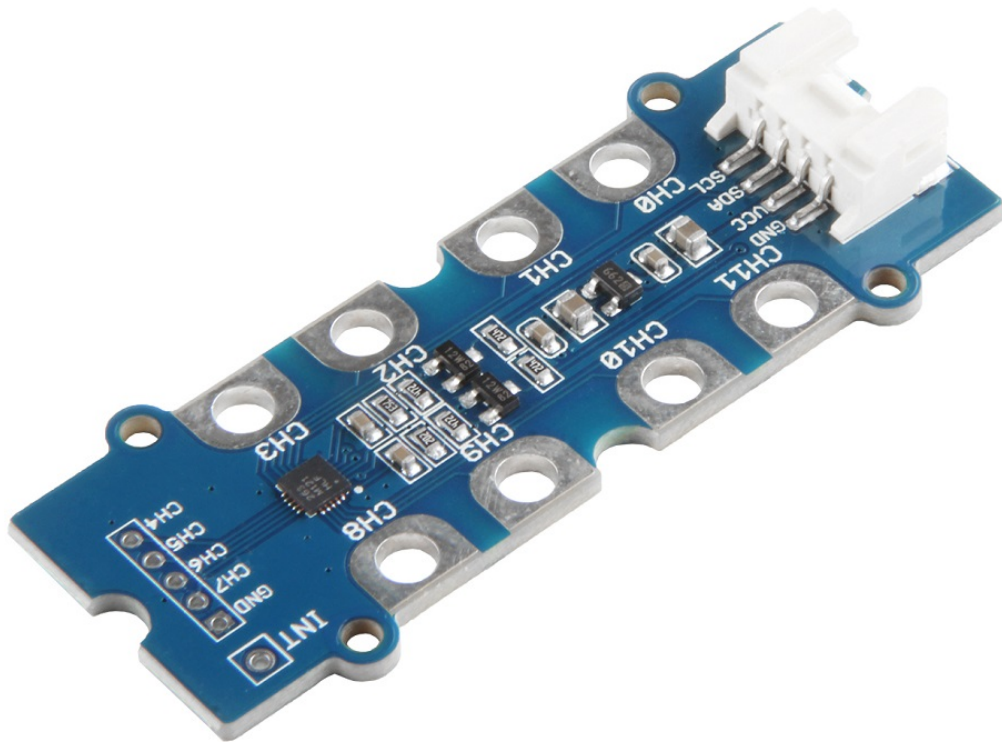


Grove - 12 Key Capacitive I2C Touch Sensor V2(MPR121)



The **Grove - 12 Key Capacitive I2C Touch Sensor V2 (MPR121)** is a multichannel proximity capacitive touch sensor. It's a 3-in-1 module with the following features: Capacitance Sensing, Touch Sensing, and Proximity Sensing.

Capacitance Sensing: This module 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.

Touch Sensing: Once the electrode capacitance data is acquired, the electrode touch/release status is determined comparing it to the capacitance baseline value.

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.

Based on Freescale MPR121, this sensor have 12 completely independent electrodes with build-in autoconfiguration. Thanks to I2C interface, you can detect all the 12 electrodes signals with only one Grove port, and the I2C address is hardware configurable, from 0X5B to 0X5D. This also makes it possible for multiple **Grove - 12 Key Capacitive I2C Touch Sensor V2 (MPR121)** to be used together for channel expansions in a single system, you can build a touch system which contains max. 36 electrodes.

This sensor is an upgrade version of [Grove - I2C Touch Sensor](https://www.seeedstudio.com/Grove-I2C-Touch-Sensor-p-840.html) [https://www.seeedstudio.com/Grove-I2C-Touch-Sensor-p-840.html], to meet Matsuzawa.Takashi's(one of our customer) needs, we make the I²C address changeable, and even cheaper than the old version. So if you have any suggestions about all the Groves, please just shoot to us. We will always listen to your voice, it may make another upgrade, even a new grove. Please kindly write your suggestions in the [Grove 100+](https://www.seeedstudio.com/grove_100) [https://www.seeedstudio.com/grove_100] page.

