA_CNT[1:0] | | NUMBER OF CONSECUTIVE NEGATIVE DELTA COUNT VALUES |
|---|---|---|
| **1** | **0** | |
| 0 | 0 | 8 |
| 0 | 1 | 16 (default) |
| 1 | 0 | 32 |
| 1 | 1 | None (disabled) |

Bits 2 - 0 - CAL_CFG[2:0] - Determines the update time and number of samples of the automatic re-calibration routine. The settings apply to all sensor inputs universally (though individual sensor inputs can be configured to support re-calibration - see Section 5.11).

**Table 5.31 CAL_CFG Bit Decode**

| CAL_CFG[2:0] | | | RECALIBRATION SAMPLES (SEE Note 5.1) | UPDATE TIME (SEE Note 5.2) |
|---|---|---|---|---|
| **2** | **1** | **0** | | |
| 0 | 0 | 0 | 16 | 16 |
| 0 | 0 | 1 | 32 | 32 |
| 0 | 1 | 0 | 64 | 64 (default) |
| 0 | 1 | 1 | 128 | 128 |

**Table 5.31 CAL_CFG Bit Decode (continued)**

| CAL_CFG[2:0] | | | RECALIBRATION SAMPLES (SEE Note 5.1) | UPDATE TIME (SEE Note 5.2) |
|---|---|---|---|---|
| **2** | **1** | **0** | | |
| 1 | 0 | 0 | 256 | 256 |
| 1 | 0 | 1 | 256 | 1024 |
| 1 | 1 | 0 | 256 | 2048 |
| 1 | 1 | 1 | 256 | 4096 |

**Note 5.1** Recalibration Samples refers to the number of samples that are measured and averaged before the Base Count is updated however does not control the base count update period.

**Note 5.2** Update Time refers to the amount of time (in polling cycle periods) that elapses before the Base Count is updated. The time will depend upon the number of channels active, the averaging setting, and the programmed cycle time.

## 5.18 Sensor Input Threshold Registers

**Table 5.32 Sensor Input Threshold Registers**

| ADDR | R/W | REGISTER | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 | DEFAULT |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 30h | R/W | Sensor Input 1 Threshold | - | 64 | 32 | 16 | 8 | 4 | 2 | 1 | 40h |
| 31h | R/W | Sensor Input 2 Threshold | - | 64 | 32 | 16 | 8 | 4 | 2 | 1 | 40h |
| 32h | R/W | Sensor Input 3 Threshold | - | 64 | 32 | 16 | 8 | 4 | 2 | 1 | 40h |
| 33h | R/W | Sensor Input 4 Threshold | - | 64 | 32 | 16 | 8 | 4 | 2 | 1 | 40h |
| 34h | R/W | Sensor Input 5 Threshold | - | 64 | 32 | 16 | 8 | 4 | 2 | 1 | 40h |
| 35h | R/W | Sensor Input 6 Threshold | - | 64 | 32 | 16 | 8 | 4 | 2 | 1 | 40h |
| 36h | R/W | Sensor Input 7 Threshold | - | 64 | 32 | 16 | 8 | 4 | 2 | 1 | 40h |
| 37h | R/W | Sensor Input 8 Threshold | - | 64 | 32 | 16 | 8 | 4 | 2 | 1 | 40h |

The Sensor Input Threshold registers store the delta threshold that is used to determine if a touch has been detected. When a touch occurs, the input signal of the corresponding sensor pad changes due to the capacitance associated with a touch. If the sensor input change exceeds the threshold settings, a touch is detected.

When the BUT_LD_TH bit is set (see Section 5.17 - bit 7), writing data to the Sensor Input 1 Threshold register will update all of the sensor input threshold registers (31h - 37h inclusive).

## 5.19 Sensor Input Noise Threshold Register

**Table 5.33 Sensor Input Noise Threshold Register**

| ADDR | R/W | REGISTER | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 | DEFAULT |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 38h | R/W | Sensor Input Noise Threshold | | | | | | | CS_BN_TH [1:0] | | 01h |

The Sensor Input Noise Threshold register controls the value of a secondary internal threshold to detect noise and improve the automatic recalibration routine. If a capacitive touch sensor input exceeds the Sensor Input Noise Threshold but does not exceed the sensor input threshold, it is determined to be caused by a noise spike. That sample is not used by the automatic re-calibration routine. This feature can be disabled by setting the DIS_DIG_NOISE bit.

Bits 1-0 - CS1_BN_TH[1:0] - Controls the noise threshold for all capacitive touch sensor inputs, as shown in Table 5.34. The threshold is proportional to the threshold setting.

**Table 5.34 CSx_BN_TH Bit Decode**

| CS_BN_TH[1:0] | | PERCENT THRESHOLD SETTING |
|---|---|---|
| **1** | **0** | |
| 0 | 0 | 25% |
| 0 | 1 | 37.5% (default) |
| 1 | 0 | 50% |
| 1 | 1 | 62.5% |

## 5.20 Standby Channel Register

**Table 5.35 Standby Channel Register**

| ADDR | R/W | REGISTER | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 | DEFAULT |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 40h | R/W | Standby Channel | CS8_ STBY | CS7_ STBY | CS6_ STBY | CS5_ STBY | CS4_ STBY | CS3_ STBY | CS2_ STBY | CS1_ STBY | 00h |

The Standby Channel register controls which (if any) capacitive touch sensor inputs are active during Standby.

Bit 7 - CS8_STBY - Controls whether the CS8 channel is active in Standby.

- ‘0’ (default) - The CS8 channel not be sampled during Standby.
- ‘1’ - The CS8 channel will be sampled during Standby. It will use the Standby threshold setting, and the standby averaging and sensitivity settings.

Bit 6 - CS7_STBY - Controls whether the CS7 channel is active in Standby.

Bit 5 - CS6_STBY - Controls whether the CS6 channel is active in Standby.

Bit 4 - CS5_STBY - Controls whether the CS5 channel is active in Standby.

Bit 3 - CS4_STBY - Controls whether the CS4 channel is active in Standby.

Bit 2 - CS3_STBY - Controls whether the CS3 channel is active in Standby.

Bit 1 - CS2_STBY - Controls whether the CS2 channel is active in Standby.

Bit 0 - CS1_STBY - Controls whether the CS1 channel is active in Standby.

## 5.21 Standby Configuration Register

**Table 5.36 Standby Configuration Register**

| ADDR | R/W | REGISTER | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 | DEFAULT |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 41h | R/W | Standby Configuration | AVG_ SUM | STBY_AVG[2:0] | | | STBY_SAMP_ TIME[1:0] | | STBY_CY_TIME [1:0] | | 39h |

The Standby Configuration register controls averaging and cycle time for those sensor inputs that are active in Standby. This register is useful for detecting proximity on a small number of sensor inputs as it allows the user to change averaging and sample times on a limited number of sensor inputs and still maintain normal functionality in the fully active state.

Bit 7 - AVG_SUM - Determines whether the active sensor inputs will average the programmed number of samples or whether they will accumulate for the programmed number of samples.

- ‘0’ - (default) - The active sensor input delta count values will be based on the average of the programmed number of samples when compared against the threshold.
- ‘1’ - The active sensor input delta count values will be based on the summation of the programmed number of samples when compared against the threshold. This bit should only be set when performing proximity detection as a physical touch will overflow the delta count registers and may result in false readings.

Bits 6 - 4 - STBY_AVG[2:0] - Determines the number of samples that are taken for all active channels during the sensor cycle as shown in Table 5.37. All samples are taken consecutively on the same channel before the next channel is sampled and the result is averaged over the number of samples measured before updating the measured results.

**Table 5.37 STBY_AVG Bit Decode**

| STBY_AVG[2:0] | | | NUMBER OF SAMPLES TAKEN PER MEASUREMENT |
|---|---|---|---|
| **2** | **1** | **0** | |
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 2 |
| 0 | 1 | 0 | 4 |
| 0 | 1 | 1 | 8 (default) |
| 1 | 0 | 0 | 16 |
| 1 | 0 | 1 | 32 |
| 1 | 1 | 0 | 64 |
| 1 | 1 | 1 | 128 |

Bit 3-2 - STBY SAMP_TIME[1:0] - Determines the time to take a single sample when the device is in Standby as shown in Table 5.38.

**Table 5.38 STBY_SAMP_TIME Bit Decode**

| STBY_SAMP_TIME[1:0] | | SAMPLING TIME |
|---|---|---|
| **1** | **0** | |
| 0 | 0 | 320us |
| 0 | 1 | 640us |
| 1 | 0 | 1.28ms (default) |
| 1 | 1 | 2.56ms |

Bits 1 - 0 - STBY_CY_TIME[2:0] - Determines the overall cycle time for all measured channels during standby operation as shown in Table 5.39. All measured channels are sampled at the beginning of the cycle time. If additional time is remaining, the device is placed into a lower power state for the remaining duration of the cycle.

**Table 5.39 STBY_CY_TIME Bit Decode**

| STBY_CY_TIME[1:0] | | OVERALL CYCLE TIME |
|---|---|---|
| **1** | **0** | |
| 0 | 0 | 35ms |
| 0 | 1 | 70ms (default) |
| 1 | 0 | 105ms |
| 1 | 1 | 140ms |

**APPLICATION NOTE:** The programmed cycle time is only maintained if the total averaging time for all samples is less than the programmed cycle. The STBY_AVG[2:0] bits will take priority so that if more samples are required than would normally be allowed during the cycle time, the cycle time will be extended as necessary to accommodate the number of samples to be measured.

## 5.22 Standby Sensitivity Register

**Table 5.40 Standby Sensitivity Register**

| ADDR | R/W | REGISTER | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 | DEFAULT |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 42h | R/W | Standby Sensitivity | - | - | - | - | - | STBY_SENSE[2:0] | | | 02h |

The Standby Sensitivity register controls the sensitivity for sensor inputs that are active in Standby.

Bits 2 - 0 - STBY_SENSE[2:0] - Controls the sensitivity for sensor inputs that are active in Standby. The sensitivity settings act to scale the relative delta count value higher or lower based on the system parameters. A setting of 000b is the most sensitive while a setting of 111b is the least sensitive. At the more sensitive settings, touches are detected for a smaller delta C corresponding to a “lighter” touch. These settings are more sensitive to noise however and a noisy environment may flag more false touches than higher sensitivity levels.

**APPLICATION NOTE:** A value of 128x is the most sensitive setting available. At the most sensitivity settings, the MSB of the Delta Count register represents 64 out of ~25,000 which corresponds to a touch of approximately 0.25% of the base capacitance (or a $\Delta$C of 25fF from a 10pF base capacitance). Conversely a value of 1x is the least sensitive setting available. At these settings, the MSB of the Delta Count register corresponds to a delta count of 8192 counts out of ~25,000 which corresponds to a touch of approximately 33% of the base capacitance (or a $\Delta$C of 3.33pF from a 10pF base capacitance).

**Table 5.41 STBY_SENSE Bit Decode**

| STBY_SENSE[2:0] | | | |
|---|---|---|---|
| **2** | **1** | **0** | **SENSITIVITY MULTIPLIER** |
| 0 | 0 | 0 | 128x (most sensitive) |
| 0 | 0 | 1 | 64x |
| 0 | 1 | 0 | 32x (default) |
| 0 | 1 | 1 | 16x |
| 1 | 0 | 0 | 8x |
| 1 | 0 | 1 | 4x |
| 1 | 1 | 0 | 2x |
| 1 | 1 | 1 | 1x - (least sensitive) |

## 5.23 Standby Threshold Register

**Table 5.42 Standby Threshold Register**

| ADDR | R/W | REGISTER | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 | DEFAULT |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 43h | R/W | Standby Threshold | - | 64 | 32 | 16 | 8 | 4 | 2 | 1 | 40h |

The Standby Threshold register stores the delta threshold that is used to determine if a touch has been detected. When a touch occurs, the input signal of the corresponding sensor pad changes due to the capacitance associated with a touch. If the sensor input change exceeds the threshold settings, a touch is detected.

## 5.24 Sensor Input Base Count Registers

**Table 5.43 Sensor Input Base Count Registers**

| ADDR | R/W | REGISTER | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 | DEFAULT |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 50h | R | Sensor Input 1 Base Count | 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 | C8h |
| 51h | R | Sensor Input 2 Base Count | 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 | C8h |
| 52h | R | Sensor Input 3 Base Count | 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 | C8h |
| 53h | R | Sensor Input 4 Base Count | 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 | C8h |
| 54h | R | Sensor Input 5 Base Count | 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 | C8h |
| 55h | R | Sensor Input 6 Base Count | 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 | C8h |
| 56h | R | Sensor Input 7 Base Count | 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 | C8h |
| 57h | R | Sensor Input 8 Base Count | 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 | C8h |

The Sensor Input Base Count registers store the calibrated “Not Touched” input value from the capacitive touch sensor inputs. These registers are periodically updated by the re-calibration routine.

The routine uses an internal adder to add the current count value for each reading to the sum of the previous readings until sample size has been reached. At this point, the upper 16 bits are taken and used as the Sensor Input Base Count. The internal adder is then reset and the re-calibration routine continues.

The data presented is determined by the BASE_SHIFT[3:0] bits (see Section 5.5).

## 5.25 LED Output Type Register

**Table 5.44 LED Output Type Register**

| ADDR | R/W | REGISTER | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 | DEFAULT |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 71h | R/W | LED Output Type | LED8_ OT | LED7_ OT | LED6_ OT | LED5_ OT | LED4_ OT | LED3_ OT | LED2_ OT | LED1_ OT | 00h |

The LED Output Type register controls the type of output for the LED pins. Each pin is controlled by a single bit. Refer to application note 21.4 CAP1188 Family LED Configuration Options for more information about implementing LEDs.

Bit 7 - LED8_OT - Determines the output type of the LED8 pin.

- ‘0’ (default) - The LED8 pin is an open-drain output with an external pull-up resistor. When the appropriate pin is set to the “active” state (logic ‘1’), the pin will be driven low. Conversely, when the pin is set to the “inactive” state (logic ‘0’), then the pin will be left in a High Z state and pulled high via an external pull-up resistor.
- ‘1’ - The LED8 pin is a push-pull output. When driving a logic ‘1’, the pin is driven high. When driving a logic ‘0’, the pin is driven low.

Bit 6 - LED7_OT - Determines the output type of the LED7 pin.

Bit 5 - LED6_OT - Determines the output type of the LED6 pin.

Bit 4 - LED5_OT - Determines the output type of the LED5 pin.

Bit 3 - LED4_OT - Determines the output type of the LED4 pin.

Bit 2 - LED3_OT - Determines the output type of the LED3 pin.

Bit 1 - LED2_OT - Determines the output type of the LED2 pin.

Bit 0 - LED1_OT - Determines the output type of the LED1 pin.

## 5.26 Sensor Input LED Linking Register

**Table 5.45 Sensor Input LED Linking Register**

| ADDR | R/W | REGISTER | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 | DEFAULT |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 72h | R/W | Sensor Input LED Linking | CS8_ LED8 | CS7_ LED7 | CS6_ LED6 | CS5_ LED5 | CS4_ LED4 | CS3_ LED3 | CS2_ LED2 | CS1_ LED1 | 00h |

The Sensor Input LED Linking register controls whether a capacitive touch sensor input is linked to an LED output. If the corresponding bit is set, then the appropriate LED output will change states defined by the LED Behavior controls (see Section 5.31) in response to the capacitive touch sensor input.

Bit 7 - CS8_LED8 - Links the LED8 output to a detected touch on the CS8 sensor input. When a touch is detected, the LED is actuated and will behave as determined by the LED Behavior controls.

- ‘0’ (default) - The LED8 output is not associated with the CS8 input. If a touch is detected on the CS8 input, then the LED will not automatically be actuated. The LED is enabled and controlled via the LED Output Control register (see Section 5.28) and the LED Behavior registers (see Section 5.31).
- ‘1’ - The LED8 output is associated with the CS8 input. If a touch is detected on the CS8 input, the LED will be actuated and behave as defined in Table 5.52.

Bit 6 - CS7_LED7 - Links the LED7 output to a detected touch on the CS7 sensor input. When a touch is detected, the LED is actuated and will behave as determined by the LED Behavior controls.

Bit 5 - CS6_LED6 - Links the LED6 output to a detected touch on the CS6 sensor input. When a touch is detected, the LED is actuated and will behave as determined by the LED Behavior controls.

Bit 4 - CS5_LED5 - Links the LED5 output to a detected touch on the CS5 sensor input. When a touch is detected, the LED is actuated and will behave as determined by the LED Behavior controls.

Bit 3 - CS4_LED4 - Links the LED4 output to a detected touch on the CS4 sensor input. When a touch is detected, the LED is actuated and will behave as determined by the LED Behavior controls.

Bit 2 - CS3_LED3 - Links the LED3 output to a detected touch on the CS3 sensor input. When a touch is detected, the LED is actuated and will behave as determined by the LED Behavior controls.

Bit 1 - CS2_LED2 - Links the LED2 output to a detected touch on the CS2 sensor input. When a touch is detected, the LED is actuated and will behave as determined by the LED Behavior controls.

Bit 0 - CS1_LED1 - Links the LED1 output to a detected touch on the CS1 sensor input. When a touch is detected, the LED is actuated and will behave as determined by the LED Behavior controls.

## 5.27 LED Polarity Register

**Table 5.46 LED Polarity Register**

| ADDR | R/W | REGISTER | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 | DEFAULT |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 73h | R/W | LED Polarity | LED8_ POL | LED7_ POL | LED6_ POL | LED5_ POL | LED4_ POL | LED3_ POL | LED2_ POL | LED1_ POL | 00h |

The LED Polarity register controls the logical polarity of the LED outputs. When these bits are set or cleared, the corresponding LED Mirror controls are also set or cleared (unless the BLK_POL_MIR bit is set - see Section 5.6, "Configuration Registers"). Table 5.48, "LED Polarity Behavior" shows the interaction between the polarity controls, output controls, and relative brightness.

**APPLICATION NOTE:** The polarity controls determine the final LED pin drive. A touch on a linked capacitive touch sensor input is treated in the same way as the LED Output Control bit being set to a logic ‘1’.

**APPLICATION NOTE:** The LED drive assumes that the LEDs are configured such that if the LED pin is driven to a logic ‘0’ then the LED will be on and that the CAP1188 LED pin is sinking the LED current. Conversely, if the LED pin is driven to a logic ‘1’, the LED will be off and there is no current flow. See Figure 4.1, "System Diagram for CAP1188".

