

Grove - LED Bar

Table of contents

Version

Features

Specification

Platforms Supported

Getting Started

 Play With Arduino

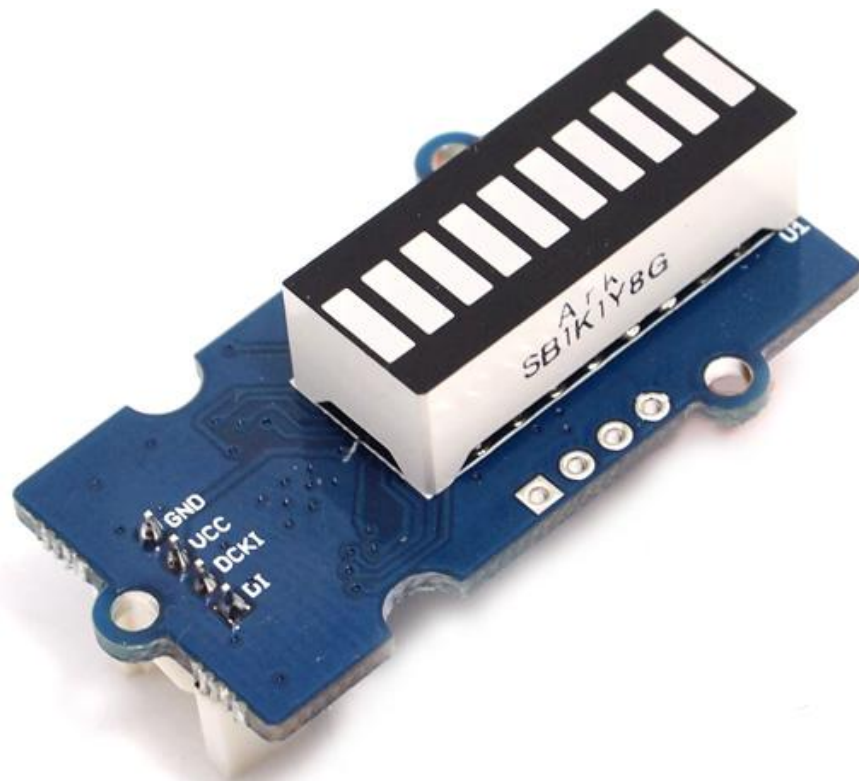
 Play With Raspberry Pi

Schematic Online Viewer

Resources

Projects

Tech Support



Grove – LED Bar is comprised of a 10 segment LED gauge bar and an MY9221 LED controlling chip. It can be used as an indicator for remaining battery life, voltage, water level, music volume or other values that require a gradient display. There are 10 LED bars in the LED bar graph: one red, one yellow, one light green, and seven green bars. Demo code is available to get you up and running quickly. It lights up the LEDs sequentially from red to green, so the entire bar graph is lit up in the end. Want to go further? Go ahead and code your own effect.

Version

Product Version	Changes	Released Date
Grove – LED Bar V1	Initial	June 2014
Grove – LED Bar V2	Improved the power supply	Oct 2015

Features

- Input Voltage: 3.3V/5V
- Each LED segment can be controlled individually via code
- Intuitive display
- Flexible power option, supports 3-5.5DC
- Available demo code
- Suli-compatible Library



Tip

More details about Grove modules please refer to [Grove System](#)

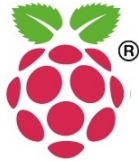
Specification

Parameter	Value/Range
Operating voltage	3.3V
Operation Temperature	-20°C to +80°C
Peak Emission Wavelength-RED(Current 20mA)	630-637nm
Peak Emission Wavelength-Yellow Green(Current 20mA)	570-573nm
Peak Emission Wavelength-Yellow(Current 20mA)	585-592nm
Luminous Intensity Per Segment-RED(Current 20mA)	50-70mcd
Luminous Intensity Per Segment-Yellow Green(Current 20mA)	28-35mcd
Luminous Intensity Per Segment-Yellow(Current 20mA)	45-60mcd
LED segment	10
Size	40mm * 20mm

Platforms Supported

Arduino

Raspberry Pi



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




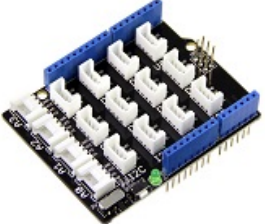

Note

If this is the first time you work with Arduino, we firmly recommend you to see [Getting Started with Arduino](#) before the start.