Using Spanner to Play Mario



Get One Now 

[<https://www.seeedstudio.com/Grove-12-Key-Capacitive-I2C-Touch-Sensor-V2-%28MPR121%29-p-3141.html>]

Version Change

Item	Grove - 12 Key Capacitive I2C Touch Sensor V2	Grove - I2C Touch Sensor
Main Chip	MPR121	MPR121
I ² C Address	changeable(0X5B ~ 0X5D)	unmodifiable(0X5A)
Touch Sensor Feeler	x	√
Input Interface	alligator interface	DIP 2Pin Female Header
Cost Performance	High	Low
Release Time	September 11-2018	October 31-2015

Features

- Internal 10-bit ADC
- Integrated independent autocalibration for each electrode input
- Completely independent electrodes with built-in autoconfiguration
- I2C interface, with IRQ Interrupt output to advise electrode status changes
- Hardware configurable I2C address
- 12 electrodes/capacitance sensing inputs in which 8 are multifunctional for LED driving and GPIO

- 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

Specification

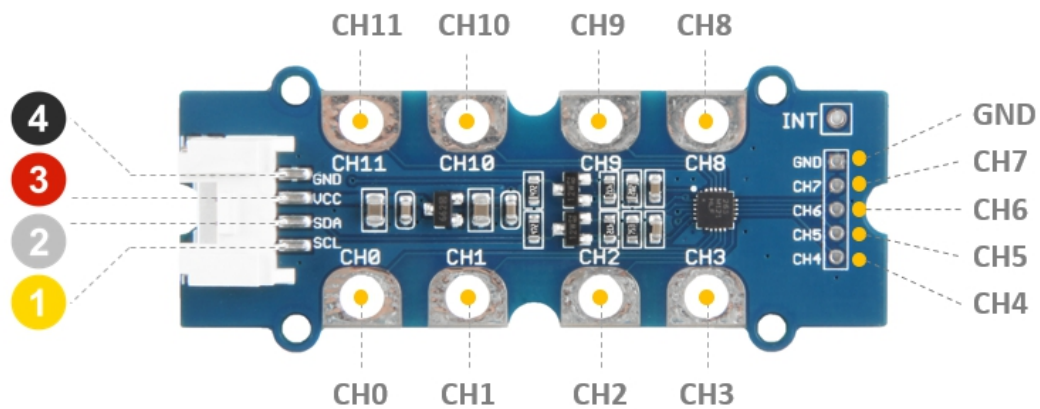
Item	Value
Operating voltage	3.3V / 5V
Operating temperature	-40°C to +85°C
Storage Temperature Range	-40°C to +125°C
Capacitances ranging	10 pF to over 2000 pF
Resolution	0.01 pF
GPIO Source Current per Pin	12 mA
GPIO Sink Current per Pin	1.2 mA
Interface	I ² C
I ² C address range	0x5B,0x5C,0x5D
Default I ² C address	0x5B

Applications

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

Hardware Overview

Pin Map



- 4 GND: connect this module to the system GND
- 3 VCC: you can use 5V or 3.3V for this module
- 2 SDA: I²C serial data
- 1 SCL: I²C serial clock

Pin Number	Pin Name	Function	Pin multiplexing
8	CH0	Channel0, Electrode 0, input the capacitance value	-
9	CH1	Channel1, Electrode 1	-

	CH1	Channel1, Electrode 1, input the capacitance valnue	
10	CH2	Channel2, Electrode 2, input the capacitance valnue	-
11	CH3	Channel3, Electrode 3, input the capacitance valnue	-
12	CH4	Channel4, Electrode 4, input the capacitance valnue	GPIO or LED driver
13	CH5	Channel5, Electrode 5, input the capacitance valnue	GPIO or LED driver
14	CH6	Channel6, Electrode 6, input the capacitance valnue	GPIO or LED driver
15	CH7	Channel7, Electrode 7, input the capacitance valnue	GPIO or LED driver
16	CH8	Channel8, Electrode 8, input the capacitance valnue	GPIO or LED driver
17	CH9	Channel9, Electrode 9, input the capacitance valnue	GPIO or LED driver
18	CH10	Channel10, Electrode 10, input the capacitance valnue	GPIO or LED driver
Pin Number	Pin Name	Function	Pin
18	CH10	Channel10, Electrode 10, input the capacitance valnue	GPIO or LED driver

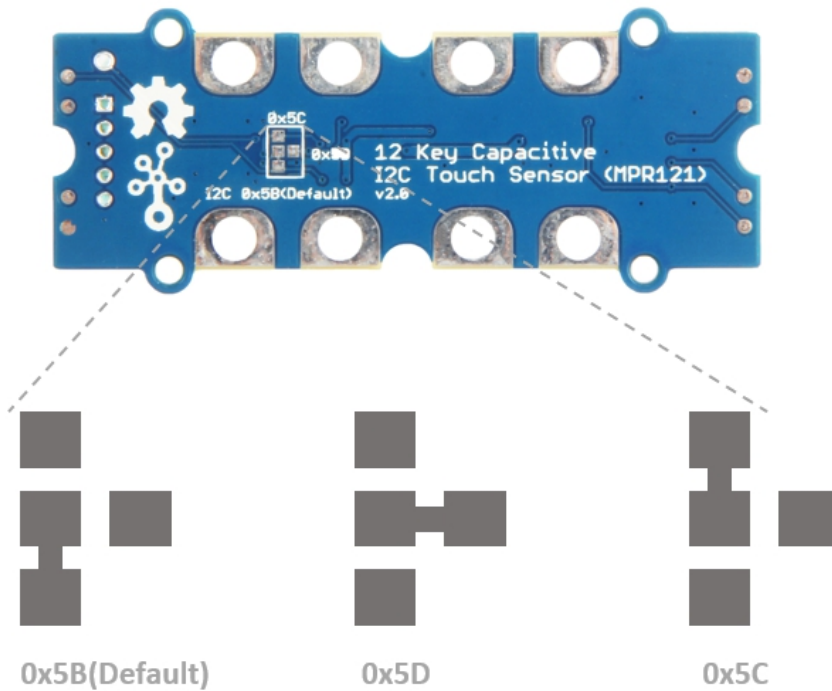
		input the capacitance value	driver
19	CH11	Channel11, Electrode 11, input the capacitance value	GPIO or LED driver



