

# Touch sensor - Using an MPR121

## What is the MPR121?

The MPR121 is a tiny microchip formerly manufactured by NXP, now under Resurgent Semiconductor, it is a tiny surface mount device that provides 12 capacitive touch electrodes through an I2C interface. We are using Adafruit MPR121 in this tutorial.

## What is capacitive touch?

Capacitive touch is the technology used on modern touch sensitive devices such as phone and tablet screens, trackpads and computer mice like Apple's Magic Mouse.

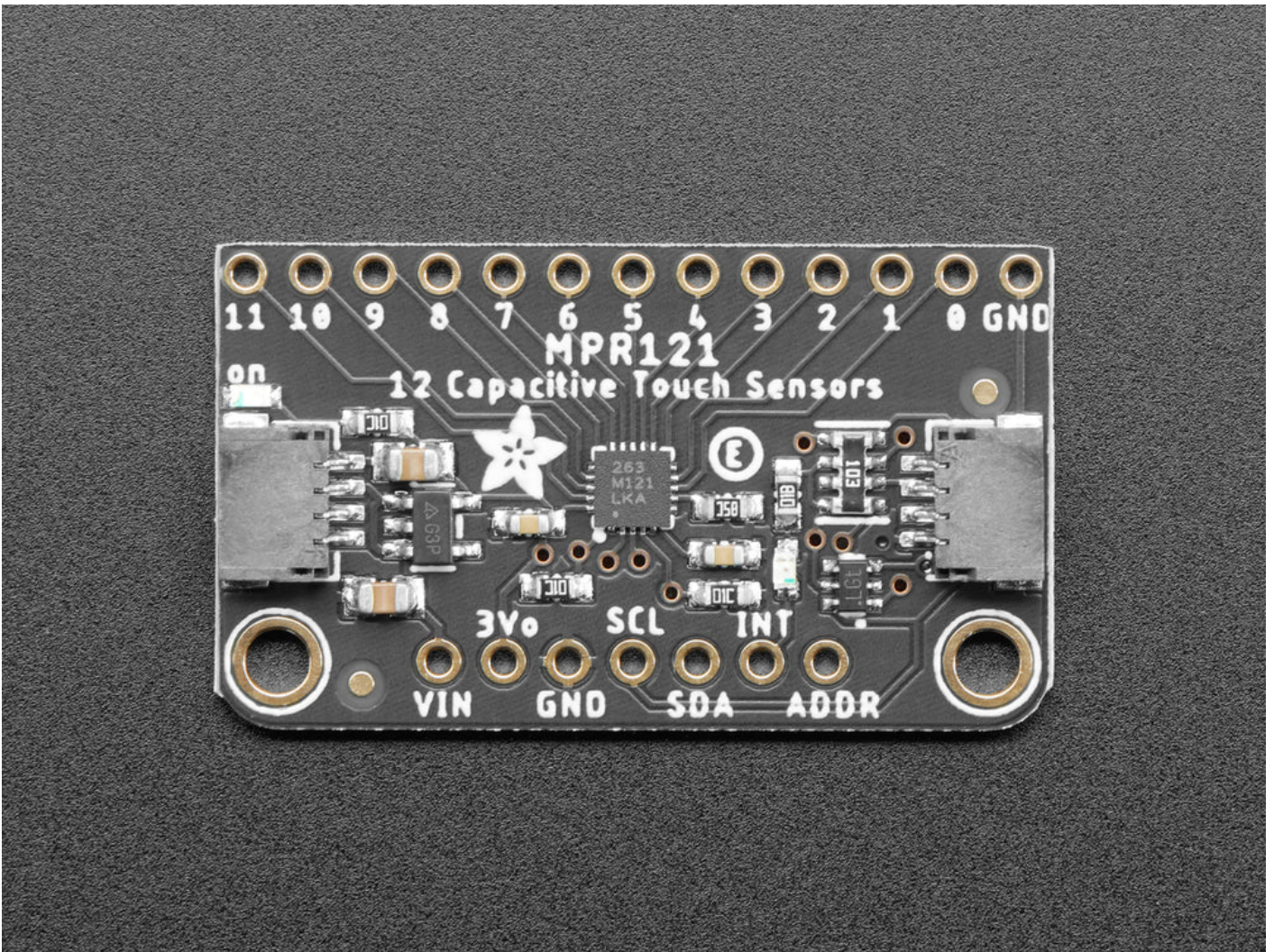
Capacitive touch takes advantage of the human body being electrically conductive, this is why using a pen on a smart phone doesn't work and styluses for these devices are made of metal so they will make an electrical connection.

