

more with a simple swipe of your hand.

Light Up Your Valentine with Infinite Matrix Sparkle



Get One Now 

[<https://www.seeedstudio.com/Grove-Gesture-p-2463.html>]

Features

- Built-in proximity detection
- Various main boards support : Arduino UNO/Seeeduino/Arduino Mega2560

- 9 Basic gestures
 - Up
 - Down
 - Left
 - Right
 - Forward
 - Backward
 - Clockwise
 - Count Clockwise
 - Wave

**Tip**

More details about Grove modules please refer to [Grove System](https://wiki.seeedstudio.com/Grove_System/)
[https://wiki.seeedstudio.com/Grove_System/]

Specification

Spec Name	Value
Sensor	PAJ7620U2
Power supply	5V
Ambient light immunity	< 100k Lux
Gesture speed in Normal Mode	60°/s to 600°/s
Gesture speed in Gaming Mode	60°/s to 1200°/s
Interface type IIC interface	up to 400 kbit/s
Operating Temperature	-40°C to +85°C
Dimensions	20 * 20mmv
Detection range	5-10cm
I2C Address	0x73

**Note**



If you want to use multiplue I2C devices, please refer to [Software I2C](https://wiki.seeedstudio.com/Arduino_Software_I2C_user_guide/) [https://wiki.seeedstudio.com/Arduino_Software_I2C_user_guide/].

Getting started

Play With Arduino

Hardware

- **Step 1.** Prepare the below stuffs:

Seeeduino V4.2	Base Shield
	
<p>Get One Now [https://www.seeedstudio.com/Seeeduino-V4.2-p-2517.html]</p>	<p>Get One Now [https://www.seeedstudio.com/Base-Shield-V2-p-1378.html]</p>

- **Step 2.** Connect Grove-Gesture_v1.0 to port I2C of Grove-Base Shield.
- **Step 3.** Plug Grove - Base Shield into Seeeduino.
- **Step 4.** Connect Seeeduino to PC via a USB cable.

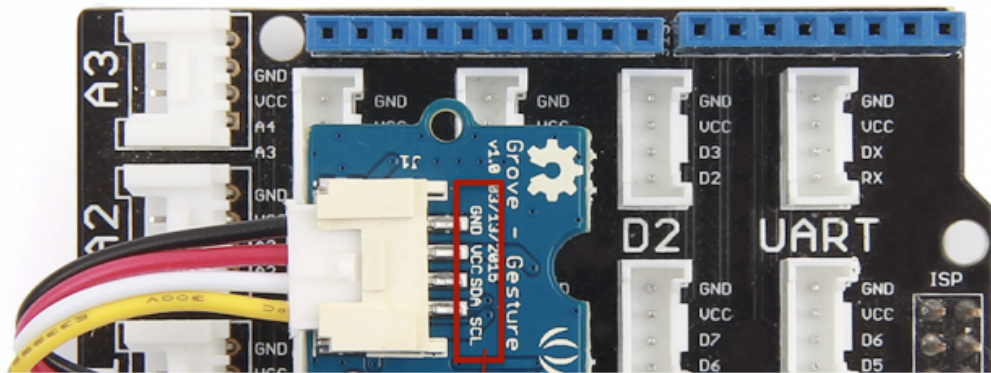
**Note**

If we don't have Grove Base Shield, We also can directly connect Grove-Gesture v1.0 to Seeeduino as below.

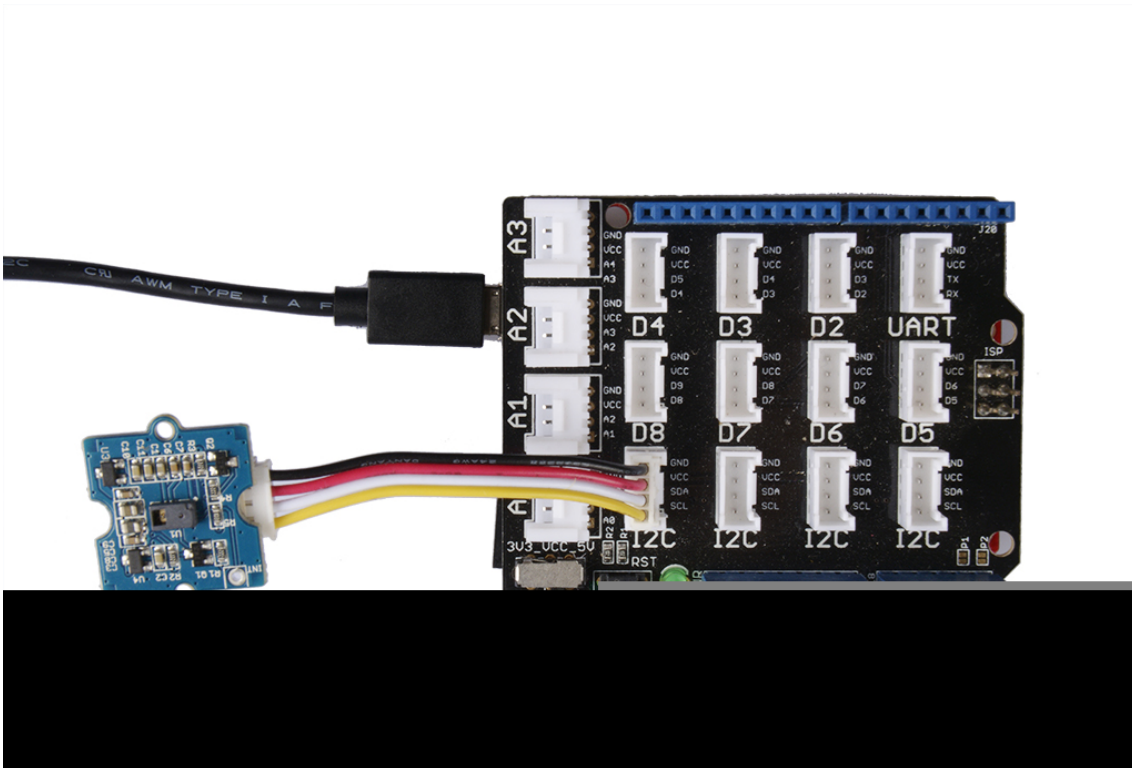
Seeeduino	Grove-Gesture v1.0
5V	VCC
GND	GND
SDA	SDA
SCL	SCL

INT: Gesture detection interrupt flag mask. You can connect INT pad to digit 2 of Arduino by using jumper wire.