Tip

For the CH0 ~ CH11, once the electrode capacitance data is acquired, the electrode touch/release status is determined comparing it to the capacitance baseline value. And you can set the baseline value for each channel separately. The Pin12 ~ Pin19 is multifunctional, which means you can configure them as GPIO or LED driver, for more detail, please refer to the freescale application note [AN3894](https://files.seeedstudio.com/wiki/Grove-12_Key_Capacitive_I2C_Touch_Sensor_V2-MPR121/res/AN3894.pdf) [https://files.seeedstudio.com/wiki/Grove-12_Key_Capacitive_I2C_Touch_Sensor_V2-MPR121/res/AN3894.pdf].

I2C Address

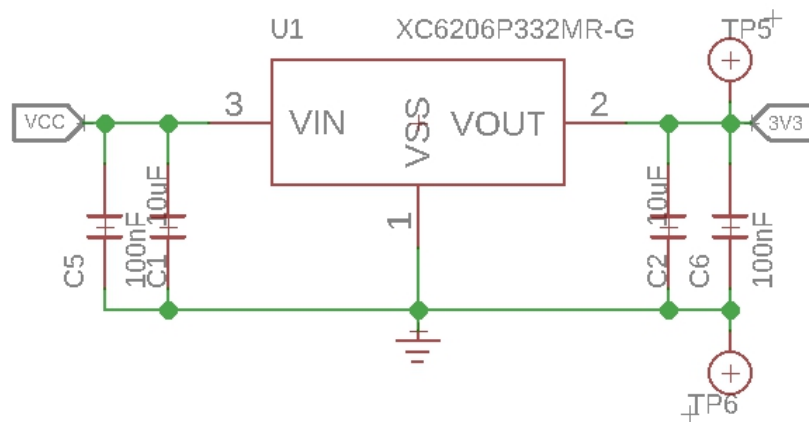


Danger

The central pad is connected to the address wire, you can change the I2C address by cutting the wire and re-welding it. For the safety of you and others, please be careful with knife or welding gun you may use.

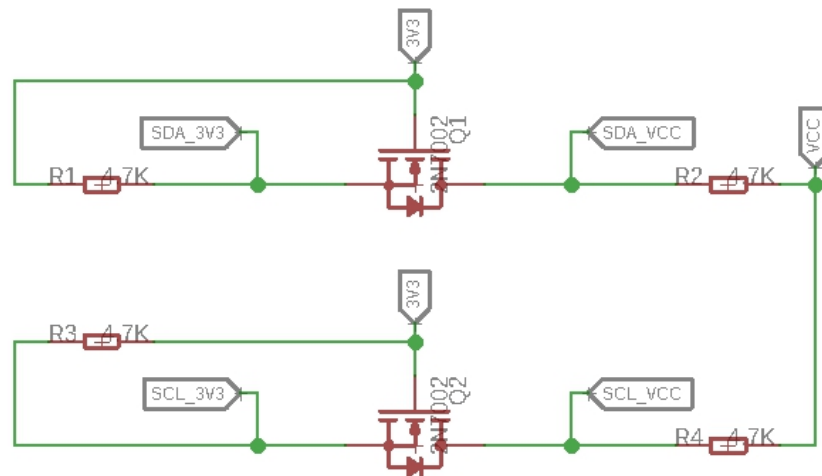
Schemaitc

Power



The operation voltage of Freescale MPR121 is 1.71V to 3.6V, however, the voltage of Arduino is 3.3V or 5V. In order to make it compaticable with 5V system, we use a voltage conversion chip to provide 3.3V for the Freescale MPR121.