**APPLICATION NOTE:** This application note applies when the LED polarity is inverted (LEDx_POL = ‘0’). For LED operation, the duty cycle settings determine the % of time that the LED pin will be driven to a logic ‘0’ state in. The Max Duty Cycle settings define the maximum % of time that the LED pin will be driven low (i.e. maximum % of time that the LED is **on**) while the Min Duty Cycle settings determine the minimum % of time that the LED pin will be driven low (i.e. minimum % of time that the LED is **on**). When there is no touch detected or the LED Output Control register bit is at a logic ‘0’, the LED output will be driven at the minimum duty cycle setting. Breathe operations will ramp the duty cycle from the minimum duty cycle to the maximum duty cycle.

**APPLICATION NOTE:** This application note applies when the LED polarity is non-inverted (LEDx_POL = ‘1’). For LED operation, the duty cycle settings determine the % of time that the LED pin will be driven to a logic ‘1’ state. The Max Duty Cycle settings define the maximum % of time that the LED pin will be driven high (i.e. maximum % of time that the LED is **off**) while the Min Duty Cycle settings determine the minimum % of time that the LED pin will be driven high (i.e. minimum % of time that the LED is **off**). When there is no touch detected or the LED Output Control register bit is at a logic ‘0’, the LED output will be driven at 100 minus the minimum duty cycle setting. Breathe operations will ramp the duty cycle from 100 minus the minimum duty cycle to 100 minus the maximum duty cycle.

**APPLICATION NOTE:** The LED Mirror controls (see Section 5.30, "LED Mirror Control Register") work with the polarity controls with respect to LED brightness but will not have a direct effect on the output pin drive.

Bit 7 - LED8_POL - Determines the polarity of the LED8 output.

- ‘0’ (default) - The LED8 output is inverted. For example, a setting of ‘1’ in the LED Output Control register will cause the LED pin output to be driven to a logic ‘0’.
- ‘1’ - The LED8 output is non-inverted. For example, a setting of ‘1’ in the LED Output Control register will cause the LED pin output to be driven to a logic ‘1’ or left in the high-z state as determined by its output type.

Bit 6 - LED7_POL - Determines the polarity of the LED7 output.

Bit 5 - LED6_POL - Determines the polarity of the LED6 output.

Bit 4 - LED5_POL - Determines the polarity of the LED5 output.

Bit 3 - LED4_POL - Determines the polarity of the LED4 output.

Bit 2 - LED3_POL - Determines the polarity of the LED3 output.

Bit 1 - LED2_POL - Determines the polarity of the LED2 output.

Bit 0 - LED1_POL - Determines the polarity of the LED1 output.

## 5.28 LED Output Control Register

**Table 5.47 LED Output Control Register**

| ADDR | R/W | REGISTER | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 | DEFAULT |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 74h | R/W | LED Output Control | LED8_ DR | LED7_ DR | LED6_ DR | LED5_ DR | LED4_ DR | LED3_ DR | LED2_ DR | LED1_ DR | 00h |

The LED Output Control Register controls the output state of the LED pins that are not linked to sensor inputs.

**Note:** If an LED is linked to a sensor input in the Sensor Input LED Linking Register (Section 5.26, "Sensor Input LED Linking Register"), the corresponding bit in the LED Output Control Register is ignored (i.e. a linked LED cannot be host controlled).

The LED Polarity Control Register will determine the non actuated state of the LED pins. The actuated LED behavior is determined by the LED behavior controls (see Section 5.31, "LED Behavior Registers").

Table 5.48 shows the interaction between the polarity controls, output controls, and relative brightness.

Bit 7 - LED8_DR - Determines whether the LED8 output is driven high or low.

- ‘0’ (default) - The LED8 output is driven at the minimum duty cycle or not actuated.
- ‘1’ - The LED8 output is driven at the maximum duty cycle or is actuated.

Bit 6 - LED7_DR - Determines whether LED7 output is driven high or low.

Bit 5 - LED6_DR - Determines whether LED6 output is driven high or low.

Bit 4 - LED5_DR - Determines whether LED5 output is driven high or low.

Bit 3 - LED4_DR - Determines whether LED4 output is driven high or low.

Bit 2 - LED3_DR - Determines whether LED3 output is driven high or low.

Bit 1 - LED2_DR - Determines whether LED2 output is driven high or low.

Bit 0 - LED1_DR - Determines whether LED1 output is driven high or low.

**Table 5.48 LED Polarity Behavior**

| LED OUTPUT CONTROL REGISTER OR TOUCH | POLARITY | MAX DUTY | MIN DUTY | BRIGHTNESS | LED APPEARANCE |
|---|---|---|---|---|---|
| 0 | inverted (‘0’) | not used | minimum % of time that the LED is on (logic 0) | maximum brightness at min duty cycle | on at min duty cycle |
| 1 | inverted (‘0’) | maximum % of time that the LED is on (logic 0) | minimum % of time that the LED is on (logic 0) | maximum brightness at max duty cycle. Brightness ramps from min duty cycle to max duty cycle | according to LED behavior |
| 0 | non-inverted (‘1’) | not used | minimum % of time that the LED is off (logic 1) | maximum brightness at 100 minus min duty cycle. | on at 100 - min duty cycle |
| 1 | non-inverted (‘1’) | maximum % of time that the LED is off (logic 1) | minimum % of time that the LED is off (logic 1) | For Direct behavior, maximum brightness is 100 minus max duty cycle. When breathing, max brightness is 100 minus min duty cycle. Brightness ramps from 100 - min duty cycle to 100 - max duty cycle. | according to LED behavior |

## 5.29 Linked LED Transition Control Register

**Table 5.49 Linked LED Transition Control Register**

| ADDR | R/W | REGISTER | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 | DEFAULT |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 77h | R/W | Linked LED Transition Control | LED8_ LTRAN | LED7_ LTRAN | LED6_ LTRAN | LED5_ LTRAN | LED4_ LTRAN | LED3_ LTRAN | LED2_ LTRAN | LED1_ LTRAN | 00h |

The Linked LED Transition Control register controls the LED drive when the LED is linked to a capacitive touch sensor input. These controls work in conjunction with the INV_LINK_TRAN bit (see Section 5.6.2, "Configuration 2 - 44h") to create smooth transitions from host control to linked LEDs.

Bit 7 - LED8_LTRAN - Determines the transition effect when LED8 is linked to CS8.

- ‘0’ (default) - When the LED output control bit for CS8 is ‘1’, and then CS8 is linked to LED8 and no touch is detected, the LED will change states.
- ‘1’ - If the INV_LINK_TRAN bit is ‘1’, when the LED output control bit for CS8 is ‘1’, and then CS8 is linked to LED8 and no touch is detected, the LED will not change states. In addition, the LED state will change when the sensor pad is touched. If the INV_LINK_TRAN bit is ‘0’, when the LED output control bit for CS8 is ‘1’, and then CS8 is linked to LED8 and no touch is detected, the LED will not change states. However, the LED state will not change when the sensor pad is touched.

**APPLICATION NOTE:** If the LED behavior is not “Direct” and the INV_LINK_TRAN bit it ‘0’, the LED will not perform as expected when the LED8_LTRAN bit is set to ‘1’. Therefore, if breathe and pulse behaviors are used, set the INV_LINK_TRAN bit to ‘1’.

Bit 6 - LED7_LTRAN - Determines the transition effect when LED7 is linked to CS7.

Bit 5 - LED6_LTRAN - Determines the transition effect when LED6 is linked to CS6.

Bit 4 - LED5_LTRAN - Determines the transition effect when LED5 is linked to CS5.

Bit 3 - LED4_LTRAN - Determines the transition effect when LED4 is linked to CS4.

Bit 2 - LED3_LTRAN - Determines the transition effect when LED3 is linked to CS3.

Bit 1 - LED2_LTRAN - Determines the transition effect when LED2 is linked to CS2.

Bit 0 - LED1_LTRAN - Determines the transition effect when LED1 is linked to CS1.

## 5.30 LED Mirror Control Register

**Table 5.50 LED Mirror Control Register**

| ADDR | R/W | REGISTER | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 | DEFAULT |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 79h | R/W | LED Mirror Control | LED8_ MIR _ EN | LED7_ MIR _ EN | LED6_ MIR _ EN | LED5_ MIR _ EN | LED4_ LMIR_ EN | LED3_ MIR_ EN | LED2_ MIR _ EN | LED1_ MIR _ EN | 00h |

The LED Mirror Control Registers determine the meaning of duty cycle settings when polarity is non-inverted for each LED channel. When the polarity bit is set to ‘1’ (non-inverted), to obtain correct steps for LED ramping, pulse, and breathe behaviors, the min and max duty cycles need to be relative to 100%, rather than the default, which is relative to 0%.

**APPLICATION NOTE:** The LED drive assumes that the LEDs are configured such that if the LED pin is driven to a logic ‘0’, the LED will be on and the CAP1188 LED pin is sinking the LED current. When the polarity bit is set to ‘1’, it is considered non-inverted. For systems using the opposite LED configuration, mirror controls would apply when the polarity bit is ‘0’.

These bits are changed automatically if the corresponding LED Polarity bit is changed (unless the BLK_POL_MIR bit is set - see Section 5.6).

Bit 7 - LED8_MIR_EN - Determines whether the duty cycle settings are “biased” relative to 0% or 100% duty cycle.

- ‘0’ (default) - The duty cycle settings are determined relative to 0% and are determined directly with the settings.
- ‘1’ - The duty cycle settings are determined relative to 100%.

Bit 6 - LED7_MIR_EN - Determines whether the duty cycle settings are “biased” relative to 0% or 100% duty cycle.

Bit 5 - LED6_MIR_EN - Determines whether the duty cycle settings are “biased” relative to 0% or 100% duty cycle.

Bit 4 - LED5_MIR_EN - Determines whether the duty cycle settings are “biased” relative to 0% or 100% duty cycle.

Bit 3 - LED4_MIR_EN - Determines whether the duty cycle settings are “biased” relative to 0% or 100% duty cycle.

Bit 2 - LED3_MIR_EN - Determines whether the duty cycle settings are “biased” relative to 0% or 100% duty cycle.

Bit 1 - LED2_MIR_EN - Determines whether the duty cycle settings are “biased” relative to 0% or 100% duty cycle.

Bit 0 - LED1_MIR_EN - Determines whether the duty cycle settings are “biased” relative to 0% or 100% duty cycle.

## 5.31 LED Behavior Registers

**Table 5.51 LED Behavior Registers**

| ADDR | R/W | REGISTER | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 | DEFAULT |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 81h | R/W | LED Behavior 1 | LED4_CTL[1:0] | | LED3_CTL[1:0] | | LED2_CTL[1:0] | | LED1_CTL[1:0] | | 00h |
| 82h | R/W | LED Behavior 2 | LED8_CTL[1:0] | | LED7_CTL[1:0] | | LED6_CTL[1:0] | | LED5_CTL[1:0] | | 00h |

The LED Behavior registers control the operation of LEDs. Each LED pin is controlled by a 2-bit field and the behavior is determined by whether the LED is linked to a capacitive touch sensor input or not.

If the corresponding LED output is linked to a capacitive touch sensor input, the appropriate behavior will be enabled / disabled based on touches and releases.

If the LED output is not associated with a capacitive touch sensor input, the appropriate behavior will be enabled / disabled by the LED Output Control register. If the respective LEDx_DR bit is set to a logic ‘1’, this will be associated as a “touch”, and if the LEDx_DR bit is set to a logic ‘0’, this will be associated as a “release”.

Table 5.52, "LEDx_CTL Bit Decode" shows the behavior triggers. The defined behavior will activate when the Start Trigger is met and will stop when the Stop Trigger is met. Note the behavior of the Breathe Hold and Pulse Release option.

The LED Polarity Control register will determine the non actuated state of the LED outputs (see Section 5.27, "LED Polarity Register").

**APPLICATION NOTE:** If an LED is not linked to a capacitive touch sensor input and is breathing (via the Breathe or Pulse behaviors), it must be unactuated and then re-actuated before changes to behavior are processed. For example, if the LED output is breathing and the Maximum duty cycle is changed, this change will not take effect until the LED output control register is set to ‘0’ and then re-set to ‘1’.

**APPLICATION NOTE:** If an LED is not linked to the capacitive touch sensor input and configured to operate using Pulse 1 Behavior, then the circuitry will only be actuated when the corresponding output control bit is set. It will not check the bit condition until the Pulse 1 behavior is finished. The device will not remember if the bit was cleared and reset while it was actuated.

**APPLICATION NOTE:** If an LED is actuated and not linked and the desired LED behavior is changed, this new behavior will take effect immediately; however, the first instance of the changed behavior may act incorrectly (e.g. if changed from Direct to Pulse 1, the LED output may ‘breathe’ 4 times and then end at minimum duty cycle). LED Behaviors will operate normally once the LED has been un-actuated and then re-actuated.

**APPLICATION NOTE:** If an LED is actuated and it is switched from linked to a capacitive touch sensor input to unlinked (or vice versa), the LED will respond to the new command source immediately if the behavior was Direct or Breathe. For Pulse behaviors, it will complete the behavior already in progress. For example, if a linked LED was actuated by a touch and the control is changed so that it is unlinked, it will check the status of the corresponding LED Output Control bit. If that bit is ‘0’, then the LED will behave as if a release was detected. Likewise, if an unlinked LED was actuated by the LED Output Control register and the control is changed so that it is linked and no touch is detected, then the LED will behave as if a release was detected.

### 5.31.1 LED Behavior 1 - 81h

Bits 7 - 6 - LED4_CTL[1:0] - Determines the behavior of LED4 as shown in Table 5.52.

Bits 5 - 4 - LED3_CTL[1:0] - Determines the behavior of LED3 as shown in Table 5.52.

Bits 3 - 2 - LED2_CTL[1:0] - Determines the behavior of LED2 as shown in Table 5.52.

Bits 1 - 0 - LED1_CTL[1:0] - Determines the behavior of LED1 as shown in Table 5.52.

### 5.31.2 LED Behavior 2 - 82h

Bits 7 - 6 - LED8_CTL[1:0] - Determines the behavior of LED8 as shown in Table 5.52.

Bits 5 - 4 - LED7_CTL[1:0] - Determines the behavior of LED7 as shown in Table 5.52.

Bits 3 - 2 - LED6_CTL[1:0] - Determines the behavior of LED6 as shown in Table 5.52.

Bits 1 - 0 - LED5_CTL[1:0] - Determines the behavior of LED5 as shown in Table 5.52.

**Table 5.52 LEDx_CTL Bit Decode**

| LEDX_CTL [1:0] | | OPERATION | DESCRIPTION | START TRIGGER | STOP TRIGGER |
|---|---|---|---|---|---|
| 1 | 0 | | | | |
| 0 | 0 | Direct | The LED is driven to the programmed state (active or inactive). See Figure 5.7 | Touch Detected or LED Output Control bit set | Release Detected or LED Output Control bit cleared |
| 0 | 1 | Pulse 1 | The LED will “Pulse” a programmed number of times. During each “Pulse” the LED will breathe up to the maximum brightness and back down to the minimum brightness so that the total “Pulse” period matches the programmed value. | Touch or Release Detected or LED Output Control bit set or cleared (see Section 5.32) | n/a |
| 1 | 0 | Pulse 2 | The LED will “Pulse” when the start trigger is detected. When the stop trigger is detected, it will “Pulse” a programmable number of times then return to its minimum brightness. | Touch Detected or LED Output Control bit set | Release Detected or LED Output Control bit cleared |
| 1 | 1 | Breathe | The LED will breathe. It will be driven with a duty cycle that ramps up from the programmed minimum duty cycle (default 0%) to the programmed maximum duty cycle duty cycle (default 100%) and then back down. Each ramp takes up 50% of the programmed period. The total period of each “breath” is determined by the LED Breathe Period controls - see Section 5.34. | Touch Detected or LED Output Control bit set | Release Detected or LED Output Control bit cleared |

**APPLICATION NOTE:** The PWM frequency is determined based on the selected LED behavior, the programmed breathe period, and the programmed min and max duty cycles. For the Direct behavior mode, the PWM frequency is calculated based on the programmed Rise and Fall times. If these are set at 0, then the maximum PWM frequency will be used based on the programmed duty cycle settings.

## 5.32 LED Pulse 1 Period Register

**Table 5.53 LED Pulse 1 Period Register**

| ADDR | R/W | REGISTER | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 | DEFAULT |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 84h | R/W | LED Pulse 1 Period | ST_ TRIG | P1_ PER6 | P1_ PER5 | P1_ PER4 | P1_ PER3 | P1_ PER2 | P1_ PER1 | P1_ PER0 | 20h |

The LED Pulse Period 1 register determines the overall period of a pulse operation as determined by the LED_CTL registers (see Table 5.52 - setting 01b). The LSB represents 32ms so that a setting of 18h (24d) would represent a period of 768ms (24 x 32ms = 768ms). The total range is from 32ms to 4.064 seconds as shown in Table 5.54 with the default being 1024ms.

**APPLICATION NOTE:** Due to constraints on the LED Drive PWM operation, any Breathe Period less than 160ms (05h) may not be achievable. The device will breathe at the minimum period possible as determined by the period and min / max duty cycle settings.

Bit 7 - ST_TRIG - Determines the start trigger for the LED Pulse behavior.

- ‘0’ (default) - The LED will Pulse when a touch is detected or the drive bit is set.
- ‘1’ - The LED will Pulse when a release is detected or the drive bit is cleared.

The Pulse 1 operation is shown in Figure 5.1 when the LED output is configured for non-inverted polarity (LEDx_POL = 1) and in Figure 5.2 for inverted polarity (LEDx_POL = 0).

.