Below image shows how to plug Grove - Gesture onto the I2C port of Base shield



Then plug Base shield onto Seeduino



Software

- **Step 1.** Download the [library](https://github.com/Seeed-Studio/Gesture_PAJ7620) [https://github.com/Seeed-Studio/Gesture_PAJ7620] from Github.
- **Step 2.** Refer [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. Software Installation.
- **Step 3.** 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 4.** Start the Arduino IDE, Let's try the up/down demo.
- **Step 5.** Copy below code and paste to arduino IDE.

```

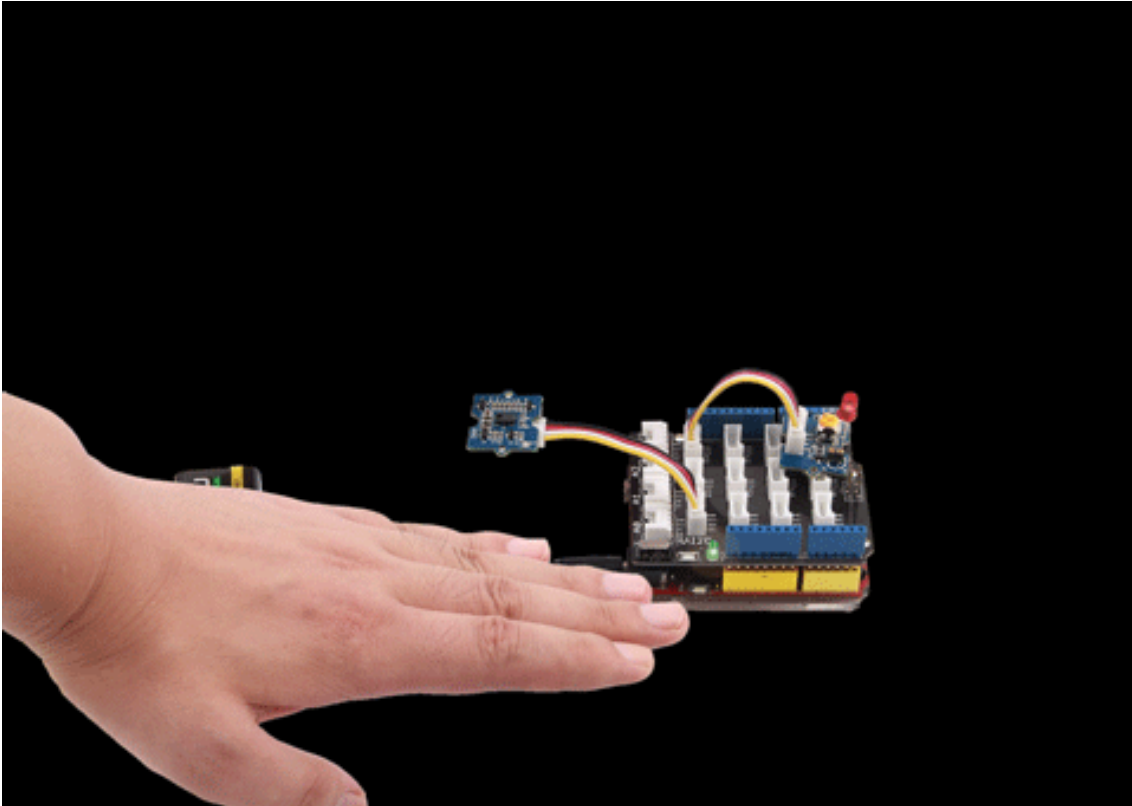
1  /*
2  The following simple demo will show you a very easy appl
3  */
4  #include <Wire.h>

```



```
5  #include "paj7620.h"
6
7  void setup()
8  {
9      paj7620Init();
10 }
11
12 void loop()
13 {
14     uint8_t data = 0; // Read Bank_0_Reg_0x43/0x44 for
15
16     paj7620ReadReg(0x43, 1, &data); // When different g
17
18     if (data == GES_UP_FLAG) // When up gesture b
19         digitalWrite(4, HIGH); // turn the LED on (i
20     if (data == GES_DOWN_FLAG) // When down gesture
21         digitalWrite(4, LOW); // turn the LED off i
22 }
```

- **Step 6.** Select Seeeduino V4 in board manager and COM port.
- **Step 7.** Click upload and put the hand over the Gesture sensor up and down. You will see the led on and off as below.



- **Step 8.** Let's try the 9 gestures by select **File->Example->Gesture_PAJ7620->paj7620_9gestures**, then select Seeeduino v4 as board manager and correct COM port.

```
1  #include <Wire.h>
2  #include "paj7620.h"
3
4  /*
5   Notice: When you want to recognize the Forward/Backward
6           You also can adjust the reaction time according
7   */
8  #define GES_REACTION_TIME      500           // You
9  #define GES_ENTRY_TIME        800           // When
10 #define GES_QUIT_TIME          1000
11
12 void setup()
13 {
14     uint8_t error = 0;
15
16     Serial.begin(9600);
```

```
17     Serial.println("\nPAJ7620U2 TEST DEMO: Recognize 9 |
18
19     error = paj7620Init();           // initialize Paj76.
20     if (error)
21     {
22         Serial.print("INIT ERROR, CODE:");
23         Serial.println(error);
24     }
25     else
26     {
27         Serial.println("INIT OK");
28     }
29     Serial.println("Please input your gestures:\n");
30 }
31
32 void loop()
33 {
34     uint8_t data = 0, data1 = 0, error;
35
36     error = paj7620ReadReg(0x43, 1, &data);
37     if (!error)
38     {
39         switch (data)
40         {
41             case GES_RIGHT_FLAG:
42                 delay(GES_ENTRY_TIME);
43                 paj7620ReadReg(0x43, 1, &data);
44                 if(data == GES_FORWARD_FLAG)
45                 {
46                     Serial.println("Forward");
47                     delay(GES_QUIT_TIME);
48                 }
49                 else if(data == GES_BACKWARD_FLAG)
50                 {
51                     Serial.println("Backward");
52                     delay(GES_QUIT_TIME);
53                 }
54                 else
55                 {
56                     Serial.println("Right");
57                 }
```

```
58         break;
59     case GES_LEFT_FLAG:
60         delay(GES_ENTRY_TIME);
61         paj7620ReadReg(0x43, 1, &data);
62         if(data == GES_FORWARD_FLAG)
63         {
64             Serial.println("Forward");
65             delay(GES_QUIT_TIME);
66         }
67         else if(data == GES_BACKWARD_FLAG)
68         {
69             Serial.println("Backward");
70             delay(GES_QUIT_TIME);
71         }
72         else
73         {
74             Serial.println("Left");
75         }
76         break;
77     case GES_UP_FLAG:
78         delay(GES_ENTRY_TIME);
79         paj7620ReadReg(0x43, 1, &data);
80         if(data == GES_FORWARD_FLAG)
81         {
82             Serial.println("Forward");
83             delay(GES_QUIT_TIME);
84         }
85         else if(data == GES_BACKWARD_FLAG)
86         {
87             Serial.println("Backward");
88             delay(GES_QUIT_TIME);
89         }
90         else
91         {
92             Serial.println("Up");
93         }
94         break;
95     case GES_DOWN_FLAG:
96         delay(GES_ENTRY_TIME);
97         paj7620ReadReg(0x43, 1, &data);
98         if(data == GES_FORWARD_FLAG)
```

```
99         {
100             Serial.println("Forward");
101             delay(GES_QUIT_TIME);
102         }
103         else if(data == GES_BACKWARD_FLAG)
104         {
105             Serial.println("Backward");
106             delay(GES_QUIT_TIME);
107         }
108         else
109         {
110             Serial.println("Down");
111         }
112         break;
113     case GES_FORWARD_FLAG:
114         Serial.println("Forward");
115         delay(GES_QUIT_TIME);
116         break;
117     case GES_BACKWARD_FLAG:
118         Serial.println("Backward");
119         delay(GES_QUIT_TIME);
120         break;
121     case GES_CLOCKWISE_FLAG:
122         Serial.println("Clockwise");
123         break;
124     case GES_COUNT_CLOCKWISE_FLAG:
125         Serial.println("anti-clockwise");
126         break;
127     default:
128         paj7620ReadReg(0x44, 1, &data1);
129         if (data1 == GES_WAVE_FLAG)
130         {
131             Serial.println("wave");
132         }
133         break;
134     }
135 }
136 delay(100);
137 }
```