Bi-directional level shifter circuit



This is a typical Bi-directional level shifter circuit to connect two different voltage section of an I²C bus. The I²C bus of this sensor use 3.3V, if the I²C bus of the Arduino use 5V, this circuit will be needed. In the schematic above, **Q1** and **Q2** are N-Channel MOSFET [2N7002A](https://files.seeedstudio.com/wiki/Grove-I2C_High_Accuracy_Temperature_Sensor-MCP9808/res/2N7002A_datasheet.pdf) [https://files.seeedstudio.com/wiki/Grove-I2C_High_Accuracy_Temperature_Sensor-MCP9808/res/2N7002A_datasheet.pdf], which act as a bidirectional switch. In order to better understand this part, you can refer to the [AN10441](https://files.seeedstudio.com/wiki/Grove-I2C_High_Accuracy_Temperature_Sensor-MCP9808/res/AN10441.pdf) [https://files.seeedstudio.com/wiki/Grove-I2C_High_Accuracy_Temperature_Sensor-MCP9808/res/AN10441.pdf]

Platforms Supported



Caution

The platforms mentioned above as supported is/are an indication of the module's software or theoretical compatibility. We only provide software library or code examples for Arduino platform in most cases. It is not possible to provide software library / demo code for all possible MCU platforms. Hence, users have to write their own software library.

Getting Started

Play With Arduino

In this part, we will show you how to use the **Grove - 12 Key Capacitive I2C Touch Sensor V2 (MPR121)** as a touch sensor, as for how to configure it as a Capacitance Sensor or Proximity Sensor, please check the [Datasheet](#)

[https://files.seeedstudio.com/wiki/Grove-12_Key_Capacitive_I2C_Touch_Sensor_V2-MPR121/res/MPR121.pdf].

Hardware

Materials required

Seeeduino V4.2



Base Shield



[Get One Now](#)

[<https://www.seeedstudio.com/Seeeduino-V4.2-p-2517.html>]

[Get One Now](#)

[<https://www.seeedstudio.com/Base-Shield-V2-p-1378.html>]

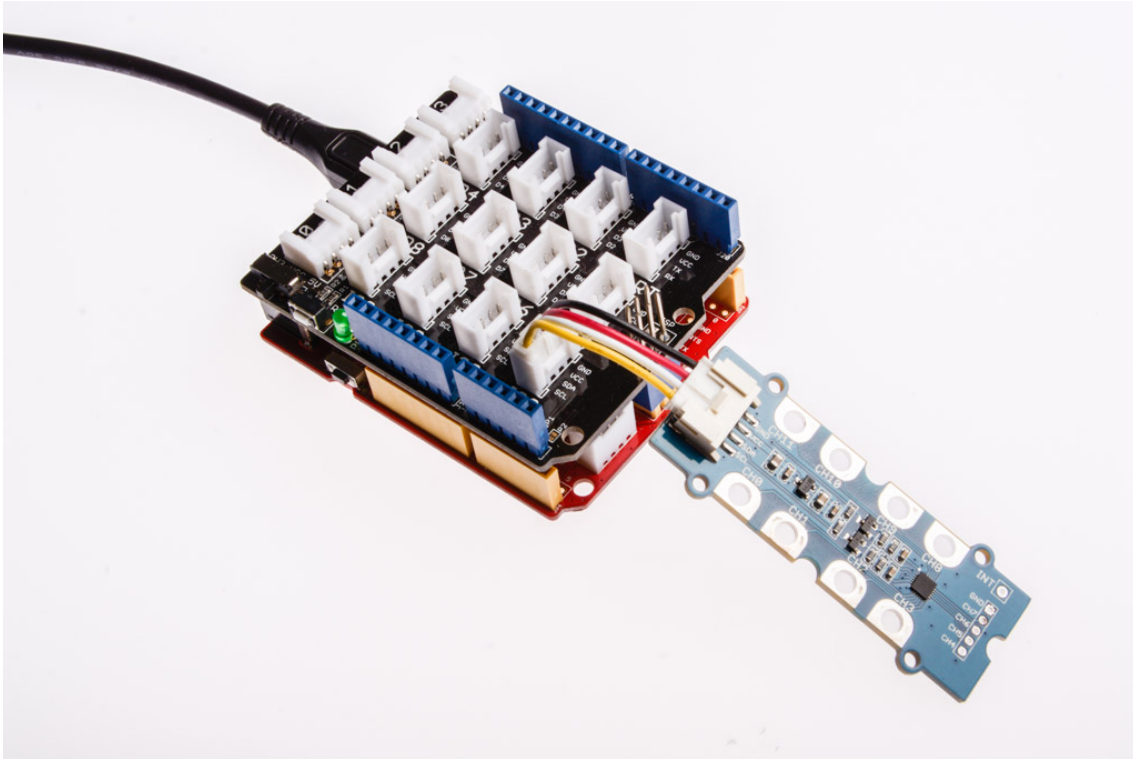


Note

1 Please plug the USB cable gently, otherwise you may damage the port. Please use the USB cable with 4 wires inside, the 2 wires cable can't transfer data. If you are not sure about the wire you have, you can click [here](https://www.seeedstudio.com/Micro-USB-Cable-48cm-p-1475.html) [<https://www.seeedstudio.com/Micro-USB-Cable-48cm-p-1475.html>] to buy

2 Each Grove module comes with a Grove cable when you buy. In case you lose the Grove cable, you can click [here](https://www.seeedstudio.com/Grove-Universal-4-Pin-Buckled-20cm-Cable-%285-PCs-pack%29-p-936.html) [<https://www.seeedstudio.com/Grove-Universal-4-Pin-Buckled-20cm-Cable-%285-PCs-pack%29-p-936.html>] to buy.

