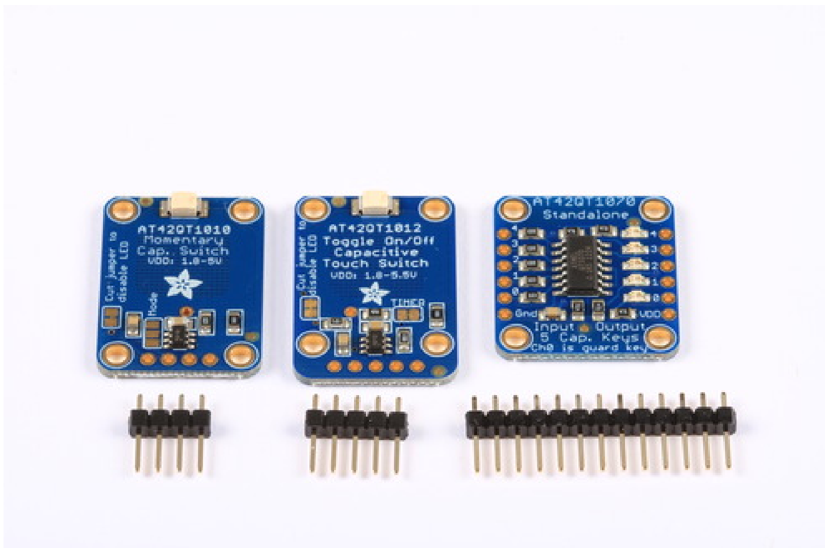
adafruit learning system

Adafruit Capacitive Touch Sensor Breakouts

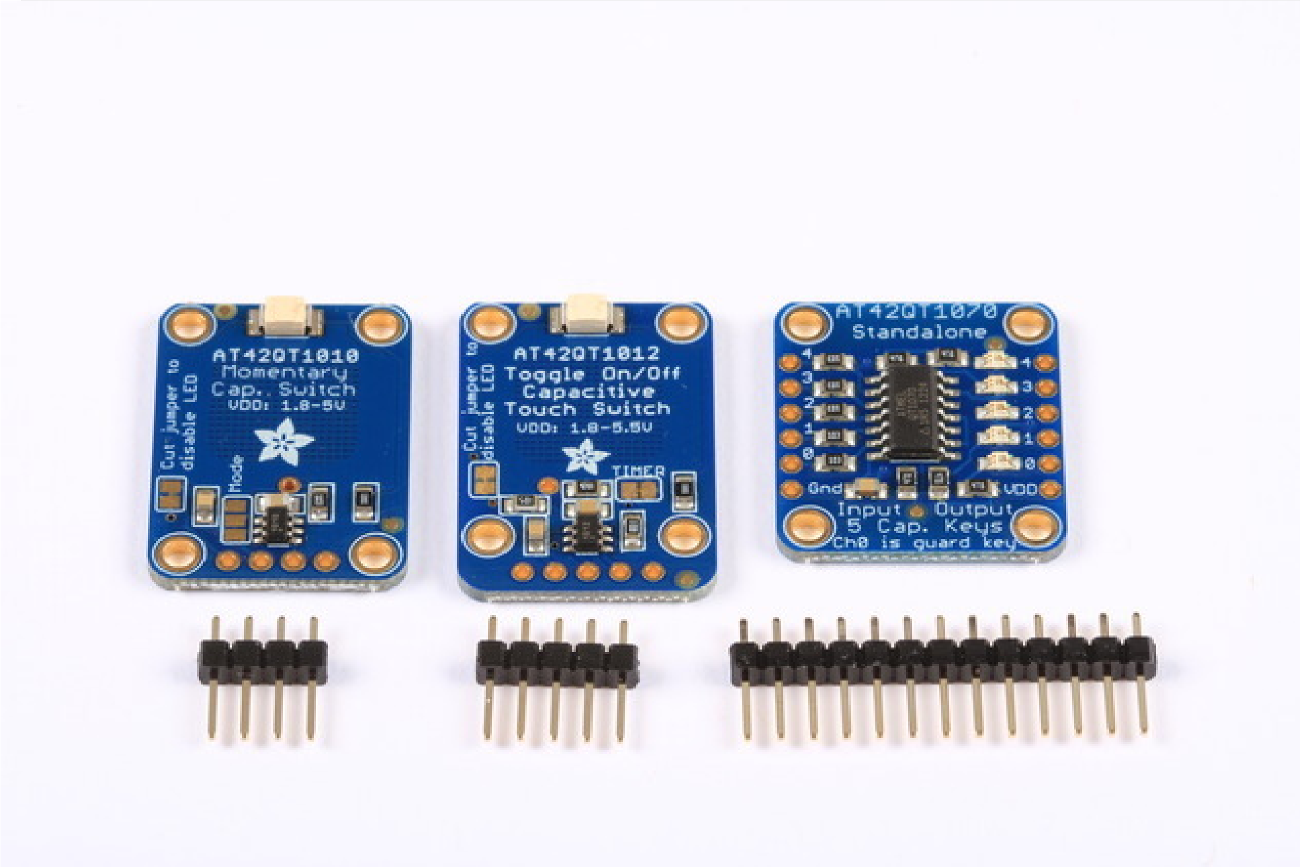
Created by Bill Earl



Last updated on 2014-12-11 11:45:16 AM EST

Guide Contents

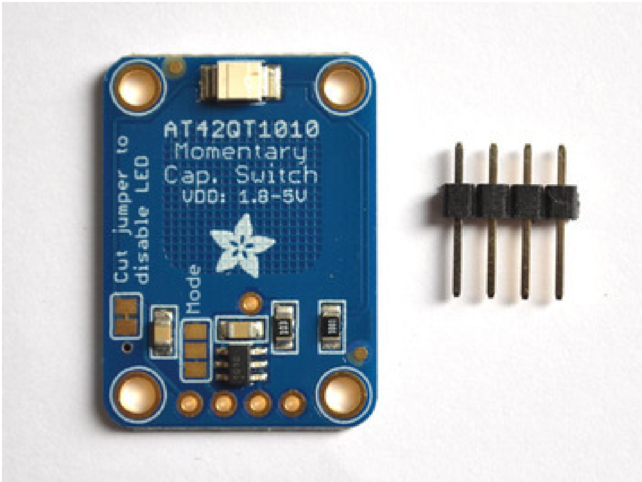
Overview



These breakout boards are a simple way to add capacitive touch to your project. Just power with 1.8 to 5.5VDC and touch the pad to activate the sensor. These touch switches interface easily to any project - with or without a microcontroller.

When a capacitive load (such as a human hand) is in close proximity to the sense-pad, the sensor detects the change in capacitance and activates the switch. Custom sense-pads can be made form nearly any conductive material and these sensors can detect touch through thin layers of non-conductive materials such as glass, plastic, fabric or even wood.

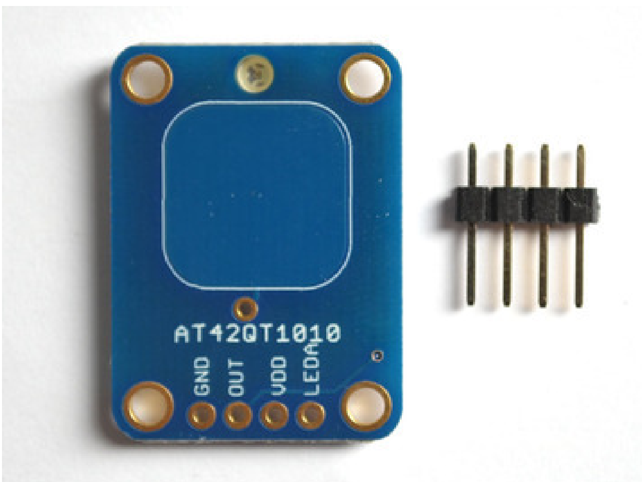
The breakouts come in three styles:

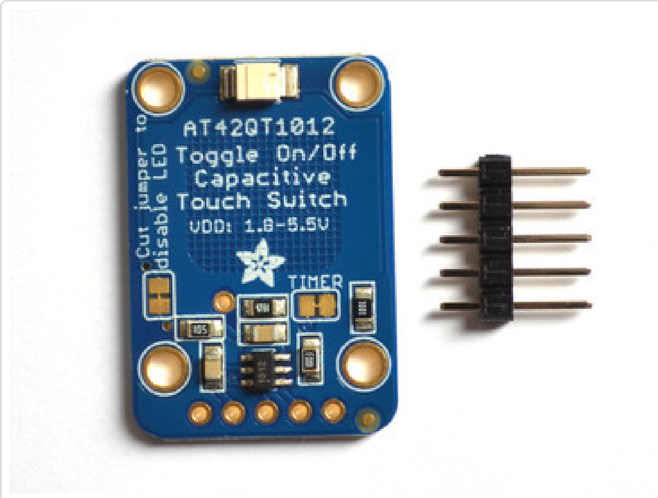


Momentary

This sensor has a built-in sense-pad and is active for as long as the sensor area is touched.

The sense-pad can be extended with wire and almost any conductive material.

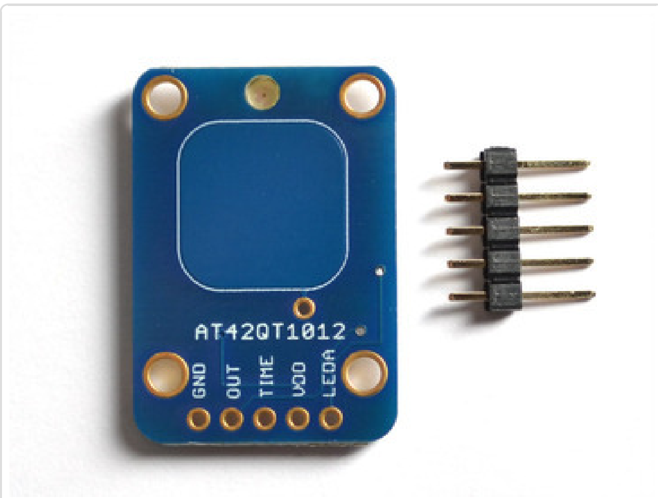


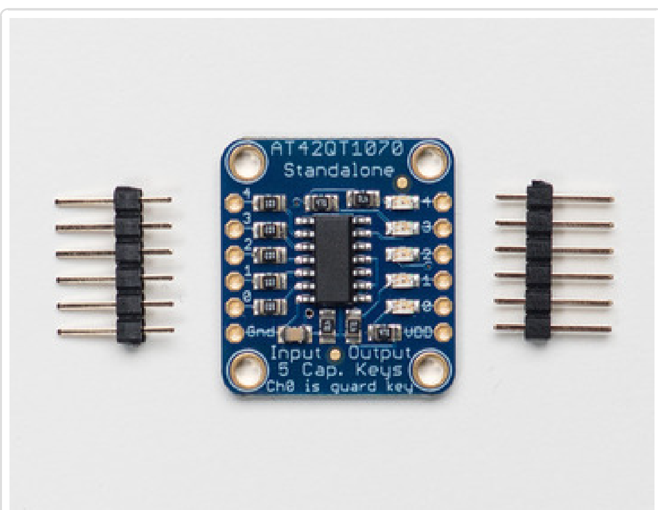


Toggle

This sensor also has a built-in sense-pad. It becomes active when touched and remains active until it is touched again.

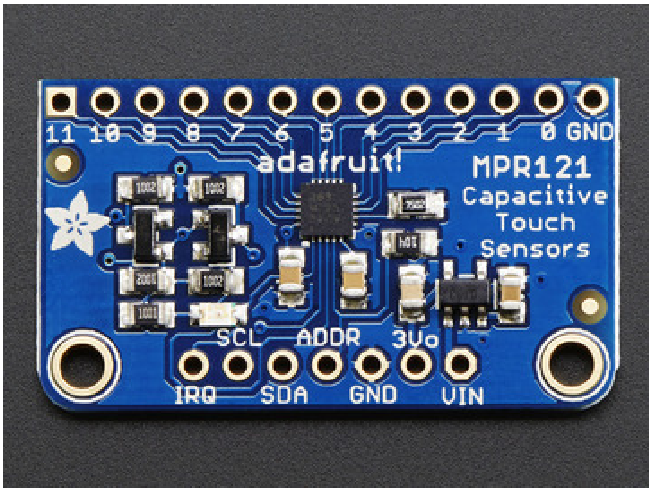
As with the momentary sensor, the sense-pad can be extended with wire and almost any conductive material.





5-Pad Momentary

This version combines 5 momentary switches into one breakout. There are 5 pins for attaching wires to up to 5 external sensor pads.



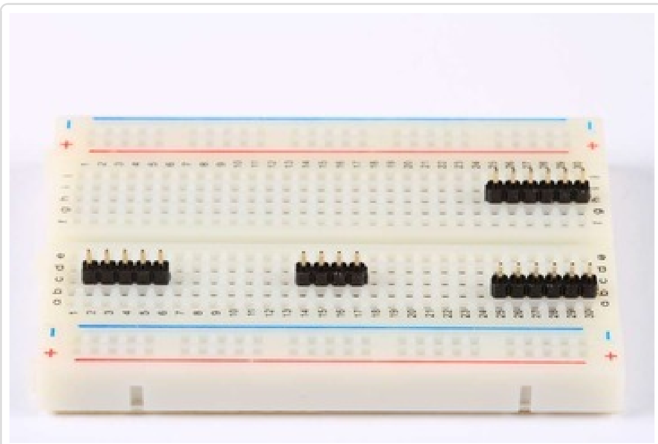
We also have a new 12-Key version with it's own tutorial over here! (http://adafru.it/dKH)

Assembly and Wiring

These breakouts come fully assembled. For use in a breadboard, you may want to take a couple minutes to install the included header strips:

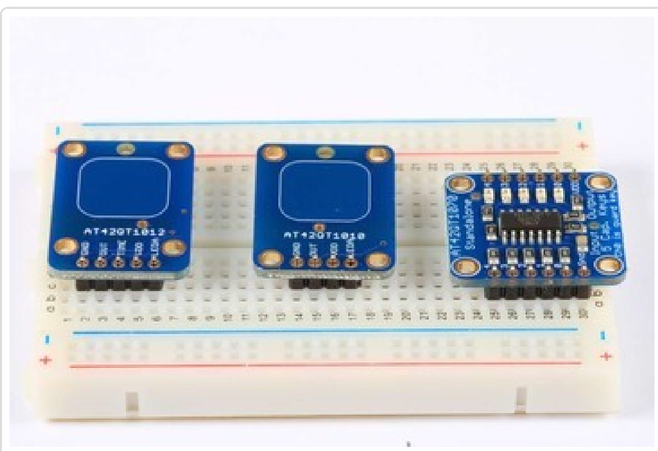
Installing the Headers:

Install the headers by following these 3 easy steps. The photographs below show one of each sensor type.



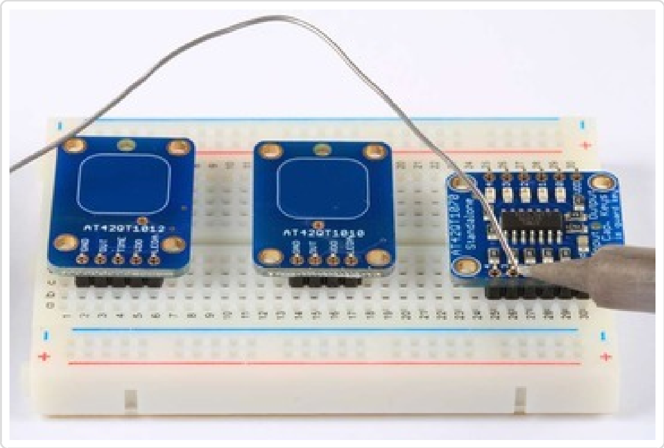
Position the header strips

Plug them long-pins down into a breadboard to stabilize them for soldering.



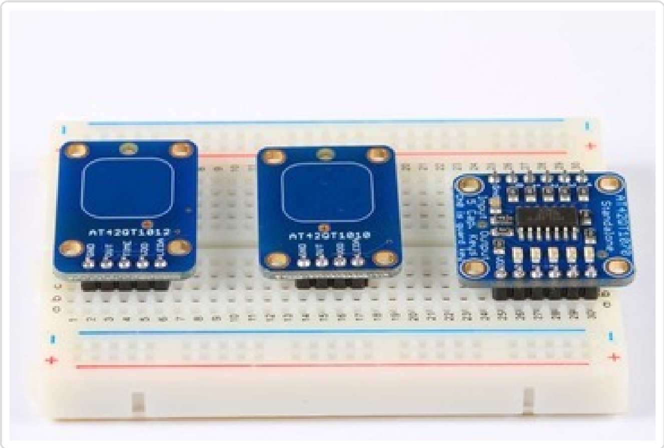
Position the breakout

Place the breakout board over the header pins.

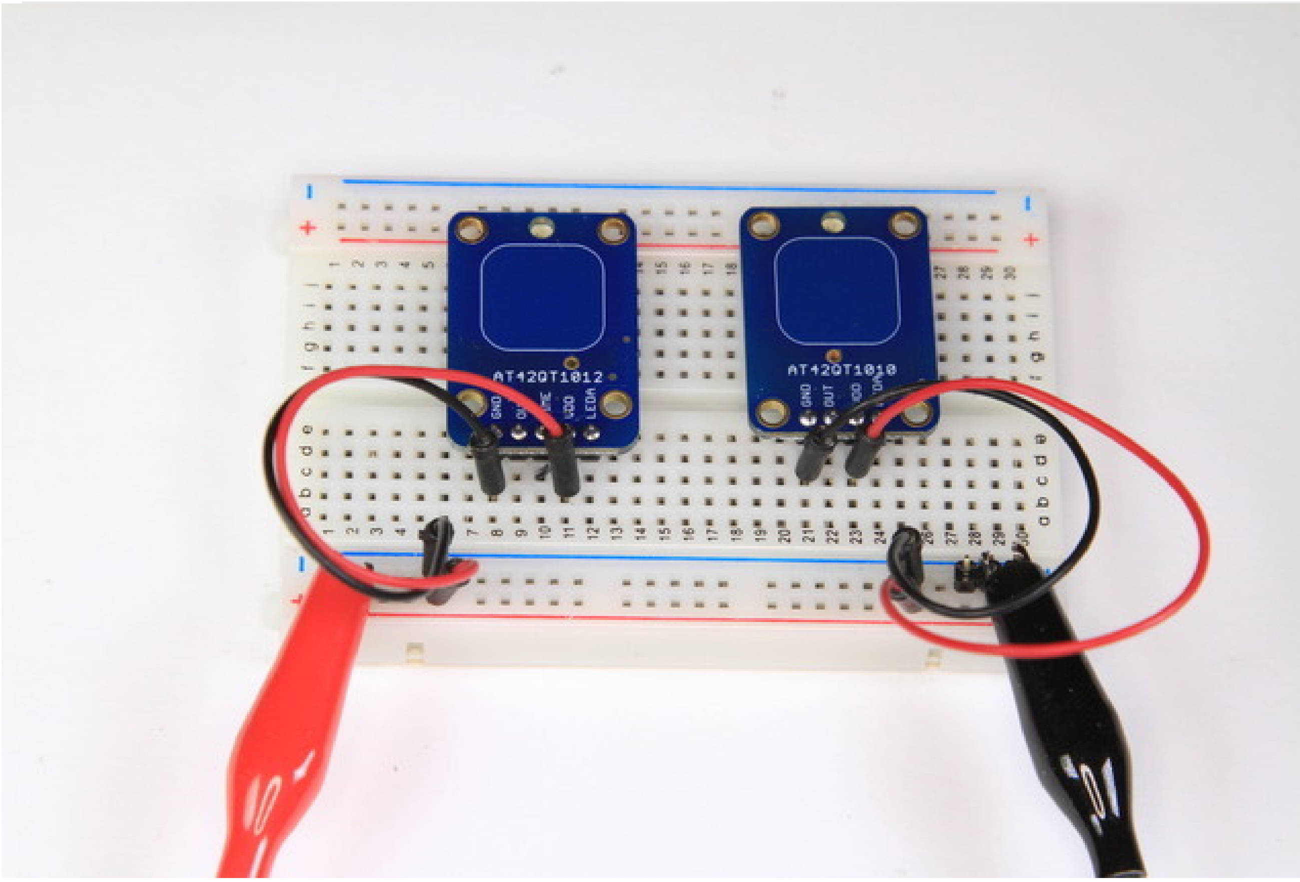


And Solder

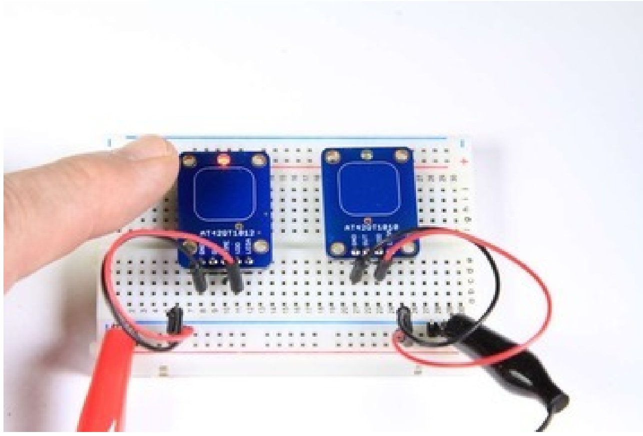
Solder each pin for solid electrical contact.



Wiring for Toggle and Momentary

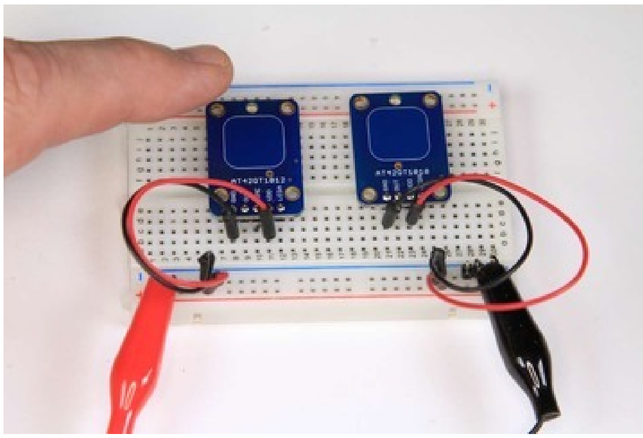


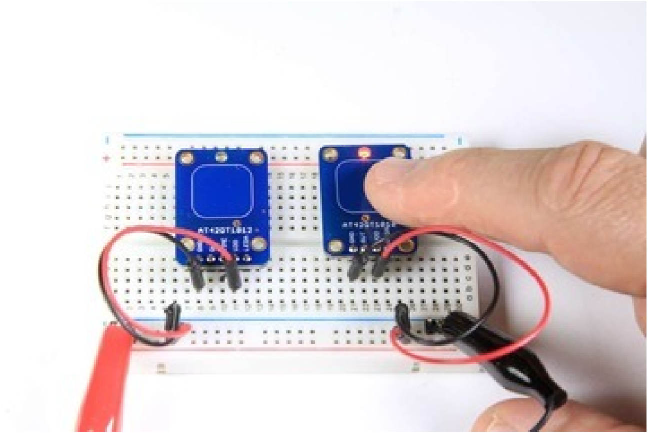
These two breakouts are very similar and can be powered by anything from 1.8V to 5.5V DC. Simply connect Ground to GND and the positive voltage to VDD. The standalone sensors are fully functional without further connections.



Toggle Operation

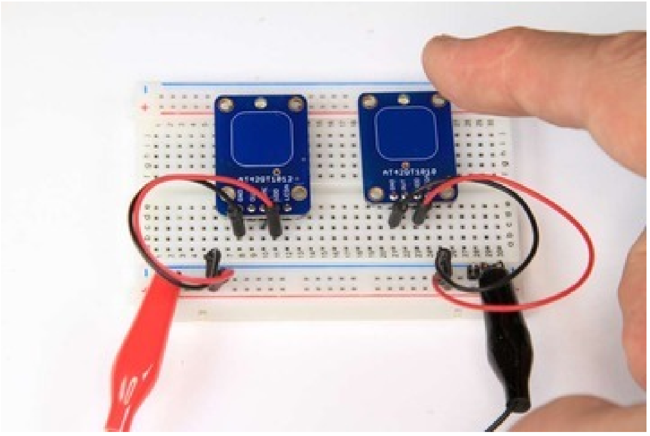
The Toggle version of the sensor turns on when you touch it once, then turns off when you touch it again. The on-board LED indicates the state of the switch.





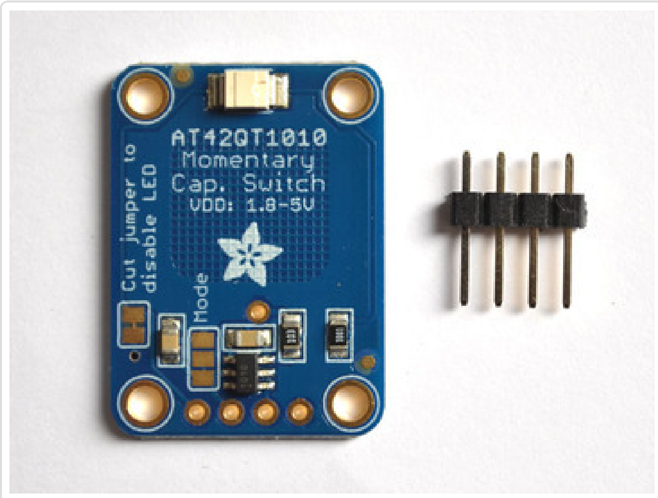
Momentary Operation

The momentary touch sensor works just like a momentary switch. It is on when you touch it and off when you move away. The on-board LED indicates the state of the switch.



Other Options:

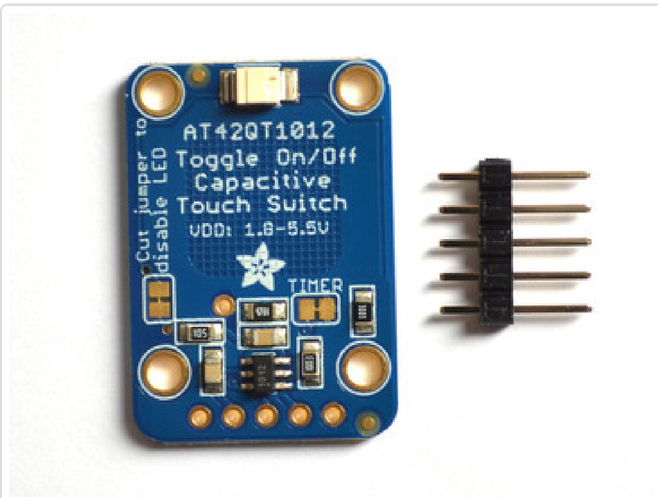
These sensors have several jumper configurable operating modes as described below:

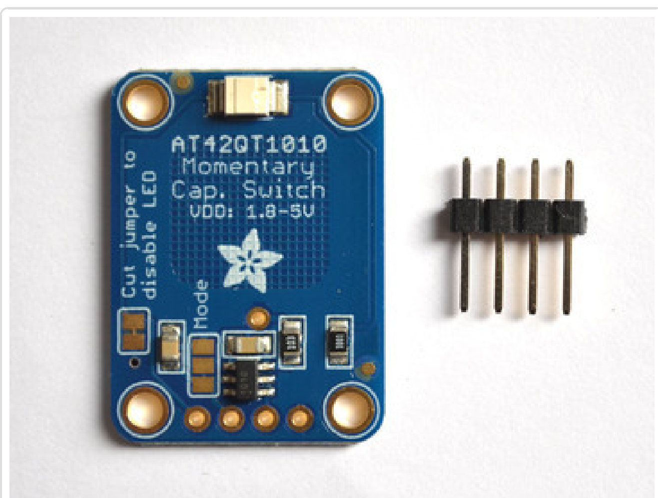


LED Control

The led indicators can be disabled for ultra-low power applications. To disable the LED, simply cut the jumper between the pads where indicated on the back of the breakout board.