- **Step 9.** Click upload and open the Serial Monitor port.
- **Step 10.** This example can recognize 9 gestures and output the result, including move up, move down, move left, move right, move forward, move backward, circle-clockwise, circle-counter clockwise, and wave. Let's try and see COM port output as below.

```
1 PAJ7620U2 TEST DEMO: Recognize 9 gestures.
2 INIT SENSOR...
3 Addr0 =20, Addr1 =76
4 wake-up finish.
5 Set up gaming mode.
6 Paj7620 initialize register finished.
7 INIT OK
8 Please input your gestures:
9
10 Left
11 Left
12 Up
13 Right
14 Up
15 Up
16 Down
17 Up
18 Down
19 Down
20 Up
21 Right
22 Left
```

**Note**

When you want to recognize the Forward/Backward gestures, your gestures' reaction time must less than GES_ENTRY_TIME(0.8s). You also can adjust the reaction time according to the actual circumstance.

Play with Codecraft

Hardware

Step 1. Connect a Grove - Gesture to I2C port, and connect a Grove - Red LED to port D4 of a Base Shield.

Step 2. Plug the Base Shield to your Seeeduino/Arduino.

Step 3. Link Seeeduino/Arduino to your PC via an USB cable.

Software

Step 1. Open [Codecraft](https://ide.chmakered.com/) [https://ide.chmakered.com/], add Arduino support, and drag a main procedure to working area.



Note

If this is your first time using Codecraft, see also [Guide for Codecraft using Arduino](https://wiki.seeedstudio.com/Guide_for_Codecraft_using_Arduino/) [https://wiki.seeedstudio.com/Guide_for_Codecraft_using_Arduino/].

Step 2. Drag blocks as picture below or open the cdc file which can be downloaded at the end of this page.



Upload the program to your Arduino/Seeeduino.




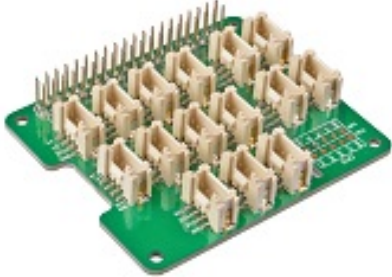
Success

When the code finishes uploaded, the LED will lights up when you make an up gesture, and it will be blacks out when you make a down gesture.

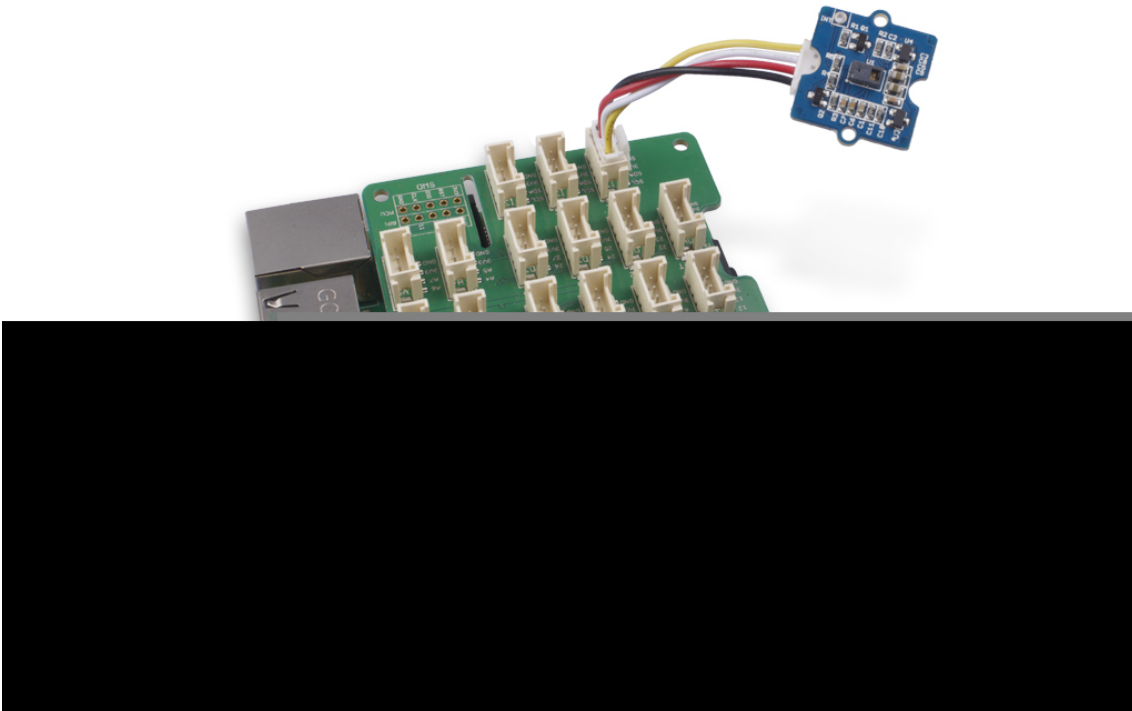
Play With Raspberry Pi (With Grove Base Hat for Raspberry Pi)

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 gesture sensor to the **I²C** port of the Base Hat.
- **Step 4.** Connect the Raspberry Pi to PC through USB cable.