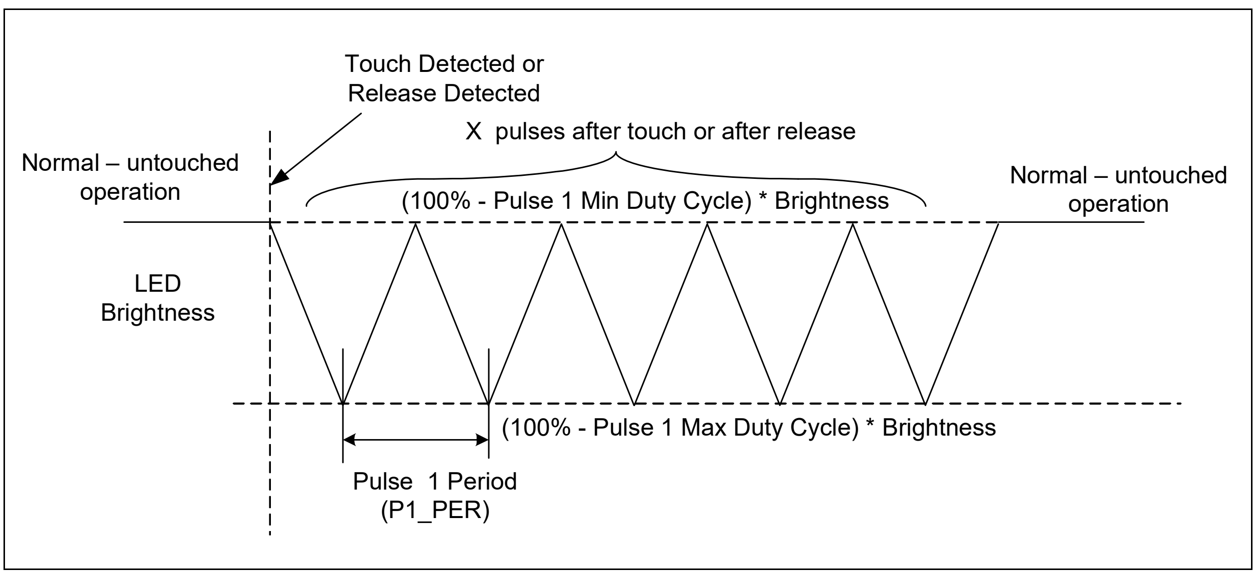



**Figure 5.1 Pulse 1 Behavior with Non-Inverted Polarity**

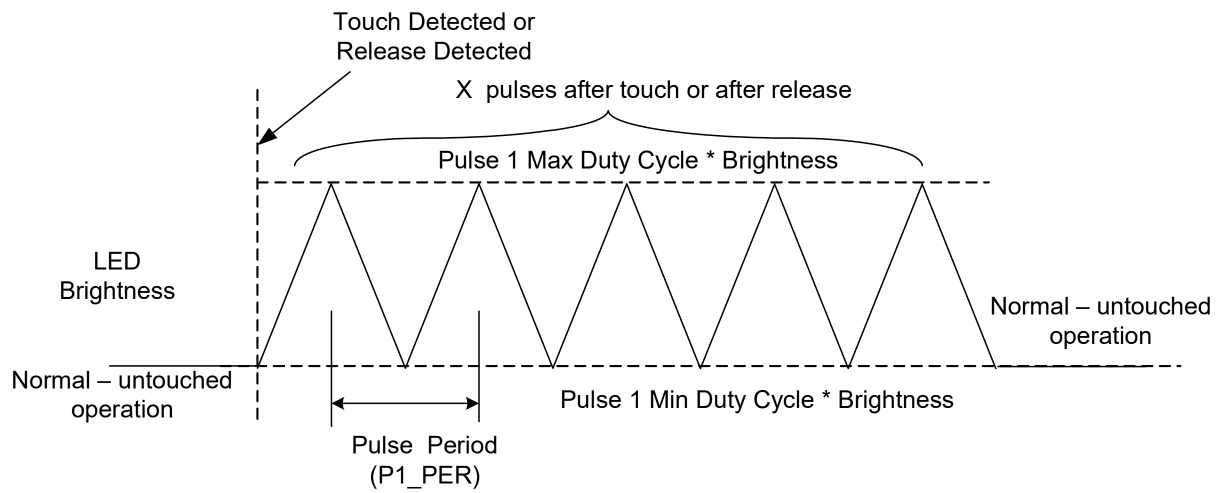



**Figure 5.2 Pulse 1 Behavior with Inverted Polarity**

**Table 5.54 LED Pulse / Breathe Period Example**

| SETTING (HEX) | SETTING (DECIMAL) | TOTAL BREATHE / PULSE PERIOD (MS) |
|---|---|---|
| 00h | 0 | 32 |
| 01h | 1 | 32 |
| 02h | 2 | 64 |
| 03h | 3 | 96 |
| . . . | . . . | . . . |
| 7Dh | 125 | 4000 |
| 7Eh | 126 | 4032 |
| 7Fh | 127 | 4064 |

## 5.33 LED Pulse 2 Period Register

**Table 5.55 LED Pulse 2 Period Register**

| ADDR | R/W | REGISTER | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 | DEFAULT |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 85h | R/W | LED Pulse 2 Period | - | P2_ PER6 | P2_ PER5 | P2_ PER4 | P2_ PER3 | P2_ PER2 | P2_ PER1 | P2_ PER0 | 14h |

The LED Pulse 2 Period register determines the overall period of a pulse operation as determined by the LED_CTL registers (see Table 5.52 - setting 10b). The LSB represents 32ms so that a setting of

18h (24d) would represent a period of 768ms. The total range is from 32ms to 4.064 seconds (see Table 5.54) with a default of 640ms.

**APPLICATION NOTE:** Due to constraints on the LED Drive PWM operation, any Breathe Period less than 160ms (05h) may not be achievable. The device will breathe at the minimum period possible as determined by the period and min / max duty cycle settings.

The Pulse 2 Behavior is shown in Figure 5.3 for non-inverted polarity (LEDx_POL = 1) and in Figure 5.4 for inverted polarity (LEDx_POL = 0).

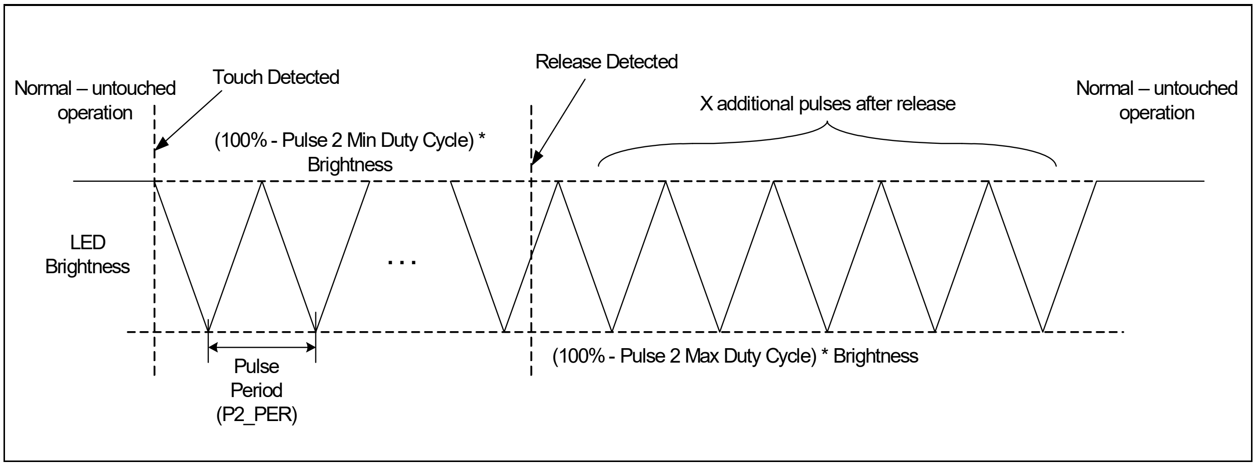



**Figure 5.3 Pulse 2 Behavior with Non-Inverted Polarity**

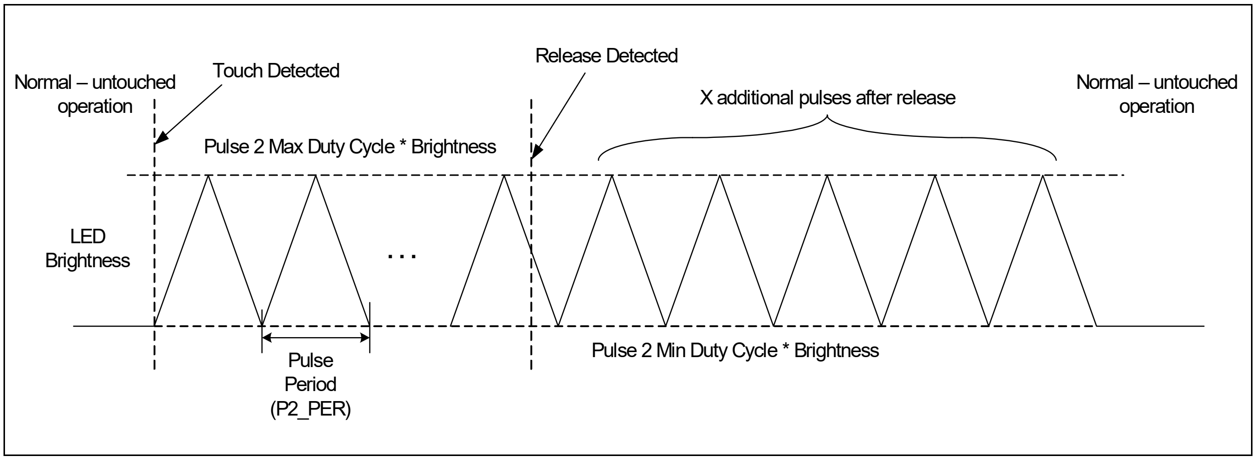



**Figure 5.4 Pulse 2 Behavior with Inverted Polarity**

## 5.34 LED Breathe Period Register

**Table 5.56 LED Breathe Period Register**

| ADDR | R/W | REGISTER | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 | DEFAULT |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 86h | R/W | LED Breathe Period | - | BR_ PER6 | BR_ PER5 | BR_ PER4 | BR_ PER3 | BR_ PER2 | BR_ PER1 | BR_ PER0 | 5Dh |

The LED Breathe Period register determines the overall period of a breathe operation as determined by the LED_CTL registers (see Table 5.52 - setting 11b). The LSB represents 32ms so that a setting of 18h (24d) would represent a period of 768ms. The total range is from 32ms to 4.064 seconds (see Table 5.54) with a default of 2976ms.

**APPLICATION NOTE:** Due to constraints on the LED Drive PWM operation, any Breathe Period less than 160ms (05h) may not be achievable. The device will breathe at the minimum period possible as determined by the period and min / max duty cycle settings.

## 5.35 LED Configuration Register

**Table 5.57 LED Configuration Register**

| ADDR | R/W | REGISTER | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 | DEFAULT |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 88h | R/W | LED Config | - | RAMP_ ALERT | PULSE2_CNT[2:0] | | | PULSE1_CNT[2:0] | | | 04h |

The LED Configuration register controls general LED behavior as well as the number of pulses that are sent for the PULSE LED output behavior.

Bit 6 - RAMP_ALERT - Determines whether the device will assert the ALERT# pin when LEDs actuated by the LED Output Control register bits have finished their respective behaviors. Interrupts will only be generated if the LED activity is generated by writing the LED Output Control registers. Any LED activity associated with touch detection will not cause an interrupt to be generated when the LED behavior has been finished.

- ‘0’ (default) - The ALERT# pin will not be asserted when LEDs actuated by the LED Output Control register have finished their programmed behaviors.
- ‘1’ - The ALERT# pin will be asserted whenever any LED that is actuated by the LED Output Control register has finished its programmed behavior.

Bits 5 - 3 - PULSE2_CNT[2:0] - Determines the number of pulses used for the Pulse 2 behavior as shown in Table 5.58.

Bits 2 - 0 - PULSE1_CNT[2:0] - Determines the number of pulses used for the Pulse 1 behavior as shown in Table 5.58.

**Table 5.58 PULSEX_CNT Decode**

| PULSEX_CNT[2:0] | | | NUMBER OF BREATHS |
|---|---|---|---|
| 2 | 1 | 0 | |
| 0 | 0 | 0 | 1 (default - Pulse 2) |
| 0 | 0 | 1 | 2 |
| 0 | 1 | 0 | 3 |
| 0 | 1 | 1 | 4 |
| 1 | 0 | 0 | 5 (default - Pulse 1) |
| 1 | 0 | 1 | 6 |
| 1 | 1 | 0 | 7 |

**Table 5.58 PULSEX_CNT Decode (continued)**

| PULSEX_CNT[2:0] | | | NUMBER OF BREATHS |
|---|---|---|---|
| **2** | **1** | **0** | |
| 1 | 1 | 1 | 8 |

## 5.36 LED Duty Cycle Registers

**Table 5.59 LED Duty Cycle Registers**

| ADDR | R/W | REGISTER | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 | DEFAULT |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 90h | R/W | LED Pulse 1 Duty Cycle | P1_MAX_DUTY[3:0] | | | | P1_MIN_DUTY[3:0] | | | | F0h |
| 91h | R/W | LED Pulse 2 Duty Cycle | P2_MAX_DUTY[3:0] | | | | P2_MIN_DUTY[3:0] | | | | F0h |
| 92h | R/W | LED Breathe Duty Cycle | BR_MAX_DUTY[3:0] | | | | BR_MIN_DUTY[3:0] | | | | F0h |
| 93h | R/W | Direct Duty Cycle | DR_MAX_DUTY[3:0] | | | | DR_MIN_DUTY[3:0] | | | | F0h |

The LED Duty Cycle registers determine the minimum and maximum duty cycle settings used for the LED for each LED behavior. These settings affect the brightness of the LED when it is fully off and fully on.

The LED driver duty cycle will ramp up from the minimum duty cycle to the maximum duty cycle and back down again.

**APPLICATION NOTE:** When operating in Direct behavior mode, changes to the Duty Cycle settings will be applied immediately. When operating in Breathe, Pulse 1, or Pulse 2 modes, the LED must be unactuated and then re-actuated before changes to behavior are processed.

Bits 7 - 4 - X_MAX_DUTY[3:0] - Determines the maximum PWM duty cycle for the LED drivers as shown in Table 5.60.

Bits 3 - 0 - X_MIN_DUTY[3:0] - Determines the minimum PWM duty cycle for the LED drivers as shown in Table 5.60.

**Table 5.60 LED Duty Cycle Decode**

| X_MAX/MIN_DUTY [3:0] | | | | MAXIMUM DUTY CYCLE | MINIMUM DUTY CYCLE |
|---|---|---|---|---|---|
| **3** | **2** | **1** | **0** | | |
| 0 | 0 | 0 | 0 | 7% | 0% |
| 0 | 0 | 0 | 1 | 9% | 7% |
| 0 | 0 | 1 | 0 | 11% | 9% |
| 0 | 0 | 1 | 1 | 14% | 11% |
| 0 | 1 | 0 | 0 | 17% | 14% |

**Table 5.60 LED Duty Cycle Decode (continued)**

| X_MAX/MIN_DUTY [3:0] | | | | MAXIMUM DUTY CYCLE | MINIMUM DUTY CYCLE |
|---|---|---|---|---|---|
| **3** | **2** | **1** | **0** | | |
| 0 | 1 | 0 | 1 | 20% | 17% |
| 0 | 1 | 1 | 0 | 23% | 20% |
| 0 | 1 | 1 | 1 | 26% | 23% |
| 1 | 0 | 0 | 0 | 30% | 26% |
| 1 | 0 | 0 | 1 | 35% | 30% |
| 1 | 0 | 1 | 0 | 40% | 35% |
| 1 | 0 | 1 | 1 | 46% | 40% |
| 1 | 1 | 0 | 0 | 53% | 46% |
| 1 | 1 | 0 | 1 | 63% | 53% |
| 1 | 1 | 1 | 0 | 77% | 63% |
| 1 | 1 | 1 | 1 | 100% | 77% |

## 5.37 LED Direct Ramp Rates Register

**Table 5.61 LED Direct Ramp Rates Register**

| ADDR | R/W | REGISTER | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 | DEFAULT |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 94h | R/W | LED Direct Ramp Rates | - | - | RISE_RATE[2:0] | | | FALL_RATE[2:0] | | | 00h |

The LED Direct Ramp Rates register control the rising and falling edge time of an LED that is configured to operate in Direct behavior mode. The rising edge time corresponds to the amount of time the LED takes to transition from its minimum duty cycle to its maximum duty cycle. Conversely, the falling edge time corresponds to the amount of time that the LED takes to transition from its maximum duty cycle to its minimum duty cycle.

Bits 5 - 3 - RISE_RATE[2:0] - Determines the rising edge time of an LED when it transitions from its minimum drive state to its maximum drive state as shown in Table 5.62.

Bits 2 - 0 - FALL_RATE[2:0] - Determines the falling edge time of an LED when it transitions from its maximum drive state to its minimum drive state as shown in Table 5.62.

**Table 5.62 Rise / Fall Rate Decode**

| RISE_RATE/ FALL_RATE/ BIT DECODE | | | RISE / FALL TIME ($T_{RISE}$ / $T_{FALL}$)) |
|---|---|---|---|
| **2** | **1** | **0** | |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 250ms |

**Table 5.62 Rise / Fall Rate Decode (continued)**

| RISE_RATE/ FALL_RATE/ BIT DECODE | | | RISE / FALL TIME ($T_{RISE}$ / $T_{FALL}$) |
|---|---|---|---|
| **2** | **1** | **0** | |
| 0 | 1 | 0 | 500ms |
| 0 | 1 | 1 | 750ms |
| 1 | 0 | 0 | 1s |
| 1 | 0 | 1 | 1.25s |
| 1 | 1 | 0 | 1.5s |
| 1 | 1 | 1 | 2s |

## 5.38 LED Off Delay Register

**Table 5.63 LED Off Delay Register**

| ADDR | R/W | REGISTER | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 | DEFAULT |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 95h | R/W | LED Off Delay Register | - | BR_OFF_DLY[2:0] | | | DIR_OFF_DLY[3:0] | | | | 00h |

The LED Off Delay register determines the amount of time that an LED remains at its maximum duty cycle (or minimum as determined by the polarity controls) before it starts to ramp down. If the LED is operating in Breathe mode, this delay is applied at the top of each “breath”. If the LED is operating in the Direct mode, this delay is applied when the LED is unactuated.

Bits 6 - 4 - BR_OFF_DLY[2:0] - Determines the Breathe behavior mode off delay, which is the amount of time an LED in Breathe behavior mode remains inactive after it finishes a breathe pulse (ramp on and ramp off), as shown in Figure 5.5 (non-inverted polarity LEDx_POL = 1) and Figure 5.6 (inverted polarity LEDx_POL = 0). Available settings are shown in Table 5.64.