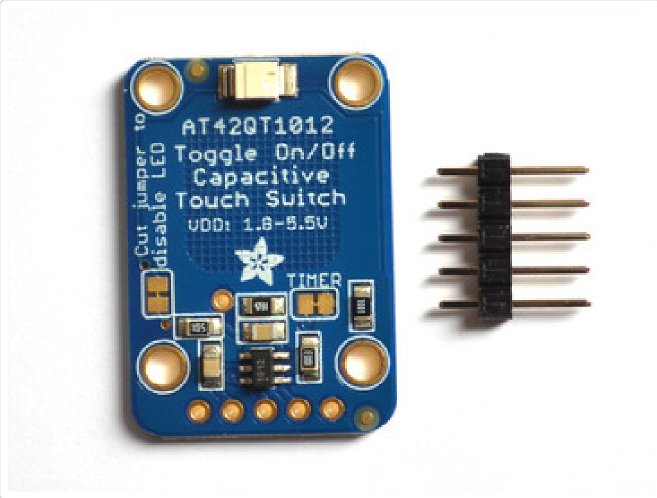
With the jumper cut, the LED can be controlled externally via the LED pin on the header.





Speed vs. Power (Momentary Only)

The Momentary version can be configured for "Fast" mode (default) or low-power mode. Fast mode requires 0.5mA. Low Power mode requires just 50uA. To switch between the two, cut the jumper on one side of the "mode" pads and bridge the other side with a drop of solder.



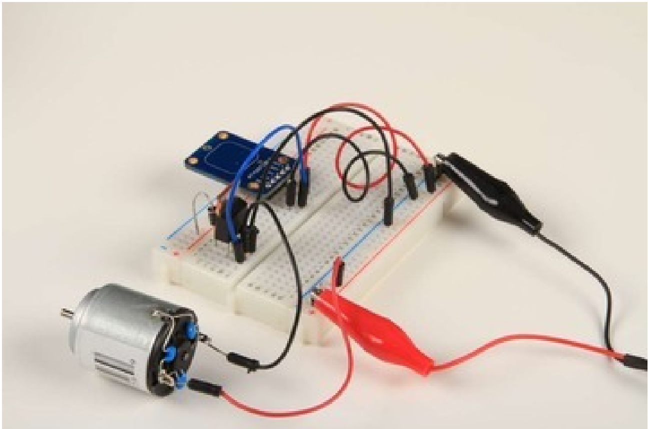
Timer (Toggle Only)

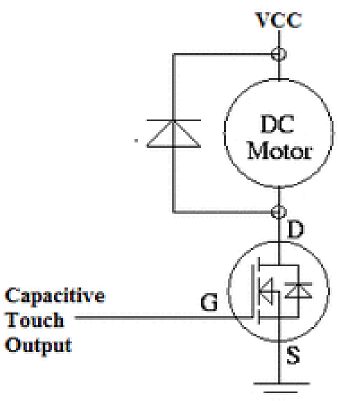
By default, the toggle sensor is configured for infinte time-out. it will stay on until you touch the sensor to turn it off. It also supports a configurable time-out to turn off the output automatically after a delay. To select this mode, cut the 'TIMER' jumper and connect a resistor & capacitor to the TIME pin. For a circuit diagram and resistor/capacitor calculations, see page 13 of the datasheet (http://adafru.it/cgW).

You can also just connect TIME to Vdd and the chip will turn off approx 15 minutes after being turned on. Connect TIME to OUT and the chip will time-out approx one hour after being turned on.

Connecting to your Circuit.

The outputs of these touch switches are 'active high'. Use them like a positive logic signal, or a pushbutton that shorts to VCC.

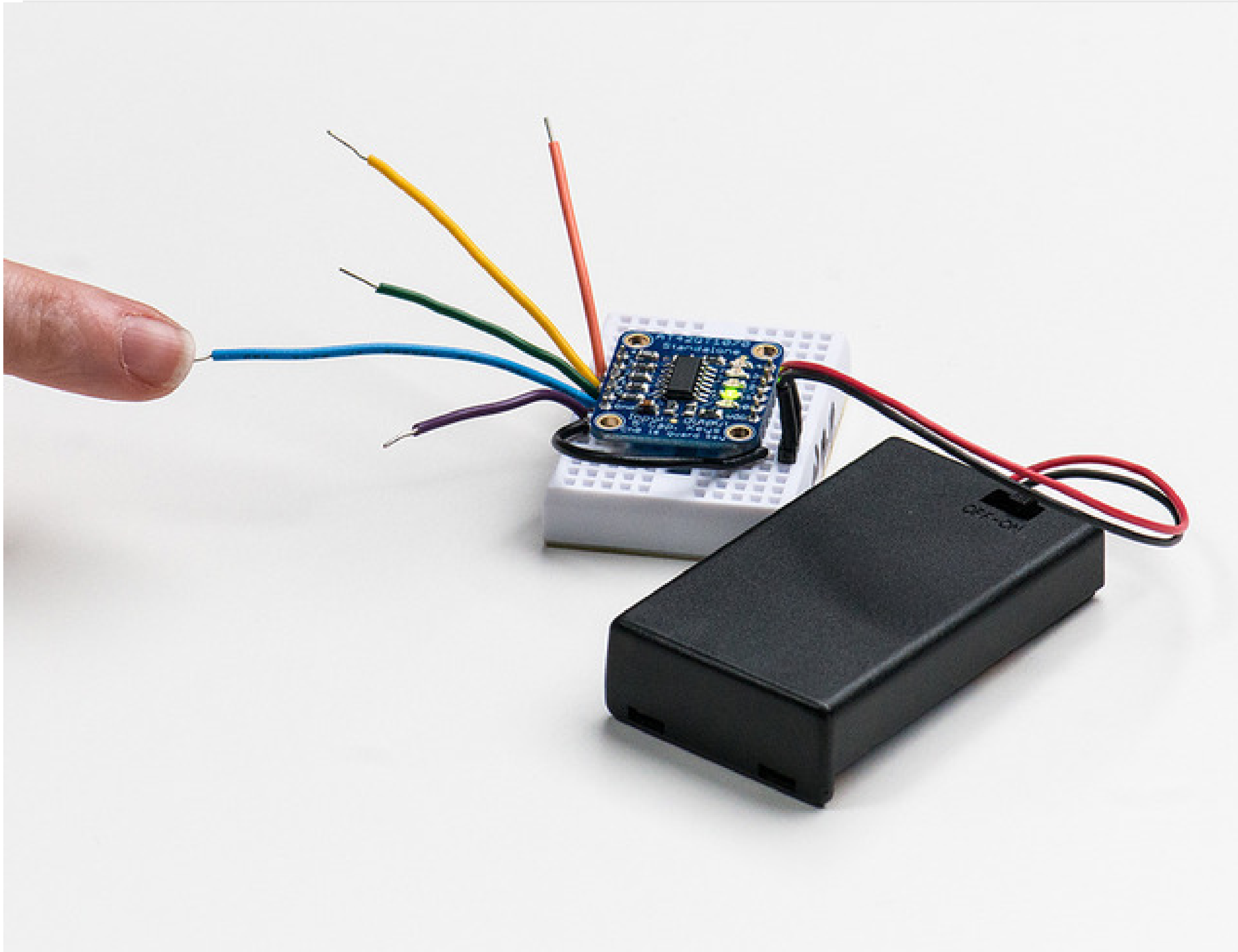




Simple Motor Control

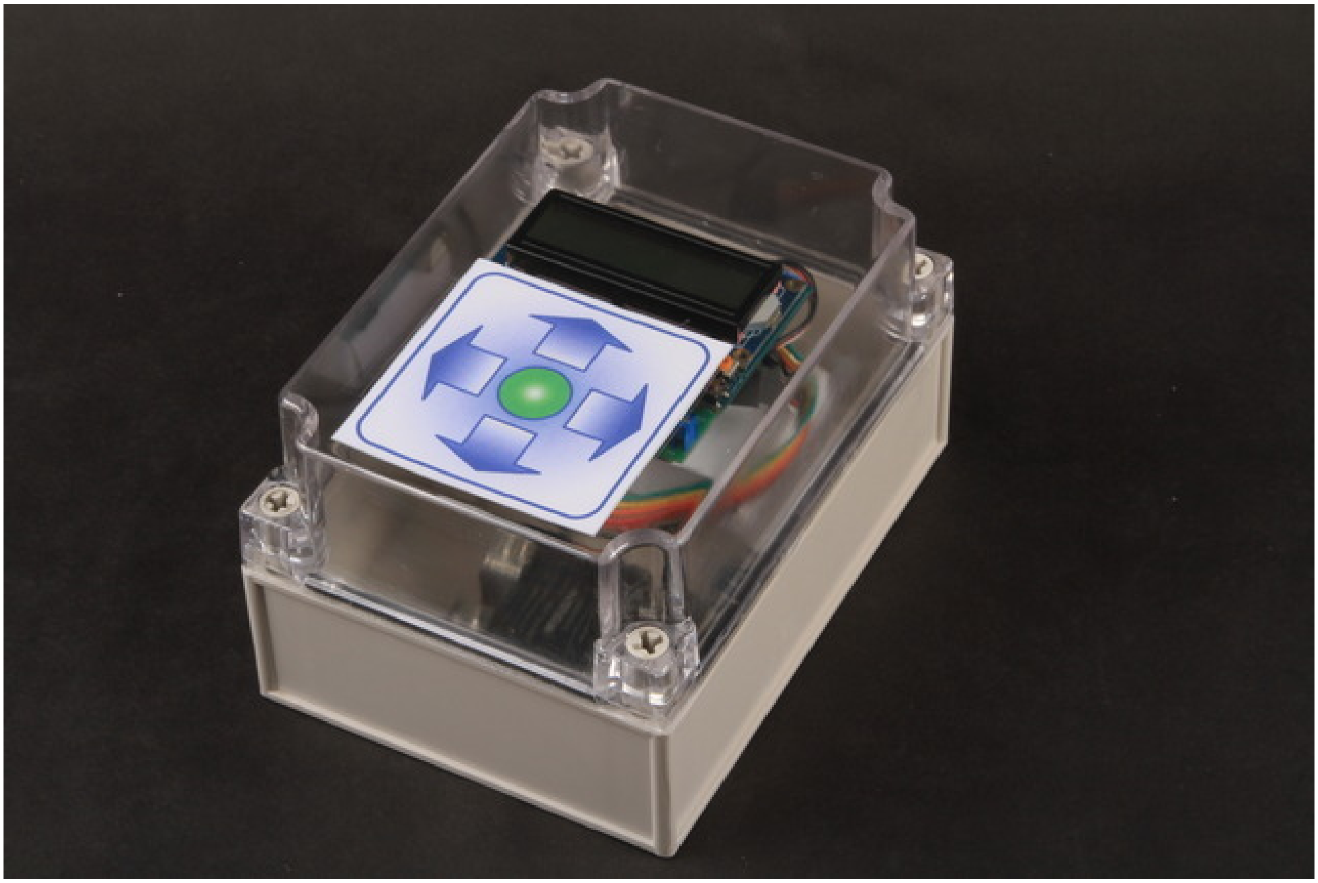
You can use it just like a pushbutton or logic signal with a transistor or MOSFET to drive high-current loads like a DC motor.

Wiring for 5-pad Momentary



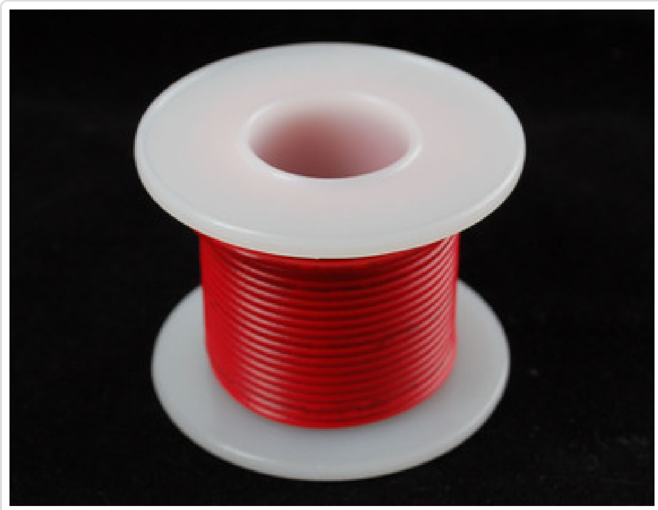
The 5-pin momentary breakout can be powered with anything from 1.8V to 5.5VDC. Just connect ground to GND and the positive voltage to VDD. This sensor does not have built-in touch pads, but you can create your own pads in any size or shape from wire, foil or any other conductive material. Simply connect your touch-pads to each of the 5 sense pins. When you touch the pad, the corresponding LED on the other side will light up. The chip only detects one touch-pad at a time to prevent false readings.

5 touch inputs in one device makes this the perfect component for building your own custom capacitive touch panel!



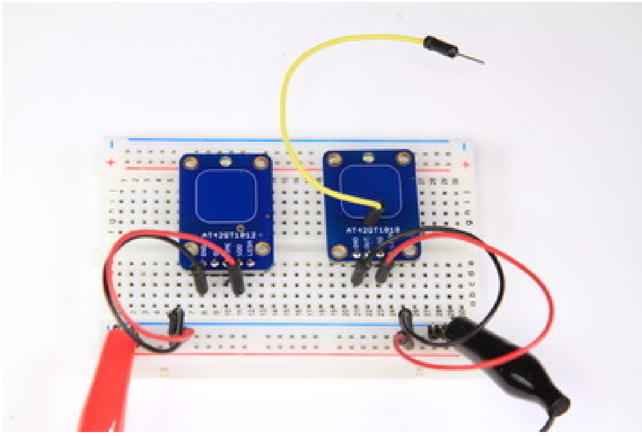
Adding Custom Touch Pads

Custom touch pads are easy to make. You can use almost any conductive material:



Wire, Thread, Foil, Fabric, Paint

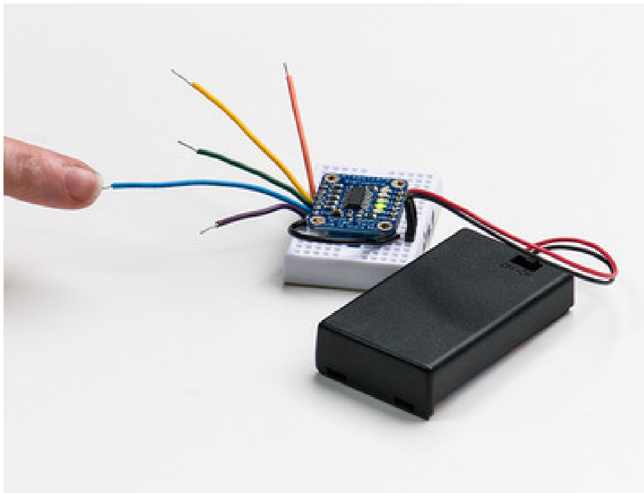
If it will conduct electricity, it will work as a touch sensor!

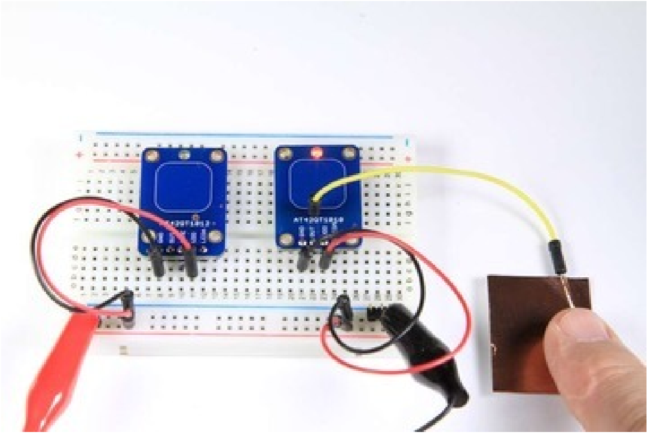


Connections:

The Toggle and Momentary boards have a solder hole located just below the sensor pad for attaching a wire to an external sensor.

The 5-pad breakout has pins numbered 0-4 on the left side of the board.

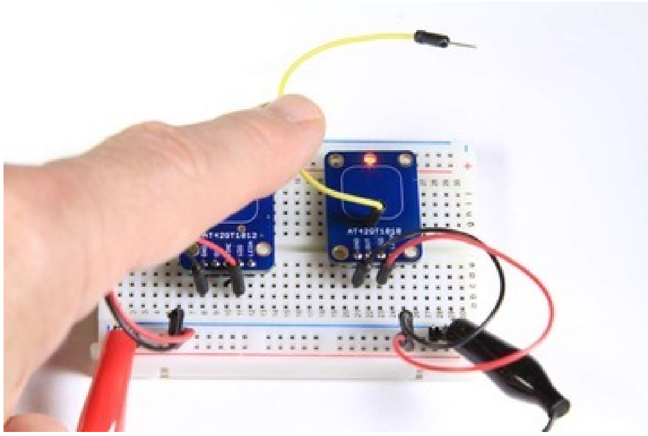




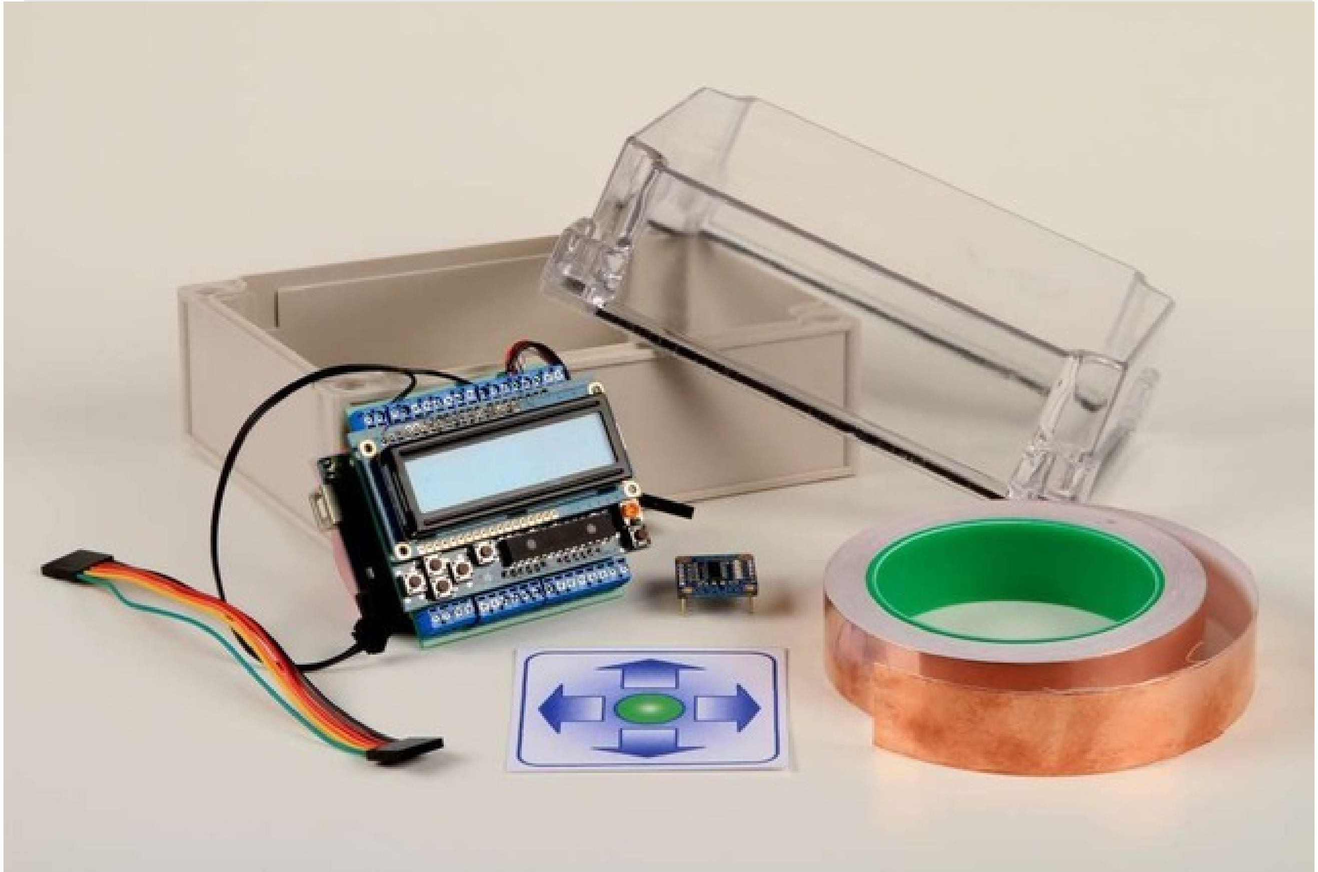
Sensor Pads

Attach the connecting wire to any conductive object or surface. That surface will become touch sensitive. Larger surfaces tend to be more sensitive. You will be able to sense through fabric, plastic glass and many other non-conductive materials.

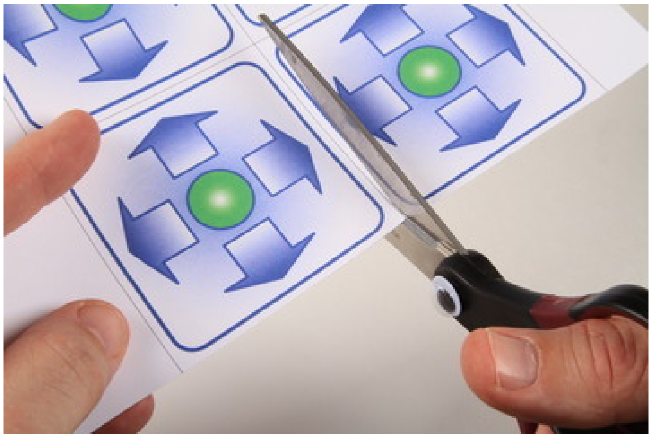
Note that the wire will be touch sensitive too! Be sure to route any connecting wires away from areas where they might create an accidental touch input.



Build a Touch Control Panel

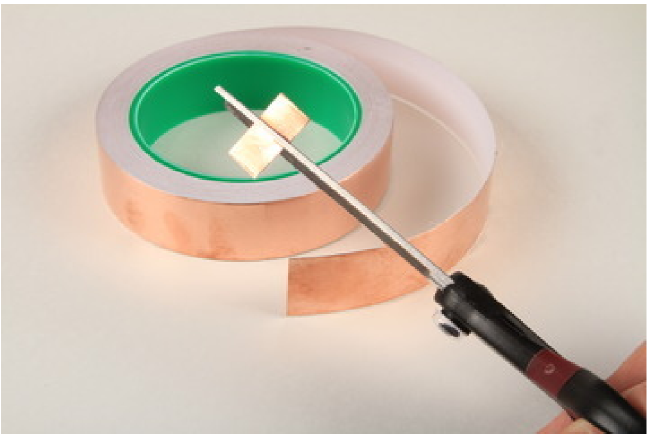


Capacitive touch sensors are a great way to add external controls to a waterproof enclosure. There is no need to drill holes or worry about gaskets and O-rings. These sensors will detect your touch right through the plastic case!



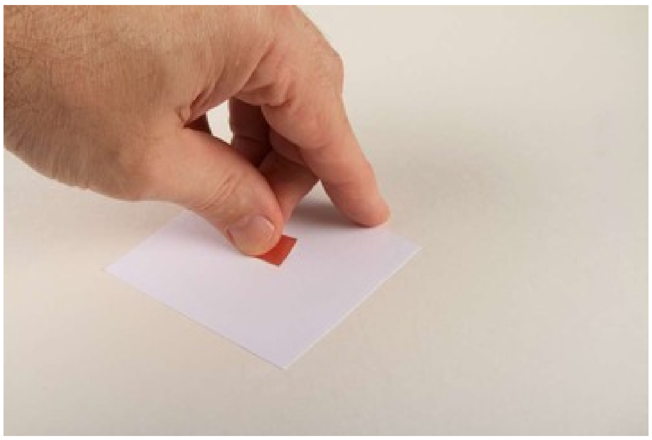
Design your panel

You can draw it by hand, or with your favorite drawing tool and print it on some heavy card-stock.



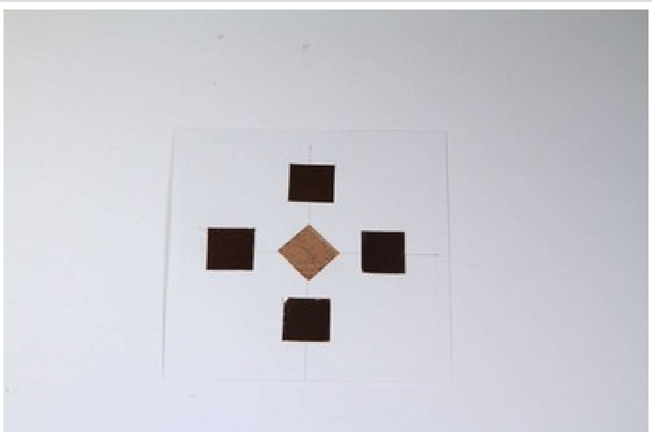
Cut the touch-pads

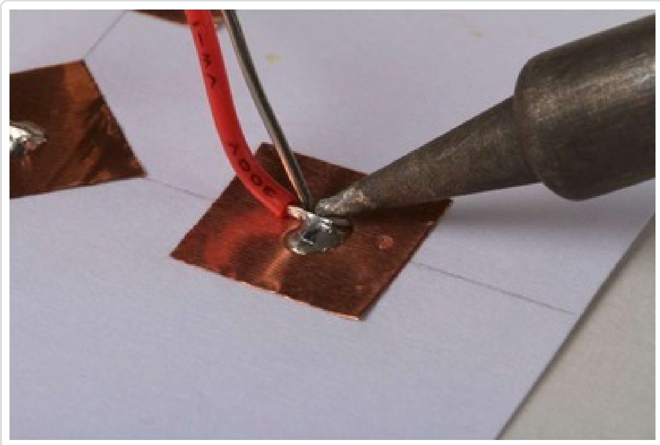
Cut pads from copper tape. About 1/2" square is a good size for buttons on a touch-pad.