Software

- **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 commands to run the code.

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

Following is the grove_gesture_sensor.py code.

```

1  import time,sys
2  import RPi.GPIO as GPIO
3  from grove.i2c import Bus
4
5  # use the bus that matches your raspi version
6  rev = GPIO.RPI_REVISION
7  if rev == 2 or rev == 3:
8      bus = Bus(1)
9  else:
10     bus = Bus(0)
11
12  class gesture:
13     #Registers and variables for the gesture sensor
14     GES_REACTION_TIME      =.500          # You c
15     GES_ENTRY_TIME        =.800          # When
16     GES_QUIT_TIME         =1.000
17
18     BANK0 = 0
19     BANK1 = 1
20
21     PAJ7620_ADDR_BASE =0x00
22
23     #REGISTER BANK SELECT
24     PAJ7620_REGITER_BANK_SEL      =(PAJ7620_ADDR_BASE
25
26     #DEVICE ID
27     PAJ7620_ID =0x73
28
29     #REGISTER BANK 0
30     PAJ7620_ADDR_SUSPEND_CMD      =(PAJ7620_ADDR_BASE
31     PAJ7620_ADDR_GES_PS_DET_MASK_0 =(PAJ7620_ADDR_BASE
32     PAJ7620_ADDR_GES_PS_DET_MASK_1 =(PAJ7620_ADDR_BASE
33     PAJ7620_ADDR_GES_PS_DET_FLAG_0 =(PAJ7620_ADDR_BASE
34     PAJ7620_ADDR_GES_PS_DET_FLAG_1 =(PAJ7620_ADDR_BASE
35     PAJ7620_ADDR_STATE_INDICATOR  =(PAJ7620_ADDR_BASE
36     PAJ7620_ADDR_PS_HIGH_THRESHOLD =(PAJ7620_ADDR_BASE
37     PAJ7620_ADDR_PS_LOW_THRESHOLD  =(PAJ7620_ADDR_BASE
38     PAJ7620_ADDR_PS_APPROACH_STATE =(PAJ7620_ADDR_BASE

```

```

39 PAJ7620_ADDR_PS_RAW_DATA      =(PAJ7620_ADDR_BASE
40
41 #REGISTER BANK 1
42 PAJ7620_ADDR_PS_GAIN          =(PAJ7620_ADDR_BASE
43 PAJ7620_ADDR_IDLE_S1_STEP_0  =(PAJ7620_ADDR_BASE
44 PAJ7620_ADDR_IDLE_S1_STEP_1  =(PAJ7620_ADDR_BASE
45 PAJ7620_ADDR_IDLE_S2_STEP_0  =(PAJ7620_ADDR_BASE
46 PAJ7620_ADDR_IDLE_S2_STEP_1  =(PAJ7620_ADDR_BASE
47 PAJ7620_ADDR_OP_TO_S1_STEP_0 =(PAJ7620_ADDR_BASE
48 PAJ7620_ADDR_OP_TO_S1_STEP_1 =(PAJ7620_ADDR_BASE
49 PAJ7620_ADDR_OP_TO_S2_STEP_0 =(PAJ7620_ADDR_BASE
50 PAJ7620_ADDR_OP_TO_S2_STEP_1 =(PAJ7620_ADDR_BASE
51 PAJ7620_ADDR_OPERATION_ENABLE =(PAJ7620_ADDR_BASE
52
53 #PAJ7620_REGITER_BANK_SEL
54 PAJ7620_BANK0=0
55 PAJ7620_BANK1=1
56
57 #PAJ7620_ADDR_SUSPEND_CMD
58 PAJ7620_I2C_WAKEUP =1
59 PAJ7620_I2C_SUSPEND =0
60
61 #PAJ7620_ADDR_OPERATION_ENABLE
62 PAJ7620_ENABLE=1
63 PAJ7620_DISABLE=0
64
65 #ADC, delete
66 REG_ADDR_RESULT = 0x00
67 REG_ADDR_ALERT  = 0x01
68 REG_ADDR_CONFIG = 0x02
69 REG_ADDR_LIMITL = 0x03
70 REG_ADDR_LIMITH = 0x04
71 REG_ADDR_HYST   = 0x05
72 REG_ADDR_CONVL  = 0x06
73 REG_ADDR_CONVH  = 0x07
74
75 GES_RIGHT_FLAG  =1<<0
76 GES_LEFT_FLAG   =1<<1
77 GES_UP_FLAG     =1<<2
78 GES_DOWN_FLAG   =1<<3
79 GES_FORWARD_FLAG =1<<4

