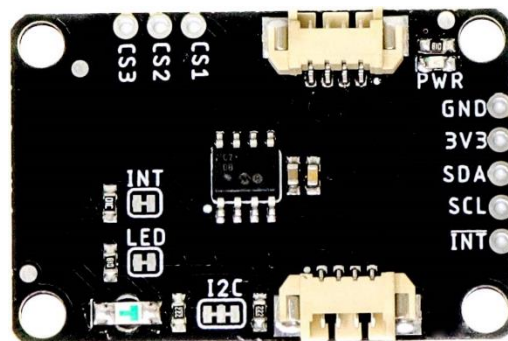




SmartElex Capacitive Touch Slider - CAP1203



Introduction

Do you want to replace a slider or a button on your art project or science experiment with a more interesting interface? The Capacitive Touch Slider - CAP1203 is a quick and easy way to add capacitive touch to your next project. With the board's built in touch pads, you can immediately start playing with the touch capabilities as three unique touch inputs or as a slider. You can also enable a touch input to act as a power button, customize the sensitivity for your own touch pads, and play with the interrupt alert LED. We've added breakout pins for the capacitive touch inputs, so you can connect to your own touch pads.

Hardware Overview

Note: If you want to do anything outside of what is covered in this tutorial or the example code, please refer to the CAP1203 datasheet for exact details on the sensor functionality.

Power

You can provide **3.3V** through the polarized connectors on the board or through the 3V3 labeled pin on the through-hole header. The connector system is meant to use **3.3V**, so be sure you are **NOT** using another voltage when using the connectors. When you have correctly powered the board, the red power LED will turn on.

Interrupt Pin and LED

The interrupt pin is an active low output which is triggered each time a sensor pad is touched. You can connect to this pin if you want to check when an interrupt occurs. On the front side of the board, the green LED in the bottom right corner also signals when an interrupt occurs.