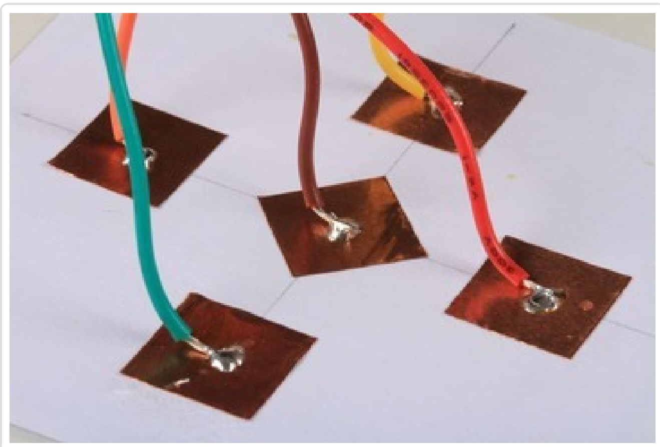


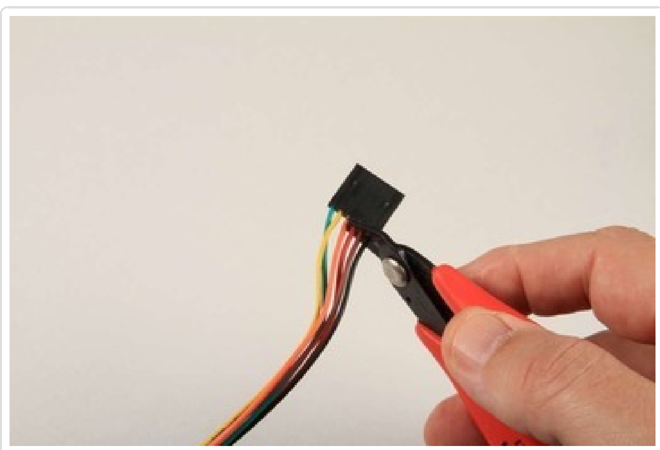
Attach the touch-pads

Peel the release paper from the back of the copper tape and stick the touch-pads to be back of the panel so that they align with the buttons on the front.



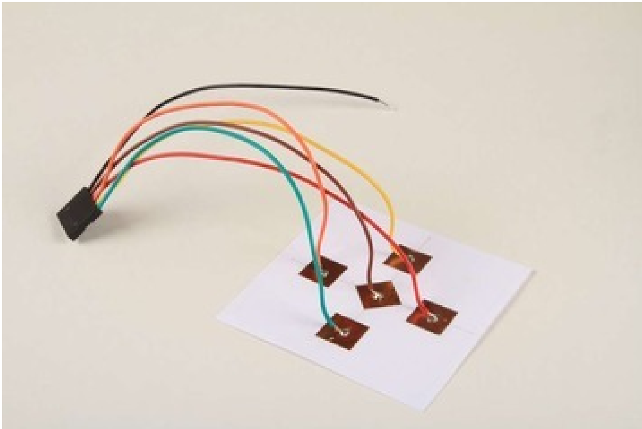






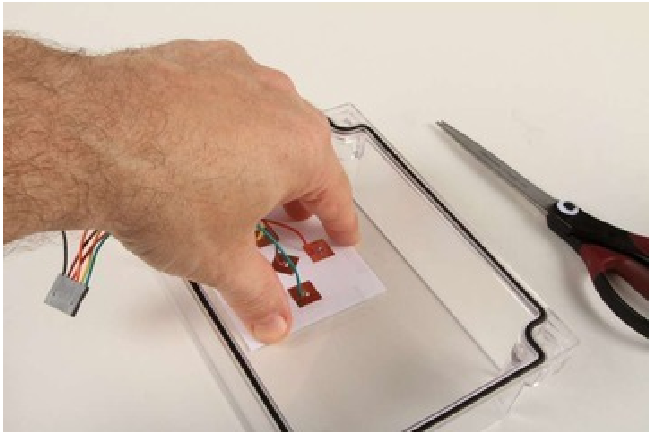
Attach the Wires

Solder wires to the copper touch-pads. For this example, I used a 6-conductor 0.1" socket cable (http://adafru.it/206) with one end cut off to simplify connections to the breakout.



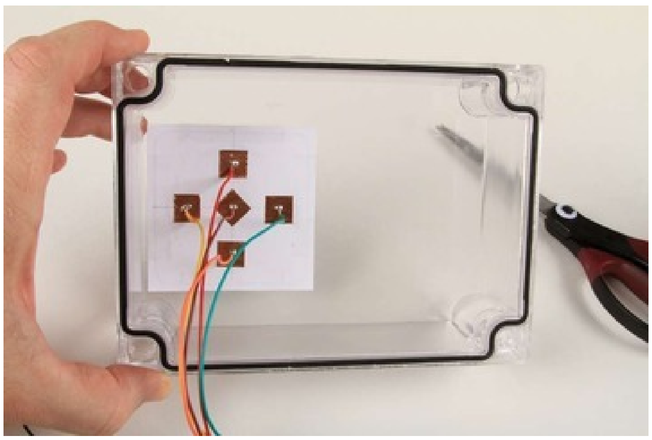
Adjust the Wires

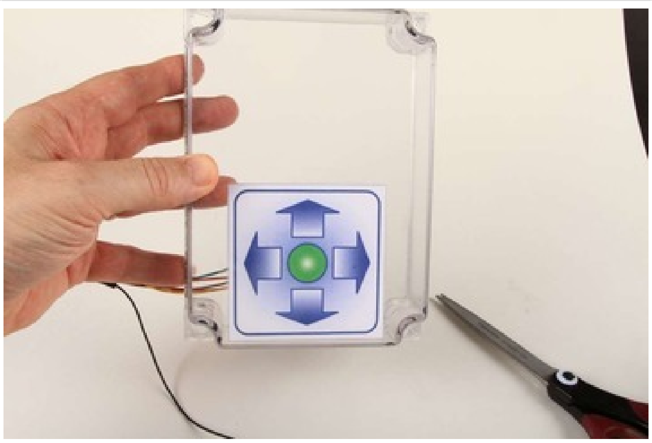
Bend the wires away from the panel. The wires will be touch-sensitive too. To prevent accidental false touches, we want to keep them away from the panel surface.



Install the Panel

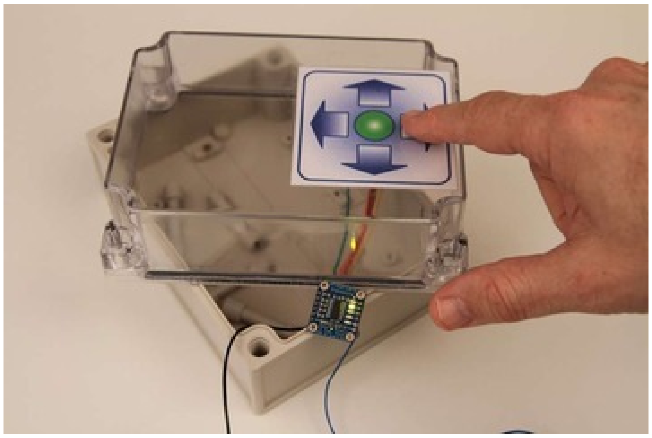
Tape the panel to the inside of the polycarbonate cover using clear packing tape.

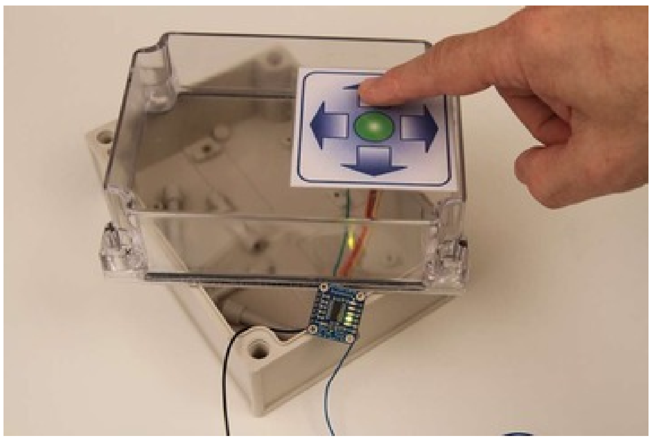


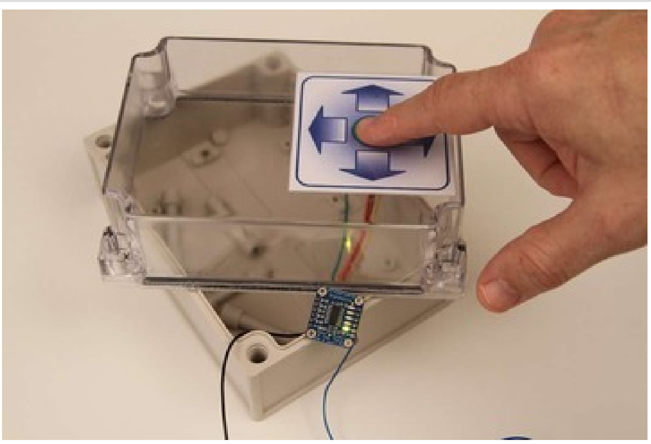


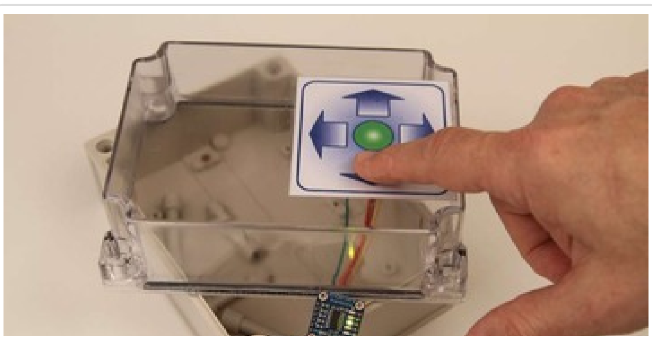
And Test

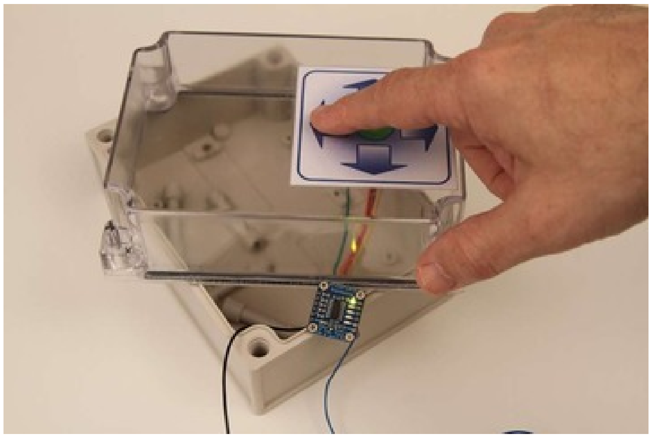
Connect the cable to the breakout. Power it up and test your control panel. Touching each button should cause a different LED to light up.

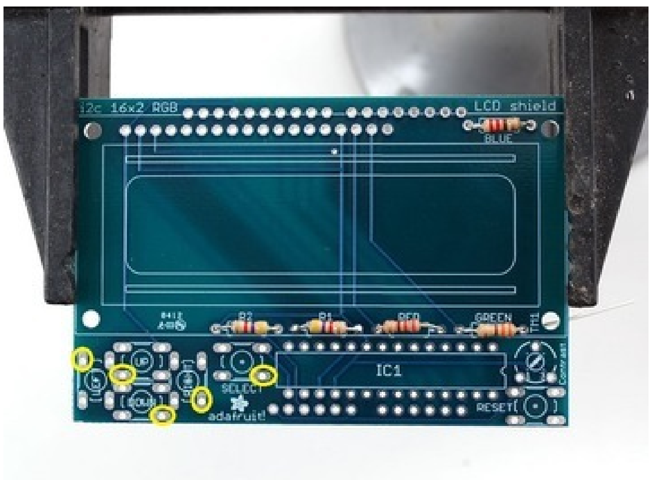








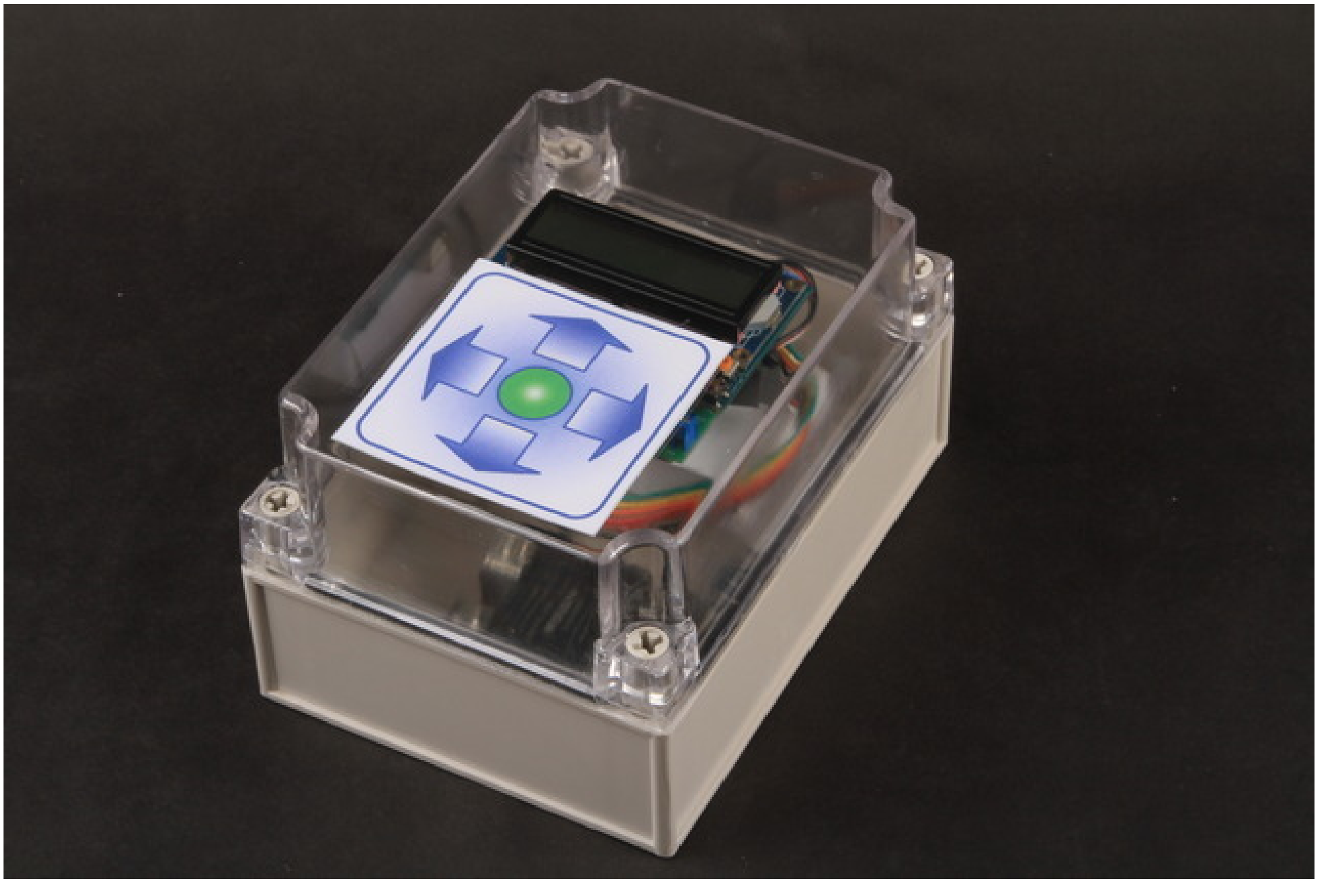




Connect to your Circuit

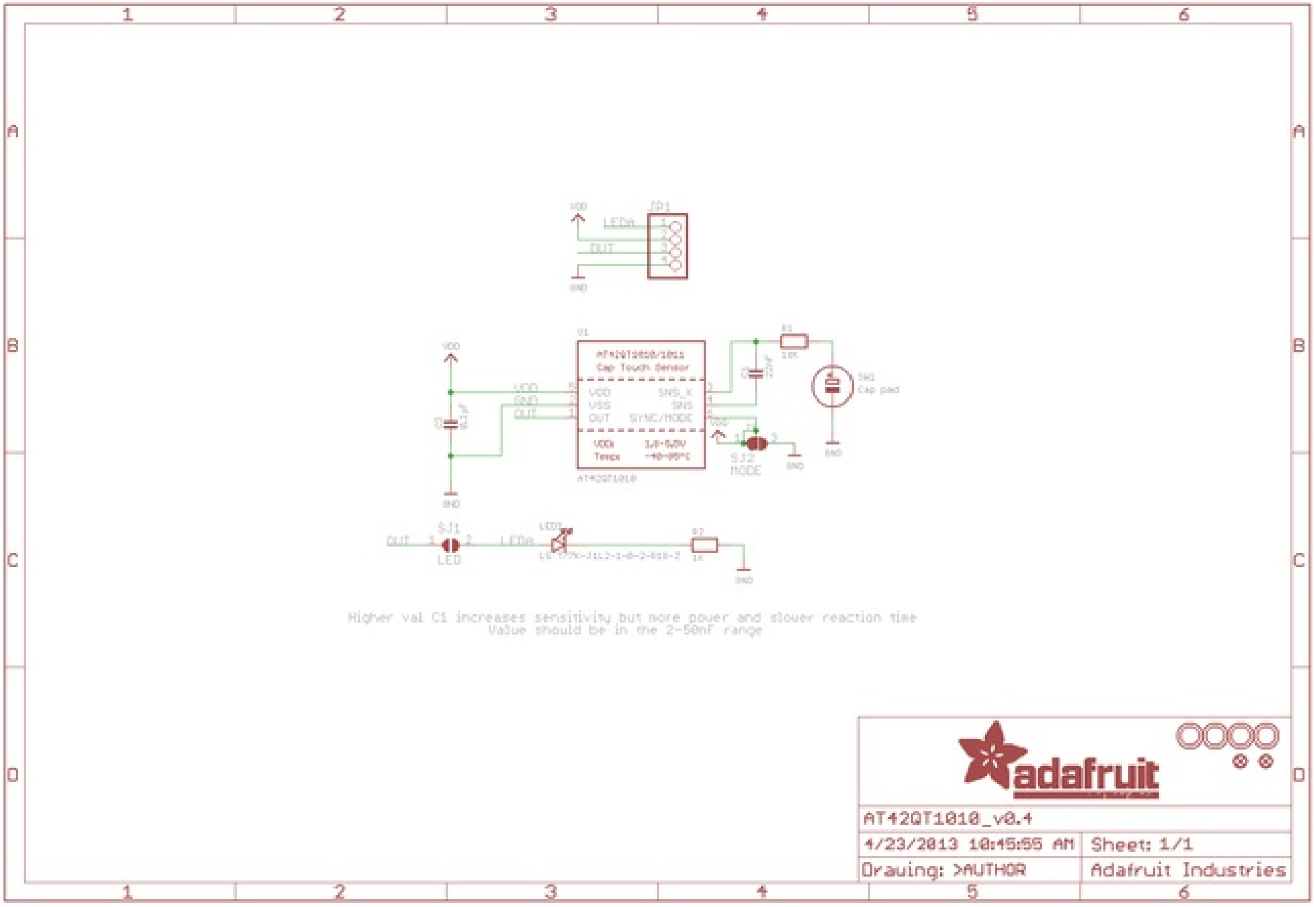
The output signals are 'active low', so they can replace any pushbutton that shorts to ground - such as the buttons on the RGB LCD shield. You can leave off the buttons and solder directly to the circled pads, or (if your shield is already built), just 'tack-solder' the wires to the legs of the buttons.

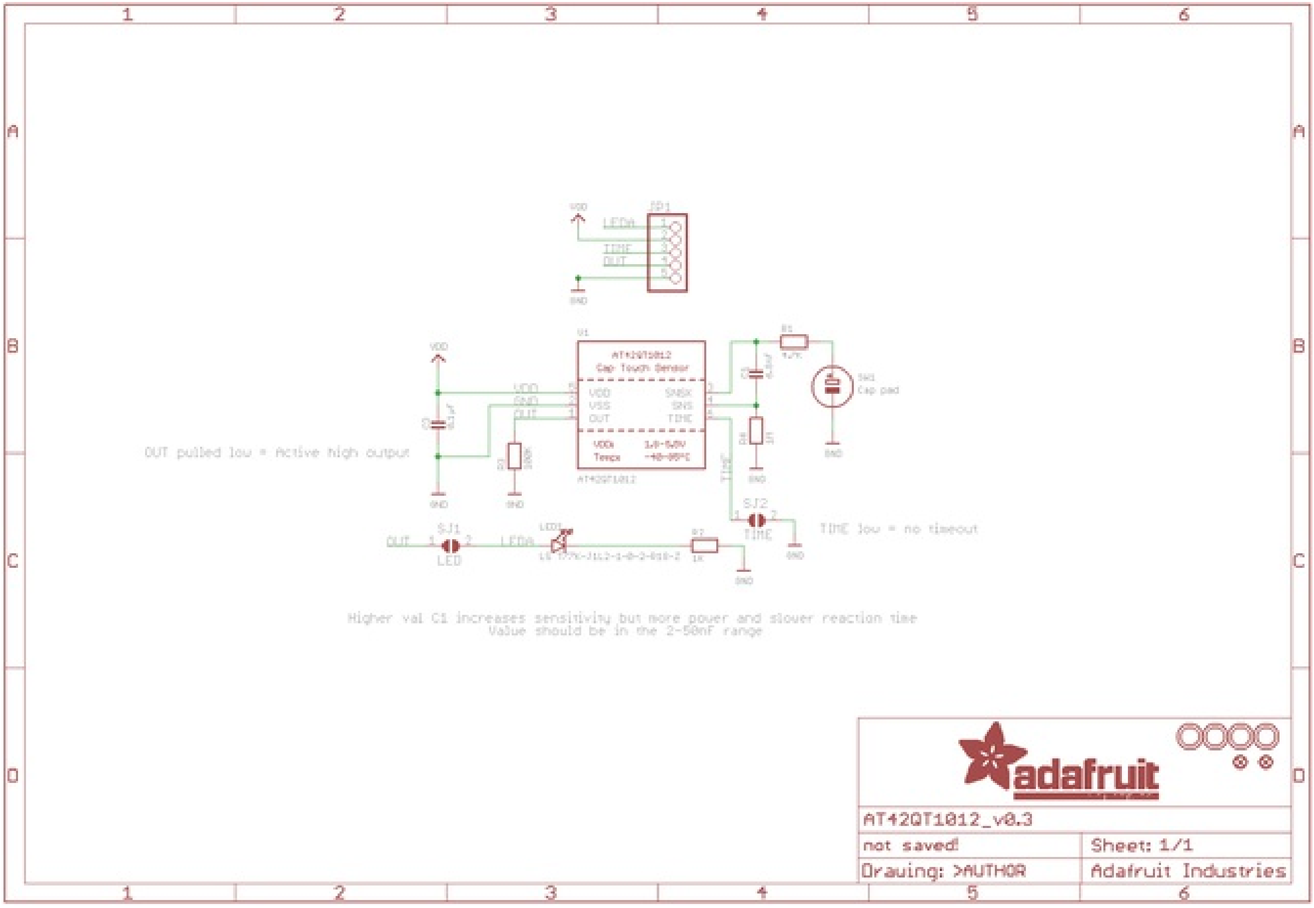
When you put it all together, you will have a completely sealed, touch sensitive control panel!



Downloads

Schematics for '1010 and '1012 breakouts (click to enlarge)





OUT pulled low = Active high output
TIME low = no timeout
Higher val C1 increases sensitivity but more power and slower reaction time
Value should be in the 2-50nF range
AT42QT1012_v0.3
not saved!
Sheet: 1/1
Drawing: >AUTHOR
Adafruit Industries

Freescale Semiconductor
Data Sheet: Technical Data
An Energy Efficient Solution by Freescale

Document Number: MPR121
Rev. 4, 02/2013

Proximity Capacitive Touch Sensor Controller

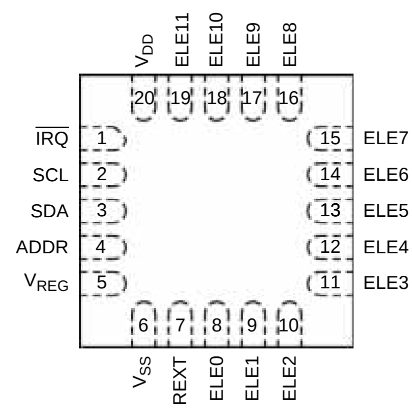
The MPR121 is the second generation sensor controller following the initial release of the MPR03x series of devices. The MPR121 features an increased internal intelligence plus Freescale's second generation capacitance detection engine. Some major enhancements include an increased electrode count, a hardware configurable I^2C address, an expanded filtering system with debounce, and completely independent electrodes with built-in autoconfiguration. The device also features a 13th simulated electrode which represents the simultaneous charging of all the electrodes connected together. When used with a touch panel or touch screen array, the 13th simulated electrode allows a greater near proximity detection distance and an increased sensing area.

MPR121

Bottom View

20-PIN QFN
CASE 2059-01

Top View



Pin Connections

Features

- 1.71V to 3.6V operation
- 29 µA typical run current at 16 ms sampling interval
- 3 µA in scan stop mode current
- 12 electrodes/capacitance sensing inputs in which 8 are multifunctional for LED driving and GPIO
- Integrated independent autocalibration for each electrode input
- Autoconfiguration of charge current and charge time for each electrode input
- Separate touch and release trip thresholds for each electrode, providing hysteresis and electrode independence
- I^2C interface, with $\overline{IRQ}$ Interrupt output to advise electrode status changes
- 3 mm x 3 mm x 0.65 mm 20 lead QFN package
- -40°C to +85°C operating temperature range

Implementations

- General Purpose Capacitance Detection
- Switch Replacements
- Touch Pads, Touch Wheel, Touch Slide Bar, Touch Screen Panel
- Capacitance Near Proximity Detection

Typical Applications

- PC Peripherals
- MP3 Players
- Remote Controls
- Mobile Phones
- Lighting Controls

ORDERING INFORMATION					
Device Name	**Temperature Range**	**Case Number**	**Touch Pads**	**I^2C Address**	**Shipping**
MPR121QR2	-40°C to +85°C	2059 (20-Pin QFN)	12-pads	0x5A - 0x5D	Tape & Reel

1 Pin Descriptions

Table 1. Pin Descriptions

Pin No.	Pin Name	Description
1	$\overline{IRQ}$	Open Collector Interrupt Output Pin, active low
2	SCL	I^2C Clock
3	SDA	I^2C Data
4	ADDR	I^2C Address Select Input Pin. Connect the ADDR pin to the VSS, VDD, SDA or SCL line, the resulting I^2C addresses are 0x5A, 0x5B, 0x5C and 0x5D respectively
5	VREG	Internal Regulator Node – Connect a 0.1 μF bypass cap to VSS
6	VSS	Ground
7	REXT	External Resistor – Connect a 75 kΩ 1% resistor to VSS to set internal reference current
8	ELE0	Electrode 0
9	ELE1	Electrode 1
10	ELE2	Electrode 2
11	ELE3	Electrode 3
12	ELE4	Electrode 4
13	ELE5	Electrode 5
14	ELE6	Electrode 6
15	ELE7	Electrode 7
16	ELE8	Electrode 8
17	ELE9	Electrode 9
18	ELE10	Electrode 10
19	ELE11	Electrode 11
20	VDD	Connect a 0.1 μF bypass cap to VSS

2 Schematic Drawings and Implementation

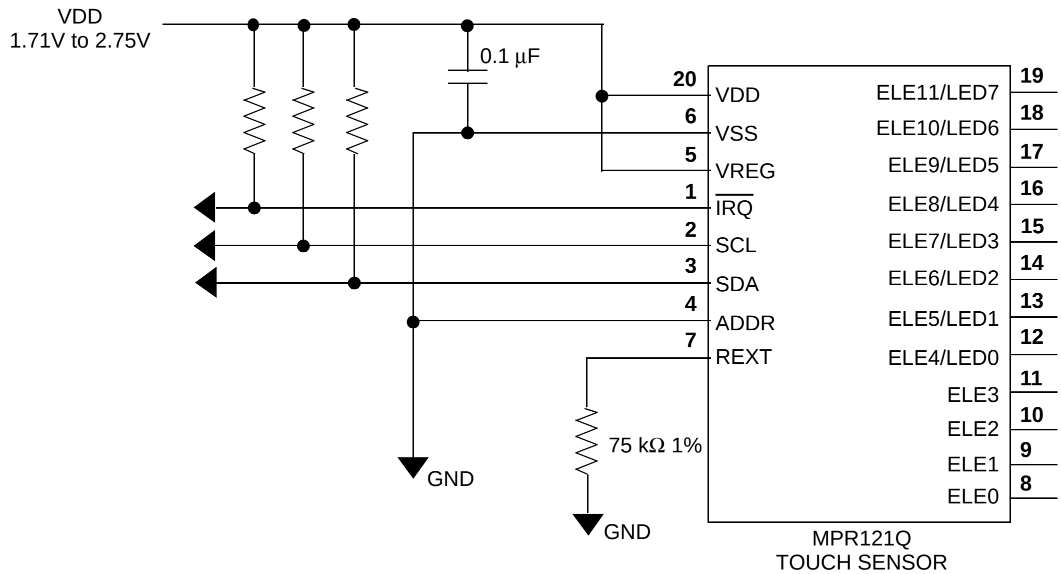


Figure 1. Power Configuration 1: MPR121 runs from a 1.71V to 2.75V supply.