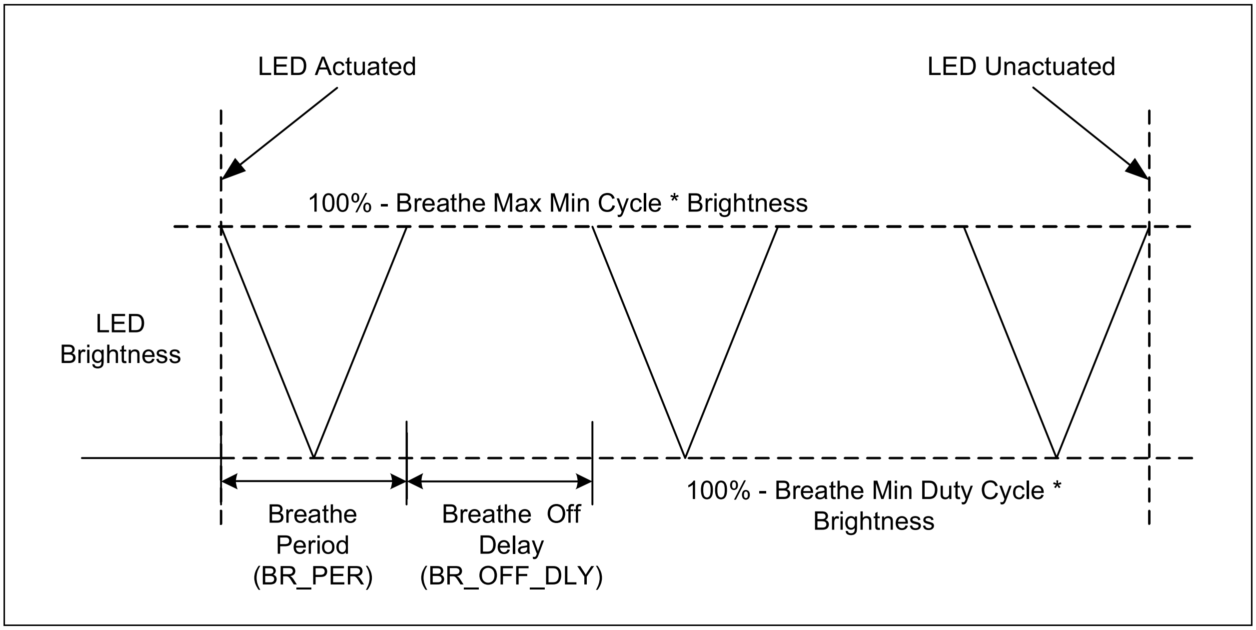



**Figure 5.5 Breathe Behavior with Non-Inverted Polarity**

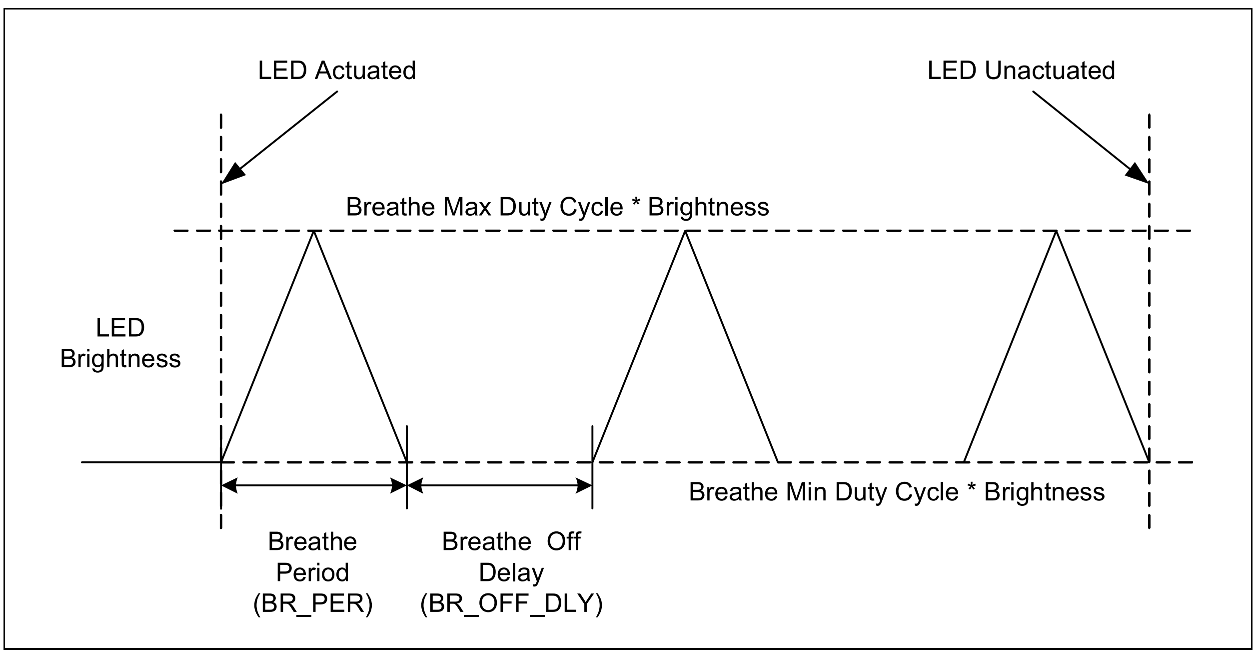



**Figure 5.6 Breathe Behavior with Inverted Polarity**

**Table 5.64 Breathe Off Delay Settings**

| BR_OFF_DLY [2:0] | | | OFF DELAY |
|---|---|---|---|
| **2** | **1** | **0** | |
| 0 | 0 | 0 | 0 (default) |
| 0 | 0 | 1 | 0.25s |
| 0 | 1 | 0 | 0.5s |
| 0 | 1 | 1 | 0.75s |
| 1 | 0 | 0 | 1.0s |
| 1 | 0 | 1 | 1.25s |
| 1 | 1 | 0 | 1.5s |
| 1 | 1 | 1 | 2.0s |

Bits 3 - 0 - DIR_OFF_DLY[3:0] - Determines the turn-off delay, as shown in Table 5.65, for all LEDs that are configured to operate in Direct behavior mode.

The Direct behavior operation is determined by the combination of programmed Rise Time, Fall Time, Min and Max Duty cycles, Off Delay, and polarity. Figure 5.7 shows the behavior for non-inverted polarity (LEDx_POL = 1) while Figure 5.8 shows the behavior for inverted polarity (LEDx_POL = 0).

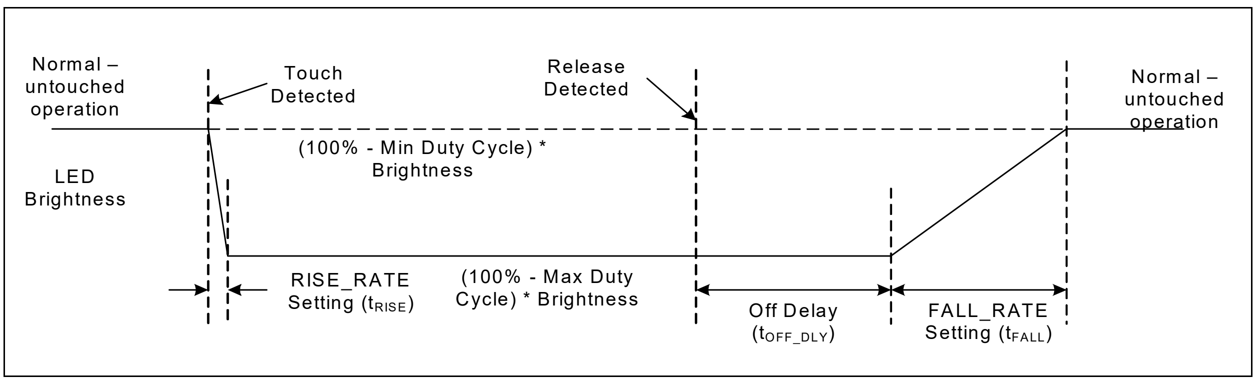



**Figure 5.7 Direct Behavior for Non-Inverted Polarity**

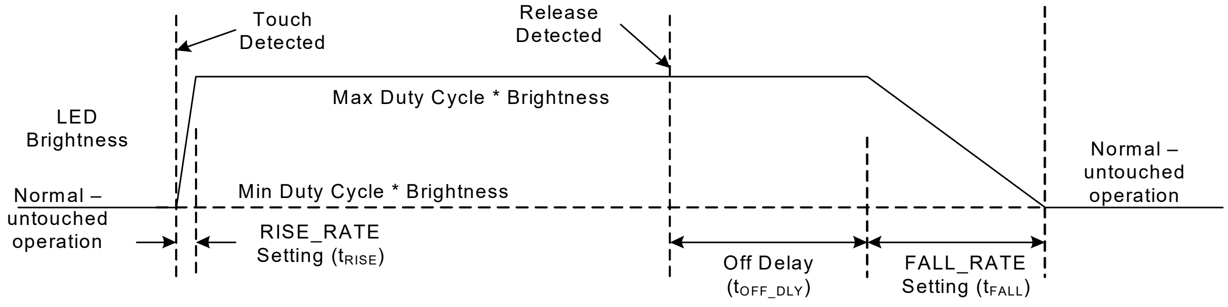



**Figure 5.8 Direct Behavior for Inverted Polarity**

**Table 5.65 Off Delay Decode**

| OFF DELAY[3:0] BIT DECODE | | | | |
|---|---|---|---|---|
| **3** | **2** | **1** | **0** | **OFF DELAY ($T_{OFF\_DLY}$)** |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 250ms |
| 0 | 0 | 1 | 0 | 500ms |
| 0 | 0 | 1 | 1 | 750ms |
| 0 | 1 | 0 | 0 | 1s |
| 0 | 1 | 0 | 1 | 1.25s |
| 0 | 1 | 1 | 0 | 1.5s |
| 0 | 1 | 1 | 1 | 2s |
| 1 | 0 | 0 | 0 | 2.5s |
| 1 | 0 | 0 | 1 | 3.0s |
| 1 | 0 | 1 | 0 | 3.5s |
| 1 | 0 | 1 | 1 | 4.0s |
| 1 | 1 | 0 | 0 | 4.5s |
| All others | | | | 5.0s |

## 5.39 Sensor Input Calibration Registers

**Table 5.66 Sensor Input Calibration Registers**

| ADDR | REGISTER | R/W | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 | DEFAULT |
|---|---|---|---|---|---|---|---|---|---|---|---|
| B1h | Sensor Input 1 Calibration | R | CAL1_9 | CAL1_8 | CAL1_7 | CAL1_6 | CAL1_5 | CAL1_4 | CAL1_3 | CAL1_2 | 00h |
| B2h | Sensor Input 2 Calibration | R | CAL2_9 | CAL2_8 | CAL2_7 | CAL2_6 | CAL2_5 | CAL2_4 | CAL2_3 | CAL2_2 | 00h |
| B3h | Sensor Input 3 Calibration | R | CAL3_9 | CAL3_8 | CAL3_7 | CAL3_6 | CAL3_5 | CAL3_4 | CAL3_3 | CAL3_2 | 00h |
| B4h | Sensor Input 4 Calibration | R | CAL4_9 | CAL4_8 | CAL4_7 | CAL4_6 | CAL4_5 | CAL4_4 | CAL4_3 | CAL4_2 | 00h |
| B5h | Sensor Input 5 Calibration | R | CAL5_9 | CAL5_8 | CAL5_7 | CAL5_6 | CAL5_5 | CAL5_4 | CAL5_3 | CAL5_2 | 00h |
| B6h | Sensor Input 6 Calibration | R | CAL6_9 | CAL6_8 | CAL6_7 | CAL6_6 | CAL6_5 | CAL6_4 | CAL6_3 | CAL6_2 | 00h |
| B7h | Sensor Input 7 Calibration | R | CAL7_9 | CAL7_8 | CAL7_7 | CAL7_6 | CAL7_5 | CAL7_4 | CAL7_3 | CAL7_2 | 00h |
| B8h | Sensor Input 8 Calibration | R | CAL8_9 | CAL8_8 | CAL8_7 | CAL8_6 | CAL8_5 | CAL8_4 | CAL8_3 | CAL8_2 | 00h |
| B9h | Sensor Input Calibration LSB 1 | R | CAL4_1 | CAL4_0 | CAL3_1 | CAL3_0 | CAL2_1 | CAL2_0 | CAL1_1 | CAL1_0 | 00h |
| BAh | Sensor Input Calibration LSB 2 | R | CAL8_1 | CAL8_0 | CAL7_1 | CAL7_0 | CAL6_1 | CAL6_0 | CAL5_1 | CAL5_0 | 00h |

The Sensor Input Calibration registers hold the 10-bit value that represents the last calibration value.

## 5.40 Product ID Register

**Table 5.67 Product ID Register**

| ADDR | R/W | REGISTER | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 | DEFAULT |
|---|---|---|---|---|---|---|---|---|---|---|---|
| FDh | R | Product ID | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 50h |

The Product ID register stores a unique 8-bit value that identifies the device.

## 5.41 Manufacturer ID Register

**Table 5.68 Vendor ID Register**

| ADDR | R/W | REGISTER | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 | DEFAULT |
|---|---|---|---|---|---|---|---|---|---|---|---|
| FEh | R | Manufacturer ID | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 5Dh |

The Vendor ID register stores an 8-bit value that represents SMSC.

## 5.42 Revision Register

**Table 5.69 Revision Register**

| ADDR | R/W | REGISTER | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 | DEFAULT |
|---|---|---|---|---|---|---|---|---|---|---|---|
| FFh | R | Revision | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 83h |

The Revision register stores an 8-bit value that represents the part revision.

# Chapter 6 Package Information

## 6.1 CAP1188 Package Drawings

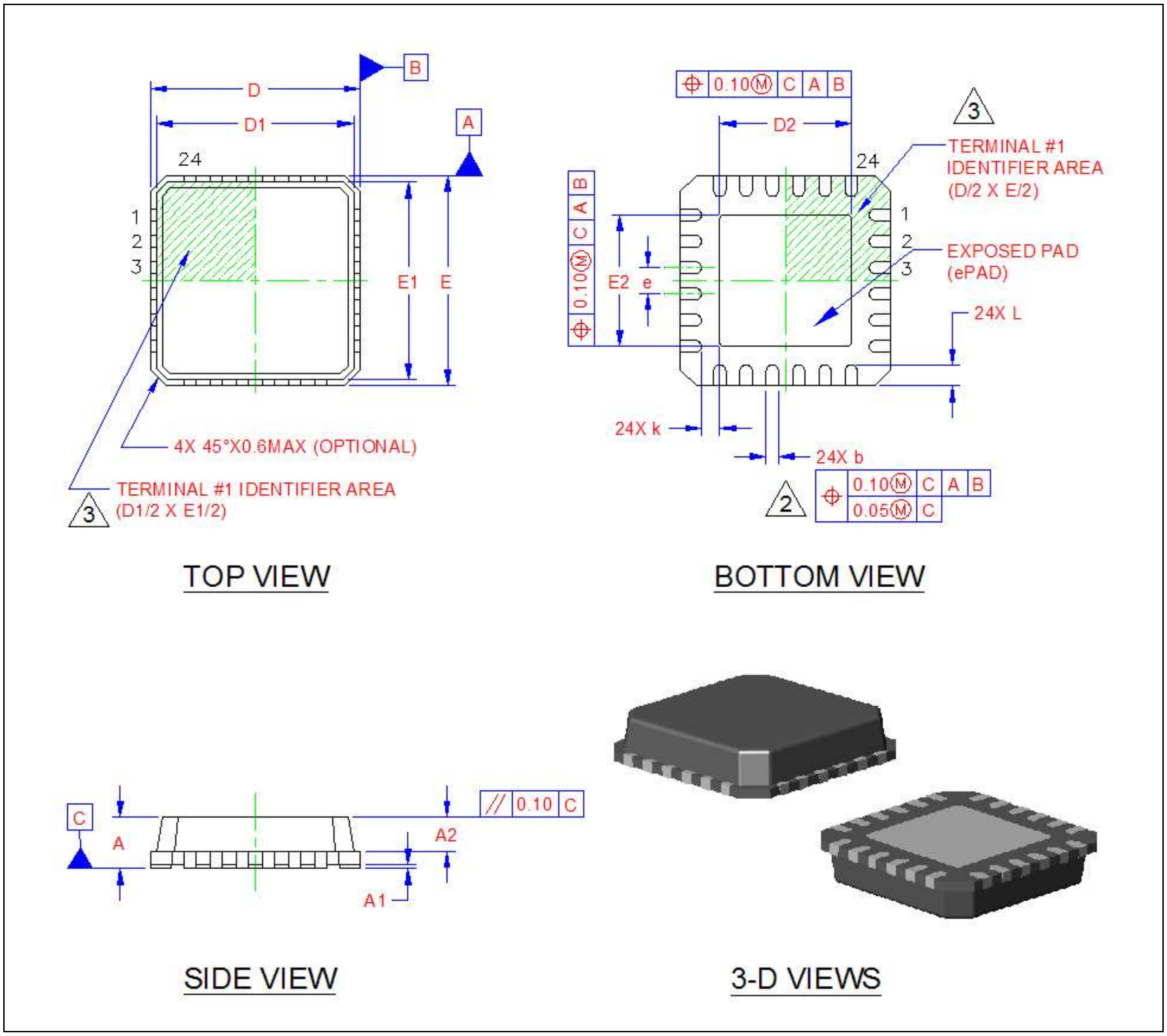



**Figure 6.1 CAP1188 Package Drawing - 24-Pin QFN 4mm x 4mm**

| COMMON DIMENSIONS | | | | | |
|---|---|---|---|---|---|
| SYMBOL | MIN | NOM | MAX | NOTE | REMARK |
| A | 0.70 | 0.85 | 1.00 | - | OVERALL PACKAGE HEIGHT |
| A1 | 0 | 0.02 | 0.05 | - | STANDOFF |
| A2 | - | - | 0.90 | - | MOLD CAP THICKNESS |
| D/E | 3.90 | 4.00 | 4.10 | - | X/Y BODY SIZE |
| D1/E1 | 3.55 | 3.75 | 3.95 | - | X/Y MOLD CAP SIZE |
| D2/E2 | 2.40 | 2.50 | 2.60 | - | X/Y EXPOSED PAD SIZE |
| L | 0.30 | 0.40 | 0.50 | - | TERMINAL LENGTH |
| b | 0.18 | 0.25 | 0.30 | 2 | TERMINAL WIDTH |
| k | 0.25 | - | - | - | PIN TO ePAD CLEARANCE |
| e | 0.50 BSC | | | - | TERMINAL PITCH |

**NOTES:**

1. ALL DIMENSIONS ARE IN MILLIMETERS.
2. DIMENSIONS "b" APPLIES TO PLATED TERMINALS AND IT IS MEASURED BETWEEN 0.15 AND 0.30 mm FROM THE TERMINAL TIP.
3. DETAILS OF TERMINAL #1 IDENTIFIER ARE OPTIONAL BUT MUST BE LOCATED WITHIN THE AREA INDICATED.