- **Step 1.** Connect the Grove - 12 Key Capacitive I2C Touch Sensor V2 (MPR121) to the **I²C** port of the Base Shield.
- **Step 2.** Plug Grove - Base Shield into Seeeduino.
- **Step 3.** Connect Seeeduino to PC via a USB cable.

**Note**

If we don't have Grove Base Shield, We also can directly connect this module to Seeeduino as below.

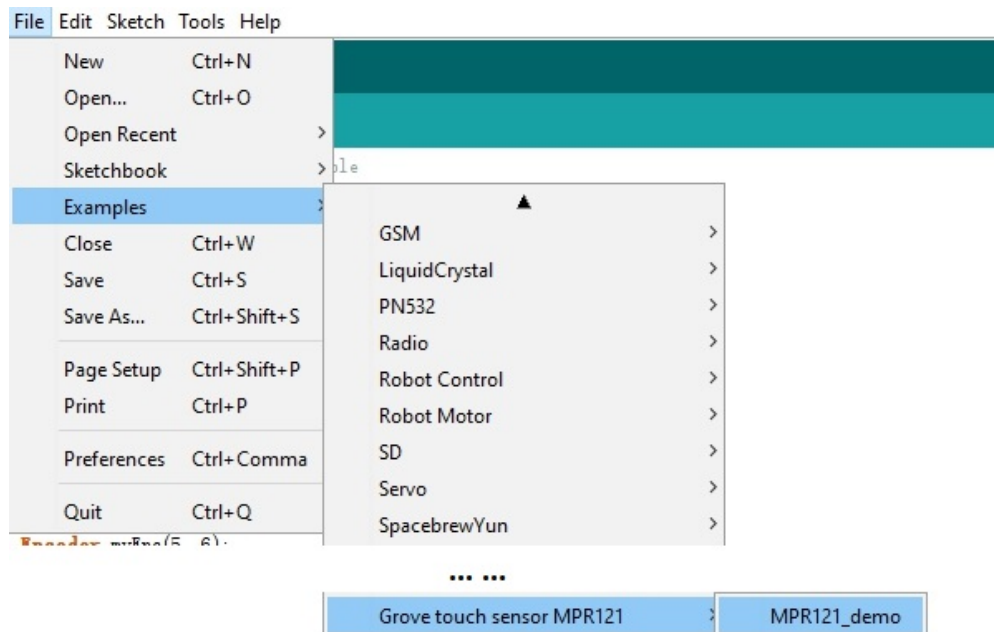
Seeeduino	Grove-MPR121
5V	Red
GND	Black
SDA	White
SCL	Yellow

Software**Note**

If this is the first time you work with Arduino, we strongly recommend you to see [Getting Started with Arduino](https://wiki.seeedstudio.com/Getting_Started_with_Arduino/) [https://wiki.seeedstudio.com/Getting_Started_with_Arduino/] before the start.

- **Step 1.** Download the [Grove touch sensor MPR121](https://github.com/linux-downey/Grove_touch_sensor_MPR121) [https://github.com/linux-downey/Grove_touch_sensor_MPR121] Library from Github.
- **Step 2.** Refer to [How to install library](https://wiki.seeedstudio.com/How_to_install_Arduino_Library) [https://wiki.seeedstudio.com/How_to_install_Arduino_Library] to install library for Arduino.
- **Step 3.** Restart the Arduino IDE. Open the example, you can open it in the following three ways:


- Open it directly in the Arduino IDE via the path: **File** → **Examples** → **Grove touch sensor MPR121** → **MPR121_demo.**




- Open it in your computer by click the **MPR121_demo.ino** which you can find in the **xxx\Arduino\libraries\Grove_touch_sensor_MPR121-**

master, XXXX is the location you installed the Arduino IDE.

s PC > Core (C:) > Users > seeed > Documents > Arduino > libraries > Grove_touch_sensor_MPR121-master > examples > MPR121_demo