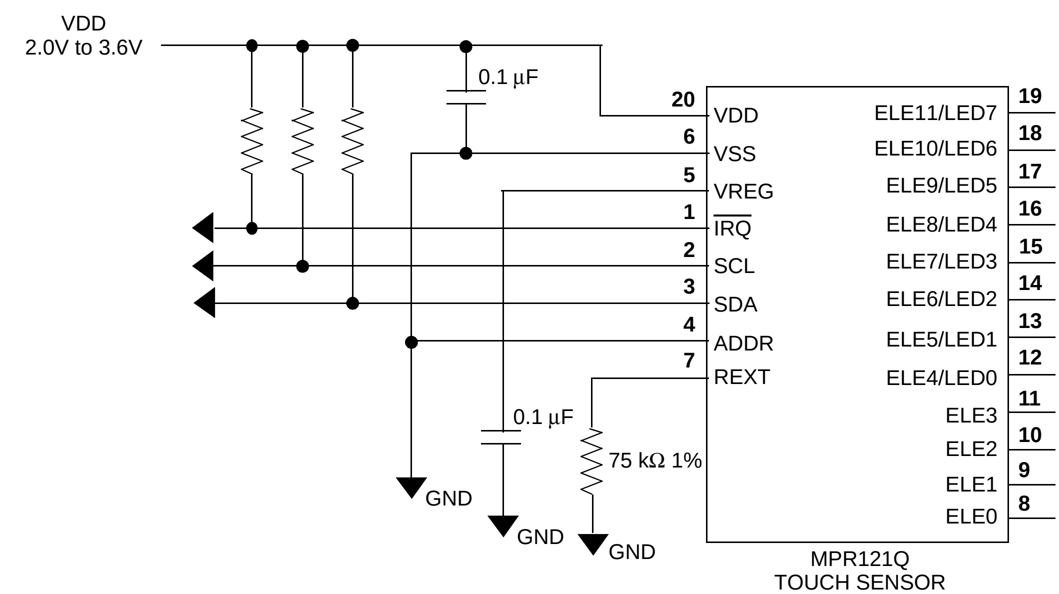


Figure 2. Power Configuration 2: MPR121 runs from a 2.0V to 3.6V supply.

3 Device Operation Overview

Power Supply

The VDD pin is the main power supply input to the MPR121 and is always decoupled with a 0.1 μF ceramic capacitor to the VSS. Excessive noise on the VDD should be avoided.

The VDD pin has an operational voltage range specification between 1.71V to 3.6V. The internal voltage regulator, which generates current to internal circuitry, operates with an input range from 2.0V to 3.6V. To work with a power supply below 2.0V and to avoid the unnecessary voltage drop, the internal voltage regulator can be bypassed, refer to Figure 1 and Figure 2.

When a power supply is in the range of 1.71V to 2.75V, the VDD and VREG pins can be connected together (Figure 1) so that internal voltage regulator is bypassed. In this configuration, the supply voltage cannot be higher than 2.75V as this is the maximum voltage limit for VREG pin.

When a power supply is higher than 2.75V, it must be connected to the VDD, i.e. configuration as in Figure 2. In this configuration, a separate 0.1 μF decoupling ceramic capacitor on VREG to VSS is applied as a bypass cap for internal circuitry. This configuration can work with a VDD supply voltage down to 2.0V. For a typical two dry cell 1.5V batteries application, this configuration covers the entire expected working voltage range from 2.0V to 3.0V.

Capacitance Sensing

The MPR121 uses a constant DC current capacitance sensing scheme. It can measure capacitances ranging from 10 pF to over 2000 pF with a resolution up to 0.01 pF. The device does this by varying the amount of charge current and charge time applied to the sensing inputs.

The 12 electrodes are controlled independently; this allows for a great deal of flexibility in electrode pattern design. An automatic configuration system is integrated as part of the device, this greatly simplifies the individual register setup. Please refer to the Freescale application note, AN3889, for more details.

The voltage measured on the input sensing node is inversely proportional to the capacitance. At the end of each charge circle, this voltage is sampled by an internal 10-bit ADC. The sampled data is then processed through several stages of digital filtering. The digital filtering process allows for good noise immunity in different environments. For more information on the filtering system, refer to application note AN3890.

Touch Sensing

Once the electrode capacitance data is acquired, the electrode touch/release status is determined comparing it to the capacitance baseline value. The capacitance baseline is tracked by MPR121 automatically based on the background capacitance variation.

The baseline value is compared with the current immediate electrode data to determine if a touch or release has occurred. A designer has the ability to set the touch/release thresholds, as well as a touch/release debounce time. This is to eliminate jitter and false touches due to noise. Additional information on baseline capacitance system is covered in application notes AN3891 and AN3892.

Proximity Sensing

One new feature of the MPR121 is the near proximity sensing system. This means that all of the system's electrodes can be summed together to create a single large electrode. The major advantage of the large electrode is that is can cover a much larger sensing area. The near proximity sensing system can be used while at the same time having separate electrodes by using touch button sensing.

Proximity detection is read as an independent channel and has configuration registers similar to the other 12 channels. When proximity detection is enabled, this "13^{th}" measurement channel will be included at the beginning of a normal detection cycle. This system is described in application note AN3893.

LED Driver

Among the 12 electrode inputs, 8 inputs are designed as multifunctional pins. When these pins are not configured as electrodes, they may be used to drive LEDs or used for general purpose input or output. For more details on this feature, please refer to application note AN3894.

Serial Communication

The MPR121 is an Inter-Integrated Circuit (I^2C) compliant device with an interrupt $\overline{IRQ}$ pin. This pin is triggered any time a touch or release is detected. The device has a configurable I^2C address by connecting the ADDR pin to the VSS, VDD, SDA or SCL lines This results in I^2C addresses of 0x5A, 0x5B, 0x5C and 0x5D. The specific details of this system are described in AN3895. For reference, the register map of the MPR121 is included in **Table 2**.

Table 2. Register Map

REGISTER	Fields								Register Address	Initial Value	Auto-Increment Address
ELE0 - ELE7 Touch Status	ELE7	ELE6	ELE5	ELE4	ELE3	ELE2	ELE1	ELE0	0x00	0x00	Register Address + 1
ELE8 - ELE11, ELEPROX Touch Status	OVCF			ELEPROX	ELE11	ELE10	ELE9	ELE8	0x01	0x00	
ELE0-7 OOR Status	E7_OOR	E6_OOR	E5_OOR	E4_OOR	E3_OOR	E2_OOR	E1_OOR	E0_OOR	0x02	0x00	
ELE8-11, ELEPROX OOR Status	ACFF	ARFF		PROX_OOR	E11_OOR	E10_OOR	E9_OOR	E8_OOR	0x03	0x00	
ELE0 Electrode Filtered Data LSB	EFD0LB								0x04	0x00	
ELE0 Electrode Filtered Data MSB							EFD0HB		0x05	0x00	
ELE1 Electrode Filtered Data LSB	EFD1LB								0x06	0x00	
ELE1 Electrode Filtered Data MSB							EFD1HB		0x07	0x00	
ELE2 Electrode Filtered Data LSB	EFD2LB								0x08	0x00	
ELE2 Electrode Filtered Data MSB							EFD2HB		0x09	0x00	
ELE3 Electrode Filtered Data LSB	EFD3LB								0x0A	0x00	
ELE3 Electrode Filtered Data MSB							EFD3HB		0x0B	0x00	
ELE4 Electrode Filtered Data LSB	EFD4LB								0x0C	0x00	
ELE4 Electrode Filtered Data MSB							EFD4HB		0x0D	0x00	
ELE5 Electrode Filtered Data LSB	EFD5LB								0x0E	0x00	
ELE5 Electrode Filtered Data MSB							EFD5HB		0x0F	0x00	
ELE6 Electrode Filtered Data LSB	EFD6LB								0x10	0x00	
ELE6 Electrode Filtered Data MSB							EFD6HB		0x11	0x00	
ELE7 Electrode Filtered Data LSB	EFD7LB								0x12	0x00	
ELE7 Electrode Filtered Data MSB							EFD7HB		0x13	0x00	
ELE8 Electrode Filtered Data LSB	EFD8LB								0x14	0x00	
ELE8 Electrode Filtered Data MSB							EFD8HB		0x15	0x00	
ELE9 Electrode Filtered Data LSB	EFD9LB								0x16	0x00	
ELE9 Electrode Filtered Data MSB							EFD9HB		0x17	0x00	
ELE10 Electrode Filtered Data LSB	EFD10LB								0x18	0x00	
ELE10 Electrode Filtered Data MSB							EFD10HB		0x19	0x00	
ELE11 Electrode Filtered Data LSB	EFD11LB								0x1A	0x00	
ELE11 Electrode Filtered Data MSB							EFD11HB		0x1B	0x00	
ELEPROX Electrode Filtered Data LSB	EFDPROXLB								0x1C	0x00	
ELEPROX Electrode Filtered Data MSB							EFDPROXHB		0x1D	0x00	
ELE0 Baseline Value	E0BV								0x1E	0x00	
ELE1 Baseline Value	E1BV								0x1F	0x00	
ELE2 Baseline Value	E2BV								0x20	0x00	
ELE3 Baseline Value	E3BV								0x21	0x00	
ELE4 Baseline Value	E4BV								0x22	0x00	
ELE5 Baseline Value	E5BV								0x23	0x00	
ELE6 Baseline Value	E6BV								0x24	0x00	
ELE7 Baseline Value	E7BV								0x25	0x00	
ELE8 Baseline Value	E8BV								0x26	0x00	
ELE9 Baseline Value	E9BV								0x27	0x00	
ELE10 Baseline Value	E10BV								0x28	0x00	
ELE11 Baseline Value	E11BV								0x29	0x00	
ELEPROX Baseline Value	EPROXBV								0x2A	0x00	
MHD Rising			MHDR						0x2B	0x00	
NHD Amount Rising			NHDR						0x2C	0x00	
NCL Rising	NCLR								0x2D	0x00	
FDL Rising	FDLR								0x2E	0x00	
MHD Falling			MHDF						0x2F	0x00	
NHD Amount Falling			NHDF						0x30	0x00	

Table 2. Register Map

<table>
<tr><th>REGISTER</th><th colspan="8">Fields</th><th>Register Address</th><th>Initial Value</th><th>Auto-Increment Address</th></tr>
<tr><td>NCL Falling</td><td colspan="8">NCLF</td><td>0x31</td><td>0x00</td><td rowspan="49">Register Address + 1</td></tr>
<tr><td>FDL Falling</td><td colspan="8">FDLF</td><td>0x32</td><td>0x00</td></tr>
<tr><td>NHD Amount Touched</td><td></td><td></td><td colspan="6">NHDT</td><td>0x33</td><td>0x00</td></tr>
<tr><td>NCL Touched</td><td colspan="8">NCLT</td><td>0x34</td><td>0x00</td></tr>
<tr><td>FDL Touched</td><td colspan="8">FDLT</td><td>0x35</td><td>0x00</td></tr>
<tr><td>ELEPROX MHD Rising</td><td></td><td></td><td colspan="6">MHDPROXR</td><td>0x36</td><td>0x00</td></tr>
<tr><td>ELEPROX NHD Amount Rising</td><td></td><td></td><td colspan="6">NHDPROXR</td><td>0x37</td><td>0x00</td></tr>
<tr><td>ELEPROX NCL Rising</td><td colspan="8">NCLPROXR</td><td>0x38</td><td>0x00</td></tr>
<tr><td>ELEPROX FDL Rising</td><td colspan="8">FDLPROXR</td><td>0x39</td><td>0x00</td></tr>
<tr><td>ELEPROX MHD Falling</td><td></td><td></td><td colspan="6">MHDPROXF</td><td>0x3A</td><td>0x00</td></tr>
<tr><td>ELEPROX NHD Amount Falling</td><td></td><td></td><td colspan="6">NHDPROXF</td><td>0x3B</td><td>0x00</td></tr>
<tr><td>ELEPROX NCL Falling</td><td colspan="8">NCLPROXF</td><td>0x3C</td><td>0x00</td></tr>
<tr><td>ELEPROX FDL Falling</td><td colspan="8">FDLPROXF</td><td>0x3D</td><td>0x00</td></tr>
<tr><td>ELEPROX NHD Amount Touched</td><td></td><td></td><td colspan="6">NHDPROXT</td><td>0x3E</td><td>0x00</td></tr>
<tr><td>ELEPROX NCL Touched</td><td colspan="8">NCLPROXT</td><td>0x3F</td><td>0x00</td></tr>
<tr><td>ELEPROX FDL Touched</td><td colspan="8">FDLPROXT</td><td>0x40</td><td>0x00</td></tr>
<tr><td>ELE0 Touch Threshold</td><td colspan="8">E0TTH</td><td>0x41</td><td>0x00</td></tr>
<tr><td>ELE0 Release Threshold</td><td colspan="8">E0RTH</td><td>0x42</td><td>0x00</td></tr>
<tr><td>ELE1 Touch Threshold</td><td colspan="8">E1TTH</td><td>0x43</td><td>0x00</td></tr>
<tr><td>ELE1 Release Threshold</td><td colspan="8">E1RTH</td><td>0x44</td><td>0x00</td></tr>
<tr><td>ELE2 Touch Threshold</td><td colspan="8">E2TTH</td><td>0x45</td><td>0x00</td></tr>
<tr><td>ELE2 Release Threshold</td><td colspan="8">E2RTH</td><td>0x46</td><td>0x00</td></tr>
<tr><td>ELE3 Touch Threshold</td><td colspan="8">E3TTH</td><td>0x47</td><td>0x00</td></tr>
<tr><td>ELE3 Release Threshold</td><td colspan="8">E3RTH</td><td>0x48</td><td>0x00</td></tr>
<tr><td>ELE4 Touch Threshold</td><td colspan="8">E4TTH</td><td>0x49</td><td>0x00</td></tr>
<tr><td>ELE4 Release Threshold</td><td colspan="8">E4RTH</td><td>0x4A</td><td>0x00</td></tr>
<tr><td>ELE5 Touch Threshold</td><td colspan="8">E5TTH</td><td>0x4B</td><td>0x00</td></tr>
<tr><td>ELE5 Release Threshold</td><td colspan="8">E5RTH</td><td>0x4C</td><td>0x00</td></tr>
<tr><td>ELE6 Touch Threshold</td><td colspan="8">E6TTH</td><td>0x4D</td><td>0x00</td></tr>
<tr><td>ELE6 Release Threshold</td><td colspan="8">E6RTH</td><td>0x4E</td><td>0x00</td></tr>
<tr><td>ELE7 Touch Threshold</td><td colspan="8">E7TTH</td><td>0x4F</td><td>0x00</td></tr>
<tr><td>ELE7 Release Threshold</td><td colspan="8">E7RTH</td><td>0x50</td><td>0x00</td></tr>
<tr><td>ELE8 Touch Threshold</td><td colspan="8">E8TTH</td><td>0x51</td><td>0x00</td></tr>
<tr><td>ELE8 Release Threshold</td><td colspan="8">E8RTH</td><td>0x52</td><td>0x00</td></tr>
<tr><td>ELE9 Touch Threshold</td><td colspan="8">E9TTH</td><td>0x53</td><td>0x00</td></tr>
<tr><td>ELE9 Release Threshold</td><td colspan="8">E9RTH</td><td>0x54</td><td>0x00</td></tr>
<tr><td>ELE10 Touch Threshold</td><td colspan="8">E10TTH</td><td>0x55</td><td>0x00</td></tr>
<tr><td>ELE10 Release Threshold</td><td colspan="8">E10RTH</td><td>0x56</td><td>0x00</td></tr>
<tr><td>ELE11 Touch Threshold</td><td colspan="8">E11TTH</td><td>0x57</td><td>0x00</td></tr>
<tr><td>ELE11 Release Threshold</td><td colspan="8">E11RTH</td><td>0x58</td><td>0x00</td></tr>
<tr><td>ELEPROX Touch Threshold</td><td colspan="8">EPROXTTH</td><td>0x59</td><td>0x00</td></tr>
<tr><td>ELEPROX Release Threshold</td><td colspan="8">EPROXRTH</td><td>0x5A</td><td>0x00</td></tr>
<tr><td>Debounce Touch & Release</td><td></td><td colspan="3">DR</td><td></td><td colspan="3">DT</td><td>0x5B</td><td>0x00</td></tr>
<tr><td>Filter/Global CDC Configuration</td><td colspan="2">FFI</td><td colspan="6">CDC</td><td>0x5C</td><td>0x10</td></tr>
<tr><td>Filter/Global CDT Configuration</td><td colspan="3">CDT</td><td colspan="2">SFI</td><td colspan="3">ESI</td><td>0x5D</td><td>0x24</td></tr>
<tr><td>Electrode Configuration</td><td colspan="2">CL</td><td colspan="2">ELEPROX_EN</td><td colspan="4">ELE_EN</td><td>0x5E</td><td>0x00</td></tr>
<tr><td>ELE0 Electrode Current</td><td></td><td></td><td colspan="6">CDC0</td><td>0x5F</td><td>0x00</td></tr>
<tr><td>ELE1 Electrode Current</td><td></td><td></td><td colspan="6">CDC1</td><td>0x60</td><td>0x00</td></tr>
<tr><td>ELE2 Electrode Current</td><td></td><td></td><td colspan="6">CDC2</td><td>0x61</td><td>0x00</td></tr>
</table>

Table 2. Register Map

REGISTER	Fields								Register Address	Initial Value	Auto-Increment Address
ELE3 Electrode Current			CDC3						0x62	0x00	Register Address + 1
ELE4 Electrode Current			CDC4						0x63	0x00	
ELE5 Electrode Current			CDC5						0x64	0x00	
ELE6 Electrode Current			CDC6						0x65	0x00	
ELE7 Electrode Current			CDC7						0x66	0x00	
ELE8 Electrode Current			CDC8						0x67	0x00	
ELE9 Electrode Current			CDC9						0x68	0x00	
ELE10 Electrode Current			CDC10						0x69	0x00	
ELE11 Electrode Current			CDC11						0x6A	0x00	
ELEPROX Electrode Current			CDCPROX						0x6B	0x00	
ELE0, ELE1 Charge Time		CDT1				CDT0			0x6C	0x00	
ELE2, ELE3 Charge Time		CDT3				CDT2			0x6D	0x00	
ELE4, ELE5 Charge Time		CDT5				CDT4			0x6E	0x00	
ELE6, ELE7 Charge Time		CDT7				CDT6			0x6F	0x00	
ELE8, ELE9 Charge Time		CDT9				CDT8			0x70	0x00	
ELE10, ELE11 Charge Time		CDT11				CDT10			0x71	0x00	
ELEPROX Charge Time						CDTPROX			0x72	0x00	
GPIO Control Register 0	CTL011	CTL010	CTL09	CTL08	CTL07	CTL06	CTL05	CTL04	0x73	0x00	
GPIO Control Register 1	CTL111	CTL110	CTL19	CTL18	CTL17	CTL16	CTL15	CTL14	0x74	0x00	
GPIO Data Register	DAT11	DAT10	DAT9	DAT8	DAT7	DAT6	DAT5	DAT4	30x75	0x00	
GPIO Direction Register	DIR11	DIR10	DIR9	DIR8	DIR7	DIR6	DIR5	DIR4	0x76	0x00	
GPIO Enable Register	EN11	EN10	EN9	EN8	EN7	EN6	EN5	EN4	0x77	0x00	
GPIO Data Set Register	SET11	SET10	SET9	SET8	SET7	SET6	SET5	SET4	0x78	0x00	
GPIO Data Clear Register	CLR11	CLR10	CLR9	CLR8	CLR7	CLR6	CLR5	CLR4	0x79	0x00	
GPIO Data Toggle Register	TOG11	TOG10	TOG9	TOG8	TOG7	TOG6	TOG5	TOG4	0x7A	0x00	
AUTO-CONFIG Control Register 0	FFI		RETRY		BVA		ARE	ACE	0x7B	0x00	
AUTO-CONFIG Control Register 1	SCTS					OORIE	ARFIE	ACFIE	0x7C	0x00	
AUTO-CONFIG USL Register	USL								0x7D	0x00	
AUTO-CONFIG LSL Register	LSL								0x7E	0x00	
AUTO-CONFIG Target Level Register	TL								0x7F	0x00	0x00
Soft Reset Register	SRST								0x80		

4 Electrical Characteristics

4.1 Absolute Maximum Ratings

Absolute maximum ratings are stress ratings only and functional operation at the maxima is not guaranteed. Stress beyond the limits specified in Table 3 may affect device reliability or cause permanent damage to the device. For functional operating conditions, refer to the remaining tables in this section. This device contains circuitry protecting against damage due to high-static voltage or electrical fields; however, it is advised that normal precautions be taken to avoid application of any voltages higher than maximum-rated voltages to this high-impedance circuit.

Table 3. Absolute Maximum Ratings - Voltage (with respect to V_{SS})

Rating	Symbol	Value	Unit
Supply Voltage	V_{DD}	-0.3 to +3.6	V
Supply Voltage	V_{REG}	-0.3 to +2.75	V
Input Voltage SCL, SDA, $\overline{IRQ}$	V_{IN}	V_{SS} - 0.3 to V_{DD} + 0.3	V
Operating Temperature Range	T_O	-40 to +85	°C
GPIO Source Current per Pin	i_{GPIO}	12	mA
GPIO Sink Current per Pin	i_{GPIO}	1.2	mA
Storage Temperature Range	T_S	-40 to +125	°C

4.2 ESD and Latch-up Protection Characteristics

Normal handling precautions should be used to avoid exposure to static discharge.

Qualification tests are performed to ensure that these devices can withstand exposure to reasonable levels of static without suffering any permanent damage. During the device qualification, ESD stresses were performed for the Human Body Model (HBM), the Machine Model (MM) and the Charge Device Model (CDM).

A device is defined as a failure if after exposure to ESD pulses, the device no longer meets the device specification. Complete DC parametric and functional testing is performed per the applicable device specification at room temperature followed by hot temperature, unless specified otherwise in the device specification.

Table 4. ESD and Latch-up Test Conditions

Rating	Symbol	Value	Unit
Human Body Model (HBM)	V_{ESD}	±2000	V
Machine Model (MM)	V_{ESD}	±200	V
Charge Device Model (CDM)	V_{ESD}	±500	V
Latch-up current at T_A = 85°C	I_{LATCH}	±100	mA

4.3 DC Characteristics

This section includes information about power supply requirements and I/O pin characteristics.

Table 5. DC Characteristics

(Typical Operating Circuit, V_{DD} and V_{REG} = 1.8V, T_A = 25°C, unless otherwise noted.)

Parameter	Symbol	Conditions	Min	Typ	Max	Units
High Supply Voltage	V_{DD}		2.0	3.3	3.6	V
Low Supply Voltage	V_{REG}		1.71	1.8	2.75	V
Average Supply Current(1)	I_{DD}	Run Mode @ 1 ms sample period		393		µA
		Run Mode @ 2 ms sample period		199		µA
		Run Mode @ 4 ms sample period		102		µA
		Run Mode @ 8 ms sample period		54		µA
		Run Mode @ 16 ms sample period		29		µA
		Run Mode @ 32 ms sample period		17		µA
		Run Mode @ 64 ms sample period		11		µA
		Run Mode @ 128 ms sample period		8		µA
Measurement Supply Current	I_{DD}	Peak of measurement duty cycle		1		mA
Idle Supply Current	I_{DD}	Stop Mode		3		µA
Input Leakage Current ELE_	I_{IH}, I_{IL}			0.025		µA
Input Self-Capacitance on ELE_					15	pF
Input High Voltage SDA, SCL	V_{IH}		0.7 x V_{DD}			V
Input Low Voltage SDA, SCL	V_{IL}				0.3 x V_{DD}	V
Input Leakage Current SDA, SCL	I_{IH}, I_{IL}			0.025	1	µA
Input Capacitance SDA, SCL					7	pF
Output Low Voltage SDA, $\overline{IRQ}$	V_{OL}	I_{OL} = 6mA			0.5V	V
Output High Voltage ELE4 - ELE11 (GPIO mode)	V_{OHGPIO}	V_{DD} = 2.7V to 3.6V: I_{OHGPIO} = -10 mA V_{DD} = 2.3V to 2.7V: I_{OHGPIO} = -6 mA V_{DD} = 1.8V to 2.3V: I_{OHGPIO} = -3 mA	V_{DD} - 0.5			V
Output Low Voltage ELE4 - ELE11 (GPIO mode)	V_{OLGPIO}	$I_{OLGPIOD}$ = 1 mA			0.5	V
Power On Reset	V_{TLH}	V_{DD} rising	1.08	1.35	1.62	V
	V_{THL}	V_{DD} falling	0.88	1.15	1.42	V

1.: ECR set to 0x2C and all 12 channels plus one proximity channel activated. Measurement current CDC is set at maximum of 0x3F.

4.4 AC Characteristics

Table 6. AC Characteristics

(Typical Operating Circuit, V_{DD} and V_{REG} = 1.8V, T_A = 25°C, unless otherwise noted.)

Parameter	Symbol	Conditions	Min	Typ	Max	Units
8 MHz Internal Oscillator	f_H		7.44	8	8.56	MHz
1 kHz Internal Oscillator	f_L		0.65	1	1.35	kHz

4.5 I²C AC Characteristics

Table 7. I²C AC Characteristics

(Typical Operating Circuit, V_{DD} and V_{REG} = 1.8V, T_A = 25°C, unless otherwise noted.)

Parameter	Symbol	Conditions	Min	Typ	Max	Units
Serial Clock Frequency	f_{SCL}				400	kHz
Bus Free Time Between a STOP and a START Condition	t_{BUF}		1.3			µs
Hold Time, (Repeated) START Condition	$t_{HD, STA}$		0.6			µs
Repeated START Condition Setup Time	$t_{SU, STA}$		0.6			µs
STOP Condition Setup Time	$t_{SU, STO}$		0.6			µs
Data Hold Time	$t_{HD, DAT}$				0.9	µs
Data Setup Time	$t_{SU, DAT}$		100			ns
SCL Clock Low Period	t_{LOW}		1.3			µs
SCL Clock High Period	t_{HIGH}		0.7			µs
Rise Time of Both SDA and SCL Signals, Receiving	t_R			$20+0.1C_b$	300	ns
Fall Time of Both SDA and SCL Signals, Receiving	t_F			$20+0.1C_b$	300	ns
Fall Time of SDA Transmitting	$t_{F.TX}$			$20+0.1C_b$	250	ns
Pulse Width of Spike Suppressed	t_{SP}			25		ns
Capacitive Load for Each Bus Line	C_b				400	pF

5 Register Operation Descriptions

5.1 Register Read/Write Operations and Measurement Run/Stop Mode

After power on reset (POR) or soft reset by command, all registers are in reset default initial value (see Table 2). All the registers, except registers 0x5C (default 0x10) and 0x5D (default 0x24), are cleared.