**Figure 6.2 CAP1188 Package Dimensions - 24-Pin QFN 4mm x 4mm**

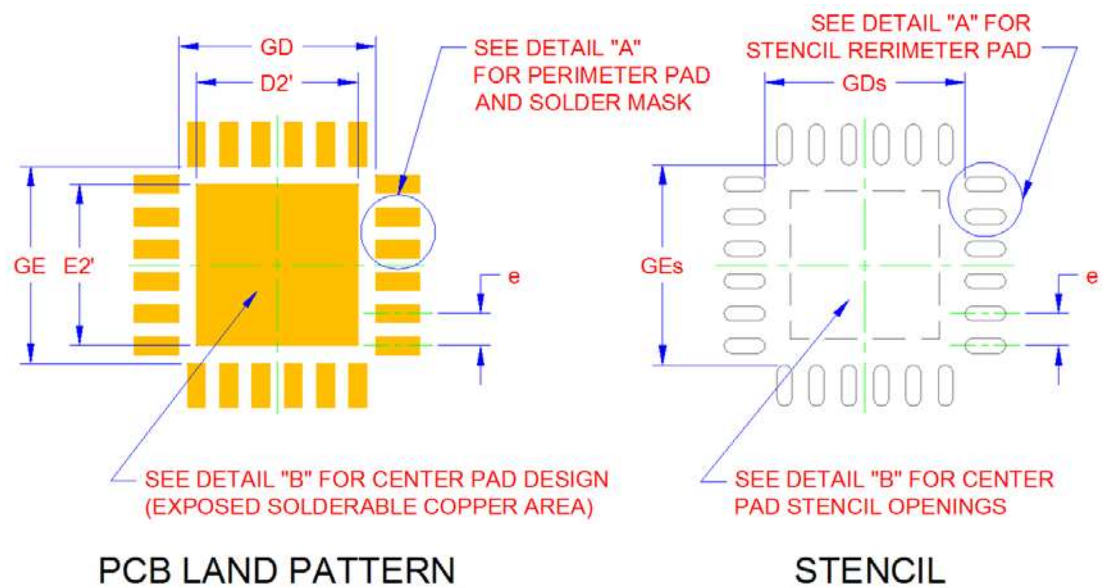



**Figure 6.3 CAP1188 PCB Land Pattern and Stencil - 24-Pin QFN 4mm x 4mm**

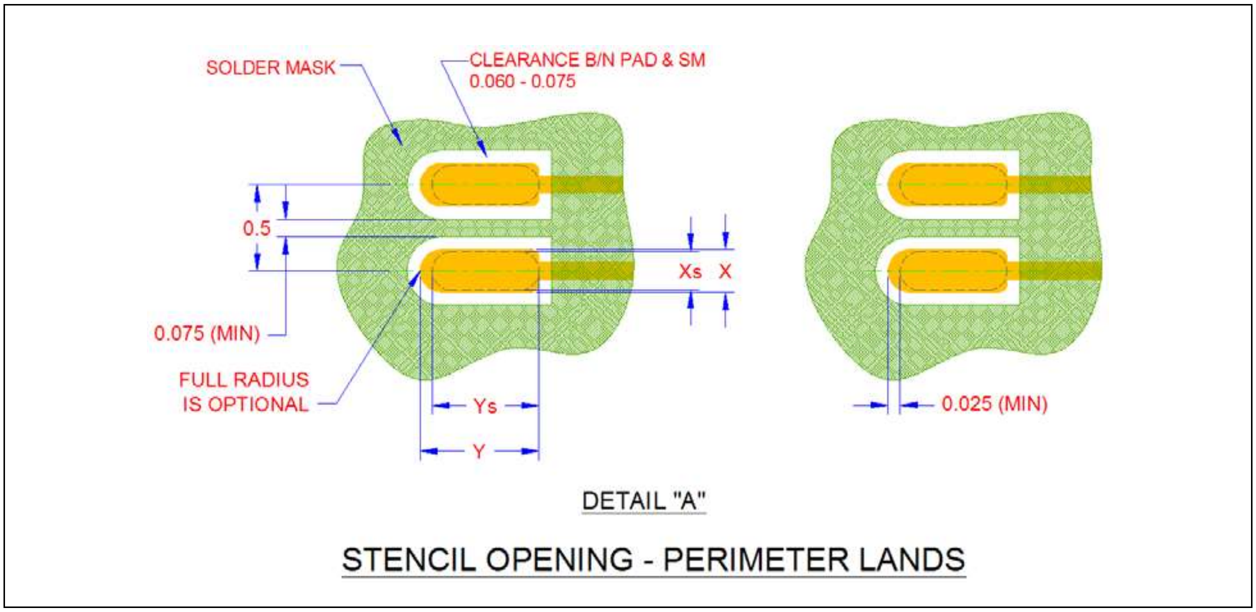



**Figure 6.4 CAP1188 PCB Detail A - 24-Pin QFN 4mm x 4mm**

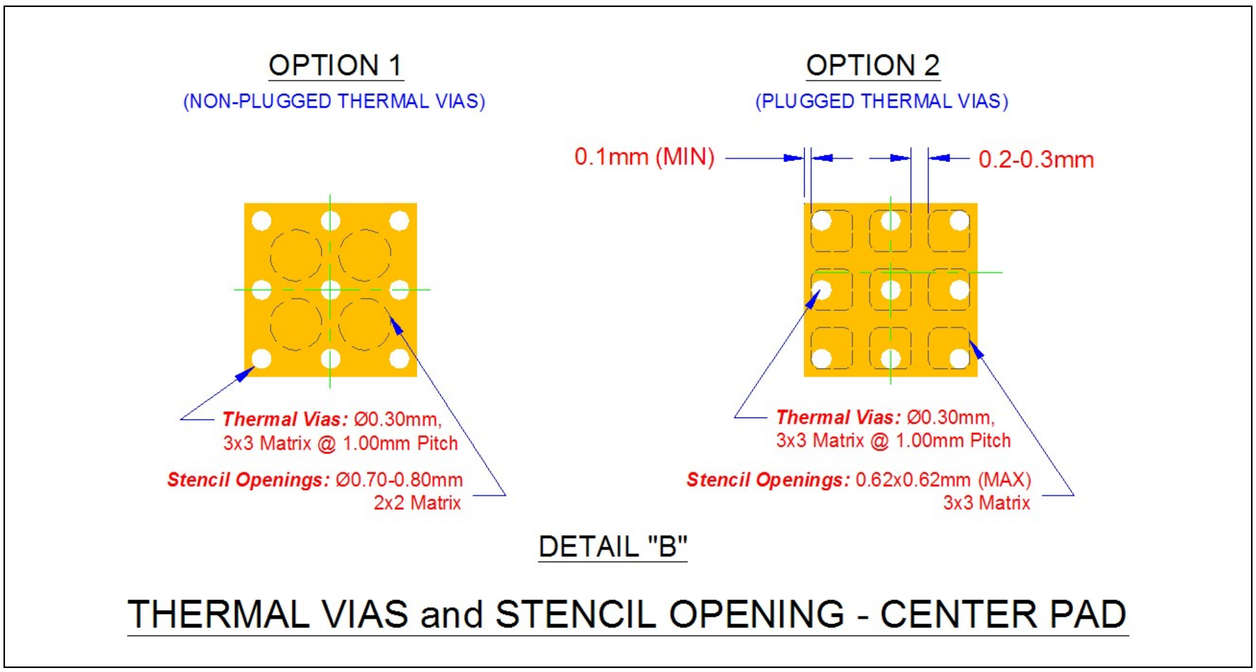



**Figure 6.5 CAP1188 PCB Detail B - 24-Pin QFN 4mm x 4mm**

| LAND PATTERN DIMENSIONS | | | |
|---|---|---|---|
| SYMBOL | MIN | NOM | MAX |
| GD/GE | 3.05 | - | 3.10 |
| GDs/GEs | 3.10 | - | - |
| D2'/E2' | - | 2.50 | 2.50 |
| Pad: X | - | 0.28 | 0.28 |
| Stencil: Xs | - | 0.23 | 0.25 |
| Pad: Y | - | 0.69 | 0.69 |
| Stencil: Ys | - | 0.62 | 0.64 |
| e | 0.50 | | |

**Figure 6.6 CAP1188 Land Dimensions - 24-Pin QFN 4mm x 4mm**

SMT APPLICATION NOTES (QFN)

1. THE USER MAY MODIFY THE PCB LAND PATTERN DIMENSIONS BASED ON THEIR EXPERIENCE AND/OR PROCESS CAPABILITY.
2. THE LAND PATTERN CORRESPONDING TO THE PACKAGE EXPOSED PAD (IN THE CENTER) CAN BE LARGER, AND WITH DIFFERENT SHAPE THAN THE EXPOSED PAD ON THE PACKAGE. HOWEVER, THE SOLDERABLE AREA, AS DEFINED BY THE SOLDER MASK (SMD), OR NON-SOLDER MASK DEFINED (NSMD), SHOULD BE AS SHOWN FOR THE BEST THERMAL & ELECTRICAL PERFORMANCE.
3. MAXIMUM THERMAL AND ELECTRICAL PERFORMANCE IS ACHIEVED WHEN AN ARRAY OF SOLID VIAS IS INCORPORATED IN THE CENTER LAND PATTERN. (See Options 1 & 2)
4. THE VIAS SHOULD BE AT 0.8 to 1.2MM PITCH WITH 0.30 TO 0.40MM DIAMETER, AND 1 OZ COPPER VIA BARREL PLATING.
5. NON SOLDER MASK DEFINED (NSMD) PAD DESIGN IS RECOMMENDED FOR PERIMETER LANDS.
6. A LASER-CUT STAINLESS STEEL STENCIL IS RECOMMENDED WITH ELECTRO POLISHED TRAPEZOIDAL WALLS. THE RECOMMENDED STENCIL THICKNESS IS 0.125 mm FOR PITCHES 0.4 and 0.5 mm.
7. RECOMMENDED STENCIL AREA & ASPECT RATIOS ARE 0.66 & 1.5 (MIN) RESPECTIVELY.
8. RECOMMENDED STENCIL APERTURES ARE AS SHOWN.
9. IT IS RECOMMENDED TO USE "NO-CLEAN", TYPE 3 SOLDER PASTE.
10. THE REFLOW PROFILE DEPENDS ON THE EXACT SOLDER PASTE USED AND THE GIVEN BOARD DETAILS, SUCH AS GEOMETRY, COMPONENTS ETC.

**Figure 6.7 QFN Application Notes**

## 6.2 Package Marking

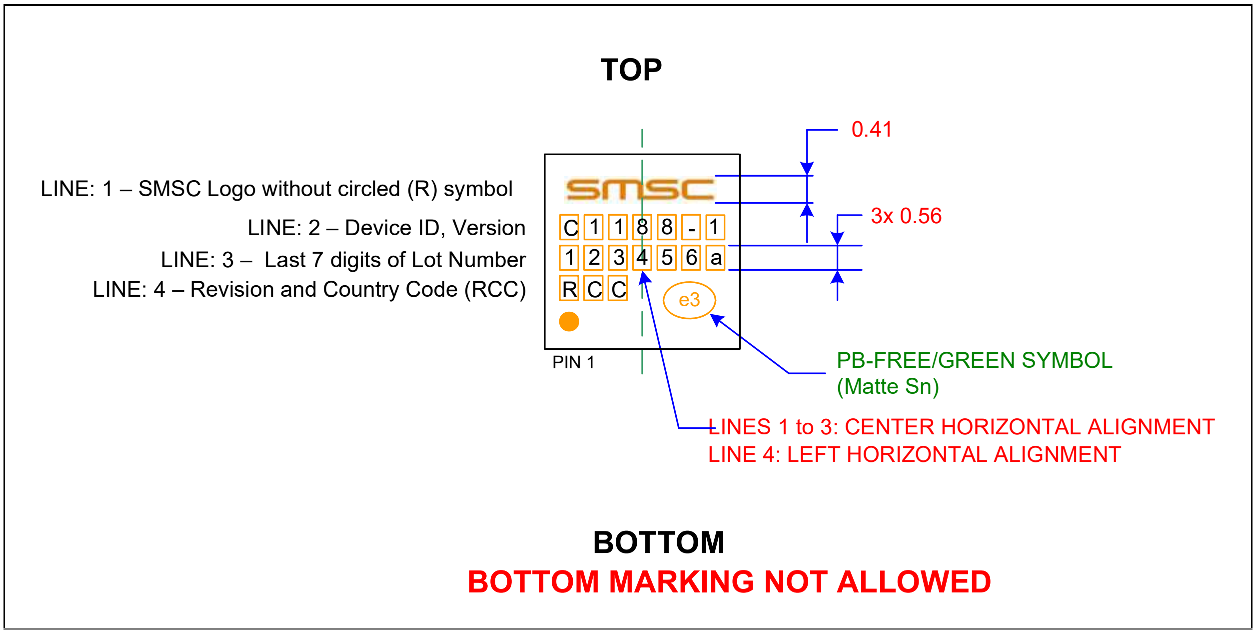



**Figure 6.8 CAP1188 Package Markings**

# Appendix A Device Delta

## A.1 Delta from CAP1088 to CAP1188

1. Updated circuitry to improve power supply rejection.
2. Updated LED driver duty cycle decode values to have more distribution at lower values - closer to a logarithmic curve. See Table 5.60, "LED Duty Cycle Decode".
3. Updated bug that breathe periods were not correct above 2.6s. This includes rise / fall time decodes above 1.5s.
4. Added filtering on RESET pin to prevent errant resets.
5. Updated controls so that the RESET pin assertion places the device into the lowest power state available and causes an interrupt when released. See Section 4.2, "RESET Pin".
6. Added 1 bit to the LED Off Delay register (see Section 5.38, "LED Off Delay Register") to extend times from 2s to 5s in 0.5s intervals.
7. Breathe behavior modified. A breathe off delay control was added to the LED Off Delay Register (see Section 5.38, "LED Off Delay Register") so the LEDs can be configured to remain inactive between breathes.
8. Added controls for the LED transition effects when linking LEDs to capacitive sensor inputs. See Section 5.29, "Linked LED Transition Control Register".
9. Added controls to “mirror” the LED duty cycle outputs so that when polarity changes, the LED brightness levels look right. These bits are automatically set when polarity is set. Added control to break this auto-set behavior. See Section 5.30, "LED Mirror Control Register".
10. Added Multiple Touch Pattern detection circuitry. See Section 5.15, "Multiple Touch Pattern Configuration Register".
11. Added General Status register to flag Multiple touches, Multiple Touch Pattern issues and general touch detections. See Section 5.2, "Status Registers".
12. Added bits 6 and 5 to the Recalibration Configuration register (2Dh - see Section 5.17, "Recalibration Configuration Register"). These bits control whether the accumulation of intermediate data and the consecutive negative delta counts counter are cleared when the noise status bit is set.
13. Added Configuration 2 register for LED linking controls, noise detection controls, and control to interrupt on press but not on release. Added control to change alert pin polarity. See Section 5.6, "Configuration Registers".
14. Updated Deep Sleep behavior so that device does not clear DSLEEP bit on received communications but will wake to communicate.
15. Changed PWM frequency for LED drivers. The PWM frequency was derived from the programmed breathe period and duty cycle settings and it ranged from ~4Hz to ~8000 Hz. The PWM frequency has been updated to be a fixed value of ~2000Hz.
16. Register delta:

**Table A.1 Register Delta From CAP1088 to CAP1188**

| ADDRESS | REGISTER DELTA | DELTA | DEFAULT |
|---|---|---|---|
| 00h<br>Page 41 | Changed - Main Status / Control | added bits 7-6 to control gain | 00h |
| 02h<br>Page 42 | New - General Status | new register to store MTP, MULT, LED, RESET, and general TOUCH bits | 00h |

**Table A.1 Register Delta From CAP1088 to CAP1188 (continued)**

| ADDRESS | REGISTER DELTA | DELTA | DEFAULT |
|---|---|---|---|
| 44h<br>Page 47 | New - Configuration 2 | new register to control alert polarity, LED touch linking behavior, LED output behavior, and noise detection, and interrupt on release | 40h |
| 24h<br>Page 52 | Changed - Averaging Control | updated register bits - moved SAMP_AVG[2:0] bits and added SAMP_TIME bit 1. Default changed | 39h |
| 2Bh<br>Page 56 | New - Multiple Touch Pattern Configuration | new register for Multiple Touch Pattern configuration - enable and threshold settings | 80h |
| 2Dh<br>Page 58 | New - Multiple Touch Pattern Register | new register for Multiple Touch Pattern detection circuitry - pattern or number of sensor inputs | FFh |
| 2Fh<br>Page 58 | Changed - Recalibration Configuration | updated register - updated CAL_CFG bit decode to add a 128 averages setting and removed highest time setting. Default changed. Added bit 6 NO_CLR_INTD and bit 5 NO_CLR_NEG. | 8Ah |
| 38h<br>Page 61 | Changed - Sensor Input Noise Threshold | updated register bits - removed bits 7 - 3 and consolidated all controls into bits 1 - 0. These bits will set the noise threshold for all channels. Default changed | 01h |
| 39h | Removed - Noise Threshold Register 2 | removed register | n/a |
| 41h<br>Page 62 | Changed - Standby Configuration | updated register bits - moved STBY_AVG[2:0] bits and added STBY_TIME bit 1. Default changed | 39h |
| 77h<br>Page 69 | New - Linked LED Transition Control | new register to control transition effect when LED linked to sensor inputs | 00h |
| 79h<br>Page 70 | New - LED Mirror Control | new register to control LED output mirroring for brightness control when polarity changed | 00h |
| 90h<br>Page 77 | Changed - LED Pulse 1 Duty Cycle | changed bit decode to be more logarithmic | F0h |
| 91h<br>Page 77 | Changed - LED Pulse 2 Duty Cycle | changed bit decode to be more logarithmic | F0h |
| 92h<br>Page 77 | Changed - LED Breathe Duty Cycle | changed bit decode to be more logarithmic | F0h |
| 93h<br>Page 77 | Changed - LED Direct Duty Cycle | changed bit decode to be more logarithmic | F0h |
| 95h | Added controls - LED Off Delay | Added bits 6-4 BR_OFF_DLY[2:0]<br>Added bit 3 DIR_OFF_DLY[3] | 00h |
| FDh<br>Page 83 | Changed - Product ID | Changed bit decode for CAP1188 | 50h |