Capacitive touch is relatively complex to understand but fundamentally the sensor can detect when you are in proximity or actually touching the electrode, or any conductive part between the chip and the end of the wire.

If you extend a wire from the electrode, even if it is shielded with plastic, it will probably detect you touching the wire just the same as the exposed metal part.

## Adafruit MPR121

- Power input: 3.3V to 5V
- Address: 0x5A, it is also relatively easy to configure the address to 0x5B, 0x5C or 0x5D allowing up to 48 electrodes (12 x 4).



## Library

Adafruit, Sparkfun, Seeedstudio and Bare Conductive all provide Arduino libraries, however by far the best which includes tools for visualising the data is the **Bare Conductive Touch Board library** and **grapher**. All libraries should all work interchangeably as long as you get the correct address and IRQ pin.

The key thing to remember when using examples from the Bare Conductive library is that they use the address `0x5C` rather than the default `0x5A` address used on both the Sparkfun and Adafruit boards, also ensure that you use the correct IRQ pin based on the example you are using.

We have a tutorial on **how to install a library** here.

## Wiring

Wiring is pretty simple, it's an I2C component so it's relatively standard.

1. VIN -> 5V
2. GND -> GND

3. SCL -> SCL / A5
4. SDA -> SDA / A4

#### *Older Arduino boards*

Some older Arduino boards do not have SDA and SCL pins as shown in the diagrams, in this case you'll need to look it up on the boards documentation, however most Arduino boards used A4 as SDA and A5 as SCL.

## Getting started

After installing the library and wiring the board, go ahead and use the examples in the File > Examples menu in Arduino, the Simple Touch example is particularly good as it's simple to check it's working.

## Basic Example

This is a basic example of using the `MPR121` with the Bare Conductive MPR121 library to show which touch point is touched.

```
#include <MPR121.h>
#include <Wire.h>

#define numElectrodes 12

void setup() {
  Serial.begin(9600);
  Wire.begin();

  // Setup MPR121
  MPR121.begin(0x5A);
  MPR121.setInterruptPin(4);
  MPR121.setTouchThreshold(20); //0 (most sensitive) to 255 (least sensitive)
  MPR121.setReleaseThreshold(10); //half of TouchThreshold
}

void loop() {
  // Update touch data
  MPR121.updateTouchData();

  // Loop through all electrodes
```

```

for (int i = 0; i < numElectrodes; i++) {
  if (MPR121.getTouchData(i)) {
    Serial.print("Touched electrode: ");
    Serial.println(i);
  }
}

delay(100); // Small delay to reduce serial output frequency
}

```

## Getting proximity / raw data

This example code is only reading from **electrode 11**.

```

#include <MPR121.h>
#include <Wire.h>

void setup() {
  Serial.begin(9600);
  Wire.begin();

  // Initialize MPR121
  MPR121.begin(0x5A); // Default I2C address
  MPR121.setTouchThreshold(40);
  MPR121.setReleaseThreshold(20);
}

void loop() {
  // Update sensor readings
  MPR121.updateTouchData();
  MPR121.updateFilteredData();

  // Print filtered data for electrode 11
  Serial.print("FilteredData[11]: ");
  Serial.println(MPR121.getFilteredData(11));

  delay(100); // Adjust for readability
}

```

# CSV Example

This is an example using the MPR121 with the Bare Conductive MPR121 library to output each electrode's touch status to the serial port as a comma separated string.

This example can be modified to work with MaxMSP easily by changing the delimiter line to a space instead of a comma:

```
#define DELIMITER " "
```

You could also modify this example to output the proximity data, instead of the touch data by modifying the `updateTouchData` line to:

```
MPR121.updateFilteredData();
```

And the `getTouchData` line to:

```
Serial.print( MPR121.getFilteredData( i ) );
```

## **Sketch: CSV Example**

---

Revision #14

Created 12 August 2016 17:55:30 by Tom Lynch

Updated 25 April 2025 14:36:10 by Joanne Leung