```

```
80     GES_BACKWARD_FLAG           =1<<5
81     GES_CLOCKWISE_FLAG         =1<<6
82     GES_COUNT_CLOCKWISE_FLAG   =1<<7
83     GES_WAVE_FLAG              =1<<0
84
85     #Gesture output
86     FORWARD                    = 1
87     BACKWARD                   = 2
88     RIGHT                      = 3
89     LEFT                       = 4
90     UP                         = 5
91     DOWN                      = 6
92     CLOCKWISE                  = 7
93     ANTI_CLOCKWISE            = 8
94     WAVE                       = 9
95
96     #Initial register state
97     initRegisterArray=( [0xEF,0x00],
98                          [0x32,0x29],
99                          [0x33,0x01],
100                         [0x34,0x00],
101                         [0x35,0x01],
102                         [0x36,0x00],
103                         [0x37,0x07],
104                         [0x38,0x17],
105                         [0x39,0x06],
106                         [0x3A,0x12],
107                         [0x3F,0x00],
108                         [0x40,0x02],
109                         [0x41,0xFF],
110                         [0x42,0x01],
111                         [0x46,0x2D],
112                         [0x47,0x0F],
113                         [0x48,0x3C],
114                         [0x49,0x00],
115                         [0x4A,0x1E],
116                         [0x4B,0x00],
117                         [0x4C,0x20],
118                         [0x4D,0x00],
119                         [0x4E,0x1A],
120                         [0x4F,0x14],
```

```
121 [0x50,0x00],
122 [0x51,0x10],
123 [0x52,0x00],
124 [0x5C,0x02],
125 [0x5D,0x00],
126 [0x5E,0x10],
127 [0x5F,0x3F],
128 [0x60,0x27],
129 [0x61,0x28],
130 [0x62,0x00],
131 [0x63,0x03],
132 [0x64,0xF7],
133 [0x65,0x03],
134 [0x66,0xD9],
135 [0x67,0x03],
136 [0x68,0x01],
137 [0x69,0xC8],
138 [0x6A,0x40],
139 [0x6D,0x04],
140 [0x6E,0x00],
141 [0x6F,0x00],
142 [0x70,0x80],
143 [0x71,0x00],
144 [0x72,0x00],
145 [0x73,0x00],
146 [0x74,0xF0],
147 [0x75,0x00],
148 [0x80,0x42],
149 [0x81,0x44],
150 [0x82,0x04],
151 [0x83,0x20],
152 [0x84,0x20],
153 [0x85,0x00],
154 [0x86,0x10],
155 [0x87,0x00],
156 [0x88,0x05],
157 [0x89,0x18],
158 [0x8A,0x10],
159 [0x8B,0x01],
160 [0x8C,0x37],
161 [0x8D,0x00],
```

```
162 [0x8E,0xF0],
163 [0x8F,0x81],
164 [0x90,0x06],
165 [0x91,0x06],
166 [0x92,0x1E],
167 [0x93,0x0D],
168 [0x94,0x0A],
169 [0x95,0x0A],
170 [0x96,0x0C],
171 [0x97,0x05],
172 [0x98,0x0A],
173 [0x99,0x41],
174 [0x9A,0x14],
175 [0x9B,0x0A],
176 [0x9C,0x3F],
177 [0x9D,0x33],
178 [0x9E,0xAE],
179 [0x9F,0xF9],
180 [0xA0,0x48],
181 [0xA1,0x13],
182 [0xA2,0x10],
183 [0xA3,0x08],
184 [0xA4,0x30],
185 [0xA5,0x19],
186 [0xA6,0x10],
187 [0xA7,0x08],
188 [0xA8,0x24],
189 [0xA9,0x04],
190 [0xAA,0x1E],
191 [0xAB,0x1E],
192 [0xCC,0x19],
193 [0xCD,0x0B],
194 [0xCE,0x13],
195 [0xCF,0x64],
196 [0xD0,0x21],
197 [0xD1,0x0F],
198 [0xD2,0x88],
199 [0xE0,0x01],
200 [0xE1,0x04],
201 [0xE2,0x41],
202 [0xE3,0xD6],
```

```
203 [0xE4,0x00],
204 [0xE5,0x0C],
205 [0xE6,0x0A],
206 [0xE7,0x00],
207 [0xE8,0x00],
208 [0xE9,0x00],
209 [0xEE,0x07],
210 [0xEF,0x01],
211 [0x00,0x1E],
212 [0x01,0x1E],
213 [0x02,0x0F],
214 [0x03,0x10],
215 [0x04,0x02],
216 [0x05,0x00],
217 [0x06,0xB0],
218 [0x07,0x04],
219 [0x08,0x0D],
220 [0x09,0x0E],
221 [0x0A,0x9C],
222 [0x0B,0x04],
223 [0x0C,0x05],
224 [0x0D,0x0F],
225 [0x0E,0x02],
226 [0x0F,0x12],
227 [0x10,0x02],
228 [0x11,0x02],
229 [0x12,0x00],
230 [0x13,0x01],
231 [0x14,0x05],
232 [0x15,0x07],
233 [0x16,0x05],
234 [0x17,0x07],
235 [0x18,0x01],
236 [0x19,0x04],
237 [0x1A,0x05],
238 [0x1B,0x0C],
239 [0x1C,0x2A],
240 [0x1D,0x01],
241 [0x1E,0x00],
242 [0x21,0x00],
243 [0x22,0x00],
```



```
244 [0x23,0x00],
245 [0x25,0x01],
246 [0x26,0x00],
247 [0x27,0x39],
248 [0x28,0x7F],
249 [0x29,0x08],
250 [0x30,0x03],
251 [0x31,0x00],
252 [0x32,0x1A],
253 [0x33,0x1A],
254 [0x34,0x07],
255 [0x35,0x07],
256 [0x36,0x01],
257 [0x37,0xFF],
258 [0x38,0x36],
259 [0x39,0x07],
260 [0x3A,0x00],
261 [0x3E,0xFF],
262 [0x3F,0x00],
263 [0x40,0x77],
264 [0x41,0x40],
265 [0x42,0x00],
266 [0x43,0x30],
267 [0x44,0xA0],
268 [0x45,0x5C],
269 [0x46,0x00],
270 [0x47,0x00],
271 [0x48,0x58],
272 [0x4A,0x1E],
273 [0x4B,0x1E],
274 [0x4C,0x00],
275 [0x4D,0x00],
276 [0x4E,0xA0],
277 [0x4F,0x80],
278 [0x50,0x00],
279 [0x51,0x00],
280 [0x52,0x00],
281 [0x53,0x00],
282 [0x54,0x00],
283 [0x57,0x80],
284 [0x59,0x10],
```

```
285         [0x5A, 0x08],
286         [0x5B, 0x94],
287         [0x5C, 0xE8],
288         [0x5D, 0x08],
289         [0x5E, 0x3D],
290         [0x5F, 0x99],
291         [0x60, 0x45],
292         [0x61, 0x40],
293         [0x63, 0x2D],
294         [0x64, 0x02],
295         [0x65, 0x96],
296         [0x66, 0x00],
297         [0x67, 0x97],
298         [0x68, 0x01],
299         [0x69, 0xCD],
300         [0x6A, 0x01],
301         [0x6B, 0xB0],
302         [0x6C, 0x04],
303         [0x6D, 0x2C],
304         [0x6E, 0x01],
305         [0x6F, 0x32],
306         [0x71, 0x00],
307         [0x72, 0x01],
308         [0x73, 0x35],
309         [0x74, 0x00],
310         [0x75, 0x33],
311         [0x76, 0x31],
312         [0x77, 0x01],
313         [0x7C, 0x84],
314         [0x7D, 0x03],
315         [0x7E, 0x01])
316
317     #Enable debug message
318     debug=0
319
320     #Initialize the sensors
321     def init(self):
322         time.sleep(.001)
323         self.paj7620SelectBank(self.BANK0)
324         self.paj7620SelectBank(self.BANK0)
325
```

```

326     data0 = self.paj7620ReadReg(0, 1)[0]
327     data1 = self.paj7620ReadReg(1, 1)[0]
328     if self.debug:
329         print("data0:",data0,"data1:",data1)
330     if data0 != 0x20 :#or data1 <> 0x76:
331         print("Error with sensor")
332         #return 0xff
333     if data0 == 0x20:
334         print("wake-up finish.")
335
336     for i in range(len(self.initRegisterArray)):
337         self.paj7620WriteReg(self.initRegisterArray
338
339     self.paj7620SelectBank(self.BANK0)
340
341     print("Paj7620 initialize register finished.")
342
343     #Write a byte to a register on the Gesture sensor
344     def paj7620WriteReg(self,addr,cmd):
345         bus.write_word_data(self.PAJ7620_ID, addr, cmd)
346
347     #Select a register bank on the Gesture Sensor
348     def paj7620SelectBank(self,bank):
349         if bank==self.BANK0:
350             self.paj7620WriteReg(self.PAJ7620_REGITER_B
351
352     #Read a block of bytes of length "qty" starting at
353     def paj7620ReadReg(self,addr,qty):
354         return bus.read_i2c_block_data(self.PAJ7620_ID,
355
356     #Print the values from the gesture sensor
357     def print_gesture(self):
358         data=self.paj7620ReadReg(0x43,1)[0]
359         if data==self.GES_RIGHT_FLAG:
360             time.sleep(self.GES_ENTRY_TIME)
361             data=self.paj7620ReadReg(0x43, 1)[0]
362             if data == self.GES_FORWARD_FLAG:
363                 print("Forward")
364                 time.sleep(self.GES_QUIT_TIME)
365             elif data == self.GES_BACKWARD_FLAG:
366                 print("Backward")