Registers 0x2B ~ 0x7F are control and configuration registers which need to be correctly configured before any capacitance measurement and touch detection.

Registers 0x00 ~ 0x2A are output registers updating periodically by the MPR121 in Run Mode. Among these output registers, Baseline Value Registers 0x1D ~ 0x2A are also writable, this is sometimes useful when user specific baseline values are desired.

The MPR121's Run Mode and Stop Mode are controlled by control bits in Electrode Configuration Register (ECR, 0x5E). When all ELEPROX_EN and ELE_EN bits are zeros, the MPR121 is in Stop Mode. While in Stop Mode, there are no capacitance or touch detection measurement on any of the 13 channels. When any of the ELEPROX_EN and ELE_EN bits are set to '1', the MPR121 is in Run Mode. The MPR121 will continue to run on its own until it is set again to Stop Mode by the user.

The MPR121 registers read operation can be done at any time, either in Run Mode or in Stop Mode. However, the register write operation can only be done in Stop Mode. The ECR (0x5E) and GPIO/LED control registers (0x73~0x7A) can be written at anytime.

5.2 Touch Status Registers (0x00~0x01)

ELE0-ELE7 Touch Status (0x00)

Bit	D7	D6	D5	D4	D3	D2	D1	D0
Read	ELE7	ELE6	ELE5	ELE4	ELE3	ELE4	ELE1	ELE0
Write	—	—	—	—	—	—	—	—

ELE8-ELE11 ELEPROX Touch Status (0x01)

Bit	D7	D6	D5	D4	D3	D2	D1	D0
Read	OVCF	—	—	ELEPROX	ELE11	ELE10	ELE9	ELE8
Write	—	—	—	—	—	—	—	—

These two registers indicate the detected touch/release status of all of the 13 sensing input channels. ELEPROX is the status for the 13^{th} proximity detection channel. The update rate of these status bits will be {ESI x SFI}.

ELEx, ELEPROX: Touch or Release status bit of each respective channel (read only).

1, the respective channel is currently deemed as touched.

0, the respective channel is deemed as released.

Note: When an input is not configured as an electrode and enabled as GPIO input port, the corresponding status bit shows the input level, but these GPIO status changes will not cause any IRQ interrupt. This feature is for ELE4~ELE11 only.

OVCF: Over Current Flag (read and write)

1, over current was detected on REXT pin.

0, normal condition.

When over current is detected, the OVCF is set to '1' and the MPR121 goes to Stop Mode. All other bits in status registers 0x00~0x03, output registers 0x04~0x2A, and bits D5~D0 in ECR (0x5E) will also be cleared. When the bit is set at '1', the write to the ECR register to enter Run Mode will be discarded. The write to '1'of the OVCF will clear this bit and the MPR121 fault condition will be cleared. The MPR121 can then be configured to return to the Run Mode again.

5.3 Electrode Filtered Data Register (0x04~0x1D)

Electrode Filtered Data Low Byte (0x04,0x06,...,0x1C)

Bit	D7	D6	D5	D4	D3	D2	D1	D0
Read	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Write	—	—	—	—	—	—	—	—

Electrode Filtered Data High Byte (0x05,0x07,...,0x1D)

Bit	D7	D6	D5	D4	D3	D2	D1	D0
Read	—	—	—	—	—	—	Bit 9	Bit 8
Write	—	—	—	—	—	—	—	—

The MPR121 provides filtered electrode output data for all 13 channels. The output data is 10-bit and comes from the internal 2nd stage filter output. The data range is 0~1024 or 0x000~0x400 in Hex. Bit 0~7 of the 10-bit data are stored in the low byte and bit 9 and bit 8 are stored in the high byte. The data is the measured voltage on each channel and inversely proportional to the capacitance on that channel.

These registers are read only and are updated every {ESI x SFI}. A multibyte read operation to read both LSB and MSB is recommended to keep the data coherency (i.e, LSB and MSB matching). A multibyte reading of 0x00~0x2A returns results of a single moment without mixing up old and new data.

5.4 Baseline Value Register (0x1E~0x2A)

Electrode Baseline Value (0x1E~0x2A)

Bit	D7	D6	D5	D4	D3	D2	D1	D0
Read / Write	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2

Along with the 10-bit electrode filtered data output, each channel also has a 10-bit baseline value. The update rate of these registers is {ESI x SFI} if baseline tracking operation is enabled. These values are the output of the internal baseline filter operation tracking the slow-voltage variation of the background capacitance change. Touch/release detection is made based on the comparison between the 10-bit electrode filtered data and the 10-bit baseline value.

Note: Although internally the baseline value is 10-bit, users can only access the 8 MSB of the 10-bit baseline value through the baseline value registers. The read out from the baseline register must be left shift two bits before comparing it with the 10-bit electrode data.

The Baseline Value register is writable in Stop Mode. Note: when the user writes into the baseline value register, the lower two bits of the 10-bit baseline value are automatically cleared internally upon write operation. The Write to Baseline Value Register by specific values can be sometimes useful if user wants to manipulate the touch/release status. For example, manually setting the target channel from a touch locked state into a touch released state is easily done by setting the baseline value above the signal data.

Refer to the Electrode Configuration Register (ECR, 0x5E) on how to control the on/off operation of baseline tracking and further details on how the initial baseline data is loaded into Run Mode. Refer to Baseline Filtering Control registers(0x2B~0x2A) on how to control the filtering of the baseline value.

5.5 Baseline Filtering Control Register (0x2B~0x40)

All12 of the electrode baseline values are controlled by the same set of filtering control registers, 0x2B ~ 0x35. The 13th channel ELEPROX is controlled by registers 0x36 ~ 0x40. Both sets of registers have the same structure using three different scenarios; rising, falling, and touched.

Rising is defined as when the electrode data is greater than the baseline value. Falling is defined as when the electrode data is less than the baseline value. Touched is when the electrode is in touched status. For each scenario, the filtering characteristic is further defined by four parameters: the maximum half delta (MHD), noise half delta (NHD), noise count limit (NCL) and filter delay count limit (FDL). Note: there is no maximum half delta for the touched scenario.

Maximum Half Delta (MHD): Determines the largest magnitude of variation to pass through the baseline filter. The range of the effective value is 1~63.

Noise Half Delta (NHD): Determines the incremental change when non-noise drift is detected. The range of the effective value is 1~63.

Noise Count Limit (NCL): Determines the number of samples consecutively greater than the Max Half Delta value. This is necessary to determine that it is not noise. The range of the effective value is 0~255.

Filter Delay Count Limit (FDL): Determines the operation rate of the filter. A larger count limit means the filter delay is operating more slowly. The range of the effective value is 0~255.

The setting of the filter is depended on the actual application. For more information on these registers, refer to application note AN3891.

5.6 Touch / Release Threshold (0x41~0x5A)

ELEx, ELEProx Touch Threshold (0x41,0x43,...,0x59)

Bit	D7	D6	D5	D4	D3	D2	D1	D0
Read	ExTTH							
Write								

ELEx, ELEProx Release Threshold (0x42,0x44,...,0x5A)

Bit	D7	D6	D5	D4	D3	D2	D1	D0
Read	ExRTH							
Write								

ExTTH: Electrode touch threshold, in range of 0~0xFF.

ExRTH: Electrode release threshold, in range of 0~0xFF.

Each of the 13 channels can be set with its own set of touch and release thresholds. Touch and release are detected by comparing the electrode filtered data to the baseline value. The amount of deviation from the baseline value represents a immediate capacitance change detected by possible a touch/release action.

Touch condition: Baseline - Electrode filtered data > Touch threshold

Release condition: Baseline - Electrode filtered data < Release threshold

Threshold settings are dependant on the touch/release signal strength, system sensitivity and noise immunity requirements. In a typical touch detection application, threshold is typically in the range 0x04~0x10. The touch threshold is several counts larger than the release threshold. This is to provide hysteresis and to prevent noise and jitter. For more information, refer to the application note AN3892 and the MPR121 design guidelines.

5.7 Debounce Register (0x5B)

Debounce Register (0x5B)

Bit	D7	D6	D5	D4	D3	D2	D1	D0
Read	—	DR			—	DT		
Write								

DT: Debounce number for touch. The value range is 0~7.

DR: Debounce number for release. The value range is 0~7.

All 13 channels use the same set of touch and release debounce numbers. The status bits in Status Register 0x00 and 0x01 will only take place after the number of consecutive touch or release detection meets the debounce number setting. The debounce setting can be very useful in avoiding possible noise glitches. Using the debounce setting, the status bit change will have a delay of {ESI x SFI x DR (or DT)}.

5.8 Filter and Global CDC CDT Configuration (0x5C, 0x5D)

Filter/Global CDC Configuration Register (0x5C)

Bit	D7	D6	D5	D4	D3	D2	D1	D0
Read	FFI		CDC					
Write								

Filter/Global CDT Configuration Register (0x5D)

Bit	D7	D6	D5	D4	D3	D2	D1	D0
Read	CDT			SFI		ESI		
Write								

Table 8. Bit Descriptions

Field	Description
FFI	First Filter Iterations - The first filter iterations field selects the number of samples taken as input to the first level of filtering.
	00 Encoding 0 - Sets samples taken to 6 (Default)
	01 Encoding 1 - Sets samples taken to 10
	10 Encoding 2 - Sets samples taken to 18
	11 Encoding 3 - Sets samples taken to 34
CDC	Charge Discharge Current - Selects the global value of charge discharge current applied to electrode. The maximum is 63 μA, 1 μA step.
	000000 Encoding 0 - Disable Electrode Charging
	000001 Encoding 1 - Sets the current to 1 μA
	~
	010000 Encoding 16 - Sets the current to 16 μA (Default)
	~
	111111 Encoding 63 - Sets the current to 63 μA
CDT	Charge Discharge Time - Selects the global value of charge time applied to electrode. The maximum is 32 μs, programmable as 2 ^(n-2) μs.
	000 Encoding 0 - Disables Electrode Charging
	001 Encoding 1 - Time is set to 0.5 μs (Default)
	010 Encoding 2 - Time is set to 1 μs
	~
	111 Encoding 7 - Time is set to 32 μs
SFI	Second Filter Iterations - Selects the number of samples taken for the second level filter
	00 Encoding 0 - Number of samples is set to 4 (Default)
	01 Encoding 1 - Number of samples is set to 6
	10 Encoding 2 - Number of samples is set to 10
	11 Encoding 3 - Number of samples is set to 18
ESI	Electrode Sample Interval - Selects the period between samples used for the second level of filtering. The maximum is 128ms, Programmable to 2^n ms
	000 Encoding 0 - Period set to 1 ms
	001 Encoding 1 - Period set to 2 ms
	~
	100 Encoding 4 - Period set to 16 ms (Default)
	~
	111 Encoding 7 - Period set to 128 ms

These two registers set the global AFE settings. This includes global electrode charge/discharge current CDC, global charge/discharge time CDT, as well as a common filtering setting (FFI, SFI, ESI) for all 13 channels, including the 13th Eleprox channel.

The register 0x5C holds the global CDC and the first level filter configuration for all 13 channels. For each enabled channel, the global CDC will be used for that channel if the respective charge discharge current CDCx setting in 0x5F~0x6B for that channel is zero. If it is not zero, the individual CDCx value will be used in place of the global CDC value. If the MPR121's auto-configuration feature is enabled, CDCx will be automatically set up during system start stage and used for the actual measurement.

The register 0x5D holds the global CDT and the second level filter configuration for all 13 channels. For each enabled channel, the global CDT will be used for that channel if the respective charge discharge time CDTx setting in 0x6C~0x72 for that channel is zero. If it is not zero, the individual CDTx value will be used in place of the global CDT value. If the SCTS bit (Skip Charge Time Search) in the MPR121's autoconfiguration is set, then the current global CDT and CDTx will be used for each channel measurements. If not, then the individual CDTx will be automatically set up during the system start stage and used for the actual measurement.

Using only the global CDC and/or global CDT is acceptable where the capacitance values from all 13 channels are similar. If the electrode pattern, size, or even overlay and base material type changes from one channel to another, then using individual CDCx (and CDTx) will have a better result on sensing sensitivity as each electrode is charged up to a point closing to the supply voltage rail so that the highest sensing field is built for each channel.

The settings for the FFI, SFI, and ESI must be selected according to the system design noise filtering requirement. These settings must also balance the need for power consumption and response time.

When the total time required by scanning and charging/discharge all the enabled channels is longer than the ESI setting, then the actual time will override the ESI setting. For example if the ESI = 4 (16 mS), when FFI = 3 (34 samples), CDT = 7 (32 µS), with all 13 channels enabled, the scan time needed is 34 x (32 µS + 32 µS) x 13 = 28 mS. This 28 mS will be the actual sampling interval instead of ESI (16 mS).

5.9 Electrode Charge Current Register (0x5F~0x6B)

Electrode Charge Current (0x5F~0x6B)

Bit	D7	D6	D5	D4	D3	D2	D1	D0
Read	—	—	CDCx					
Write	—	—						

CDCx: Sets the charge current applied to each channel. Similar to global CDC value, the range is 0~63 µA, from 0x00~0x3F in 1 µA step. When the CDCx is zero, the global CDC value will be used for that channel.

The individual CDCx bit can either be set manually or automatically (if autoconfiguration is enabled). When the autoconfiguration is enabled, during the first transition from Stop Mode to Run Mode, the system will automatically run a trial search for the appropriate CDCx (and CDTx if SCTS = 0). The individual CDCx will be automatically updated by the MPR121 into the respective registers once autoconfiguration is finished. CDCx is used in the following capacitance measurement and touch detection.

5.10 Electrode Charge Time Register (0x6C~0x72)

Electrode Charge Time (0x6C~0x72)

Bit	D7	D6	D5	D4	D3	D2	D1	D0
Read	—	CDTx+1			—	CDTx		
Write	—				—			

CDTx: Sets the charge time applied to each channel. Similar to the global CDT value, the range is 0~32 µS, from 2b000~2b111. When the CDTx is zero, the global CDT value is used for that channel.

The individual CDTx bit can be set manually or automatically (if autoconfiguration is enabled). When autoconfiguration is enabled, during the first transition from Stop Mode to Run Mode, the system will automatically run a trial search for the appropriate CDCx (and CDTx if SCST = 0). This means the autoconfiguration will include a search on the CDTx. The individual CDTx will be automatically updated by the MPR121 into the respective registers once the autoconfiguration is finished. This data is used in the following capacitance measurement and touch detection. If SCTS bit is 1, the search on CDTx will be skipped.

5.11 Electrode Configuration Register (ECR, 0x5E)

Electrode Configuration Register (0x5E)

Bit	D7	D6	D5	D4	D3	D2	D1	D0
Read	CL		ELEPROX_EN		ELE_EN			
Write								

Table 9. Bit Descriptions

Field	Description
CL	Calibration Lock - Controls the baseline tracking and how the baseline initial value is loaded
	00 - Baseline tracking enabled, initial baseline value is current value in baseline value register (Default)
	01 - Baseline tracking is disabled
	10 - Baseline tracking enabled, initial baseline value is loaded with the 5 high bits of the first 10-bit electrode data value
	11 - Baseline tracking enabled, initial baseline value is loaded with all 10 bits of the first electrode data value
ELEPROX_EN	Proximity Enable - Controls the operation of 13th Proximity Detection
	00 - Proximity Detection is disabled (Default)
	01 - Run Mode with ELE0~ELE1 combined for proximity detection enabled
	10 - Run Mode with ELE0~ELE3 combined for proximity detection enabled
	11 - Run Mode with ELE0~ELE11combined for proximity detection enabled
ELE_EN	Electrode Enable - Controls the operation of 12 electrodes detection
	0000 - Electrode detection is disabled (Default)
	0001 - Run Mode with ELE0 for electrode detection enabled
	0010 - Run Mode with ELE0~ ELE1 for electrode detection enabled
	0011 - Run Mode with ELE0~ ELE2 for electrode detection enabled
	0100 - Run Mode with ELE0~ ELE3 for electrode detection enabled
	0101 - Run Mode with ELE0~ ELE4 for electrode detection enabled
	0110 - Run Mode with ELE0~ ELE5 for electrode detection enabled
	0111 - Run Mode with ELE0~ ELE6 for electrode detection enabled
	1000 - Run Mode with ELE0~ ELE7 for electrode detection enabled
	1001 - Run Mode with ELE0~ ELE8 for electrode detection enabled
	1010 - Run Mode with ELE0~ ELE9 for electrode detection enabled
	1011 - Run Mode with ELE0~ ELE10 for electrode detection enabled
	11xx - Run Mode with ELE0~ ELE11 for electrode detection enabled

The Electrode Configuration Register (ECR) determines if the MPR121 is in Run Mode or Stop Mode, controls the baseline tracking operation and specifies the input configurations of the 13 channels.

The ECR reset default value is 0x00, which means MPR121 is in Stop Mode without capacitance measurement on all 13 channels. Setting ELEPROX_EN and/or ELE_EN control bits to non-zero data will put the MPR121 into Run Mode. This will cause the MPR121 to operate immediately on its own. Clearing the ELEPROX_EN and ELE_EN all to zeros will set the MPR121 into Stop Mode (which is its lowest power state). The MPR121 can be switched between Stop Mode and Run Mode at anytime by configuring the ECR.

If all channels including the13th proximity detection channel are enabled, the proximity sensing channel is scanned first, followed by ELE0, ELE1..., and ELE11 respectively. The scan runs periodically at the sampling rate specified by the ESI in the Filter/CDT Configuration Register (0x5D). Refer to the table above for configuration of the different channels. Enabling specific channels will save the scan time and sensing field power spent on the unused channels.

In a typical touch detection application, baseline tracking is enabled. This is to compensate for the environment and background induced slow capacitance change to the input sensing channels. The CL bits can enable/disable the baseline tracking and specify how to load the baseline initial values. Since the baseline tracking filtering system has a very large time constant and the initial baseline value starts from zero, it will require a very long time for the baseline to ramp up. This results in a short period of no response to touch after the MPR121 is first set to Run Mode. Setting the CL = 2b10 will command the MPR121 to load the initial baseline value at the beginning of the Run Mode. This shortens the initial baseline ramp-up time so that user will not notice any delay on touch detection. The MPR121 uses the five high bits of the first measured 10 bit electrode data.

Auto-Configuration Registers (0x7B~0x7F)

For each enabled channel, both the charge time and charge current must be set properly. This is so that a specified amount of charge field can be built on the sensing pad and that the capacitance can be measured using the internal ADC. When all 13 channels are used, there are total 13 CDCx and 13 CDTx values which need to be configured.

The MPR121 provides an auto-configuration function which is able to automatically search and set the charging parameters. When autoconfiguration is run, specific CDCx and CDTx combinations for the enabled channels can be obtained automatically. This eliminates test trials on the prototype device and for further verification on final products. A key task for the design engineer is to verify if the parameter settings generated by the MPR121 are acceptable. This verification ensures that the settings are optimized each time MPR121 powers on and that the equipment can operate in many different environments.

The autoconfiguration finds the optimized CDCx and CDTx combination for each channel so that the charge level (I x T = V) on the each channel is as close as possible to the target setting specified by the designer. An upper and lower setting limit are used to provide the boundaries necessary to verify if the system is setup to operate correctly. If the autoconfiguration can not find the proper CDCx and CDTx value, an Out Of Range (OOR) status will be set for that channel.

Autoconfiguration operates each time the MPR121 transitions from Stop Mode to Run Mode. After autoconfiguration is completed, a set of CDCx and CDTx values for each channel are calculated and automatically loaded into the corresponding register fields.

If autoconfiguration fails, the MPR121 has an auto-reconfiguration function. Autoreconfiguration runs at each sampling interval if a channel has OOR status from a failed autoconfiguration. Autoreconfiguration will run until the OOR status is cleared or until it is disabled.

There are five registers used to control the MPR121 auto-configuration feature. Registers 0x7B and 0x7C are used as the control registers and registers 0x07D to 0x7F are used to hold the configuration target settings. Refer to application note AN3889 for more information.

Auto-Configure Control Register 0 (0x7B)

Bit	D7	D6	D5	D4	D3	D2	D1	D0
Read	FFI		RETRY		BVA		ARE	ACE
Write								

Auto-Configure Control Register 1 (0x7C)

Bit	D7	D6	D5	D4	D3	D2	D1	D0
Read	SCTS	—	—	—	—	OORIE	ARFIE	ACFIE
Write		—	—	—	—			

FFI: The FFI bits are the same as the FFI bits in register 0x5C for correct auto-configuration and reconfiguration operations.

ACE: Auto-Configuration Enable. 1: Enable, 0: Disable. When Enabled, the autoconfiguration will operate once at the beginning of the transition from Stop Mode to Run Mode. This includes search and update of the CDCx and CDTx for each enabled channel (if SCTS = 0).

ARE: Auto-Reconfiguration Enable. 1: Enable, 0: Disable. When enabled, if the OOR is set for a channel after autoconfiguration, autoreconfiguration will operate on that channel on each sampling interval until the OOR is cleared.

BVA: Fill the BVA bits same as the CL bits in ECR (0x5E) register.

RETRY: Specifies the number of retries for autoconfiguration and autoreconfiguration if the configuration fails before setting OOR.

00 - No retry
01 - retry 2 times
10 - retry 4 times
11 - retry 8 times

SCTS: Skip Charge Time Search.

1: Skip CDTx search and update when autoconfiguration or autoreconfiguration, and current global CDT or CDTx are used for respective channels. CDT or CDTx needs to be specified by the designer manually before operation. Setting the SCTS to “1” results in a shorter time to complete autoconfiguration. This is useful for when the designer has obtained the correct CDTx / CDT, and is confident that the current CDT and CDTx settings work in all conditions.

0: Both CDTx and CDCx will be searched and set by autoconfiguration and/or autoreconfiguration.

ACFIE: Auto-configuration fail interrupt enable. 1: Enable, 0: Disable

ARFIE: Auto-reconfiguration fail interrupt enable. 1: Enable, 0: Disable

OORIE: Out-of-range interrupt enable. 1: Enable, 0: Disable

Up-Side Limit Register (0x7D)

Bit	D7	D6	D5	D4	D3	D2	D1	D0
Read	USL							
Write								

Low-Side Limit Register (0x7E)

Bit	D7	D6	D5	D4	D3	D2	D1	D0
Read	LSL							
Write								

Target Level Register (0x7F)

Bit	D7	D6	D5	D4	D3	D2	D1	D0
Read	TL							
Write								

USL: Up-Side Limit. This value sets the electrode data level up limit for the boundary check in autoconfiguration and autoreconfiguration operation.

LSL: Low-Side Limit. This value sets the electrode data level low limit for the boundary check in autoconfiguration and autoreconfiguration operation.

TL: Target Level. This value is the expected target electrode data level for autoconfiguration and autoreconfiguration, that is, after successful autoconfiguration and autoreconfiguration, the measured electrode data level when untouched shall be close to the TL value. TL shall be in between of USL and LSL.

The three parameters, USL, LSL and TL, are in the format similar to the baseline value; only the eight high bits are accessible by user and the two low bits are set to zero automatically. The USL/LSL/TL data needs to be shifted left two bits before comparing with the electrode data or the 10-bit baseline value.

In order to have a valid auto-configuration result, USL/LSL/TL values should follow the relation that 255 > USL > TL > LSL > 0. For example, USL = 200, TL = USL*0.9 = 180, LSL = USL*0.5 = 100.

It is possible that in a end user environment, the channel differences may be significant. This is because the same set of USL/LS/TL data is being used for all channels. It is important that the parameters not be set too close together. This makes it difficult for the autoconfiguration to find a suitable charge setting for a specific channel. In this case, the electrode data might easily go out of USL and LSL setting limits. Since the data is out-of-range, the channel status becomes OOR. If the channel is still OOR after the autoconfiguration has been run, it may indicate that the settings for this channel have not yet been optimized. One solution to this problem is to manually review the USL/LSL/TL settings. Another possible reason why the channel status could be OOR is a problem with the channel itself. This could be caused by a short to ground, short to the power rail, or short to the pad of the other channel.

For the TL setting, a good practice is to try to set it close to the USL. This so the charge field can be set to detect a weak touch. On the other hand, the TL should not be set too close to the USL so that it is constantly exceeding the limit. For example, the electrode data from the end user’s environment might have a much wider variance of readings. Some of the readings might exceed the USL, causing the auto-configuration to fail. For this reason, if the amount of capacitance change in the end user environment is significant, it is suggested that the USL and TL be set low enough to give some headroom for possible capacitance variations.

With above mentioned, one possible example setting is given out below using equation 1~3, with the assumption that setting TL at 90% of USL, and LSL at 65% of USL would cover most of the application case. It may need further adjustment in some cases but will be a very good start.

$$\text{USL} = (\text{VDD} - 0.7)/\text{VDD} \times 256 \qquad \text{Eqn. 1}$$

$$\text{TL} = \text{USL} \times 0.9 = (\text{VDD} - 0.7)/\text{VDD} \times 256 \times 0.9 \qquad \text{Eqn. 2}$$

$$\text{LSL} = \text{USL} \times 0.65 = (\text{VDD}-0.7) / \text{VDD} \times 256 \times 0.65 \qquad \text{Eqn. 3}$$

$$\text{Cin} = \text{I} \times \text{T} / \text{V} = \text{CDC} \times \text{CDT} / (\text{ADC counts} \times \text{VDD}/1024) \qquad \text{Eqn. 4}$$

It may not necessary to set the USL at the level of VDD - 0.7 but it is beneficial to keep the applied constant charge current as accurate as that specified in the data sheet. This so the capacitance value on the input can be calculated with high accuracy using ADC conversion Equation 4. Using VDD-0.7 as USL level allows some headroom for applications where the supply varies over a certain range. For a system where the supply changes over a range, the lowest VDD point is considered for autoconfiguration so that a relative lower charge field can be used to avoid clipping the electrode data to VDD when it drops.

5.12 Out-Of-Range Status Registers (0x02, 0x03)

ELE0~ELE7 OOR Status (0x02)

Bit	D7	D6	D5	D4	D3	D2	D1	D0
Read	E7_OOR	E6_OOR	E5_OOR	E4_OOR	E3_OOR	E2_OOR	E1_OOR	E0_OOR
Write	—	—	—	—	—	—	—	—

ELE8~ELEPROX OOR Status (0x03)

Bit	D7	D6	D5	D4	D3	D2	D1	D0
Read	ACFF	ARFF	—	EPROX_OOR	E11_OOR	E10_OOR	E9_OOR	E8_OOR
Write	—	—	—	—	—	—	—	—

Ex_OOR, EPROX_OOR: Out-Of-Range Status bits for the 13 channels. This bit set indicates that a corresponding channel has failed autoconfiguration and autoreconfiguration for range check. Those bits are cleared when they pass the auto-configuration and auto-reconfiguration range check. These bits are user read only.