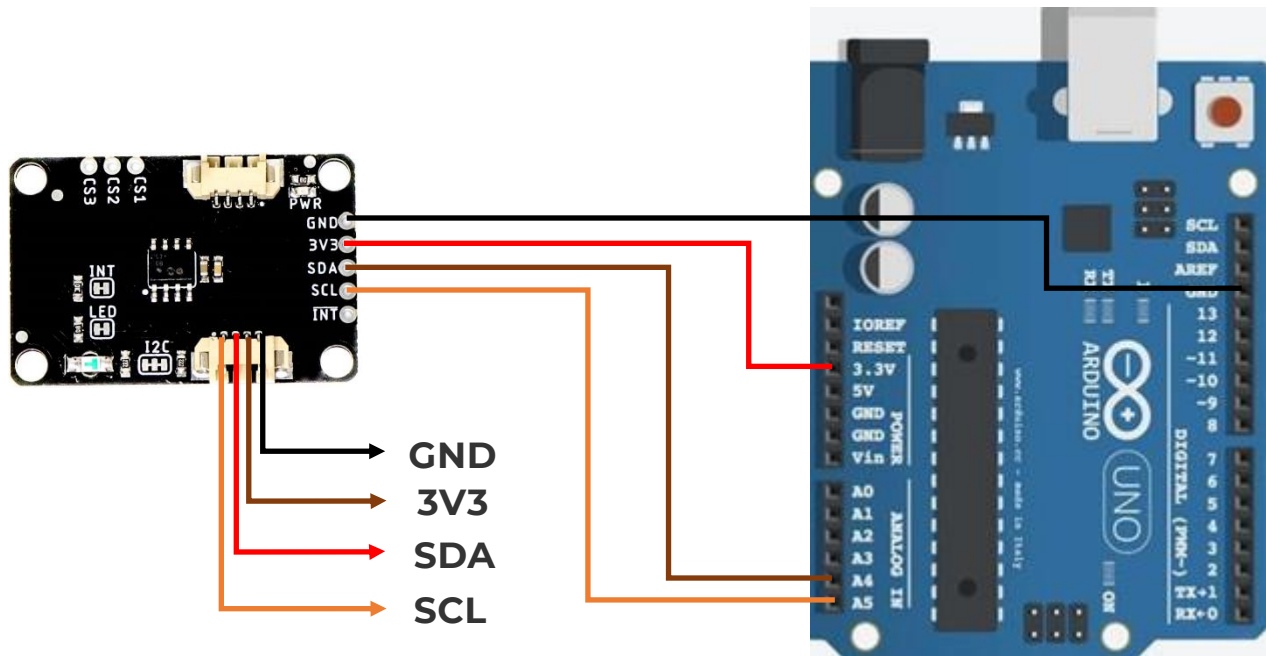
Capacitive Touch Pads and Pins

On the front of the board, there is an arrow shape which contains three separate capacitive touch pads. In order to know which pad is which, orient the board with the arrow pointing in the right direction. We will reference these capacitive touch pads as the left pad, the middle pad, and the right pad throughout the guide and in the code. We also broke out the capacitive touch sensor lines as plated through-holes on the top of the board. You can use these pins to connect to your own capacitive touch pads. The CS1 pin connects to the left pad, the CS2 pin connects to the middle pad, and the CS3 pin connects to the right pad.

Jumpers

There are 3 jumper pads on the back of the board, each labeled with its function. First, on upper left side of the board, there is a two way jumper labeled INT that connects to a 10k Ω pull-up resistor on the interrupt data line. If you cut this jumper, it will disconnect the 10k Ω pull-up resistor from the interrupt data line. Next, on the lower left side of the board, there is a two way jumper labeled LED that connects to the green interrupt LED. If you cut this jumper, it will disconnect the LED from the interrupt line, effectively disabling the green LED's functionality. Lastly, on the lower half of the board, there is a three way jumper labeled I²C that connects two 2.2k Ω pull-up resistors to the I²C data lines. If you have multiple devices on your I²C data lines, then you may consider cutting this jumper.

Wiring:



For an Arduino UNO, make the following connections:

CAP1203	Arduino
SCL	A5(SCL)
SDA	A4(SDA)
3V3	3.3V
GND	GND

Arduino Library

SparkFun have provided a library to help easily control and configure your Capacitive Touch Slider. You can download the library by searching '**SparkFun Qwiic Capacitive Touch Slider**' in the Arduino library manager. Some of the features include reading

when a pad is touched, detecting right and left swipes, enabling a power button, and setting the sensitivity for your own touch pads.

Library Functions

The Arduino library is commented and the functions should be self explanatory. However, below is a detailed list of the available library functions.

Initialization Settings: We use these to set-up and initialize the board.

`.begin()` - Initialize the sensor, returns true if correctly set-up

`.isConnected()` - Checks I²C connection, returns true if correctly connected

Sensitivity Settings: These functions allow you to change the sensitivity settings for your touch pads. Note, the default sensitivity is set to `SENSITIVITY_2X` in the `.begin()` function for the SparkFun Capacitive Touch Slider board. We only recommend changing the sensitivity setting if you are breaking out your own touch pads.

`.setSensitivity(sensitivity)` - Sets the sensitivity multiplier for the touch padsPossible sensitivity argument values:

`SENSITIVITY_128X` - Most sensitive

`SENSITIVITY_64X`

`SENSITIVITY_32X`

`SENSITIVITY_16X`

`SENSITIVITY_8X`

`SENSITIVITY_4X`

`SENSITIVITY_2X`

`SENSITIVITY_1X` - Least sensitive

`.getSensitivity()` - Returns the sensitivity multiplier for the current sensitivity settings, returns as an integer value

Power Button Settings: These functions allow you to enable or disable the power button on a specific pad. The power button requires a longer touch before registering a touch has been detected on the designated pad. Note, when the power button is enabled, the designated pad will only act as the power button.