Name	Date modified	Type	Size
 MPR121_demo.ino	9/6/2018 2:26 PM	INO File	3 KB

- c. Or, you can just click the icon  in upper right corner of the code block to copy the following code into a new sketch in the Arduino IDE.

```

1  #include "Seeed_MPR121_driver.h"
2
3  Mpr121 mpr121;
4
5  u16 touch_status_flag[CHANNEL_NUM]={0};
6  void setup()
7  {
8      s32 ret=0;
9      Serial.begin(115200);
10     if(mpr121.begin())<0)
11     {
12         Serial.println("Can't detect device!!!!");
13     }
14     else
15     {
16         Serial.println("mpr121 init OK!");
17     }
18     delay(100);
19 }
20 void loop()
21 {
22     u16 result=0;
23     u16 filtered_data_buf[CHANNEL_NUM]={0};
24     u8 baseline_buf[CHANNEL_NUM]={0};
25
26     result=mpr121.check_status_register();
27
28     mpr121.get_filtered_reg_data(&result,filtered_data_buf
29

```

```

30   for(int i=0;i<CHANNEL_NUM;i++)
31   {
32     if(result&(1<<i))                               /*key
33     {
34       if(0==touch_status_flag[i])
35       {
36         touch_status_flag[i]=1;
37         Serial.print("key ");
38         Serial.print(i);
39         Serial.println("pressed");
40       }
41     }
42     else
43     {
44       if(1==touch_status_flag[i])
45       {
46         touch_status_flag[i]=0;
47         Serial.print("key ");
48         Serial.print(i);
49         Serial.println("release");
50       }
51     }
52   }
53   delay(50);
54 }

```

- **Step 4.** Upload the demo. If you do not know how to upload the code, please check [How to upload code](https://wiki.seeedstudio.com/Upload_Code/) [https://wiki.seeedstudio.com/Upload_Code/].
- **Step 5.** Open the **Serial Monitor** of Arduino IDE by click **Tool->Serial Monitor**. Or tap the **Ctrl + Shift + M** key at the same time. Set the baud rate to **115200**.



Success

If everything goes well, you will get the result. When you touch the CH0 ~ CH11 pads, it will trigger **key ?pressed** and **key ?release**


```

1 mpr121 inmpr121 init OK!
2 key 11pressed
3 key 11release
4 key 10pressed
5 key 10release
6 key 0pressed
7 key 0release
8 key 2pressed
9 key 2release


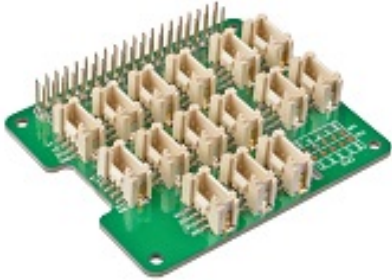
```



Play With Raspberry Pi

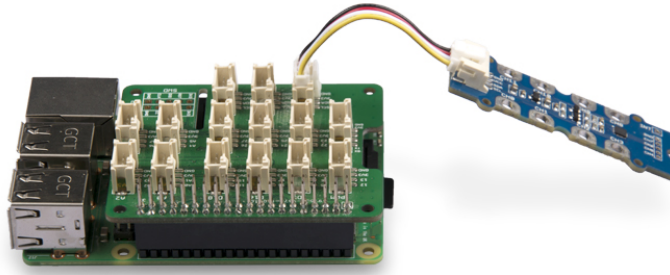
Hardware

- **Step 1.** Things used in this project:

Raspberry pi	Grove Base Hat for RasPi
	
<p>Get ONE Now [https://www.seeedstudio.com/Raspberry-Pi-3-Model-B-p-2625.html]</p>	<p>Get ONE Now [https://www.seeedstudio.com/Grove-Base-Hat-for-Raspberry-Pi-p-3186.html]</p>

- **Step 2.** Plug the Grove Base Hat into Raspberry.

- **Step 3.** Connect the Grove - 12 Key Capacitive I2C Touch Sensor V2 (MPR121) to the **I²C** port of the Base Hat.
- **Step 4.** Connect the Raspberry Pi to PC through USB cable.



Software



Attention

If you are using **Raspberry Pi with Raspberrypi OS >= Bullseye**, you have to use this command line **only with Python3**.

- **Step 1.** Follow [Setting Software](https://wiki.seeedstudio.com/Grove_Base_Hat_for_Raspberry_Pi/#installation) [https://wiki.seeedstudio.com/Grove_Base_Hat_for_Raspberry_Pi/#installation] to configure the development environment.
- **Step 2.** Download the source file by cloning the grove.py library.

```
1 cd ~
2 git clone https://github.com/Seeed-Studio/grove.py
```



- **Step 3.** Excute below command to run the code.

```
1 cd grove.py/grove
2 python3 grove_12_chan_touch_sensor_MPR121.py
```