ACFF: Auto-Configuration Fail Flag. When autoconfiguration fails, this bit is set. This bit is user read only.

ARFF: Auto-Reconfiguration Fail Flag. When autoreconfiguration fails, this is bit set. This bit is user read only.

When autoconfiguration and/or autoreconfiguration are enabled, MPR121 checks the electrode data after each auto-configuration, auto-reconfiguration operation to see if it is still in the range set by USL and LSL. When electrode data goes out of the range, corresponding Ex_OORx bit becomes “1” to indicate the failed channels. One example of triggering OOR error is shorting the measurement sensing pad to power rails, or shorting it with other channels.

5.13 Soft Rest Register (0x80)

Write 0x80 with 0x63 asserts soft reset. The soft reset does not effect the I^2C module, but all others reset the same as POR.

5.14 GPIO Registers (0x73~0x7A)

GPIO Registers (0x73~0x7A)

GPIO Registers	D7	D6	D5	D4	D3	D2	D1	D0
Control Register 0(0x73)	GTL0_E11	GTL0_E10	GTL0_E9	GTL0_E8	GTL0_E7	GTL0_E6	GTL0_E5	GTL0_E4
Control Register 1(0x74)	GTL1_E11	GTL1_E10	GTL1_E9	GTL1_E8	GTL1_E7	GTL1_E6	GTL1_E5	GTL1_E4
Data Register(0x75)	DAT_E11	DAT_E10	DAT_E9	DAT_E8	DAT_E7	DAT_E6	DAT_E5	DAT_E4

GPIO Registers (0x73~0x7A)

Direction Register(0x76)	DIR_E11	DIR_E10	DIR_E9	DIR_E8	DIR_E7	DIR_E6	DIR_E5	DIR_E4
Enable Register(0x77)	EN_E11	EN_E10	EN_E9	EN_E8	EN_E7	EN_E6	EN_E5	EN_E4
Data Set Register(0x78)	SET_E11	SET_E10	SET_E9	SET_E8	SET_E7	SET_E6	SET_E5	SET_E4
Data Clear Register(0x79)	CLR_E11	CLR_E10	CLR_E9	CLR_E8	CLR_E7	CLR_E6	CLR_E5	CLR_E4
Data Toggle Register(0x7A)	TOG_E11	TOG_E10	TOG_E11	TOG_E8	TOG_E7	TOG_E6	TOG_E5	TOG_E4

These registers control GPIO and LED driver functions. D7~D0 bits correspond to GPIO and LED functions on ELE11~ ELE4 inputs respectively. When any of these ports are not used for electrode sensing, it can be used for GPIO or LED driver. The GPIO control registers can be write at anytime regardless Stop Mode or Run mode. The configuration of the LED driver and GPIO system is described with more detail in application note AN3894.

Note: The number of touch sensing electrodes, and therefore the number of GPIO ports left available is configured by the ECR (0x5E) and GPIO Enable Register (0x77). ECR has higher priority and overrides the GPIO enabled in 0x77, that is when a pin is enabled as GPIO but is also selected as electrode by ECR, the GPIO function is disabled immediately and it becomes an electrode during Run Mode.

In the Stop Mode just after power-on reset, all electrodes and GPIO ports are in high impedance as all the GPIO ports are default disabled and the electrodes are not enabled.

EN, DIR, CTL0, CTL1: GPIO enable and configuration bits, the functions are in description table below.

EN	DIR	CTL0:CTL1	Function Description
0	X	XX	GPIO function is disabled. Port is high-z state.
1	0	00	GPIO port becomes input port.
1	0	10	GPIO port becomes input port with internal pulldown.
1	0	11	GPIO port becomes input port with internal pullup.
1	0	01	Not defined yet (as same as CTL = 00).
1	1	00	GPIO port becomes CMOS output port.
1	1	11	GPIO port becomes high side only open drain output port for LED driver.
1	1	10	GPIO port becomes low side only open drain output port.
1	1	01	Not defined yet (as same as CTL = 00).

When the EN bit is set, the corresponding GPIO pin is enabled and the GPIO function is configured by CTL0, CTL1 and DIR bits. When the port is used as an input, it can be configured as a normal logic input with high impedance (CTL0CTL1 = 2b00), input with internal pull-down (CTL0CTL1 = 2b10) or pullup (CTL0CTL1 = 2b11). Note: the former may result in an unstable logic input state if opened without fixed logic level input.

The GPIO output configuration can be configured as either push pull (CTL0CTL1 = 2b00) or open drain. When the GPIO is used for LED drivers, the GPIO is set to high side only open drain (CTL0CTL1 = 2b11), which is can source up to 12 mA current into the LED.

DAT: GPIO Data Register bits.

When a GPIO is enabled as an output, the GPIO port outputs the corresponding DAT bit level from GPIO Data Register (0x075). The output level toggle remains on during any electrode charging. The level transition will occur after the ADC conversion takes place. It is important to note that reading this register returns the content of the GPIO Data Register, (not a level of the port). When a GPIO is configured as input, reading this register returns the latched input level of the corresponding port (not contents of the GPIO Data Register). Writing to the DAT changes content of the register, but does not effect the input function.

SET: Writing a “1” to this bit will set the corresponding bit in the Data Register.

CLR: Writing a “1” to this bit will clear the corresponding bit in the Data Register.

TOG: Writing a “1” to this bit will toggle the corresponding bit in the Data Register

Writing “1” into the corresponding bits of GPIO Data Set Register, GPIO Data Clear Register, and GPIO Data Toggle Register will set/clear/toggle contents of the corresponding DAT bit in Data Register. Writing “0” has no meaning. These registers allow any individual port(s) to be set, cleared, or toggled individually without effecting other ports. It is important to note that reading these registers returns the contents of the GPIO Data Register reading.

6 MPR121 Serial Communication

6.1 I²C Serial Communications

The MPR121 uses an I^2C Serial Interface.The MPR121 operates as a slave that sends and receives data through an I^2C two-wire interface. The interface uses a Serial Data Line (SDA) and a Serial Clock Line (SCL) to achieve bidirectional communication between master(s) and slave(s). A master (typically a microcontroller) initiates all data transfers to and from the MPR121, and it generates the SCL clock that synchronizes the data transfer.

The MPR121 SDA line operates as both an input and an open-drain output. A pullup resistor, typically 4.7 kΩ, is required on SDA. The MPR121 SCL line operates only as an input. A pullup resistor, typically 4.7 kΩ, is required on SCL if there are multiple masters on the two-wire interface, or if the master in a single-master system has an open-drain SCL output.

Each transmission consists of a START condition (Figure 3) sent by a master, followed by the MPR121's 7-bit slave address plus $R/\overline{W}$ bit, a register address byte, one or more data bytes, and finally a STOP condition.

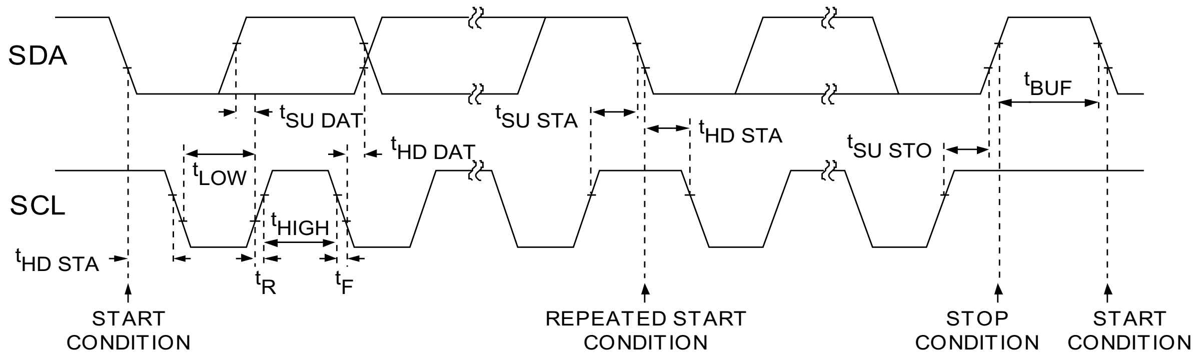


Figure 3. Two-Wire Serial Interface Timing Details

6.2 Slave Address

The MPR121 has selectable slave addresses listed by different ADDR pin connections. This also makes it possible for multiple MPR121 devices to be used together for channel expansions in a single system.

Table 10. MPR121 Slave Address

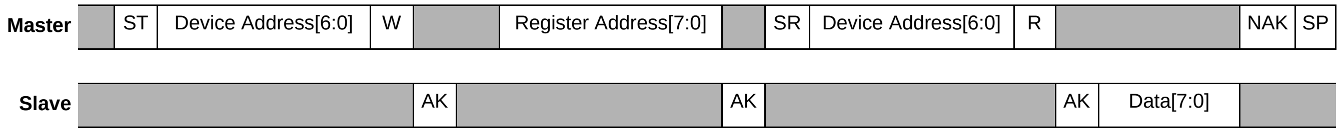
ADDR Pin Connection	I^2C Address
VSS	0x5A
VDD	0x5B
SDA	0x5C
SCL	0x5D

6.3 Operation with Multiple Master

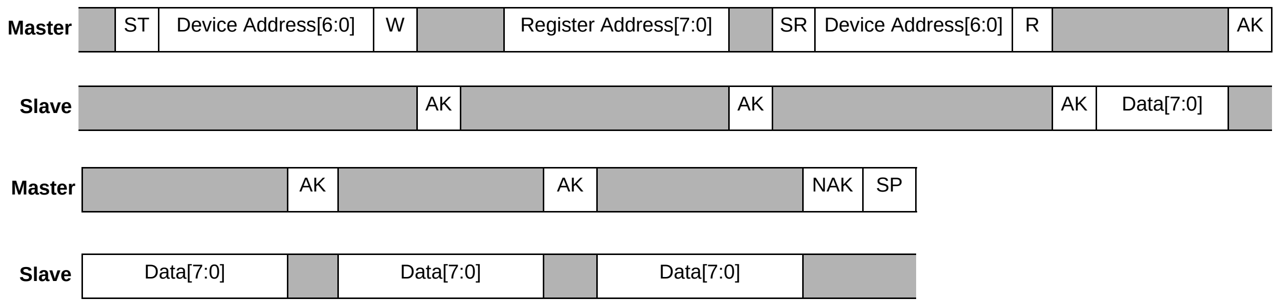
When operating with multiple masters, bus confusion between I^2C masters is sometimes a problem. One way to prevent this is to avoid using repeated starts to the MPR121. On a I^2C bus, once a master issues a start/repeated start condition, that master owns the bus until a stop condition occurs. If a master that does not own the bus attempts to take control of that bus, then improper addressing may occur. An address may always be rewritten to fix this problem. Follow I^2C protocol for multiple master configurations.

6.4 Read and Write Operation Format

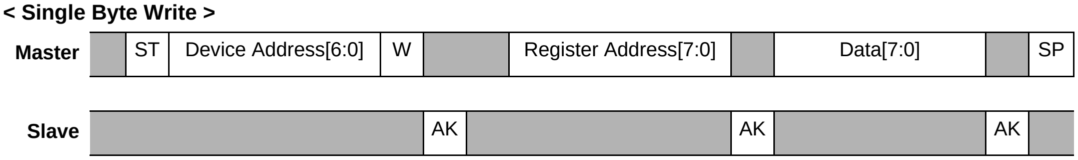
< Single Byte Read >



< Multiple Byte Read >

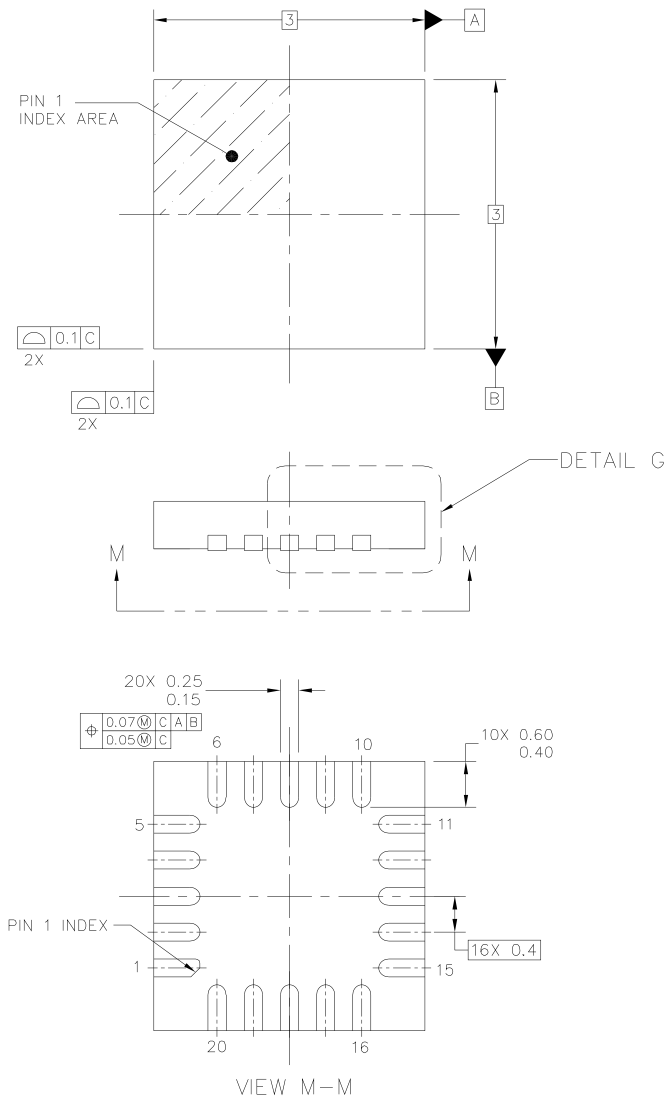


< Single Byte Write >



Legend

ST: Start Condition	SP: Stop Condition	NAK: No Acknowledge	W: Write = 0
SR: Repeated Start Condition	AK: Acknowledge	R: Read = 1	



	MECHANICAL OUTLINE	PRINT VERSION NOT TO SCALE	
TITLE: QUAD FLAT NO LEAD COL PACKAGE (QFN–COL) 20 TERMINAL, 0.4 PITCH (3 X 3 X 0.6)	DOCUMENT NO: 98ASA00021D	REV: O	
	CASE NUMBER: 2059–01	19 FEB 2009	
	STANDARD: NON JEDEC		

PAGE 1 OF 3

DETAIL G

	MECHANICAL OUTLINE	PRINT VERSION NOT TO SCALE
TITLE: QUAD FLAT NO LEAD COL PACKAGE (QFN-COL) 20 TERMINAL, 0.4 PITCH (3 X 3 X 0.6)	DOCUMENT NO: 98ASA00021D	REV: 0
	CASE NUMBER: 2059-01	19 FEB 2009
	STANDARD: NON JEDEC	

PAGE 2 OF 3

NOTES:

1. ALL DIMENSIONS ARE IN MILLIMETERS.
2. INTERPRET DIMENSIONS AND TOLERANCES PER ASME Y14.5M—1994.
3. THIS IS NON JEDEC REGISTERED PACKAGE.
4. COPLANARITY APPLIES TO LEADS AND ALL OTHR BOTTOM SURFACE METALLIZATION.
5. MIN. METAL GAP SHOULD BE 0.2MM.

© FREESCALE SEMICONDUCTOR, INC. ALL RIGHTS RESERVED.	**MECHANICAL OUTLINE**	PRINT VERSION NOT TO SCALE
TITLE: QUAD FLAT NO LEAD COL PACKAGE (QFN—COL) 20 TERMINAL, 0.4 PITCH (3 X 3 X 0.6)	DOCUMENT NO: 98ASA00021D	REV: 0
	CASE NUMBER: 2059—01	19 FEB 2009
	STANDARD: NON JEDEC	

PAGE 3 OF 3

Table 11. Revision history

Revision number	Revision date	Description of changes
3	12/2011	• On Page 1, Under Features: Changed 3 mA shutdown current to 3 mA in scan stop mode current, changed 12 electrodes to 12 electrodes/capacitance sensing inputs in which 8 are multifunctional for LED driving and GPIO, added two new bullets: Integrated independent autocalibration for each electrode input and Autoconfiguration of charge current and charge time for each electrode input, Under Implementations: added three bullets • Updated Table 1 Pin Descriptions, modified pin descriptions for Pins 4, 5, 7 • In Section 3, added Power Supply paragraph, modified remaining paragraphs • In Table 2, changed ELEPROX to PROX_OOR, changed Register Names from: AFE Configuration and Filter Configuration to: Filter/Global CDC Configuration and Filter/Global CDT Configuration, added new register for Soft Reset Register • Removed AN3889, AN3890, AN3891, AN3892, AN3893, AN3894, AN3895, and AN3944 documents • Added Sections 5.0 through 6.4
4	02/2013	• Global change to Table 5, renamed all instances Run1 to Run. Added footnote in table

How to Reach Us:

Home Page:
freescale.com

Web Support:
freescale.com/support

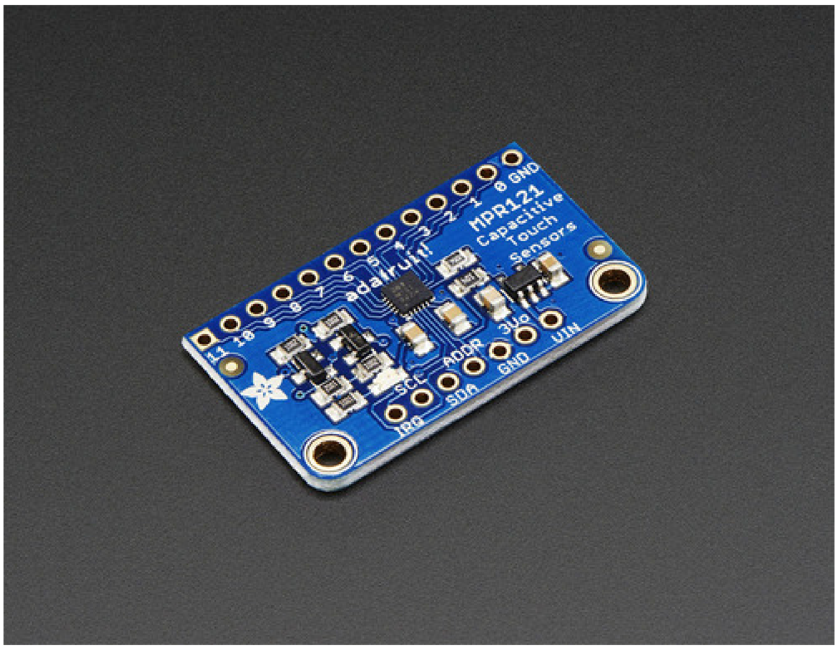
Document Number: MPR121
Rev. 4
02/2013

freescale™

adafruit learning system

Adafruit MPR121 12-Key Capacitive Touch Sensor Breakout Tutorial

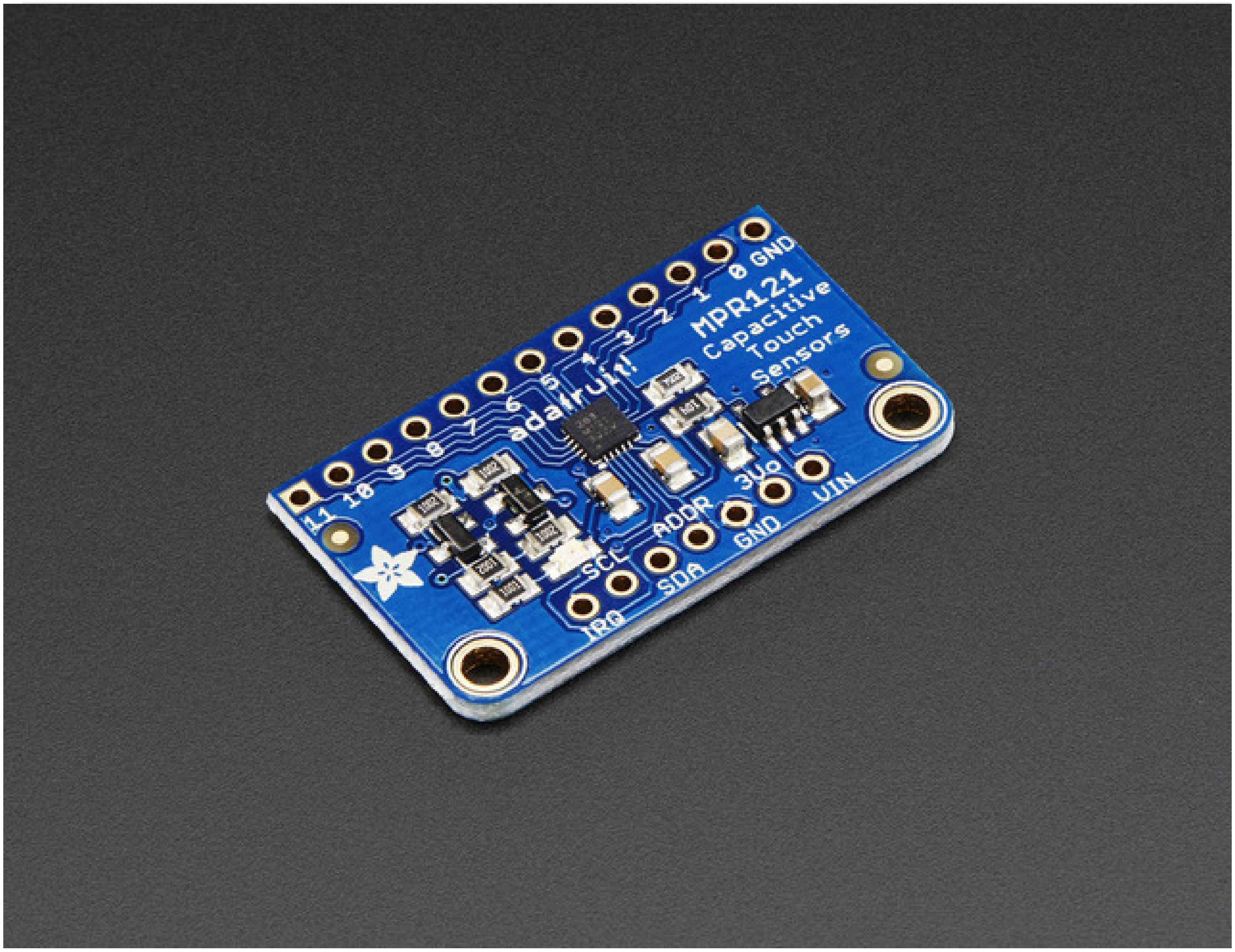
Created by lady ada



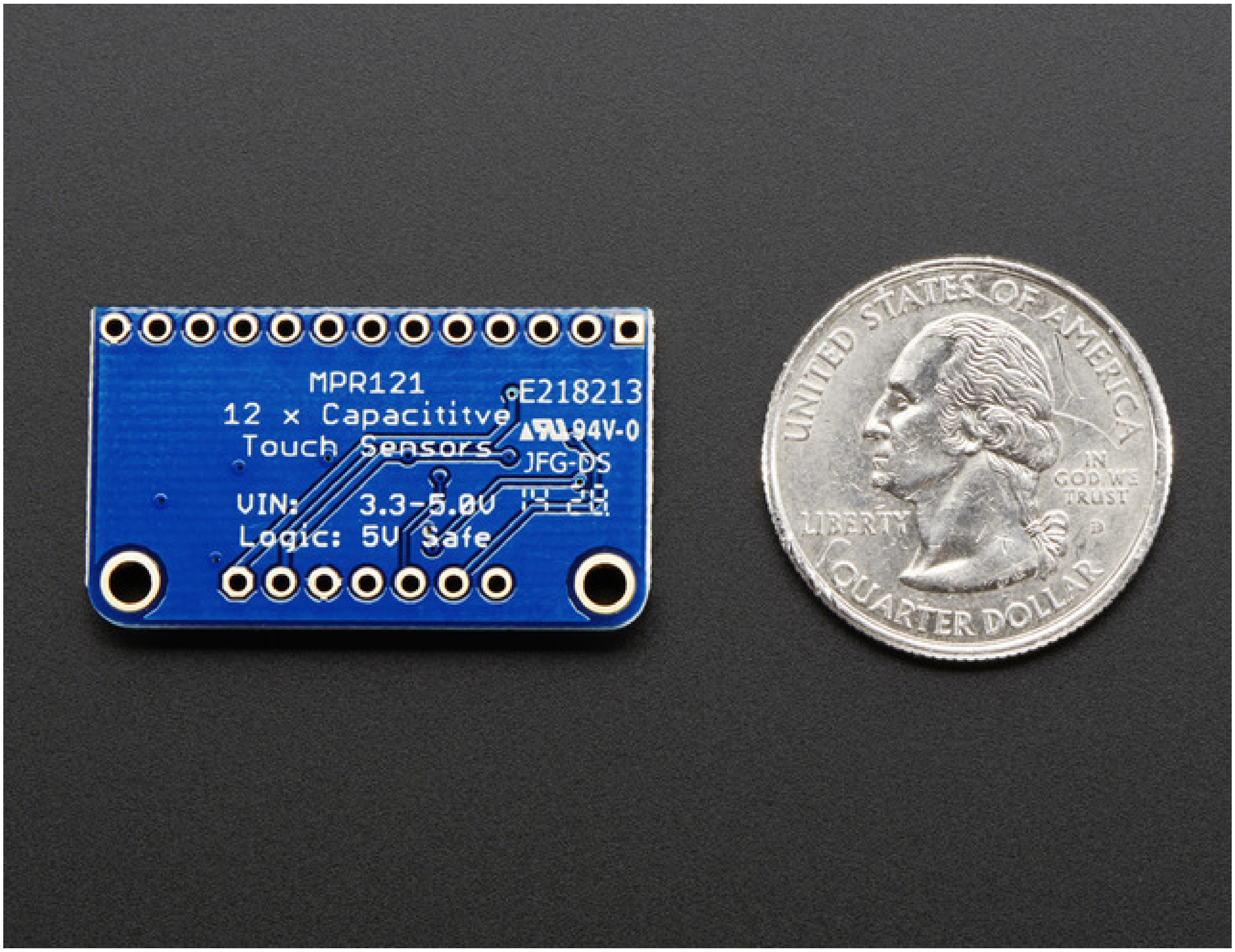
Last updated on 2014-07-25 02:45:09 PM EDT

Guide Contents

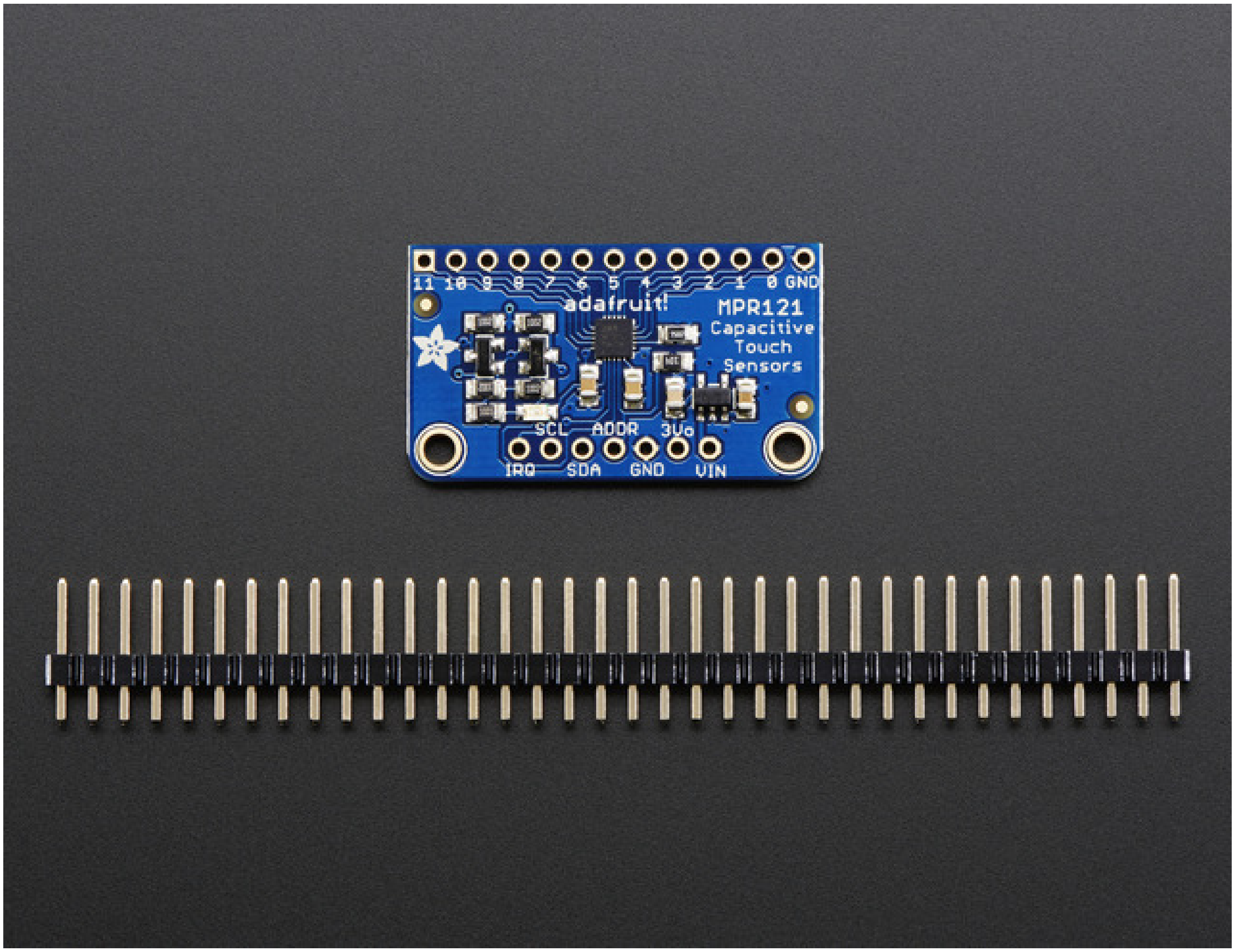
Overview



Add lots of touch sensors to your next microcontroller project with this easy-to-use 12-channel capacitive touch sensor breakout board, starring the MPR121. This chip can handle up to 12 individual touch pads.