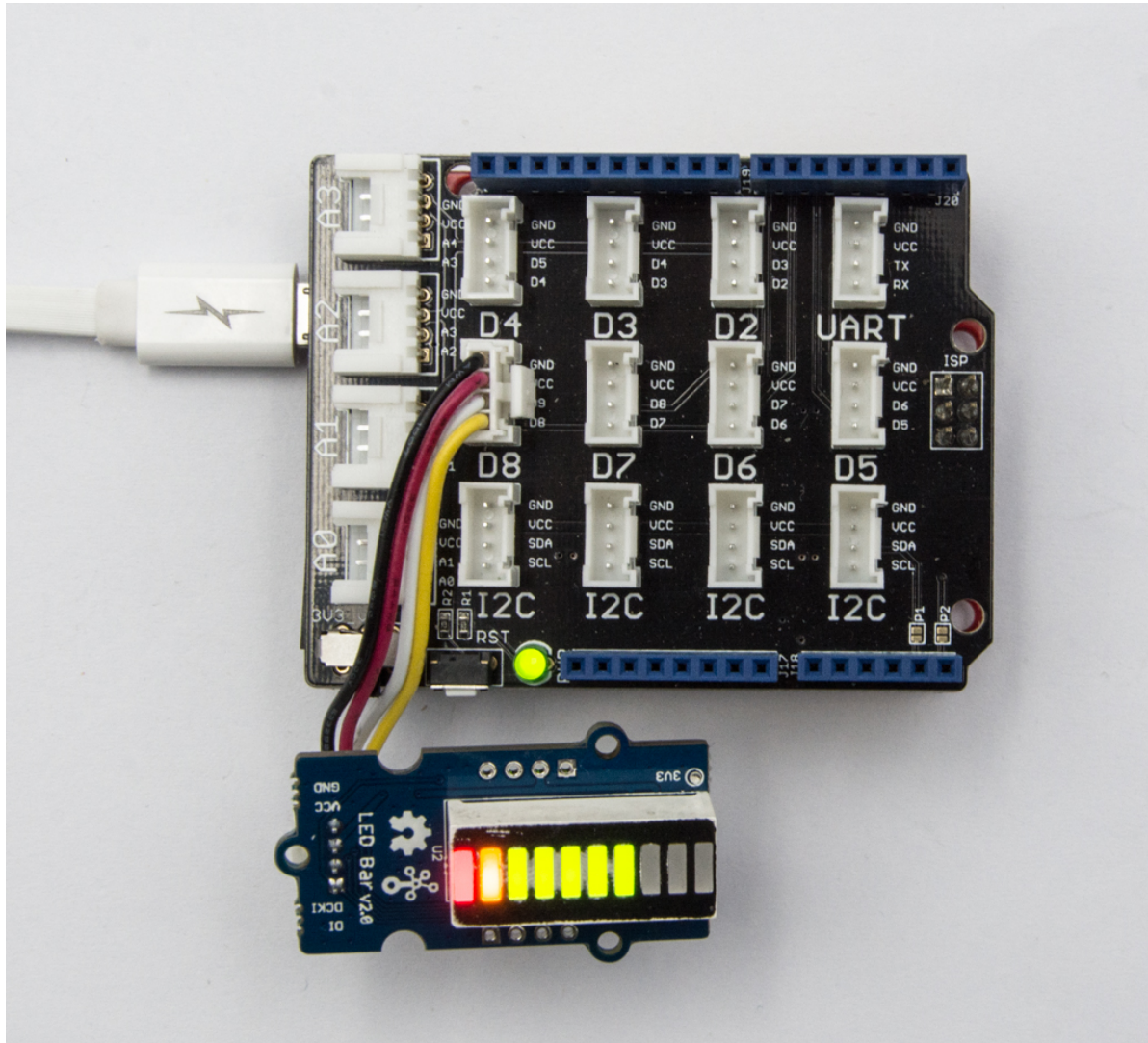
Play With Arduino


Hardware

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

Seeeduino V4.2	Base Shield	Grove-LED Bar
		
Get One Now	Get One Now	Get One Now

- **Step 2.** Connect Grove-LED Bar to port **D8** of Grove-Base Shield.
- **Step 3.** Plug Grove - Base Shield into Seeeduino.
- **Step 4.** Connect Seeeduino to PC via a USB cable.