# Chapter 7 Datasheet Revision History

**Table 7.1 Customer Revision History**

| REVISION LEVEL & DATE | SECTION/FIGURE/ENTRY | CORRECTION |
|---|---|---|
| Rev. 1.32 (01-05-12) | Table 2.2, "Electrical Specifications" | Added conditions for $t_{HD:DAT}$. |
| | Section 3.2.7, "SMBus and I2C Compatibility" | Renamed from “SMBus and I2C Compliance.” First paragraph, added last sentence: “For information on using the CAP1188 in an $I^2C$ system, refer to SMSC AN 14.0 SMSC Dedicated Slave Devices in $I^2C$ Systems.” Added: CAP1188 supports $I^2C$ fast mode at 400kHz. This covers the SMBus max time of 100kHz. |
| | Section 5.4, "Sensor Input Delta Count Registers" | Changed negative value cap from FFh to 80h. |
| Rev. 1.31 (08-18-11) | Section 3.3.3, "SMBus Send Byte" | Added an application note: The Send Byte protocol is not functional in Deep Sleep (i.e., DSLEEP bit is set). |
| | Section 3.3.4, "SMBus Receive Byte" | Added an application note: The Receive Byte protocol is not functional in Deep Sleep (i.e., DSLEEP bit is set). |
| Rev. 1.3 (05-18-11) | Section 5.42, "Revision Register" | Updated revision ID from 82h to 83h. |
| Rev. 1.2 (02-10-11) | Section A.8, "Delta from Rev B (Mask B0) to Rev C (Mask B1)" | Added. |
| | Table 1.1, "Pin Description for CAP1188" | Changed value in “Unused Connection” column for the ADDR_COMM pin from “Connect to Ground” to “n/a“. |
| | Table 2.2, "Electrical Specifications" | PSR improvements made in functional revision B. Changed PSR spec from ±100 typ and ±200 max counts / V to ±3 and ±10 counts / V. Conditions updated. |
| | Section 4.5.2, "Recalibrating Sensor Inputs" | Added more detail with subheadings for each type of recalibration. |
| | Section 5.6, "Configuration Registers" | Added bit 5 BLK_PWR_CTRL to the Configuration 2 Register 44h. The TIMEOUT bit is set to ‘1’ by default for functional revision B and is set to ‘0’ by default for functional revision C. |
| | Section 5.42, "Revision Register" | Updated revision ID in register FFh from 81h to 82h. |
| Rev. 1.1 (11-17-10) | Document | Updated for functional revision B. See Section A.7, "Delta from Rev A (Mask A0) to Rev B (Mask B0)". |

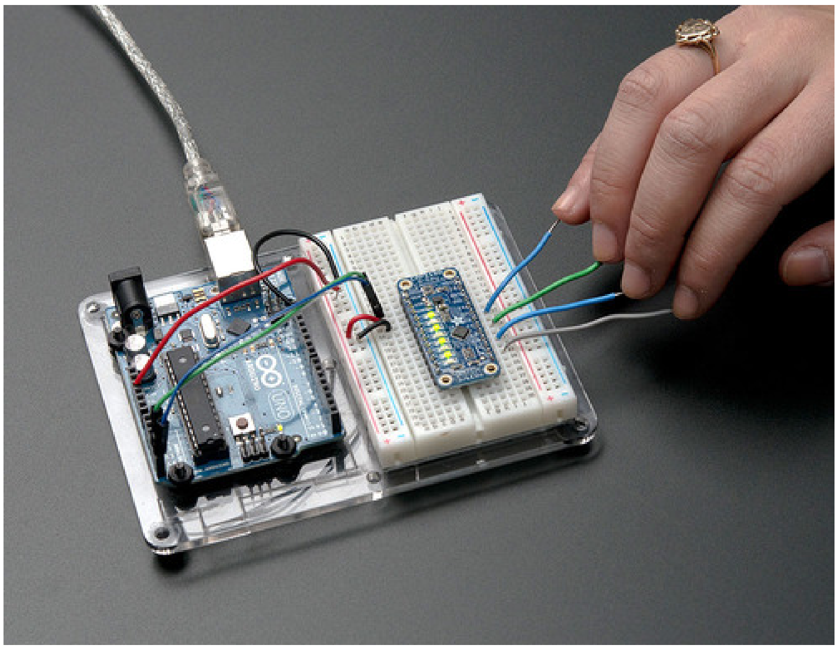
**Table 7.1 Customer Revision History (continued)**

| REVISION LEVEL & DATE | SECTION/FIGURE/ENTRY | CORRECTION |
|---|---|---|
| | Cover | Added to General Description: “includes circuitry and support for enhanced sensor proximity detection.”<br><br>Added the following Features:<br>■ Calibrates for Parasitic Capacitance<br>■ Analog Filtering for System Noise Sources<br>■ Press and Hold feature for Volume-like Applications |
| | Table 2.2, "Electrical Specifications" | Conditions for Power Supply Rejection modified adding the following:<br>Sampling time = 2.56ms<br>Averaging = 1<br>Negative Delta Counts = Disabled<br>All other parameters default |
| | Section 5.11, "Calibration Activate Register" | Updated register description to indicate which re-calibration routine is used. |
| | Section 5.14, "Multiple Touch Configuration Register" | Updated register description to indicate what will happen. |
| | Table 5.34, "CSx_BN_TH Bit Decode" | Table heading changed from “Threshold Divide Setting” to “Percent Threshold Setting”. |
| Rev. 1.0 (06-14-10) | Initial release | |

adafruit learning system

# Adafruit CAP1188 Breakout

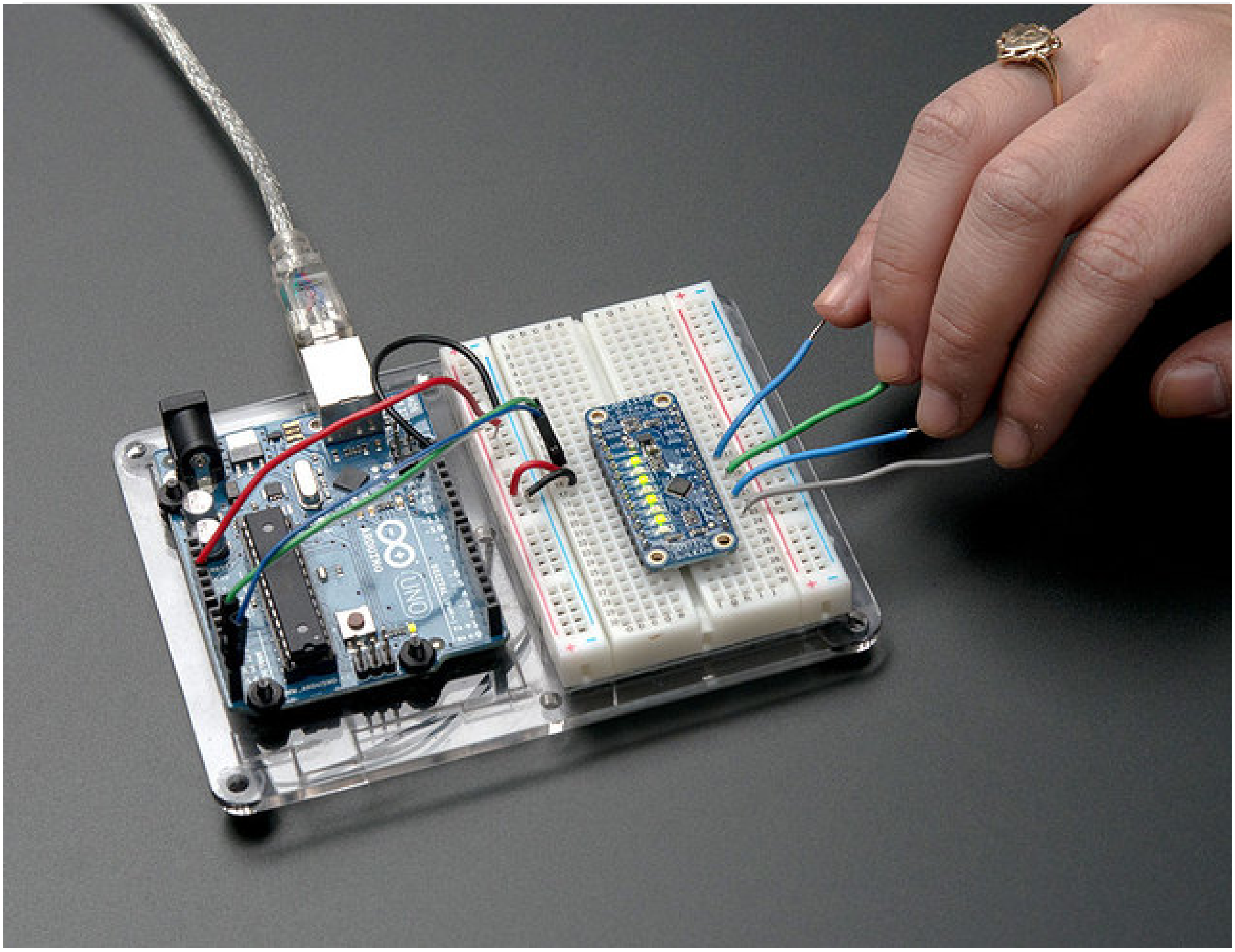
Created by lady ada



Last updated on 2014-05-14 12:00:10 PM EDT

# Guide Contents

# Overview



Add lots of touch sensors to your next microcontroller project with this easy-to-use 8-channel capacitive touch sensor breakout board, starring the CAP1188. This chip can handle up to 8 individual touch pads, and has a very nice feature that makes it stand out for us: it will light up the 8 onboard LEDs when the matching touch sensor fires to help you debug your sensor setup.

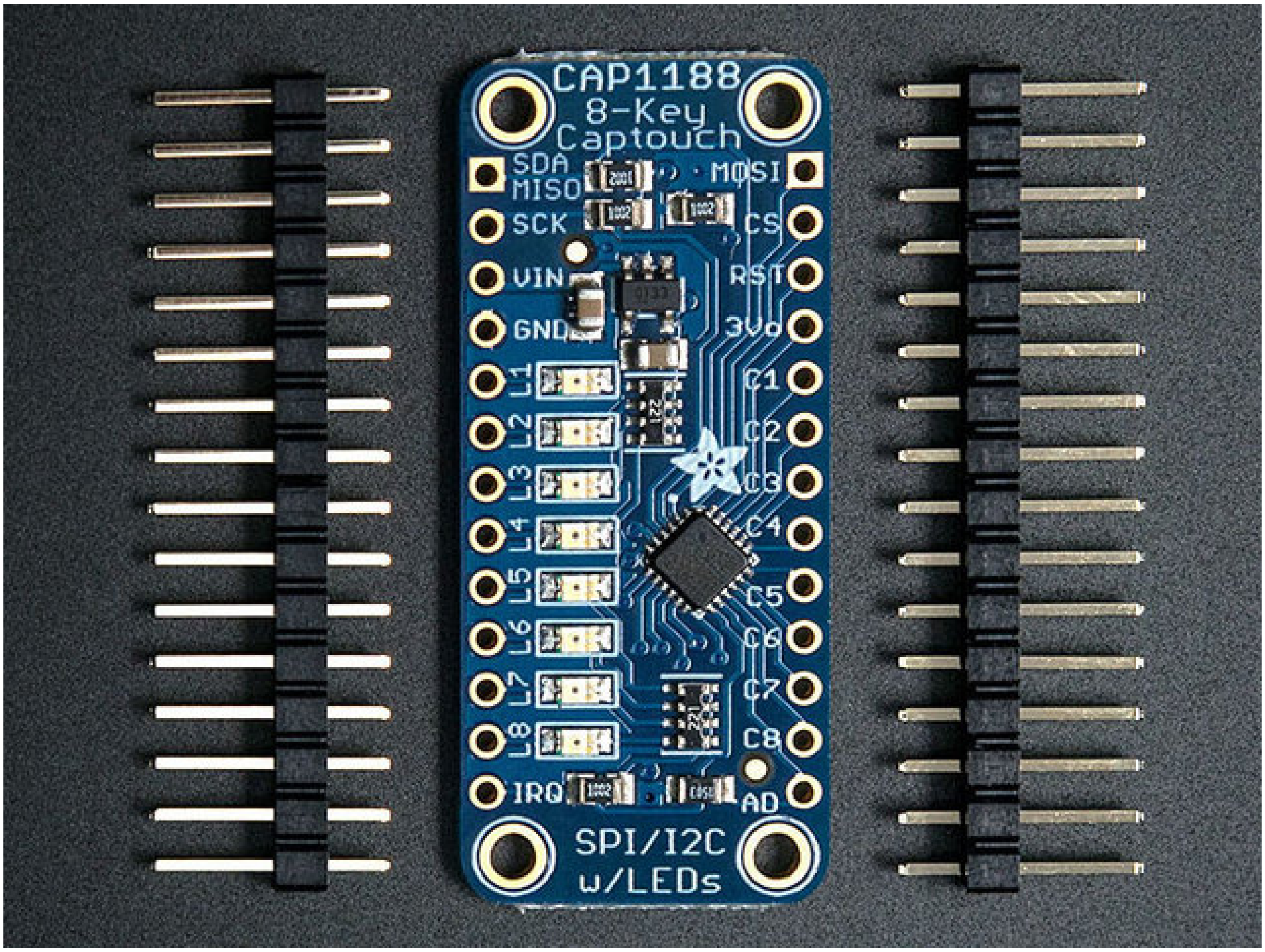
The CAP1188 has support for both I2C and SPI, so it easy to use with any microcontroller. If you are using I2C, you can select one of 5 addresses, for a total of 40 capacitive touch pads on one I2C 2-wire bus. Using this chip is a lot easier than doing the capacitive sensing with analog inputs: it handles all the filtering for you and can be configured for more/less sensitivity.

Comes with a fully assembled board, and a stick of 0.1" header so you can plug it into a breadboard. For contacts, we suggest using copper foil, then solder a wire that connects from the foil pad to the breakout.

Getting started is a breeze with our Arduino library and tutorial. You'll be up and running in a few minutes, and if you are using another microcontroller, its easy to port our code.

# Pinouts

The CAP1188 has a lot going on, so much so that we had to make the breakout double-sided! It fits nicely into a breadboard and has the sensors all in a row on one side and if you're using plain i2c you can connect to to the left side only




Here's all the pins and what they do!

## Power pins

**VIN** and **GND** are power in pins, you can use 3-5VDC so its great for any kind of microcontrollers. There's an on-board 3V regulator as well, the output is available on the **3Vo** pin (you can snag up to 150mA)

## I2C interface pins

For I2C, connect to the **SCK** (i2c clock a.k.a SCL) and **SDA** (i2c data) pins. These are 5V safe so you can use them with 3V or 5V logic

## SPI inteface pins

If you want to use SPI instead, you'll be using the SCK, MOSI, MISO and CS pins

## Other interfacing pins

The **AD** pin is used to select SPI or I2C interface, and if I2C what address to use. See the wiring page for more details

The **IRQ** pin goes *low* when a pin is touched. We don't use it on our code examples, but if you want to have an interrupt pin used, connect it to this IRQ pin and use active-low triggering.

The **RST** pin is used to reset the chip, either in I2C or SPI mode. It's optional but using it will make the system more reliable so we suggest it.

## Sensor input pins

This is the part you touch - there are 8 individual capacitive sensor pins, called **C1** thru **C8**. On restart the system recalibrates them so don't touch these when powering up!

## Sensor output pins

The **L1** thru **L8** pins are the LED driver/sensor output pins. The indicators are really useful for debugging your touch sensor system, but you can also use the indicator output pins for triggering some other electronics. Each **L** pin corresponds to the matching **C** sensor input. These pins are 3V normally, and drop to 0V when triggered.