```

```
367         time.sleep(self.GES_QUIT_TIME)
368     else:
369         print("Right")
370
371     elif data==self.GES_LEFT_FLAG:
372         time.sleep(self.GES_ENTRY_TIME)
373         data=self.paj7620ReadReg(0x43, 1)[0]
374         if data == self.GES_FORWARD_FLAG:
375             print("Forward")
376             time.sleep(self.GES_QUIT_TIME)
377         elif data == self.GES_BACKWARD_FLAG:
378             print("Backward")
379             time.sleep(self.GES_QUIT_TIME)
380         else:
381             print("Left")
382
383     elif data==self.GES_UP_FLAG:
384         time.sleep(self.GES_ENTRY_TIME)
385         data=self.paj7620ReadReg(0x43, 1)[0]
386         if data == self.GES_FORWARD_FLAG:
387             print("Forward")
388             time.sleep(self.GES_QUIT_TIME)
389         elif data == self.GES_BACKWARD_FLAG:
390             print("Backward")
391             time.sleep(self.GES_QUIT_TIME)
392         else:
393             print("Up")
394
395     elif data==self.GES_DOWN_FLAG:
396         time.sleep(self.GES_ENTRY_TIME)
397         data=self.paj7620ReadReg(0x43, 1)[0]
398         if data == self.GES_FORWARD_FLAG:
399             print("Forward")
400             time.sleep(self.GES_QUIT_TIME)
401         elif data == self.GES_BACKWARD_FLAG:
402             print("Backward")
403             time.sleep(self.GES_QUIT_TIME)
404         else:
405             print("Down")
406
407     elif data==self.GES_FORWARD_FLAG:
```

```

408         print("Forward")
409         time.sleep(self.GES_QUIT_TIME)
410
411     elif data==self.GES_BACKWARD_FLAG:
412         print("Backward")
413         time.sleep(self.GES_QUIT_TIME)
414
415     elif data==self.GES_CLOCKWISE_FLAG:
416         print("Clockwise")
417
418     elif data==self.GES_COUNT_CLOCKWISE_FLAG:
419         print("anti-clockwise")
420
421     else:
422         data1=self.paj7620ReadReg(0x44, 1)[0]
423         if (data1 == self.GES_WAVE_FLAG):
424             print("wave")
425
426     #Return a vlaue from the gestire sensor which can b
427     # 0:nothing
428     # 1:Forward
429     # 2:Backward
430     # 3:Right
431     # 4:Left
432     # 5:Up
433     # 6:Down
434     # 7:Clockwise
435     # 8:anti-clockwise
436     # 9:wave
437     def return_gesture(self):
438
439         data=self.paj7620ReadReg(0x43,1)[0]
440         if data==self.GES_RIGHT_FLAG:
441             time.sleep(self.GES_ENTRY_TIME)
442             data=self.paj7620ReadReg(0x43, 1)[0]
443             if data == self.GES_FORWARD_FLAG:
444                 return 1
445             time.sleep(self.GES_QUIT_TIME)
446             elif data == self.GES_BACKWARD_FLAG:
447                 return 2
448             time.sleep(self.GES_QUIT_TIME)

```

```
449         else:
450             return 3
451
452     elif data==self.GES_LEFT_FLAG:
453         time.sleep(self.GES_ENTRY_TIME)
454         data=self.paj7620ReadReg(0x43, 1)[0]
455         if data == self.GES_FORWARD_FLAG:
456             return 1
457             time.sleep(self.GES_QUIT_TIME)
458         elif data == self.GES_BACKWARD_FLAG:
459             return 2
460             time.sleep(self.GES_QUIT_TIME)
461         else:
462             return 4
463
464     elif data==self.GES_UP_FLAG:
465         time.sleep(self.GES_ENTRY_TIME)
466         data=self.paj7620ReadReg(0x43, 1)[0]
467         if data == self.GES_FORWARD_FLAG:
468             return 1
469             time.sleep(self.GES_QUIT_TIME)
470         elif data == self.GES_BACKWARD_FLAG:
471             return 2
472             time.sleep(self.GES_QUIT_TIME)
473         else:
474             return 5
475
476     elif data==self.GES_DOWN_FLAG:
477         time.sleep(self.GES_ENTRY_TIME)
478         data=self.paj7620ReadReg(0x43, 1)[0]
479         if data == self.GES_FORWARD_FLAG:
480             return 1
481             time.sleep(self.GES_QUIT_TIME)
482         elif data == self.GES_BACKWARD_FLAG:
483             return 2
484             time.sleep(self.GES_QUIT_TIME)
485         else:
486             return 6
487
488     elif data==self.GES_FORWARD_FLAG:
489         return 1
```

```

490         time.sleep(self.GES_QUIT_TIME)
491
492     elif data==self.GES_BACKWARD_FLAG:
493         return 2
494         time.sleep(self.GES_QUIT_TIME)
495
496     elif data==self.GES_CLOCKWISE_FLAG:
497         return 7
498
499     elif data==self.GES_COUNT_CLOCKWISE_FLAG:
500         return 8
501
502     else:
503         data1=self.paj7620ReadReg(0x44, 1)[0]
504         if (data1 == self.GES_WAVE_FLAG):
505             return 9
506         return 0
507
508 if __name__ == "__main__":
509     g=gesture()
510     g.init()
511     while True:
512         g.print_gesture()
513         time.sleep(.1)
514         # print g.return_gesture()
515         # time.sleep(.1)