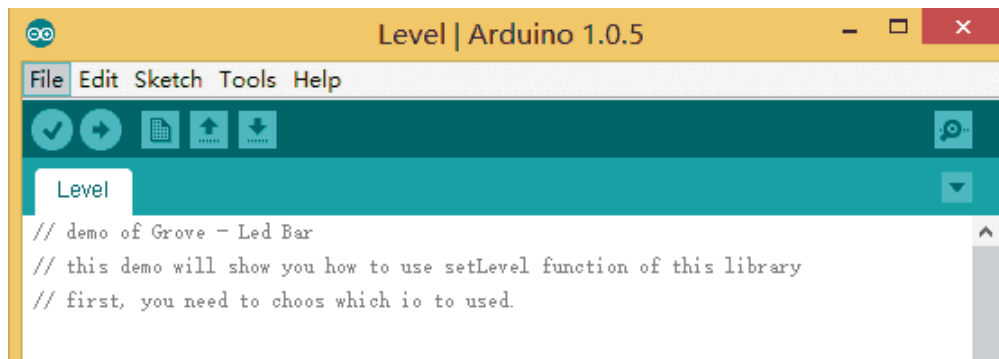
 Note

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

Seeeduino	Grove-LED Bar
5V	Red
GND	Black
D9	White
D8	Yellow

Software

- **Step 1.** Download the [Grove - LED Bar Library](#) from Github
- **Step 2.** Refer [How to install library](#) to install library for Arduino.
- **Step 3.** Restart the Arduino IDE. Open "Level" example via the path : **File** → **Examples** → **Grove LED Bar** → **Level**.



```
Level | Arduino 1.0.5
File Edit Sketch Tools Help
Level
// demo of Grove - Led Bar
// this demo will show you how to use setLevel function of this library
// first, you need to choos which io to used.
```



```
#include <LED_Bar.h>

LED_Bar bar(9, 8);           // config Io here, (clk, dio)

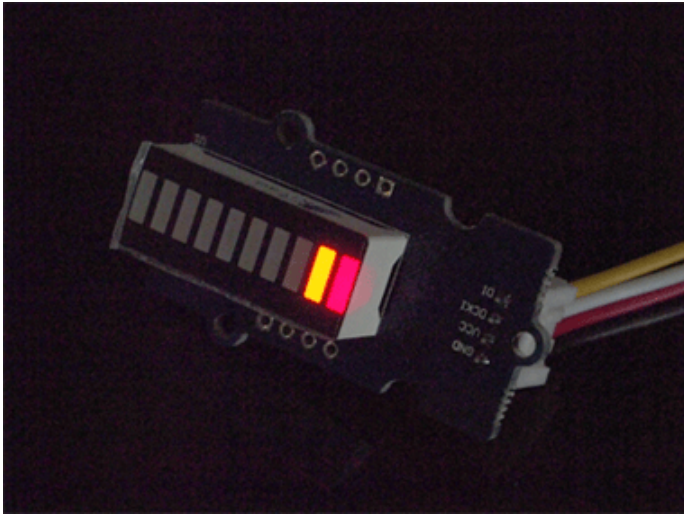
void setup()
{
  // nothing to initialize
}

void loop()
{
  for(int i=0; i<=10; i++)
  {
    bar.setLevel(i);
    delay(100);
  }
}
```



- **Step 4.** Upload the demo. If you do not know how to upload the code, please check [how to upload code](#).




The result should be like:



Play With Raspberry Pi

Hardware

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

Raspberry pi	GrovePi_Plus	Grove-LED Bar
		
Get One Now	Get One Now	Get One Now

- **Step 2.** Plug the GrovePi_Plus into Raspberry.
- **Step 3.** Connect Grove-LED Bar to **D5** port of GrovePi_Plus.
- **Step 4.** Connect the Raspberry to PC through USB cable.



Software

- **Step 1.** Follow [Setting Software](#) to configure the development environment.
- **Step 2.** Follow [Updating the Firmware](#) to update the newest firmware of GrovePi.



Tip

In this wiki we use the path `~/GrovePi/` instead of `/home/pi/Desktop/GrovePi`, you need to make sure Step 2 and Step 3 use the same path.



Note

We firmly suggest you to update the firmware, or for some sensors you may get errors.

- **Step 3.** Git clone the Github repository.