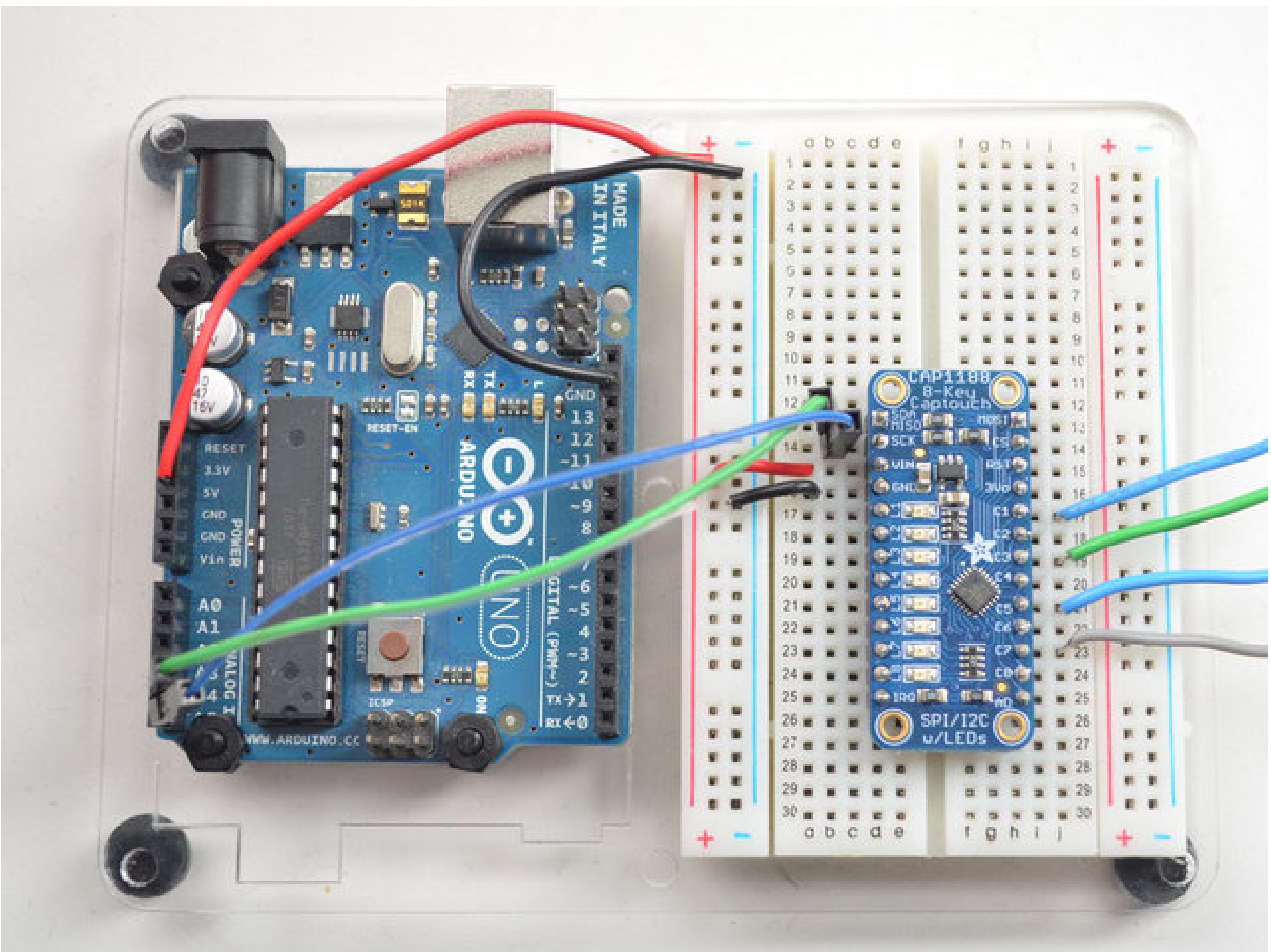
# Wiring

## Wiring for use with I2C

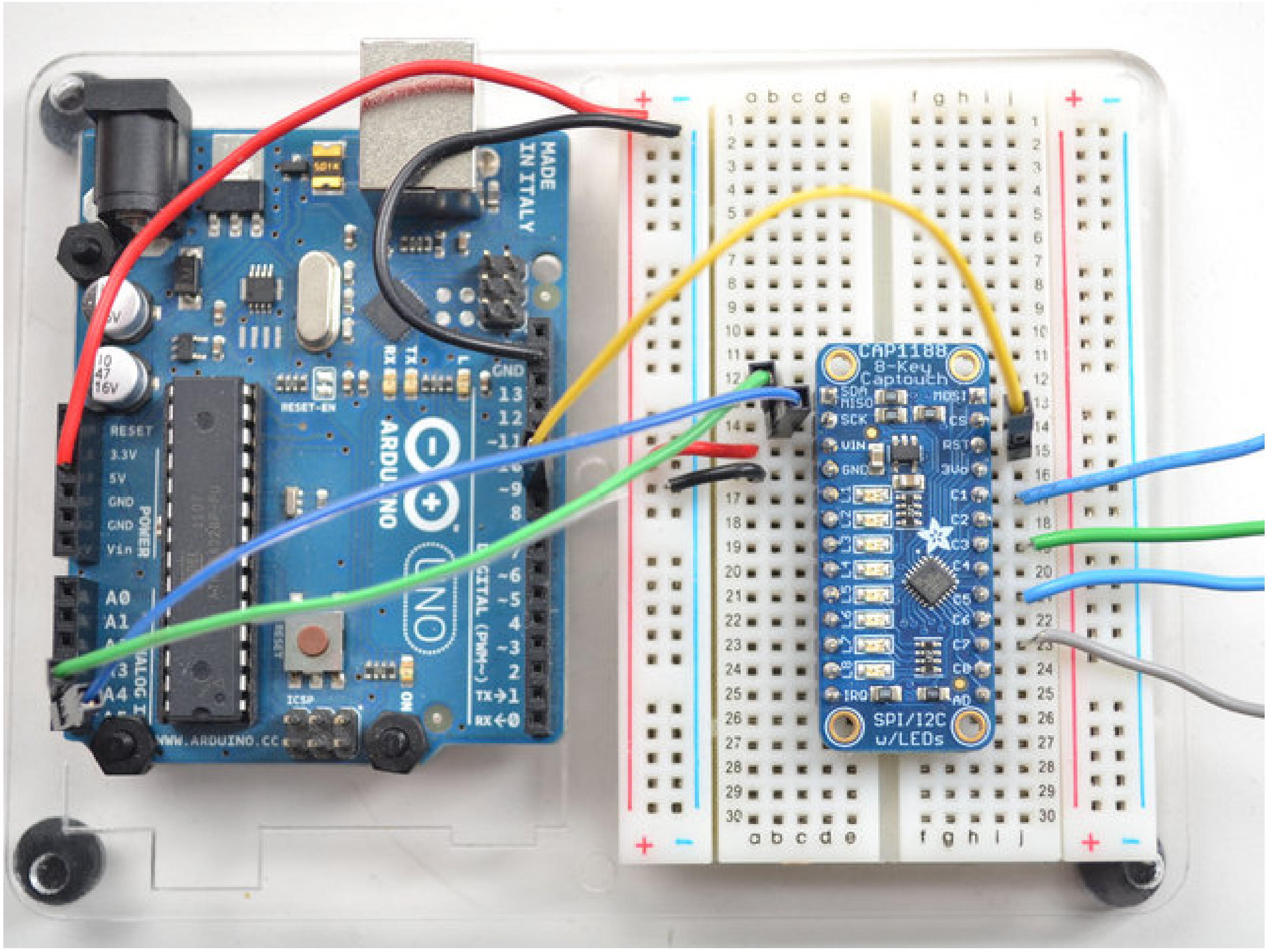
Chances are you'll use this board with the I2C interface pins. I2C is handy because you can have multiple sensors all connected on two I2C pins and share them nicely as long as each one has a unique address. I2C isn't particularly fast but that's fine for a sensor like this one (it's not good for video or audio type data)

I2C is really fast to get started, so we suggest that. Connect the **VIN** pin to 5V and **GND** pin to ground. Then connect the **SDA** pin to your I2C SDA/data line and **SCL** pin to your I2C SCL/clock line.

- On UNO/Duemilanove/etc, SDA == Analog 4, SCL == Analog 5
- On Leonardo/Micro, SDA == Digital 2, SCL == Digital 3
- On Mega/ADK/Due, SDA == Digital 20, SCL == Digital 21

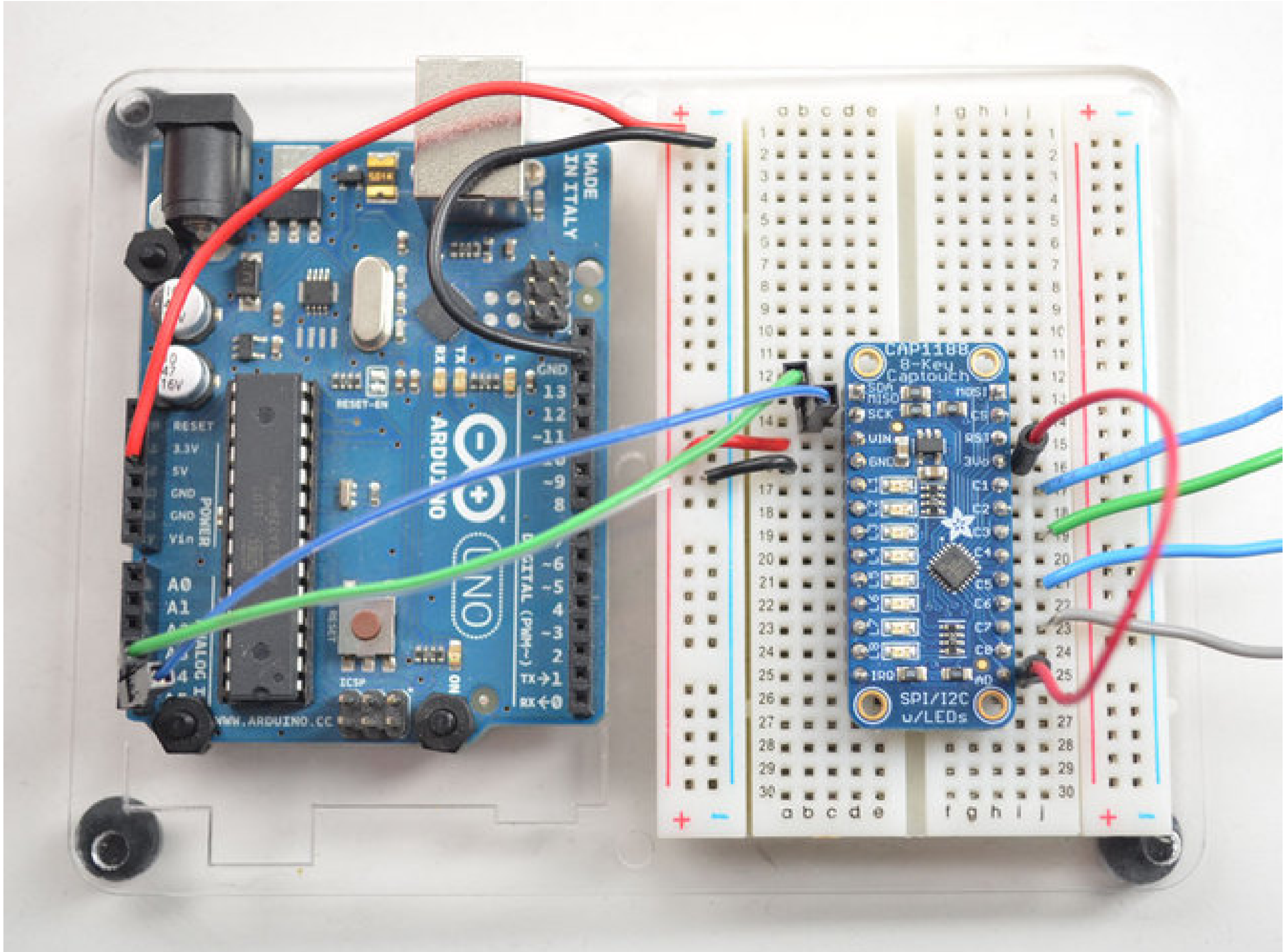



The reset pin is not required for use, but if you can spare a pin, it will make the system a little more rugged - the Arduino will do a hard reset of the CAP1188 on startup, connect **RST** to any digital I/O pin. In the library example code you can set up that pin in the CAP1188 object creation.




If you're using multiple sensors, or you just want to change the I2C address to something else, you can choose from 5 different options - **0x28**, **0x29** (default), **0x2A**, **0x2B**, **0x2C** and **0x2D**

The I2C address are selected by connecting a resistor to the **AD** pin in the lower right: different resistors set a different address. The easiest address to set is 0x28 which is just a wire from **AD** to the **3Vo** pin.



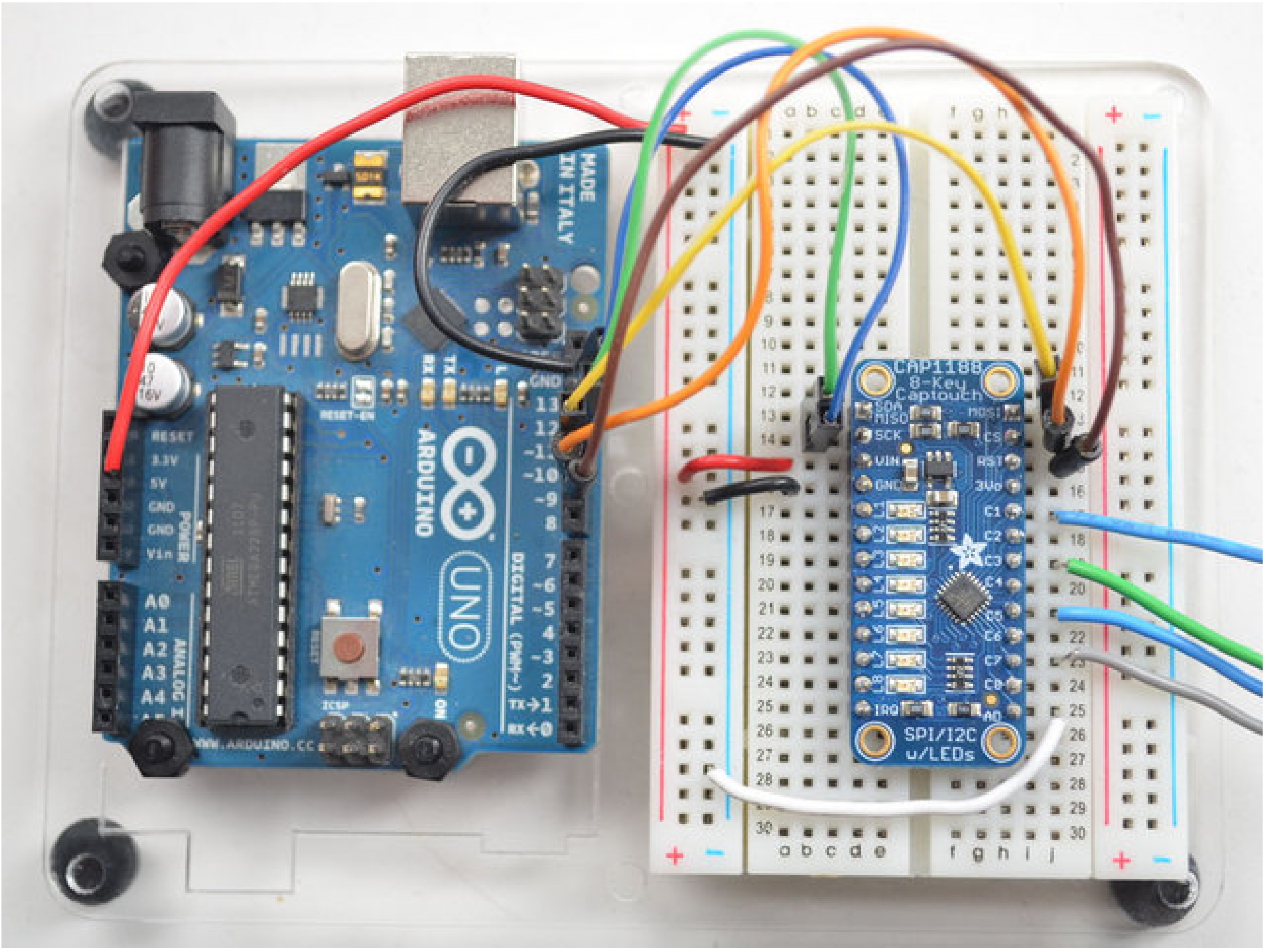
If you want to set the I2C address to a different value you'll need to connect a resistor from **AD** to ground instead of a wire to 3V. The datasheet talks about different resistor values in table 3.1 - but since the breakout board already has a 150K resistor on it, you'll need to use different values so that the parallel resistance comes out right. Here's the values you should use:

- Wire connecting **AD** to **3V** -> I2C address **0x28**
- No resistor or wire attached to **AD** -> I2C address **0x29**
- 600K resistor from **AD** to ground -> I2C address **0x2A**
- 300K resistor from **AD** to ground -> I2C address **0x2B**
- 180K resistor from **AD** to ground -> I2C address **0x2C**

## Wiring for use with SPI

You can put the CAP1188 in SPI mode by powering it up with **AD** connected to ground. Then the 4 SPI pins are used to communicate instead of I2C. SPI may be preferrable for your project if you have an I2C address collision (which is unlikely given you can choose 5 addresses!) or say if you have an UNO and you want to use the I2C pins for analog input instead of I2C, or if you are porting to a microcontroller that does not have hardware I2C.

Either way, SPI is there for you. Connect **Vin** to 5V and **GND** to ground, then tie a wire from **AD** to ground as well. Now connect the **SCK**, **MISO**, **MOSI, CS** and **RST** pins to your microcontroller. If using an Arduino you can use either the hardware SPI pins which are fixed for each Arduino, or go with software SPI where you select any 5 digital I/O pins



# Using with Arduino

Its super easy to use this sensor board with an Arduino thanks to the great Adafruit library. Once you've installed the library you can connect the sensor board via I2C or SPI to your Arduino, it will work with any kind or flavor. If you're using a different kind of microcontroller, the library is a good reference to help you port the code over.

# Download the library

First up, we'll download the Arduino library from github. The source code is in a repository there, but to make it easy, we suggest just clicking the button below to get the latest version in a Zip file.

Download CAP1188 Library

http://adafru.it/d5g

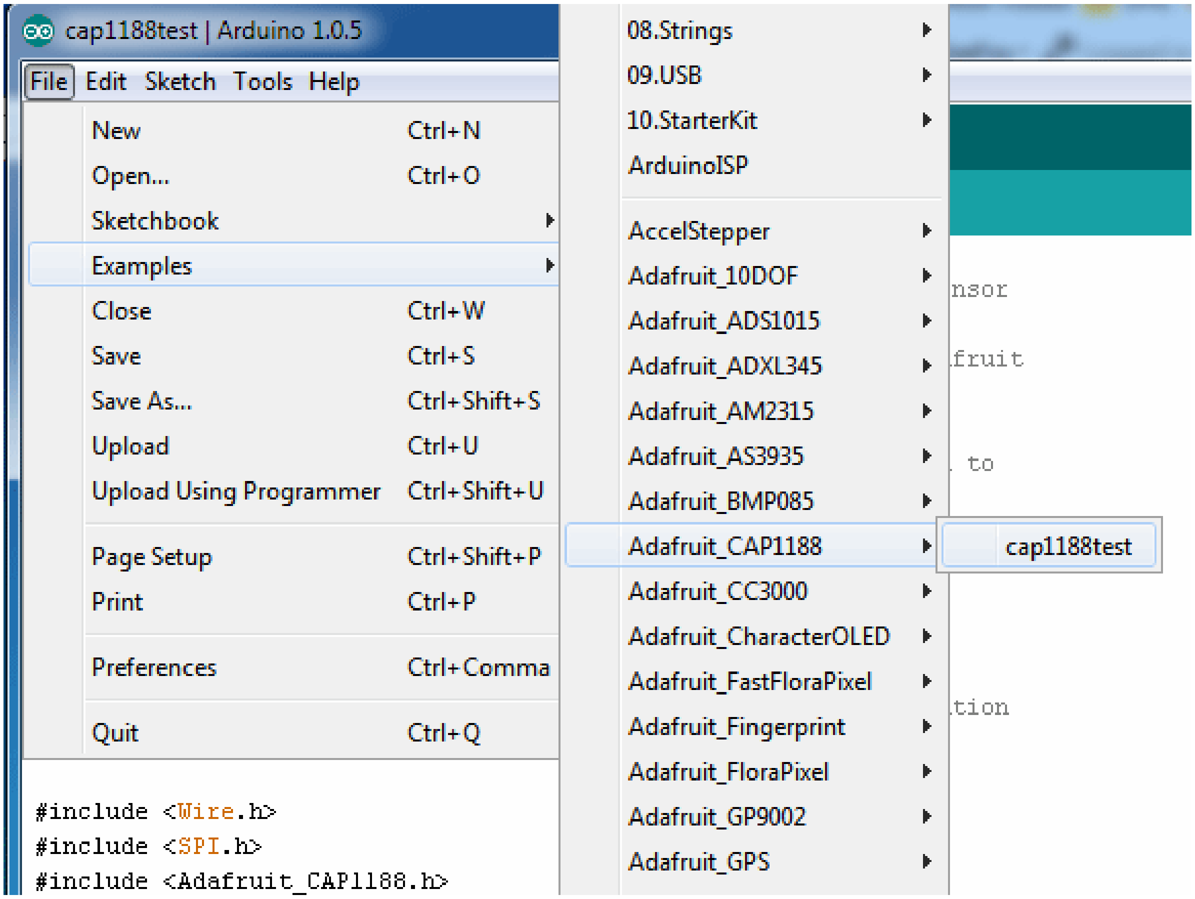
Rename the uncompressed folder **Adafruit_CAP1188** and check that the **Adafruit_CAP1188** folder contains **Adafruit_CAP1188.cpp** and **Adafruit_CAP1188.h**

Place the **Adafruit_CAP1188** library folder your **arduinosketchfolder/libraries/** folder.
You may need to create the **libraries** subfolder if its your first library. Restart the IDE.