Following is the grove_12_chan_touch_sensor_MPR121.py code.

```
1 import time
2 from grove.i2c import Bus
3
4
5 TOUCH_SENSOR_DEFAULT_ADDR = 0x5b
6
7 MODE_CONFIG_REG_ADDR = 0x5e
8 GLOBAL_PARAM_REG_ADDR_L = 0x5c
9 TOUCH_STATUS_REG_ADDR_L = 0x00
10 SET_DEBOUNCE_REG_ADDR = 0x5b
11
12 FILTERED_DATA_REG_START_ADDR_L = 0x04
13 CHANNEL_NUM = 12
14
15 STOP_MODE = 0
16 NORMAL_MODE = 0x3c
17
18 class TouchSensorMpr121():
19     def __init__(self, bus_num = 1, addr = TOUCH_SENSOR_DEI
20         self.bus = Bus(bus_num)
21         self.addr = addr
22         self.threshold = 0
23         self.touch_flag = [0]*CHANNEL_NUM
24
25     def sensor_init(self):
26         self._set_mode(STOP_MODE)
27         data = [0x23, 0x10]
28         self._set_global_param(data)
29         self._set_debounce(0x22)
30         self._set_mode(NORMAL_MODE)
31
```

```
32     def set_threshold(self, threshold):
33         self.threshold = threshold
34
35     def wait_for_ready(self):
36         time.sleep(.2)
37
38     def _set_mode(self, mode):
39         self.bus.write_byte_data(self.addr, MODE_CONFIG_R
40
41     def _set_global_param(self, data):
42         self.bus.write_i2c_block_data(self.addr, GLOBAL_P
43
44     def _set_debounce(self, data):
45         self.bus.write_byte_data(self.addr, SET_DEBOUNCE_
46
47     def _check_status_register(self):
48         data_status = self.bus.read_i2c_block_data(self.
49         return data_status
50
51     def get_filtered_touch_data(self, sensor_status):
52         result_value = []
53         for i in range(CHANNEL_NUM):
54             time.sleep(.01)
55             if(sensor_status & (1<<i)):
56                 channel_data = self.bus.read_i2c_block_d
57                 result_value.append(channel_data[0] | ch
58             else:
59                 result_value.append(0)
60         return result_value
61
62     def listen_sensor_status(self):
63         data = self._check_status_register()
64         touch_status = data[0] | (data[1]<<8)
65         touch_result_value = self.get_filtered_touch_dat
66
67         for i in range(CHANNEL_NUM):
68             if(touch_result_value[i] < self.threshold ):
69                 touch_result_value[i] = 0
70         return touch_result_value
71
72     def parse_and_print_result(self, result):
```

```

73     for i in range(CHANNEL_NUM):
74         if(result[i] != 0):
75             if(0 == self.touch_flag[i]):
76                 self.touch_flag[i] = 1
77                 print("Channel %d is pressed,value is %d" % (i, result[i]))
78             else:
79                 if(1 == self.touch_flag[i]):
80                     self.touch_flag[i] = 0
81                     print("Channel %d is released,value is %d" % (i, result[i]))
82
83
84
85 mpr121 = TouchSensorMpr121()
86 def main():
87     mpr121.sensor_init()
88     mpr121.set_threshold(0x60)
89     mpr121.wait_for_ready()
90     while 1:
91         result = mpr121.listen_sensor_status()
92         mpr121.parse_and_print_result(result)
93         time.sleep(.1)
94
95 if __name__ == '__main__':
96     main()

```



Success

If everything goes well, you will get the result. When you touch the CH0 ~ CH11 pads, it will trigger **channel # pressed** and **Channel # released** with corresponding pressure values.

```

1  >>> %Run grove_12_chan_touch_sensor_MPR121.py
2      Channel 8 is pressed, value is 308
3      Channel 8 is released, value is 0
4      Channel 9 is pressed, value is 170
5      Channel 9 is released, value is 0
6      Channel 10 is pressed, value is 340
7      Channel 8 is pressed, value is 180

```



Schematic Online Viewer



Resources

- **[Zip]** [Grove - 12 Key Capacitive I2C Touch Sensor V2 eagle files](https://files.seeedstudio.com/wiki/Grove-12_Key_Capacitive_I2C_Touch_Sensor_V2-MPR121/res/Grove-12_Key_Capacitive_I2C_Touch_Sensor_V2-MPR121.zip)
[https://files.seeedstudio.com/wiki/Grove-12_Key_Capacitive_I2C_Touch_Sensor_V2-MPR121/res/Grove-12_Key_Capacitive_I2C_Touch_Sensor_V2-MPR121.zip]