`.setPowerButtonPad(pad)` - Sets a button to act as a power button, returns true if correctly setPossible pad argument values:

`PAD_LEFT` - Left pad and CS1 breakout pin

`PAD_MIDDLE` - Middle pad and CS2 breakout pin

`PAD_RIGHT` - Right pad and CS3 breakout pin

`.getPowerButtonPad()` - Returns which pad is currently set to be the power button, returns as an integer value
Return values:

- 1 - Left pad and CS1 breakout pin
- 2 - Middle pad and CS2 breakout pin
- 3 - Right pad and CS3 breakout pin

`.setPowerButtonTime(time)` - Sets the power button touch time, returns true if correctly set
Possible time argument values:

- `PWR_TIME_280_MS`
- `PWR_TIME_560_MS`
- `PWR_TIME_1120_MS`
- `PWR_TIME_2240_MS`

`.getPowerButtonTime()` - Returns length of time (ms) power button must indicate a touch, returns as an integer value

`.setPowerButtonEnabled()` - Enables power button functionality

`.setPowerButtonDisabled()` - Disables power button functionality

`isPowerButtonEnabled()` - Returns true if power button functionality is currently enabled

Interrupt Settings: These functions allow you to control if the interrupt pin is enabled. When enabled, the green interrupt LED will turn on when any pad detects a touch. Note, the interrupt is enabled as default in the `.begin()` function.

`.setInterruptDisabled()` - Disables interrupt pin

`.setInterruptEnabled()` - Enables interrupt pin

`.isInterruptEnabled()` - Returns true if interrupt pin is currently enabled

Check if a capacitive touch pad or breakout pin has been touched: These functions allow you to check when a capacitive touch pad or breakout pin has been touched.

`.isLeftTouched()` - Returns true if left pad or CS1 pin detects a touch

`.isMiddleTouched()` - Returns true if middle pad or CS2 pin detects a touch

`.isRightTouched()` - Returns true if right pad or CS3 pin detects a touch

`.isTouched()` - Returns true if any pad or breakout pin detects a touch

`.isPowerButtonTouched()` - Returns true if designated power button pad or pin held for allotted time

Check if a swipe has occurred: The direction of the swipe is relative to the arrow on the board pointing in the right direction. For the pins, a right swipe is first CS1, then CS2, and finally CS3 and a left swipe is first CS3, then CS2, and finally CS1. Note, these functions pull most available resources. For best swipe recognition, we highly recommend not implementing other functionalities when using these two functions.

`.isRightSwipePulled()` - Returns true if board detects a right swipe

`.isLeftSwipePulled()` - Returns true if board detects a left swipe

Example Code: Basic Reading

The *Example01_BasicReading.ino* sketch works with the basic functionality of the Capacitive Touch Slider and streams which pad detects a touch. Open up the example to follow along.

First, we include the CAP1203 library and initialize an instance of the sensor. Then, in the set-up function, we check to make sure the sensor is correctly connected.

Now that the sensor is set-up, we can start checking which pads have been touched in the main loop using the `.isLeftTouched()`, `.isMiddleTouched()`, and `.isRightTouched()` functions.

```
#include <Wire.h>
#include "SparkFun_CAP1203.h" // Click here to get the library:
http://librarymanager/All#SparkFun_CAP1203

CAP1203 sensor; // Initialize sensor

void setup()
{
  Wire.begin();          // Join I2C bus
  Serial.begin(9600);    // Start serial for output

  // Setup sensor
  if (sensor.begin() == false)
  {
    Serial.println("Not connected. Please check connections and read the hookup
guide.");
    while (1)
      ;
  }
  else
  {
    Serial.println("Connected!");
  }
}

void loop()
{
  if (sensor.isLeftTouched() == true)
```

```
{  
  Serial.println("Left");  
}  
  
if (sensor.isMiddleTouched() == true)  
{  
  Serial.println("Middle");  
}  
  
if (sensor.isRightTouched() == true)  
{  
  Serial.println("Right");  
}  
}
```

//////////////////////////////////////END//////////////////////////////////////

Once you have uploaded the code, you can open your serial terminal to see when a pad has been touched. Make sure the baud rate in your serial terminal is set to **9600 baud**, otherwise you won't see the correct output.