We also have a great tutorial on Arduino library installation at:
http://learn.adafruit.com/adafruit-all-about-arduino-libraries-install-use (http://adafru.it/aYM)

# Run Test Sketch

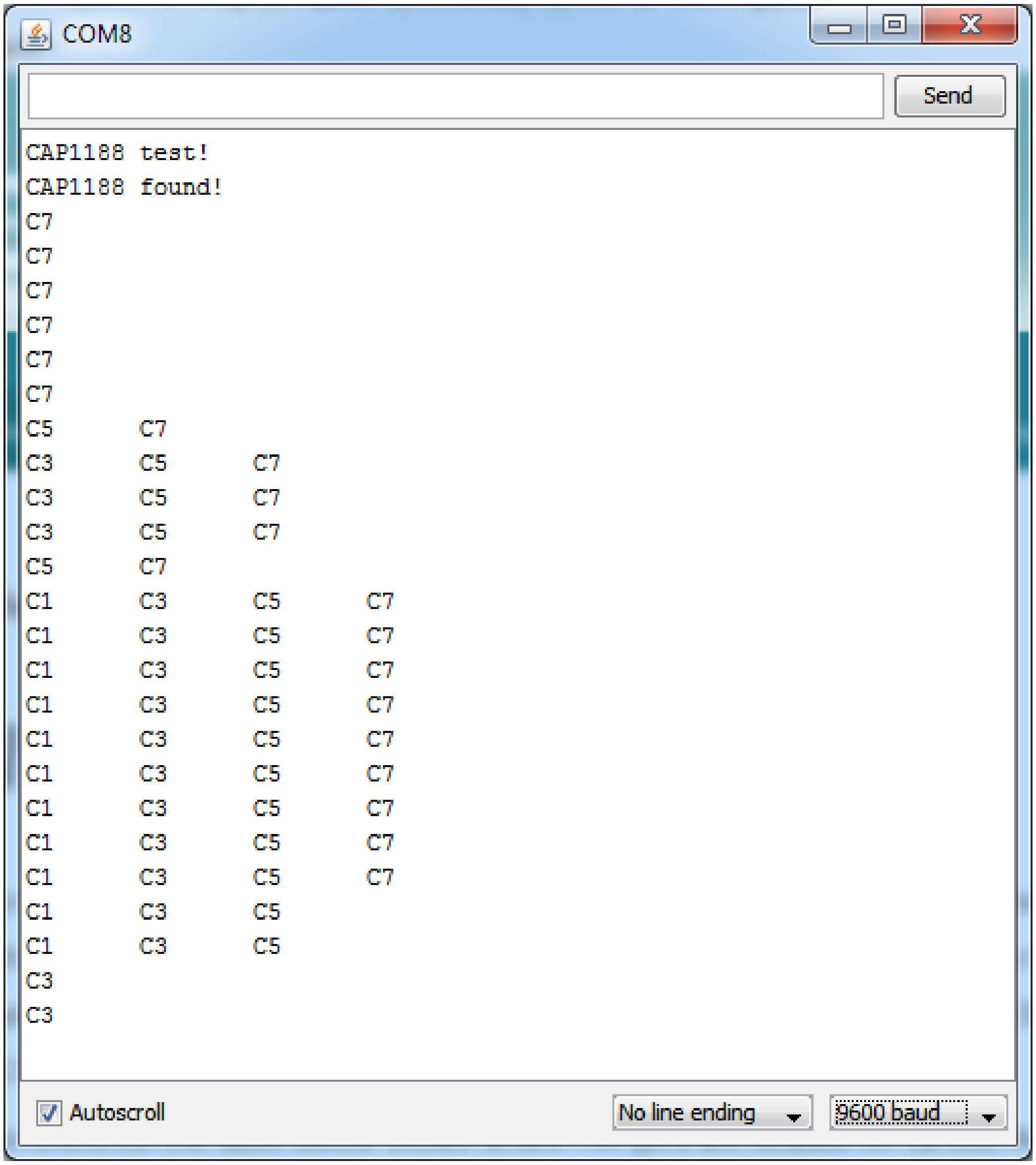
After you've restarted, you should be able to load up the **File->Examples->Adafruit_CAP1188->cap1188test** sketch

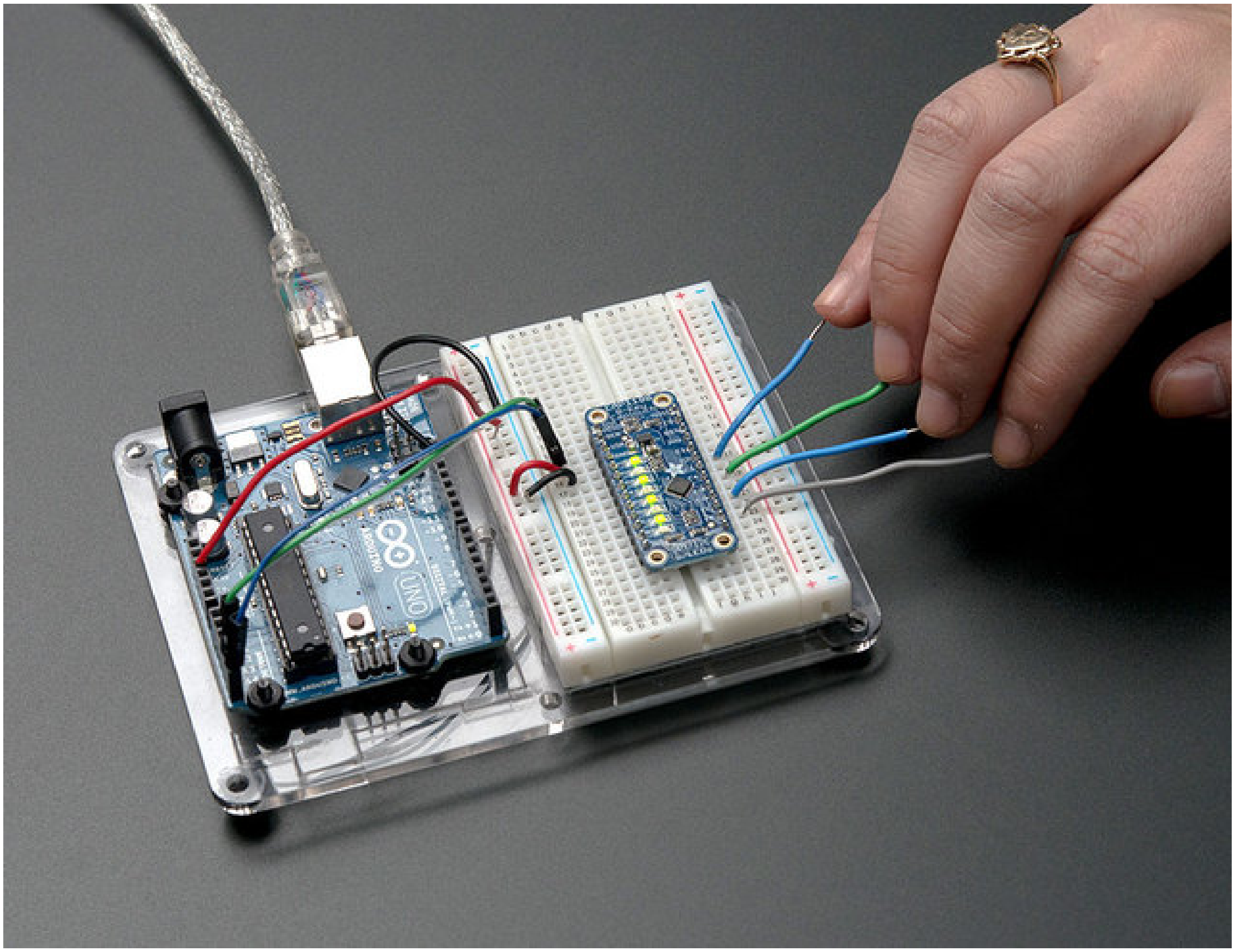



Wire up the sensor as shown in the Wiring section, for I2C. Connect the **GND** pin to ground, **VIN** pin to 5V and connect **SDA** to your Arduino's **SDA** pin, **SCL** to **SCL** pin

- On UNO/Duemilanove/etc, SDA == Analog 4, SCL == Analog 5
- On Leonardo/Micro, SDA == Digital 2, SCL == Digital 3
- On Mega/ADK/Due, SDA == Digital 20, SCL == Digital 21

Upload the sketch and open up the serial console at 9600 baud. You should see that the CAP1188 is found (good!) and then you can touch the C1 thru C8 pads with your fingers to see the touch sensor go off. When you touch a pin, you'll also see the matching LED light up.

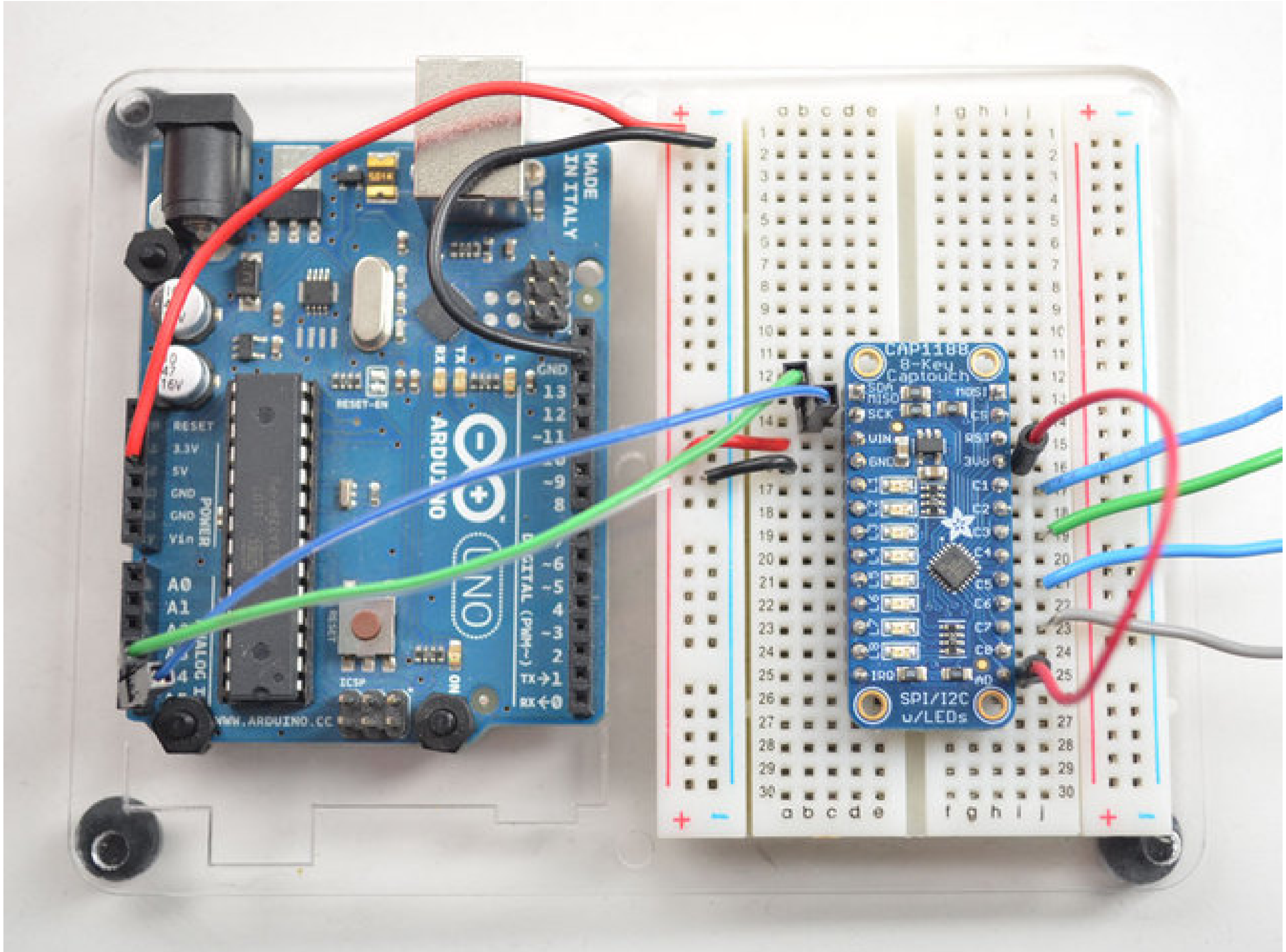

COM8
Send
CAP1188 test!
CAP1188 found!
C7
C7
C7
C7
C7
C7
C5 C7
C3 C5 C7
C3 C5 C7
C3 C5 C7
C5 C7
C1 C3 C5 C7
C1 C3 C5 C7
C1 C3 C5 C7
C1 C3 C5 C7
C1 C3 C5 C7
C1 C3 C5 C7
C1 C3 C5 C7
C1 C3 C5 C7
C1 C3 C5 C7
C1 C3 C5
C1 C3 C5
C3
C3
Autoscroll
No line ending
9600 baud



## I2C with a different address

If you're using multiple sensors, or you just want to change the I2C address to something else, you can choose from 5 different options - **0x28**, **0x29** (default), **0x2A**, **0x2B**, **0x2C** and **0x2D**

The I2C address are selected by connecting a resistor to the **AD** pin in the lower right: different resistors set a different address. The easiest address to set is 0x28 which is just a wire from **AD** to the **3Vo** pin.




Now look in the test sketch for the lines

```
 if (!cap.begin()) {
Serial.println("CAP1188 not found");
while (1);
}
```

And change the **cap.begin()** statement to **cap.begin(0x28)** to initialize it to use the **0x28** address. Don't forget to do a full power off of the Arduino & sensor, the sensor only detects the i2c address on power up so you can't change it on the fly!

# Using with SPI

You can also use SPI if you wish. The library supports both hardware SPI (using the 'hardware SPI' port on your arduino) or software/bit-bang SPI, where you can define the pins. In general, these sensors are not very fast so I2C is a good way to interface them but if you wish, SPI is there for you too! For example, if you want to have more than 5 of these connected to one board, that is possible to do with SPI, but the I2C interface on this chip doesn't support that many shared I2C addresses.

To enable SPI, be sure to wire for SPI as shown in the Wiring section, and do a power reset of your board so that the chip 'wakes up' in SPI mode

If you're using hardware SPI, check the SPI page for what pins you need to use, sometimes they are only on the ICSP header which makes them trickier to use. (http://adafru.it/d5h)

To enable the hardware SPI interface, create the Adafruit_CAP1188 object with

> Adafruit_CAP1188 cap = Adafruit_CAP1188(CAP1188_CS, CAP1188_RESET);

If you are using software SPI, you can use *any* pins that are not already used for some purpose. In that case, create the object with

> Adafruit_CAP1188 cap = Adafruit_CAP1188(CAP1188_CLK, CAP1188_MISO, CAP1188_MOSI, CAP1188_CS, CAP1188_RESET);

## Using the external IRQ Interrupt

Arduino has some basic ability to attach to pin interrupts, here's an example from Nobody123 of connecting the IRQ pin from the CAP1188 to digital pin #3 on an Uno (Interrupt #1) for tracking touches asynchronously

```
/***************************************************
  This is a library for the CAP1188 I2C/SPI 8-chan Capacitive Sensor

  Designed specifically to work with the CAP1188 sensor from Adafruit
  ----> https://www.adafruit.com/products/1602

  These sensors use I2C/SPI to communicate, 2+ pins are required to
  interface
  Adafruit invests time and resources providing this open source code,
  please support Adafruit and open-source hardware by purchasing
  products from Adafruit!

  Written by Limor Fried/Ladyada for Adafruit Industries.
  BSD license, all text above must be included in any redistribution
 ****************************************************/

#include <Wire.h>
#include <SPI.h>
#include <Adafruit_CAP1188.h>

// Reset Pin is used for I2C or SPI
#define CAP1188_RESET  4
```

```
// CS pin is used for software or hardware SPI
#define CAP1188_CS  10

// These are defined for software SPI, for hardware SPI, check your
// board's SPI pins in the Arduino documentation
#define CAP1188_MOSI  11
#define CAP1188_MISO  12
#define CAP1188_CLK  13

volatile byte interrupt = 0;

// For I2C, connect SDA to your Arduino's SDA pin, SCL to SCL pin
// On UNO/Duemilanove/etc, SDA == Analog 4, SCL == Analog 5
// On Leonardo/Micro, SDA == Digital 2, SCL == Digital 3
// On Mega/ADK/Due, SDA == Digital 20, SCL == Digital 21

// Use I2C, no reset pin!
// Adafruit_CAP1188 cap = Adafruit_CAP1188();

// Or...Use I2C, with reset pin
//Adafruit_CAP1188 cap = Adafruit_CAP1188(CAP1188_RESET);

// Or... Hardware SPI, CS pin & reset pin
Adafruit_CAP1188 cap = Adafruit_CAP1188(CAP1188_CS, CAP1188_RESET);

// Or.. Software SPI: clock, miso, mosi, cs, reset
//Adafruit_CAP1188 cap = Adafruit_CAP1188(CAP1188_CLK, CAP1188_MISO, CAP1188_MOSI, CAP1188

void setup() {
  Serial.begin(9600);
  Serial.println("CAP1188 test!");
  pinMode(3,INPUT);
  // Raise SPI slave select (SS) pins
  // Communication begins when you drop the individual select signals to LOW
  digitalWrite(10,HIGH);


  // Initialize the sensor, if using i2c you can pass in the i2c address
  // if (!cap.begin(0x28)) {
  if (!cap.begin()) {
    Serial.println("CAP1188 not found");
    while (1);
  }
  Serial.println("CAP1188 found!");
  pinMode(3, INPUT);
```

```
  // Turn off multitouch so only one button pressed at a time
  cap.writeRegister(0x2A, 0x80);  // 0x2A default 0x80 use 0x41  — Set multiple touches back to off
  cap.writeRegister(0x41, 0x39);  // 0x41 default 0x39 use 0x41  — Set "speed up" setting back to off
  cap.writeRegister(0x72, 0x00);  // 0x72 default 0x00  — Sets LED links back to off (default)
  cap.writeRegister(0x44, 0x41);  // 0x44 default 0x40 use 0x41  — Set interrupt on press but not relea
  cap.writeRegister(0x28, 0x00);  // 0x28 default 0xFF use 0x00  — Turn off interrupt repeat on button
  EIFR = 1; // clear flag for interrupt 1
  attachInterrupt(1, routine_Interrupt_CAP1188, FALLING);
}

void loop() {

  // Serial.println(digitalRead(3));
  uint8_t touched = cap.touched();

  if (touched == 0) {
    // No touch detected
    // return;
  }

  for (uint8_t i=0; i<8; i++) {
    if (touched & (1 << i)) {
      Serial.print("C"); Serial.print(i+1); Serial.print("\t");
    }
  }
  Serial.println();
  delay(50);
  Serial.print("Interrupt: ");  Serial.println(interrupt);
}

void routine_Interrupt_CAP1188()  {
  ++interrupt;
}
```

# Downloads

The CAP1188 Datasheet has lots of details about this chip (http://adafru.it/d5i)