- **[Zip]** [Grove touch sensor MPR121 Library](https://github.com/linux-downey/Grove_touch_sensor_MPR121/archive/master.zip)
[https://github.com/linux-downey/Grove_touch_sensor_MPR121/archive/master.zip]
- **[PDF]** [Datasheet of MPR121](https://files.seeedstudio.com/wiki/Grove-12_Key_Capacitive_I2C_Touch_Sensor_V2-MPR121/res/MPR121.pdf)
[https://files.seeedstudio.com/wiki/Grove-12_Key_Capacitive_I2C_Touch_Sensor_V2-MPR121/res/MPR121.pdf]
- **[PDF]** [AN3894](https://files.seeedstudio.com/wiki/Grove-12_Key_Capacitive_I2C_Touch_Sensor_V2-MPR121/res/AN3894.pdf) [https://files.seeedstudio.com/wiki/Grove-12_Key_Capacitive_I2C_Touch_Sensor_V2-MPR121/res/AN3894.pdf]

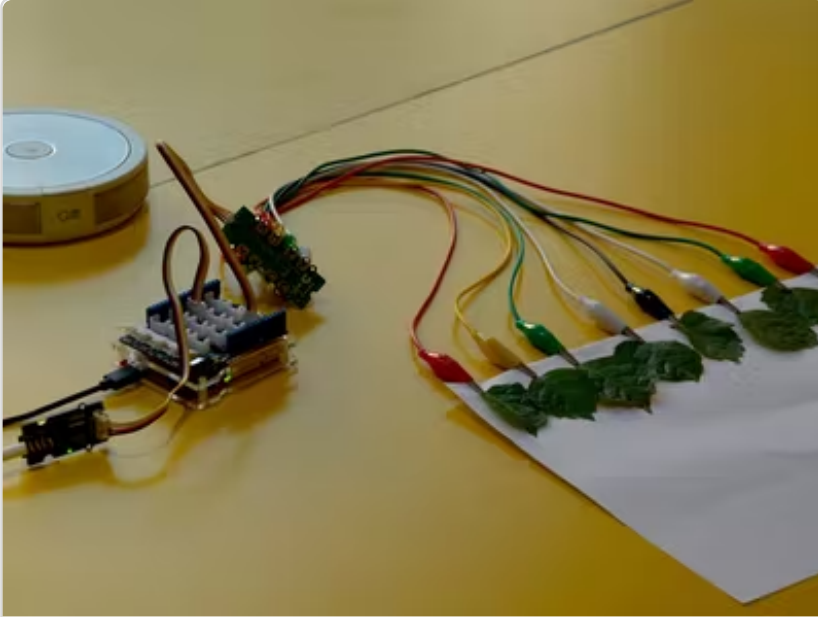
Project

This is the introduction Video of this product, simple demos, you can have a try.

All new Grove - Optical and Touch Sensor Module...

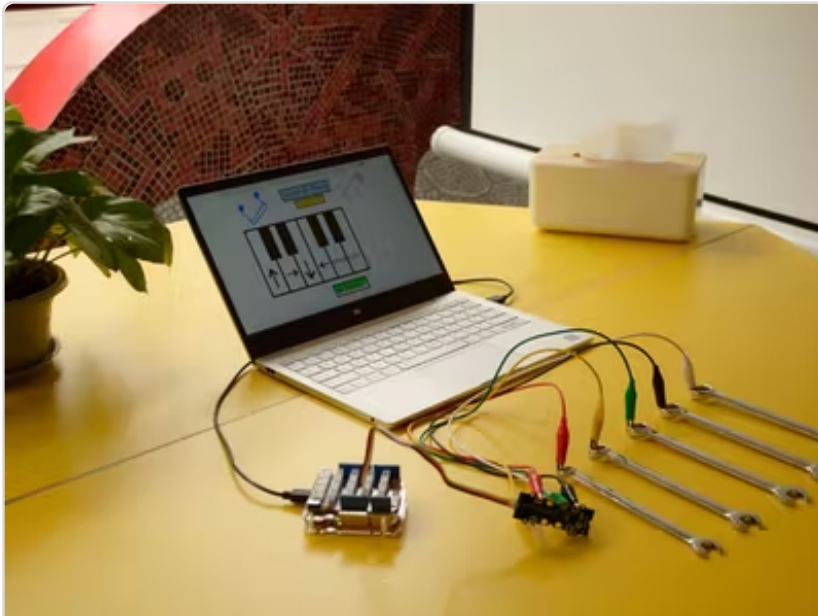


Leaf Piano: We made a piano using a touch sensor, as well as leaves for piano keys.



(<https://www.hackster.io/SeeedStudio/leaf-piano-5261a6>)

Play with Scratch: How to play a scratch game with a Touch Sensor?



(<https://www.hackster.io/SeeedStudio/play-mario-using-new-grove-touch-sensor-h3f9fc>)

Tech Support

Please do not hesitate to submit the issue into our [forum](https://forum.seeedstudio.com/) [<https://forum.seeedstudio.com/>].



[https://www.seeedstudio.com/act-4.html?utm_source=wiki&utm_medium=wikibanner&utm_campaign=newproducts]