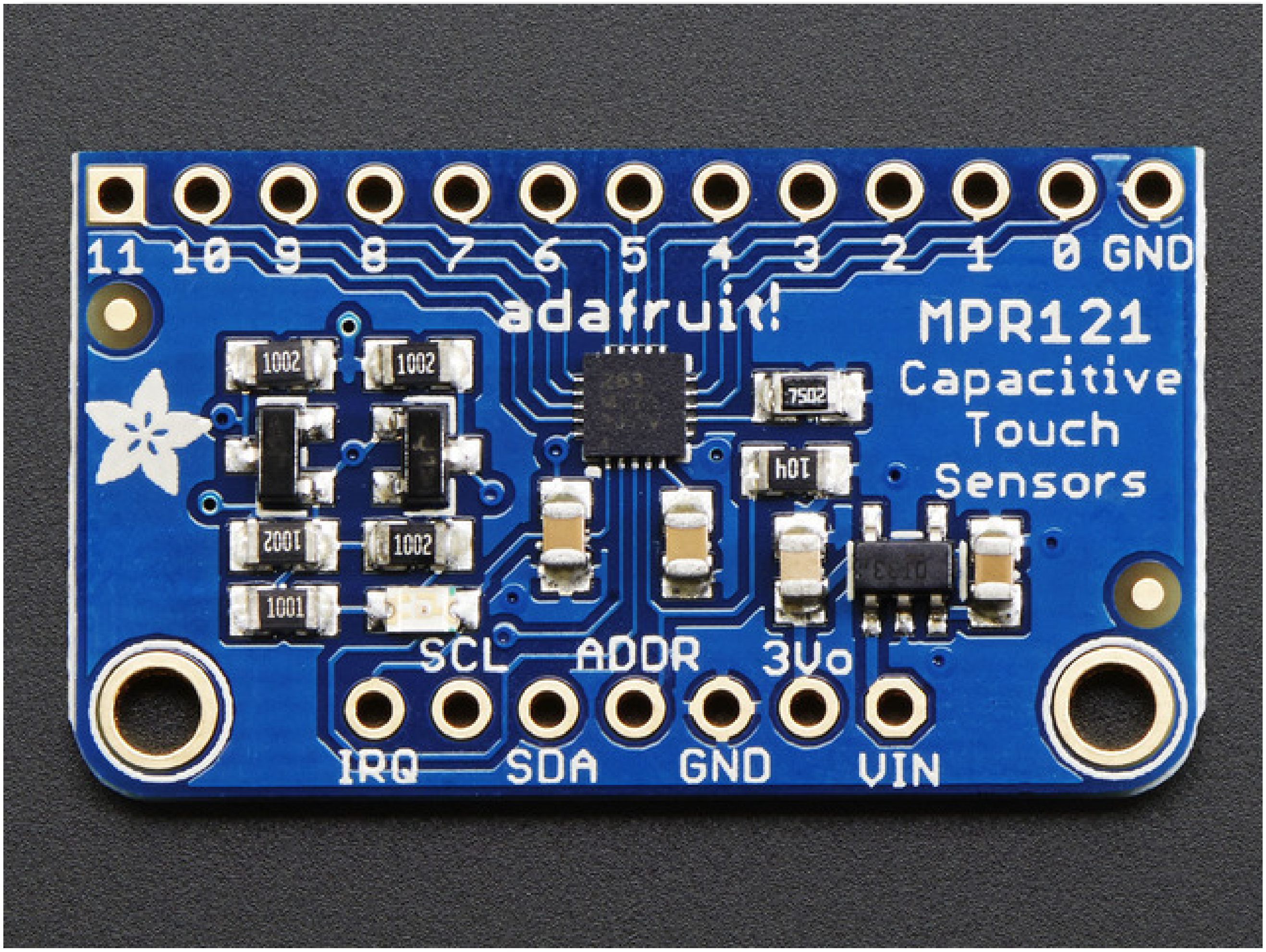


The MPR121 has support for only I2C, which can be implemented with nearly any microcontroller. You can select one of 4 addresses with the ADDR pin, for a total of 48 capacitive touch pads on one I2C 2-wire bus. Using this chip is a lot easier than doing the capacitive sensing with analog inputs: it handles all the filtering for you and can be configured for more/less sensitivity.



This sensor comes as a tiny hard-to-solder chip so we put it onto a breakout board for you. Since it's a 3V-only chip, we added a 3V regulator and I2C level shifting so its safe to use with any 3V or 5V microcontroller/processor like Arduino. We even added an LED onto the IRQ line so it will blink when touches are detected, making debugging by sight a bit easier on you. 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 or pyralux, then solder a wire that connects from the foil pad to the breakout.

Pinouts



The little chip in the middle of the PCB is the actual MPR121 sensor that does all the capacitive sensing and filtering. We add all the extra components you need to get started, and 'break out' all the other pins you may want to connect to onto the PCB. For more details you can check out the schematics in the Downloads page.

Power Pins

The sensor on the breakout requires 3V power. Since many customers have 5V microcontrollers like Arduino, we tossed a 3.3V regulator on the board. Its ultra-low dropout so you can power it from 3.3V-5V just fine.

- **Vin** - this is the power pin. Since the chip uses 3 VDC, we have included a voltage regulator on board that will take 3-5VDC and safely convert it down. To power the board, give it the same power as the logic level of your microcontroller - e.g. for a 5V micro like Arduino, use 5V
- **3Vo** - this is the 3.3V output from the voltage regulator, you can grab up to 100mA from this if you like
- **GND** - common ground for power and logic

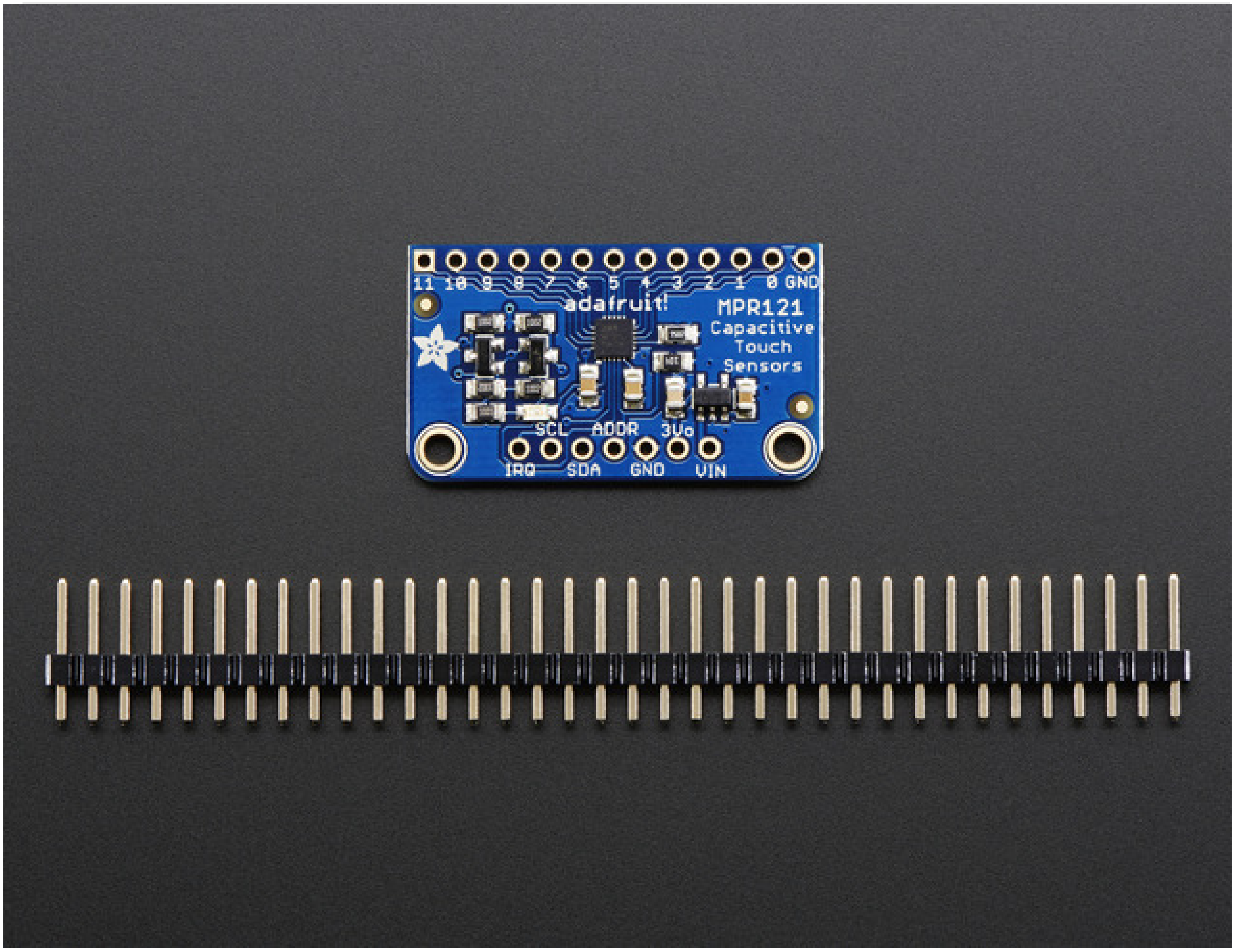
I2C Pins

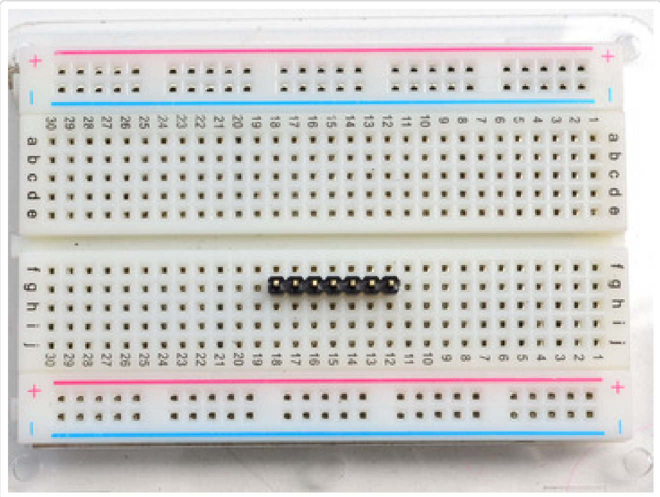
- **SCL** - I2C clock pin, connect to your microcontrollers I2C clock line.
- **SDA** - I2C data pin, connect to your microcontrollers I2C data line.

IRQ and ADDR Pins

- **ADDR** is the I2C address select pin. By default this is pulled down to ground with a 100K resistor, for an I2C address of 0x5A. You can also connect it to the 3Vo pin for an address of 0x5B, the SDA pin for 0x5C or SCL for address 0x5D
- **IRQ** is the Interrupt Request signal pin. It is pulled up to 3.3V on the breakout and when the sensor chip detects a change in the touch sense switches, the pin goes to 0V until the data is read over i2c

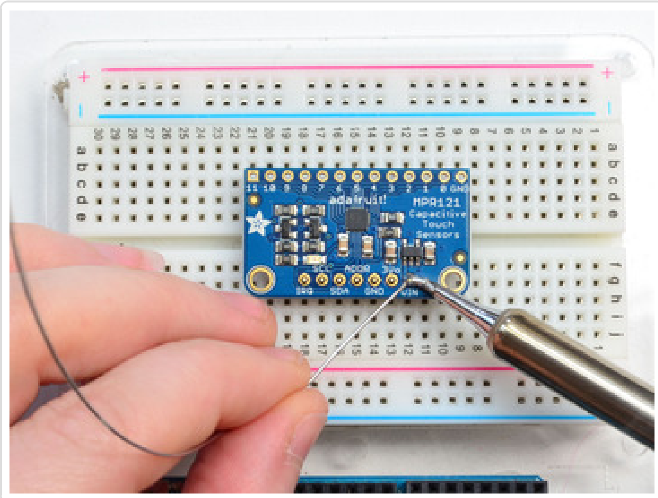
Assembly





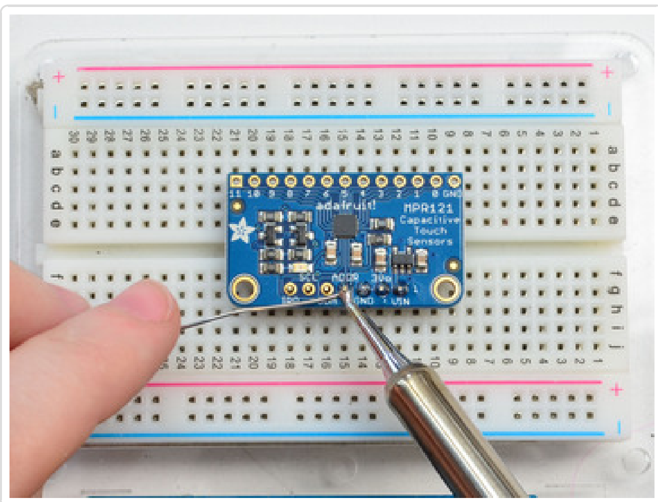
Prepare the header strip:

Cut the strip to length if necessary. It will be easier to solder if you insert it into a breadboard - **long pins down**



Add the breakout board:

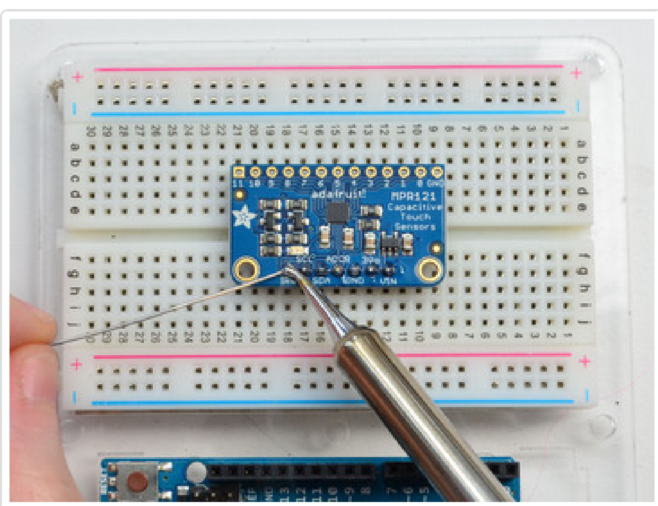
Place the breakout board over the pins so that the short pins poke through the breakout pads



And Solder!

Be sure to solder all pins for reliable electrical contact.

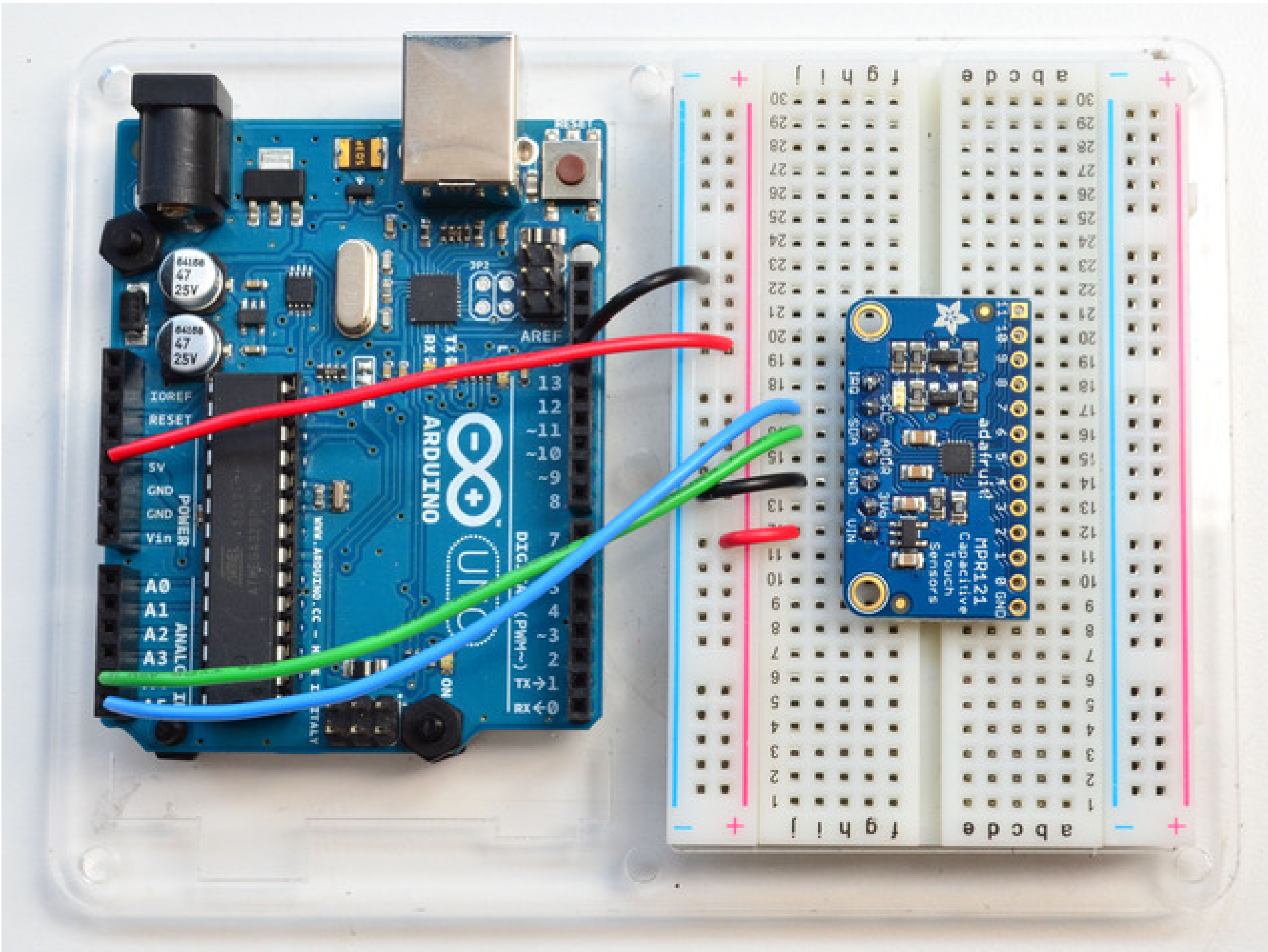
(For tips on soldering, be sure to check out our [Guide to Excellent Soldering](http://adafru.it/aTk) (http://adafru.it/aTk)).



You're done! Check your solder joints visually and continue onto the next steps

Wiring

You can easily wire this breakout to any microcontroller, we'll be using an Arduino. For another kind of microcontroller, just make sure it has I2C, then port the code - its pretty simple stuff!



- Connect **Vin** to the power supply, 3-5V is fine. Use the same voltage that the microcontroller logic is based off of. For most Arduinos, that is 5V
- Connect **GND** to common power/data ground
- Connect the **SCL** pin to the I2C clock **SCL** pin on your Arduino. On an UNO & '328 based Arduino, this is also known as **A5**, on a Mega it is also known as **digital 21** and on a Leonardo/Micro, **digital 3**
- Connect the **SDA** pin to the I2C data **SDA** pin on your Arduino. On an UNO & '328 based Arduino, this is also known as **A4**, on a Mega it is also known as **digital 20** and on a Leonardo/Micro, **digital 2**

The MPR121 **ADDR** pin is pulled to ground and has a default I2C address of **0x5A**
You can adjust the I2C address by connecting **ADDR** to other pins:

- ADDR not connected: 0x5A

- ADDR tied to 3V: 0x5B
- ADDR tied to SDA: 0x5C
- ADDR tied to SCL: 0x5D

We suggest sticking with the default for the test demo, you can always change it later.

Download Adafruit_MPR121

To begin reading sensor data, you will need to download Adafruit_MPR121_Library from our github repository (http://adafru.it/dKE). You can do that by visiting the github repo and manually downloading or, easier, just click this button to download the zip

Download Adafruit_MPR121

http://adafru.it/dKF

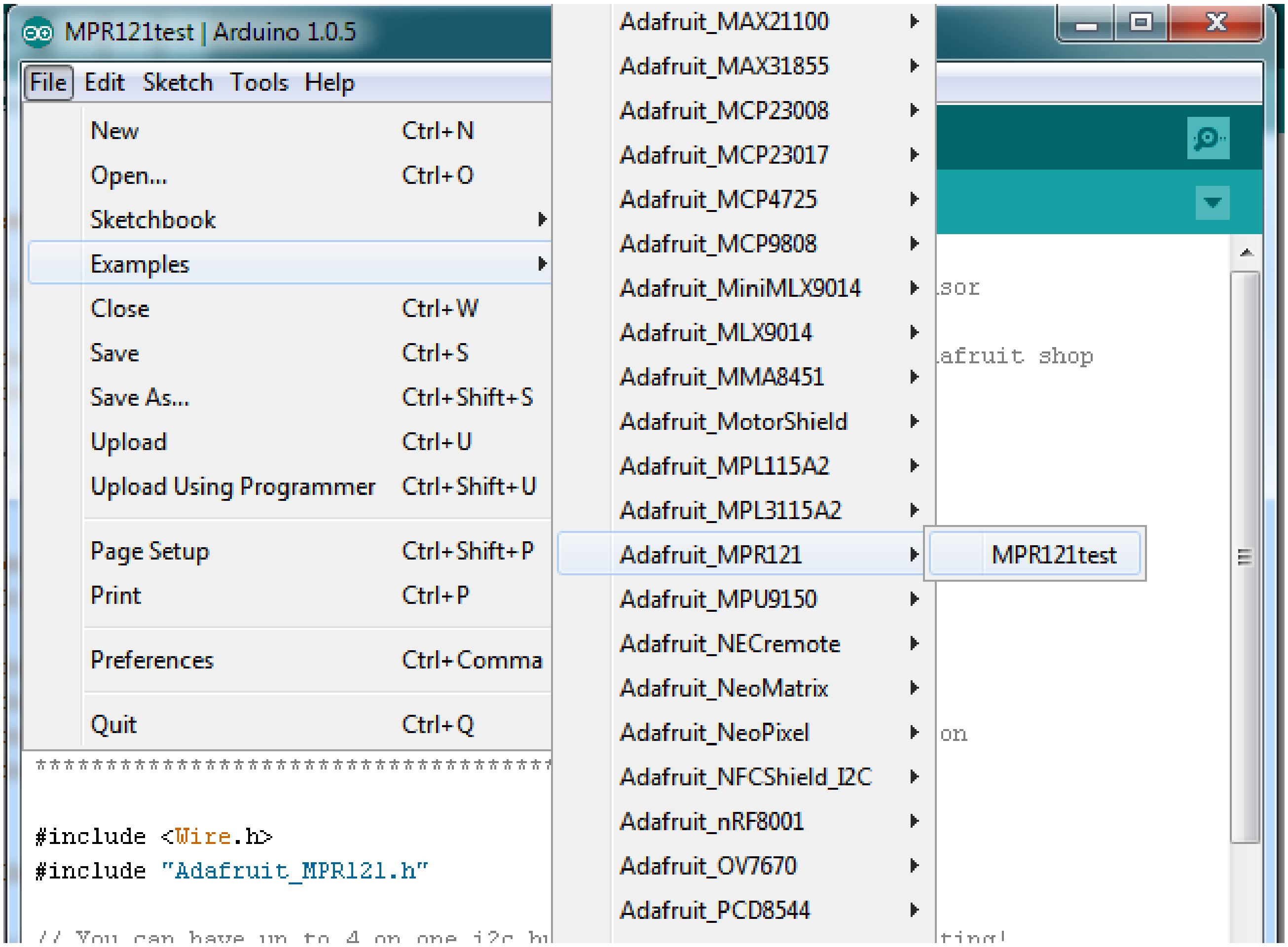
Rename the uncompressed folder **Adafruit_MPR121** and check that the **Adafruit_MPR121** folder contains **Adafruit_MPR121.cpp** and **Adafruit_MPR121.h**

Place the **Adafruit_MPR121** library folder your **arduinosketchfolder/libraries/** folder. You may need to create the **libraries** subfolder if its your first library. Restart the IDE.