```



Success

If everything goes well, you will be able to see the following result

```

1 pi@raspberrypi:~/grove.py/grove $ python3 grove_gesture_
2 wake-up finish.
3 Paj7620 initialize register finished.
4 Left
5 Forward
6 Left
7 Backward
8 Right

```

```
9 Up
10 Down
11 ^CTraceback (most recent call last):
12   File "grove_gesture_sensor.py", line 555, in <module>
13     time.sleep(.1)
14 KeyboardInterrupt
```

You can quit this program by simply press `Ctrl + C`.





Note

If you use the I2C tool to scan the I2C address of the grove module, you may find two or more address. 0x04 is the address of the *Grove Base Hat for Raspberry Pi*.

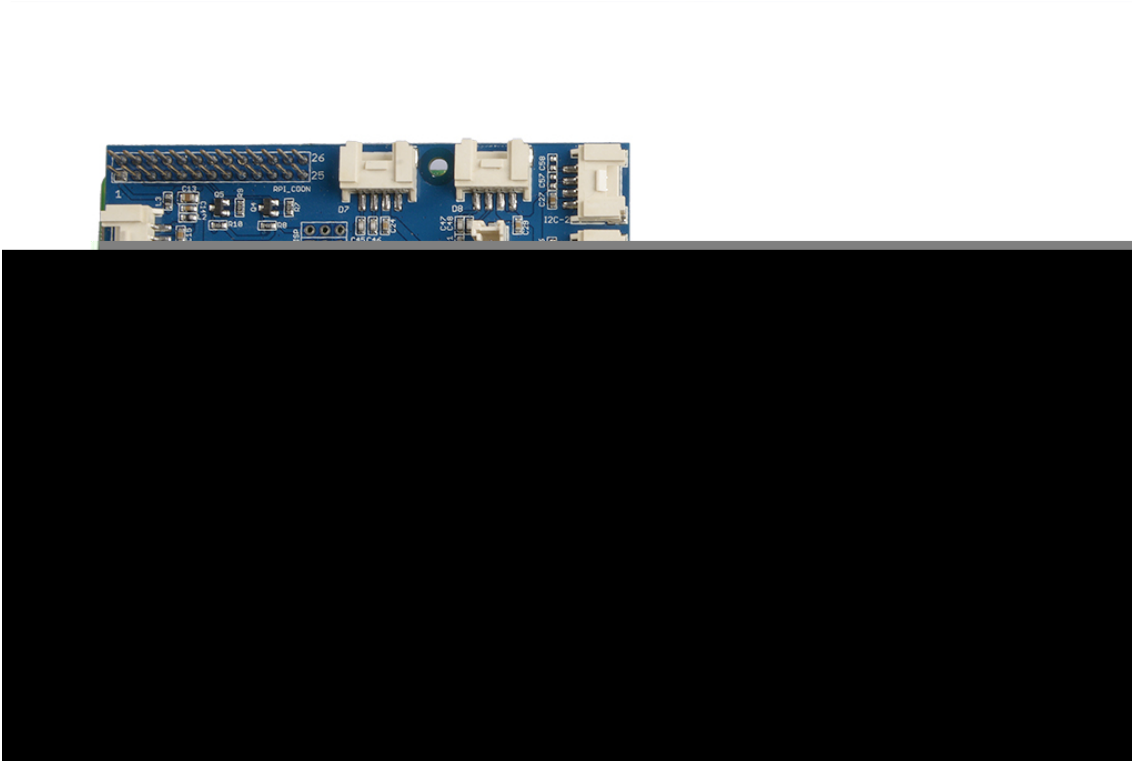
Play With Raspberry Pi (with GrovePi_Plus)

Hardware

- **Step 1.** Prepare the below stuffs:

Raspberry pi	GrovePi_Plus
	
<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/GrovePi-Plus-p-2241.html]</p>

- **Step 2.** Plug the GrovePi_Plus into Raspberry.
- **Step 3.** Connect Grove-Gesture_v1.0 to **I2C** port of GrovePi_Plus.
- **Step 4.** Connect the Raspberry to PC through USB cable.



Software

- **Step 1.** Follow [Setting Software](https://www.dexterindustries.com/GrovePi/get-started-with-the-grovepi/setting-software/) [https://www.dexterindustries.com/GrovePi/get-started-with-the-grovepi/setting-software/] to configure the development environment.
- **Step 2.** Git clone the Github repository.

```
1 cd ~
2 git clone https://github.com/DexterInd/GrovePi.git
```

- **Step 3.** Excute below commands to use this sensor

```
1 cd ~/GrovePi/Software/Python/grove_gesture_sensor
2 python3 gesture_value.py
```