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

- **Step 4.** Navigate to the demos' directory:

```
cd yourpath/GrovePi/Software/Python/
```

Here is the grove_ledbar.py code.

```
1 import time
2 import grovepi
3 import random
4
```

```
5 # Connect the Grove LED Bar to digital port D5
6 # DI,DCKI,VCC,GND
7 ledbar = 5
8
9 grovepi.pinMode(ledbar,"OUTPUT")
10 time.sleep(1)
11 i = 0
12
13 # LED Bar methods
14 # grovepi.ledBar_init(pin,orientation)
15 # grovepi.ledBar_orientation(pin,orientation)
16 # grovepi.ledBar_setLevel(pin,level)
17 # grovepi.ledBar_setLed(pin,led,state)
18 # grovepi.ledBar_toggleLed(pin,led)
19 # grovepi.ledBar_setBits(pin,state)
20 # grovepi.ledBar_getBits(pin)
21
22 while True:
23     try:
24         print "Test 1) Initialise - red to green"
25         # ledbar_init(pin,orientation)
26         # orientation: (0 = red to green, 1 = green to red)
27         grovepi.ledBar_init(ledbar, 0)
28         time.sleep(.5)
29
30
31         print "Test 2) Set level"
32         # ledbar_setLevel(pin,level)
33         # level: (0-10)
34         for i in range(0,11):
35             grovepi.ledBar_setLevel(ledbar, i)
36             time.sleep(.2)
37         time.sleep(.3)
38
39         grovepi.ledBar_setLevel(ledbar, 8)
```

```
40     time.sleep(.5)
41
42     grovepi.ledBar_setLevel(ledbar, 2)
43     time.sleep(.5)
44
45     grovepi.ledBar_setLevel(ledbar, 5)
46     time.sleep(.5)
47
48
49     print "Test 3) Switch on/off a single LED"
50     # ledbar_setLed(pin,led,state)
51     # led: which led (1-10)
52     # state: off or on (0,1)
53     grovepi.ledBar_setLed(ledbar, 10, 1)
54     time.sleep(.5)
55
56     grovepi.ledBar_setLed(ledbar, 9, 1)
57     time.sleep(.5)
58
59     grovepi.ledBar_setLed(ledbar, 8, 1)
60     time.sleep(.5)
61
62     grovepi.ledBar_setLed(ledbar, 1, 0)
63     time.sleep(.5)
64
65     grovepi.ledBar_setLed(ledbar, 2, 0)
66     time.sleep(.5)
67
68     grovepi.ledBar_setLed(ledbar, 3, 0)
69     time.sleep(.5)
70
71
72     print "Test 4) Toggle a single LED"
73     # flip a single led - if it is currently on, it will become off .
74     # ledbar_toggleLed(ledbar, led)
```

```
75     grovepi.ledBar_toggleLed(ledbar, 1)
76     time.sleep(.5)
77
78     grovepi.ledBar_toggleLed(ledbar, 2)
79     time.sleep(.5)
80
81     grovepi.ledBar_toggleLed(ledbar, 9)
82     time.sleep(.5)
83
84     grovepi.ledBar_toggleLed(ledbar, 10)
85     time.sleep(.5)
86
87
88     print "Test 5) Set state - control all leds with 10 bits"
89     # ledbar_setBits(ledbar, state)
90     # state: (0-1023) or (0x00-0x3FF) or (0b0000000000-0b1111111111)
91     for i in range(0,32):
92         grovepi.ledBar_setBits(ledbar, i)
93         time.sleep(.2)
94     time.sleep(.3)
95
96
97     print "Test 6) Get current state"
98     # state = ledbar_getBits(ledbar)
99     # state: (0-1023) a bit for each of the 10 LEDs
100    state = grovepi.ledBar_getBits(ledbar)
101    print "with first 5 leds lit, the state should be 31 or 0x1F"
102    print state
103
104    # bitwise shift five bits to the left
105    state = state << 5
106    # the state should now be 992 or 0x3E0
107    # when saved the last 5 LEDs will be lit instead of the first 5
108    time.sleep(.5)
109
```



```
110
111     print "Test 7) Set state - save the state we just modified"
112     # ledbar_setBits(ledbar, state)
113     # state: (0-1023) a bit for each of the 10 LEDs
114     grovepi.ledBar_setBits(ledbar, state)
115     time.sleep(.5)
116
117
118     print "Test 8) Swap orientation - green to red - current state"
119     # ledbar_orientation(pin,orientation)
120     # orientation: (0 = red to green, 1 = green to red)
121     # when you reverse the led bar orientation, all methods know how
122     # green to red
123     grovepi.ledBar_orientation(ledbar, 1)
124     time.sleep(.5)
125
126     # red to green
127     grovepi.ledBar_orientation(ledbar, 0)