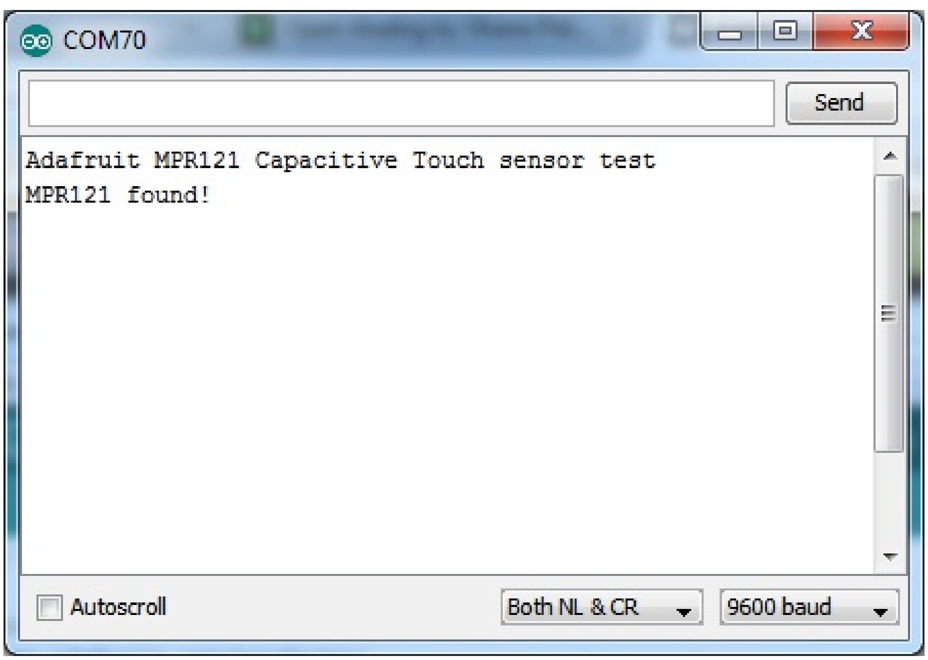
We also have a great tutorial on Arduino library installation at:
http://learn.adafruit.com/adafruit-all-about-arduino-libraries-install-use (http://adafru.it/aYM)

Load Demo

Open up **File->Examples->Adafruit_MPR121->MPR121test** and upload to your Arduino wired up to the sensor

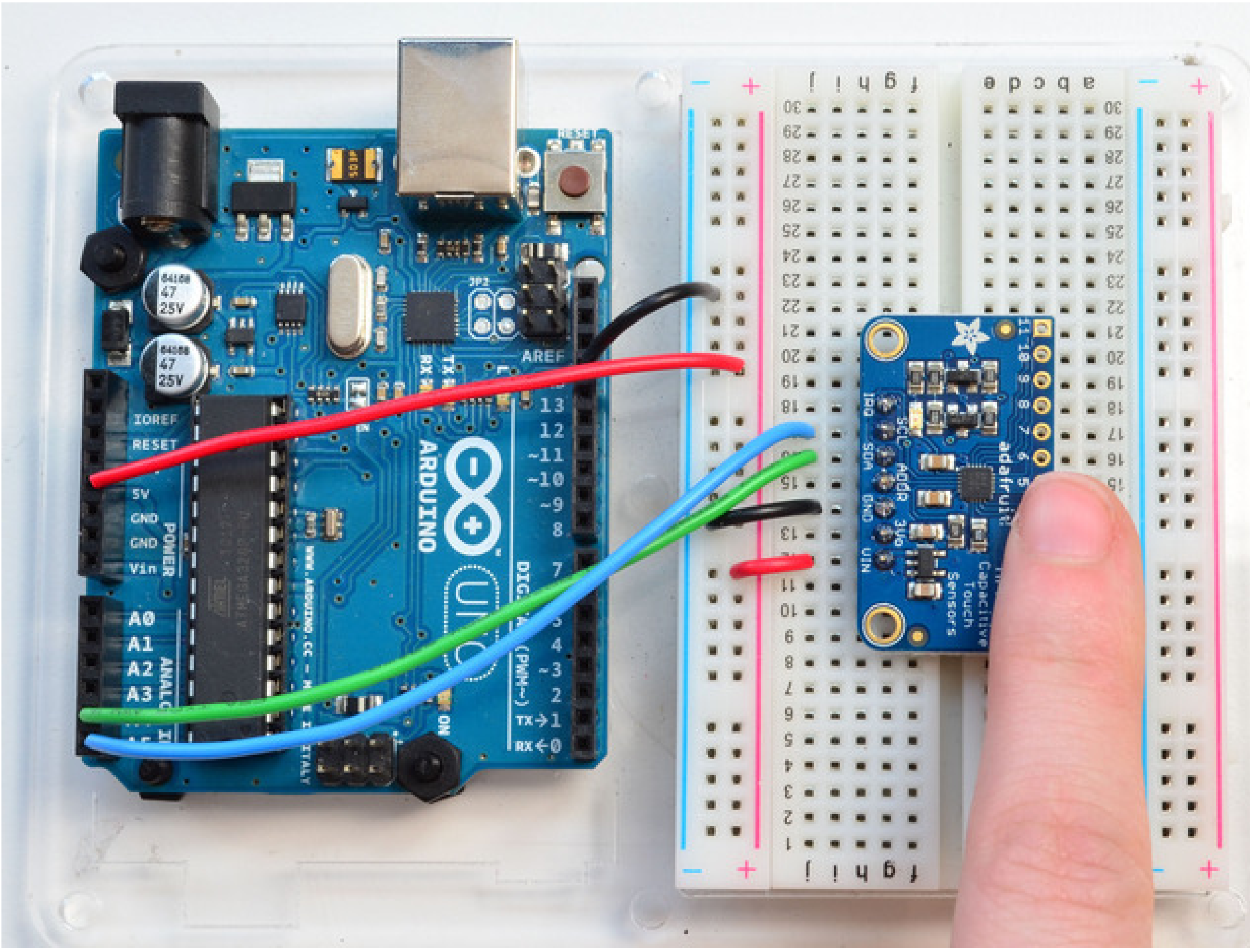


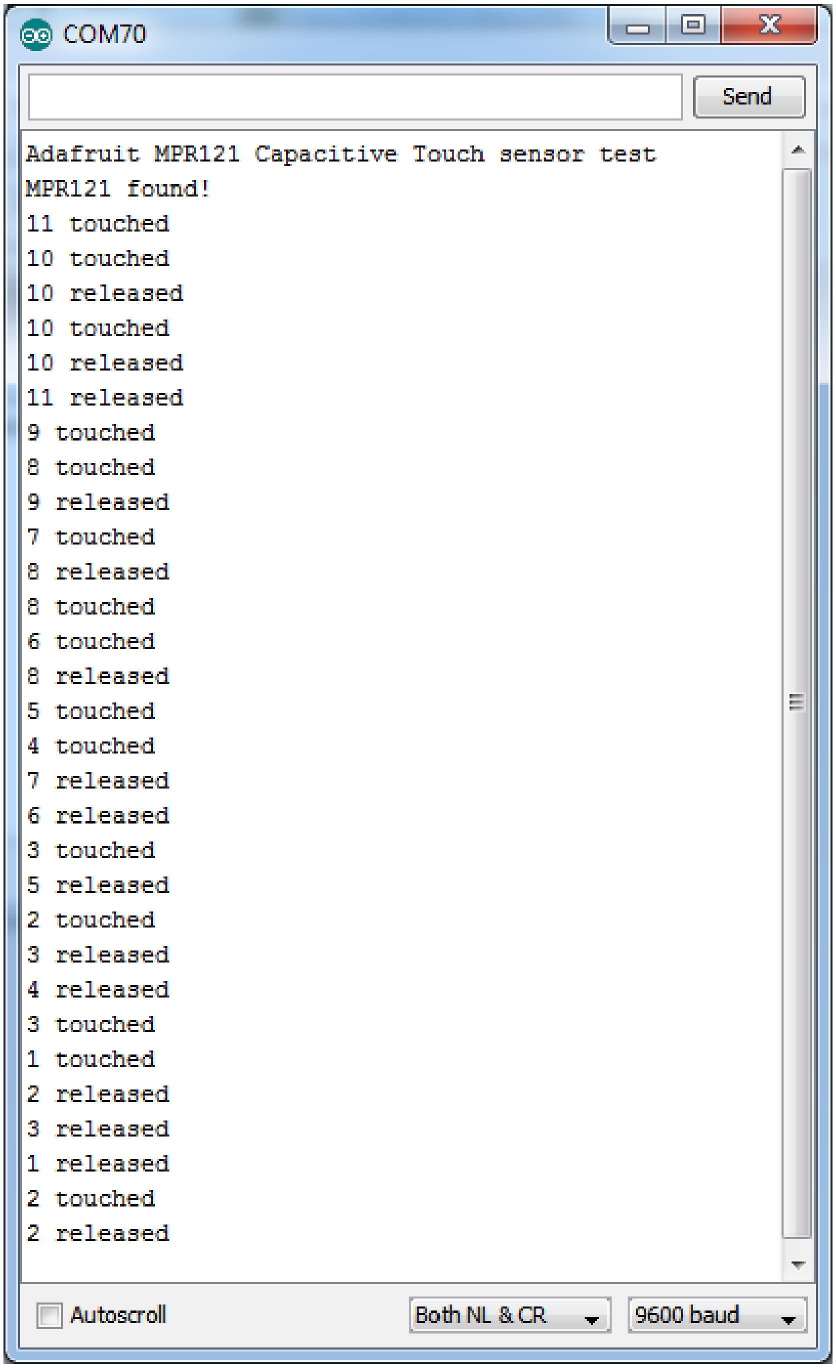
Thats it! Now open up the serial terminal window at 9600 speed to begin the test.



Make sure you see the "MPR121 found!" text which lets you know that the sensor is wired correctly.

Now touch the 12 pads with your fingertip to activate the touch-detection





For most people, that's all you'll need! Our code keeps track of the 12 'bits' for each touch and has logic to let you know when a contect is touched or released.

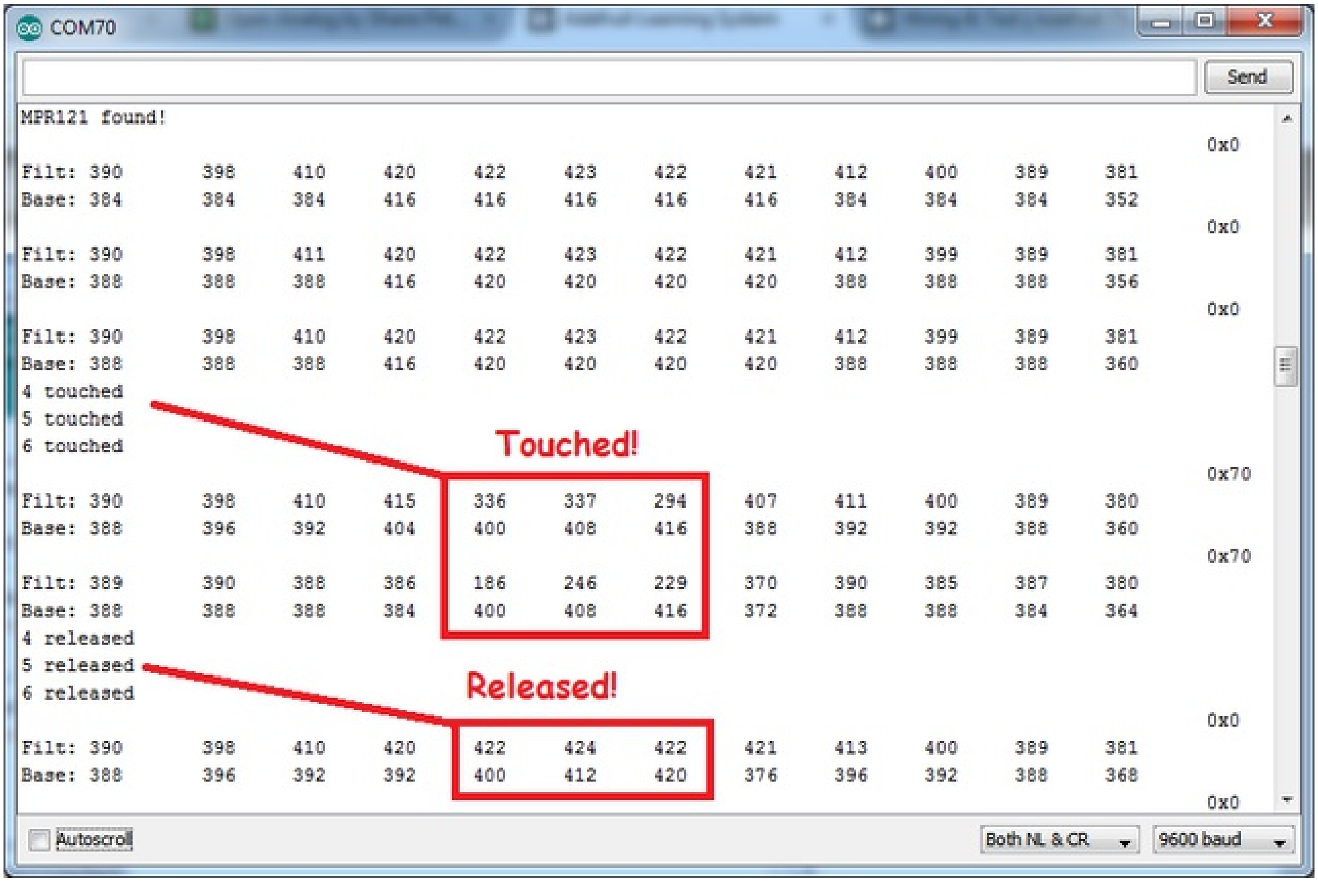
If you're feeling more advanced, you can see the 'raw' data from the chip. Basically, what it does it keep track of the capacitance it sees with "counts". There's some baseline count number that depends on the temperature, humidity, PCB, wire length etc. Where's a dramatic change in number, its considered that a person touched or released the wire.

Comment this "return" line to activate that mode:

```
// comment out this line for detailed data from the sensor!
return;
```

Then reupload. Open up the serial console again - you'll see way more text

Each reading has 12 columns. One for each sensor, #0 to #11. There's two rows, one for the 'baseline' and one for the current filtered data reading. When the current reading is within about 12 counts of the baseline, that's considered untouched. When the reading is more than 12 counts smaller than the baseline, the chip reports a touch.



Most people don't need raw data too much, but it can be handy if doing intense debugging. It can be helpful if you are tweaking your sensors to get good responsivity.

Library Reference

Since the sensors use I2C, there's no pins to be defined during instantiation. You can just use:

> Adafruit_MPR121 cap = Adafruit_MPR121();

When you initialize the sensor, pass in the I2C address. It can range from 0x5A (default) to

0x5D

```
cap.begin(0x5A)
```

begin() returns true if the sensor was found on the I2C bus, and false if not.

Touch detection

99% of users will be perfectly happy just querying what sensors are currentlt touched. You can read all at once with
cap.touched()
Which returns a 16 bit value. Each of the bottom 12 bits refers to one sensor. So if you want to test if the #4 is touched, you can use

```
if (cap.touched() & (1 << 4)) { do something }
```

You can check its not touched with:

```
if (! (cap.touched() & (1 << 4)) ) { do something }
```

Raw Data

You can grab the current baseline and filtered data for each sensor with

```
filteredData(sensornumber);
baselineData(sensornumber);
```

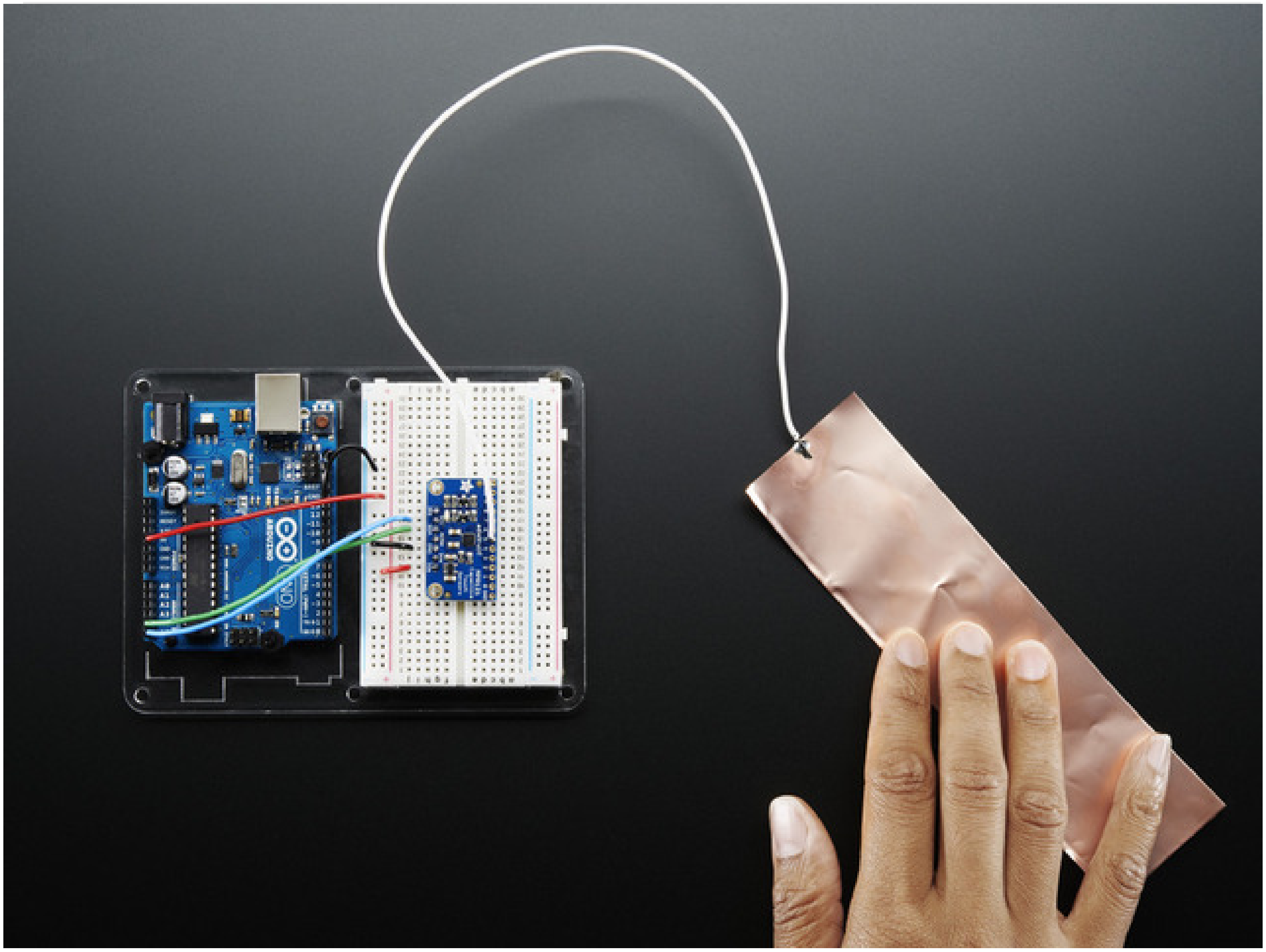
It returns a 16-bit number which is the number of counts, there's no unit like "mg" or "capacitance". The baseline is initialized to the current ambient readings when the sensor begin() is called - you can always reinitialize by re-calling begin()! The baseline will drift a bit, that's normal! It is trying to compensate for humidity and other environmental changes.

If you need to change the threshholds for touch detection, you can do that with

```
setThreshholds(uint8_t touch, uint8_t release)
```

By default, the touch threshhold is 12 counts, and the release is 6 counts. It's reset to these values whenever you call begin() by the way.

Electrodes



Once you have the MPR121 breakout working you'll want to construct electrodes. These are large conductive piece of copper, foil, paint, etc that will act as the "thing you touch"

Remember that electrodes must be electrically conductive! We suggest copper foil tape, conductive fabrics, ITO, pyralux flex PCB, etc. We have tons of great conductive materials in our Materials category. Some can be soldered to, others can be clipped to with alligator chips. (http://adafru.it/dKl)

Remember, it doesn't have to be metal to be electrically conductive. Other things that work are tap or salt water, many kinda of food, even fruit!

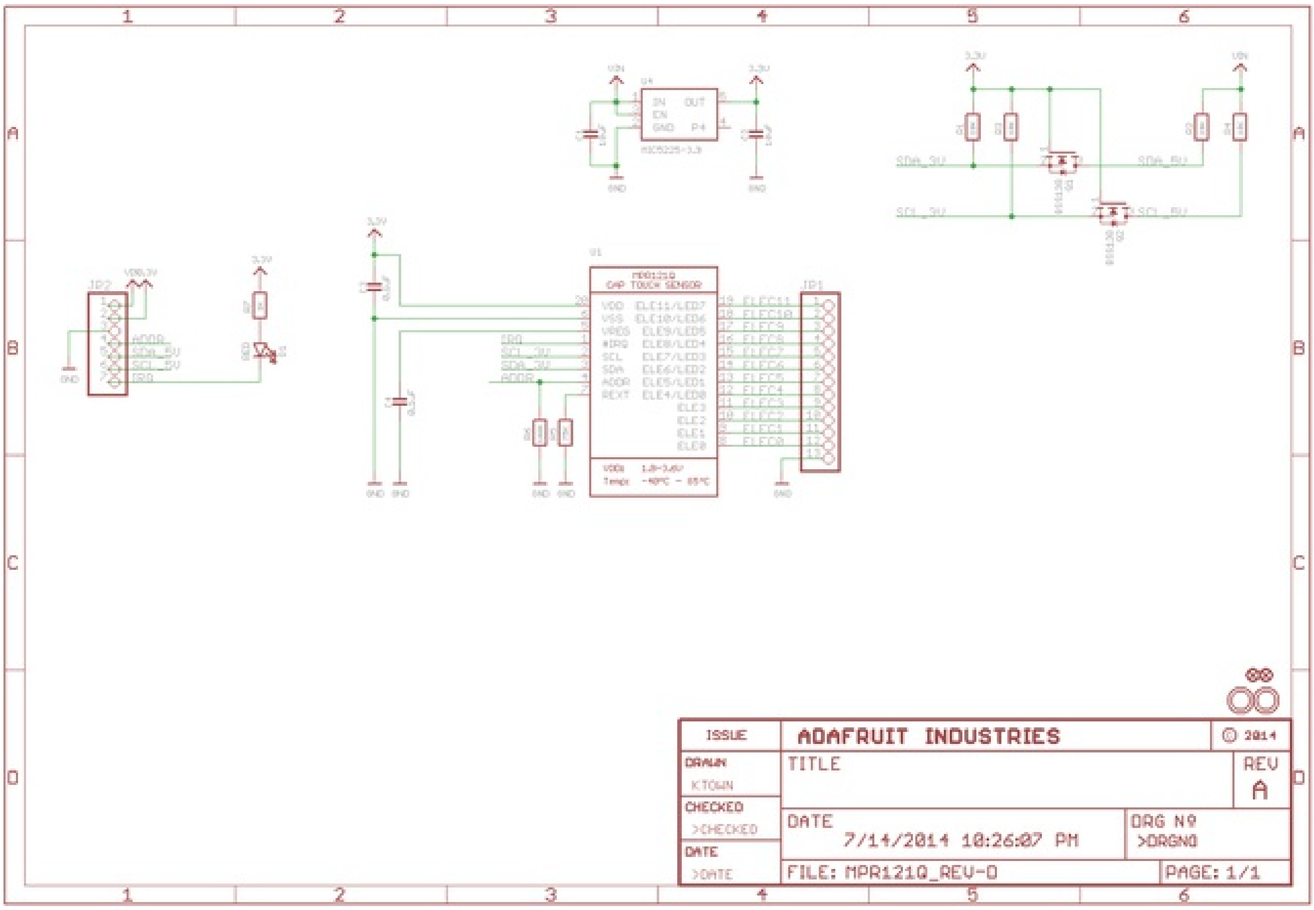
We suggest soldering a wire to the electrode pad on the breakout and then soldering or clipping it to whatever you want your electrode to be.

Downloads

Datasheets

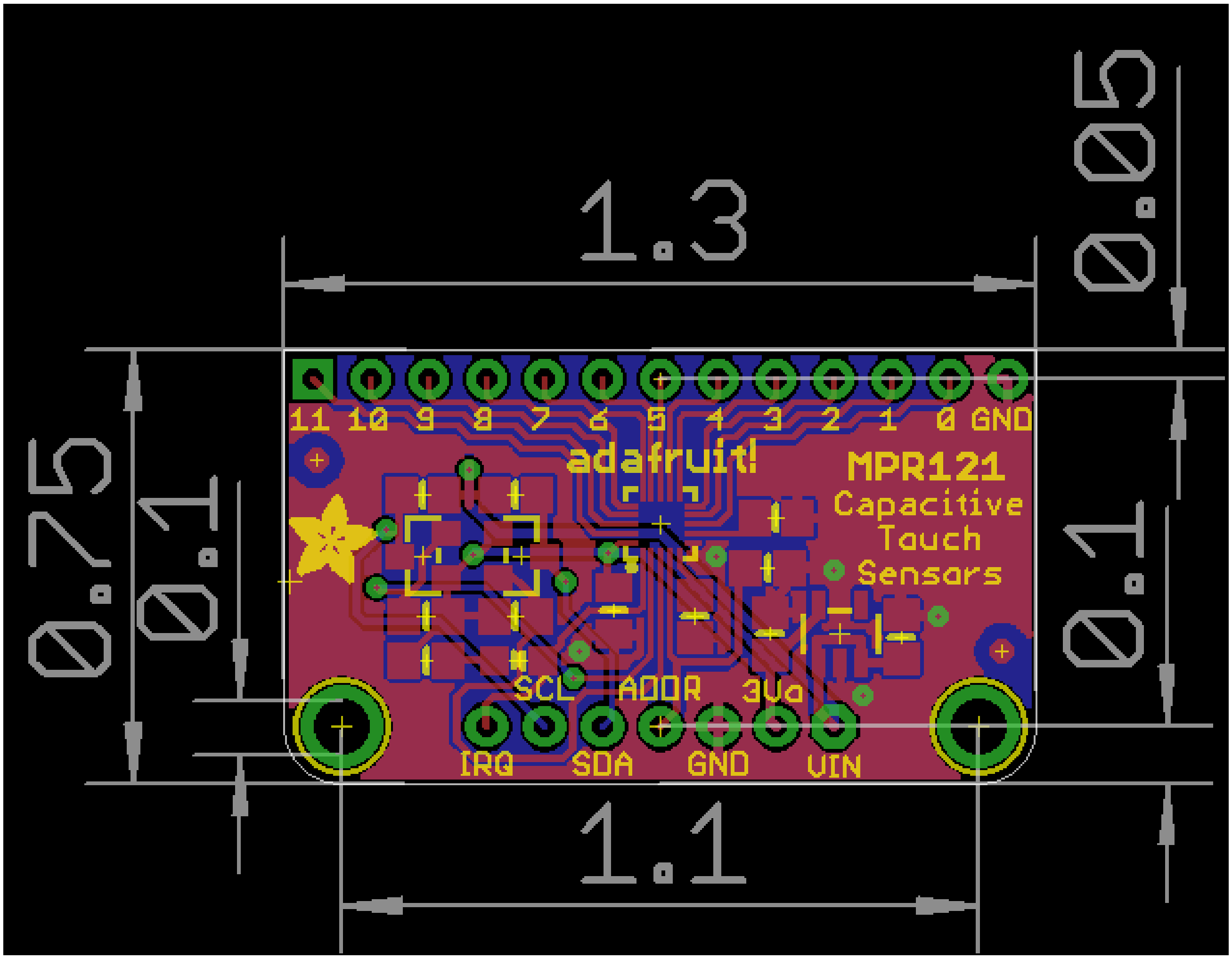
- MPR121 Datasheet (http://adafru.it/dKG)

Breakout Board Schematic



Fabrication Print

Dimensions in Inches


1.3
0.05
0.75
0.1
0.1
11 10 9 8 7 6 5 4 3 2 1 0 GND
adafruit!
MPR121
Capacitive
Touch
Sensors
SCL
ADDR
3Vo
IRQ
SDA
GND
VIN
1.1