Here is the code of gesture_value.py

```
1  #!/usr/bin/env python
2  #
3  # GrovePi Example for using the Grove - Gesture Sensor v.
4  #
5  # This example returns a value when a user does an action
6  #
7  # The GrovePi connects the Raspberry Pi and Grove sensor.
8  #
9  # Have a question about this example? Ask on the forums
10 #
11 '''
12 ## License
13 The MIT License (MIT)
14 GrovePi for the Raspberry Pi: an open source platform for
15 Copyright (C) 2017 Dexter Industries
16 Permission is hereby granted, free of charge, to any person
17 of this software and associated documentation files (the
18 "Software"), to use, copy, modify, merge, publish, distribute,
19 sublicense, and to permit persons to whom the Software is
20 furnished to do so, subject to the following conditions:
21 The above copyright notice and this permission notice shall
22 be included in all copies or substantial portions of the
23 Software.
24 THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY
25 KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE
26 WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR
27 PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
28 AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM,
29 DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF
30 CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
31 CONNECTION WITH THE SOFTWARE OR THE USE OR PERFORMANCE
32 OF THE SOFTWARE.
33 '''
34
35
36 import grove_gesture_sensor
37 import time
38
39 g=grove_gesture_sensor.gesture()
40 g.init()
41 while True:
42     gest=g.return_gesture()
43     #Match the gesture
```



```
41     if gest==g.FORWARD:
42         print("FORWARD")
43     elif gest==g.BACKWARD:
44         print("BACKWARD")
45     elif gest==g.RIGHT:
46         print("RIGHT")
47     elif gest==g.LEFT:
48         print("LEFT")
49     elif gest==g.UP:
50         print("UP")
51     elif gest==g.DOWN:
52         print("DOWN")
53     elif gest==g.CLOCKWISE:
54         print("CLOCKWISE")
55     elif gest==g.ANTI_CLOCKWISE:
56         print("ANTI_CLOCKWISE")
57     elif gest==g.WAVE:
58         print("WAVE")
59     elif gest==0:
60         print("-")
61     else:
62         print("Error")
63     time.sleep(.1)
```

- **Step 4.** Here is the screen output.

```
1 pi@raspberrypi:~/Dexter/GrovePi/Software/Python/grove_ge
2 wake-up finish.
3 Paj7620 initialize register finished.
4
5 -
6 -
7 DOWN
8 -
9 -
10 LEFT
11 -
12 -
13 -
```

```
14 -  
15 UP  
16 -  
17 -  
18 -  
19 -  
20 RIGHT  
21 -  
22 -  
23 UP  
24 -  
25 -  
26 CLOCKWISE  
27 -  
28 -  
29 CLOCKWISE
```

References

We define some register data of gesture, refer to the following table.

Gesture	Register Data	Register Address	If Ye
Up	data==GES_UP_FLAG	0x43	Gest dete
Down	data==GES_DOWN_FLAG	0x43	Gest dete
Left	data==GES_LEFT_FLAG	0x43	Gest dete
Right	data==GES_RIGHT_FLAG	0x43	Gest dete
Forward	data==GES_FORWARD_FLAG	0x43	Gest dete
Backward	data==GES_BACKWARD_FLAG	0x43	Gest dete
Clockwise	data==GES_CLOCKWISE_FLAG	0x43	Gest dete
Count Clockwise	data==GES_COUNT_CLOCKWISE_FLAG	0x43	Gest dete
Wave	data==GES_WAVE_FLAG	0x44	Gest dete



Schematic Online Viewer



Resources

- **[ZIP]** [Grove - Gesture_v1.0 sch pcb.zip](https://files.seeedstudio.com/wiki/Grove_Gesture_V_1.0/res/Grove_-_Gesture_v1.0_sch_pcb.zip)
[https://files.seeedstudio.com/wiki/Grove_Gesture_V_1.0/res/Grove_-_Gesture_v1.0_sch_pcb.zip]
- **[Datasheet]** [PAJ7620U2_Datasheet_V0.8_20140611.pdf](https://files.seeedstudio.com/wiki/Grove_Gesture_V_1.0/res/PAJ7620U2_Datasheet_V0.8_20140611.pdf)
[https://files.seeedstudio.com/wiki/Grove_Gesture_V_1.0/res/PAJ7620U2_Datasheet_V0.8_20140611.pdf]

- **[Library]** [Library Grove - Gesture](https://github.com/Seeed-Studio/Gesture_PAJ7620) [https://github.com/Seeed-Studio/Gesture_PAJ7620]
- **[Codecraft]** [CDC File](https://files.seeedstudio.com/wiki/Grove_Gesture_V_1.0/res/Grove_Gesture_CDC_File.zip) [https://files.seeedstudio.com/wiki/Grove_Gesture_V_1.0/res/Grove_Gesture_CDC_File.zip]

Project

Upcycling Workshop at Seeed x.factory [Episode 1]: Last weekend, we had an amazing upcycling hacking workshop with 20 makers, designers, and engineers from the community gathering at x.factory (I even listed it as my favorite workshop!).



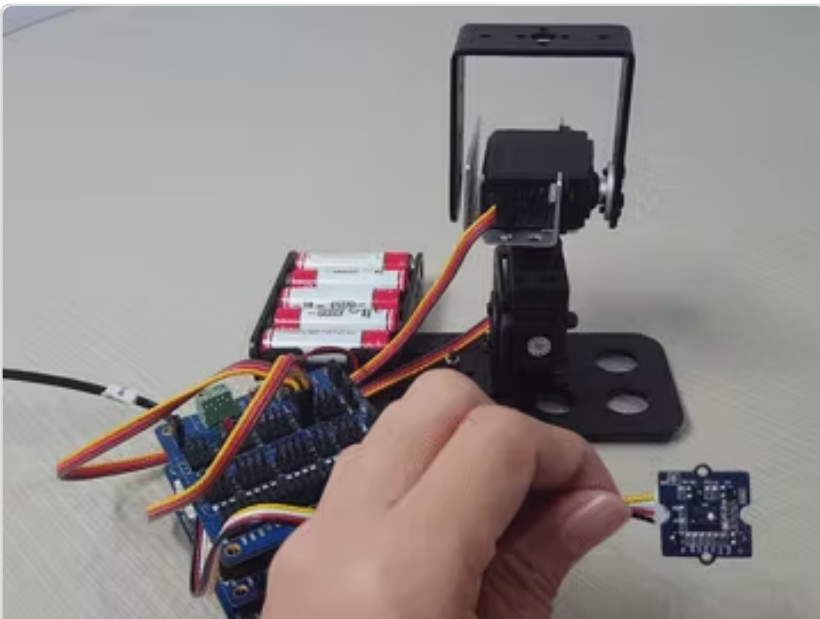
[https://blog.hackster.io/upcycling-workshop-at-seeed-x-factory-episode-1-4f2d5a61f5ad]

Jesty - Control your PC with Gestures: Control a PC with a gesture sensor, associating a key or a mouse click to each gesture.



(<https://www.hackster.io/Momy93/jesty-control-your-pc-with-gestures-5d4a3e>)

PHPoC - Control Pan-Tilt via Hand Gesture: An interesting way to control your robot: via hand gesture



(<https://www.hackster.io/Homer/phpoc-control-pan-tilt-via-hand-gesture-1daacc>)

Tech Support

Please submit any technical issue into our [forum](#)

[<https://forum.seeedstudio.com/>].



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