128     time.sleep(.5)
129
130     # green to red
131     grovepi.ledBar_orientation(ledbar, 1)
132     time.sleep(.5)
133
134
135     print "Test 9) Set level, again"
136     # ledbar_setLevel(pin,level)
137     # level: (0-10)
138     # note the red LED is now at index 10 instead of 1
139     for i in range(0,11):
140         grovepi.ledBar_setLevel(ledbar, i)
141         time.sleep(.2)
142     time.sleep(.3)
143
144
```

```
145     print "Test 10) Set a single LED, again"
146     # ledbar_setLed(pin,led,state)
147     # led: which led (1-10)
148     # state: off or on (0,1)
149     grovepi.ledBar_setLed(ledbar, 1, 0)
150     time.sleep(.5)
151
152     grovepi.ledBar_setLed(ledbar, 3, 0)
153     time.sleep(.5)
154
155     grovepi.ledBar_setLed(ledbar, 5, 0)
156     time.sleep(.5)
157
158
159     print "Test 11) Toggle a single LED, again"
160     # ledbar_toggleLed(ledbar, led)
161     grovepi.ledBar_toggleLed(ledbar, 2)
162     time.sleep(.5)
163
164     grovepi.ledBar_toggleLed(ledbar, 4)
165     time.sleep(.5)
166
167
168     print "Test 12) Get state"
169     # state = ledbar_getBits(ledbar)
170     # state: (0-1023) a bit for each of the 10 LEDs
171     state = grovepi.ledBar_getBits(ledbar)
172
173     # the last 5 LEDs are lit, so the state should be 992 or 0x3E0
174
175     # bitwise shift five bits to the right
176     state = state >> 5
177     # the state should now be 31 or 0x1F
178
179
```

```
180     print "Test 13) Set state, again"
181     # ledbar_setBits(ledbar, state)
182     # state: (0-1023) a bit for each of the 10 LEDs
183     grovepi.ledBar_setBits(ledbar, state)
184     time.sleep(.5)
185
186
187     print "Test 14) Step"
188     # step through all 10 LEDs
189     for i in range(0,11):
190         grovepi.ledBar_setLevel(ledbar, i)
191         time.sleep(.2)
192     time.sleep(.3)
193
194
195     print "Test 15) Bounce"
196     # switch on the first two LEDs
197     grovepi.ledBar_setLevel(ledbar, 2)
198
199     # get the current state (which is 0x3)
200     state = grovepi.ledBar_getBits(ledbar)
201
202     # bounce to the right
203     for i in range(0,9):
204         # bit shift left and update
205         state <<= 1;
206         grovepi.ledBar_setBits(ledbar, state)
207         time.sleep(.2)
208
209     # bounce to the left
210     for i in range(0,9):
211         # bit shift right and update
212         state >>= 1;
213         grovepi.ledBar_setBits(ledbar, state)
214         time.sleep(.2)
```

```
215     time.sleep(.3)
216
217
218     print "Test 16) Random"
219     for i in range(0,21):
220         state = random.randint(0,1023)
221         grovepi.ledBar_setBits(ledbar, state)
222         time.sleep(.2)
223     time.sleep(.3)
224
225
226     print "Test 17) Invert"
227     # set every 2nd LED on - 341 or 0x155
228     state = 341
229     for i in range(0,5):
230         grovepi.ledBar_setBits(ledbar, state)
231         time.sleep(.2)
232
233         # bitwise XOR all 10 LEDs on with the current state
234         state = 0x3FF ^ state
235
236         grovepi.ledBar_setBits(ledbar, state)
237         time.sleep(.2)
238     time.sleep(.3)
239
240
241     print "Test 18) Walk through all possible combinations"
242     for i in range(0,1024):
243         grovepi.ledBar_setBits(ledbar, i)
244         time.sleep(.1)
245     time.sleep(.4)
246
247 except KeyboardInterrupt:
248     grovepi.ledBar_setBits(ledbar, 0)
249     break
```

```
250     except IOError:  
251         print "Error"
```

- **Step 5.** Run the demo.

```
sudo python grove_ledbar.py
```



Schematic Online Viewer



Resources

- **[Eagle&PDF]**[Grove - LED Bar Eagle File](#)
- **[Library]**[Grove - LED Bar Library](#)

- **[Library]** [Suli-compatible Library](#)
- **[Datasheet]** [MY9221 Datasheet](#)
- **[More Reading]** [Wooden Laser Gun](#)

Tech Support

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

