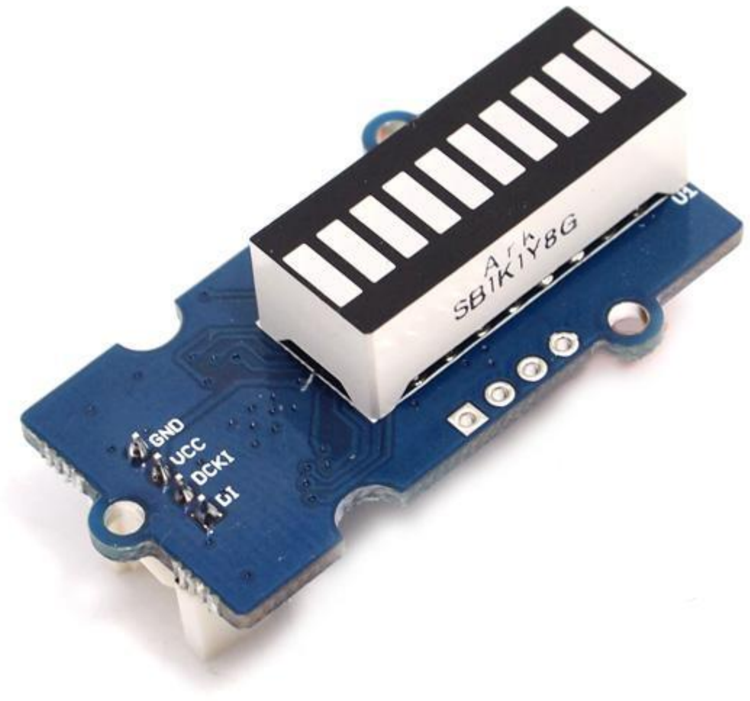
Grove - LED Bar



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	3.⅗V
Operation Temperature	-20℃ to +80℃
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	BeagleBone	Wio	LinkIt ONE
		TO DO		

Caution
The platforms mentioned above as supported is/are an indication of the module's software or theoritical 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 Arduinobefore the start.

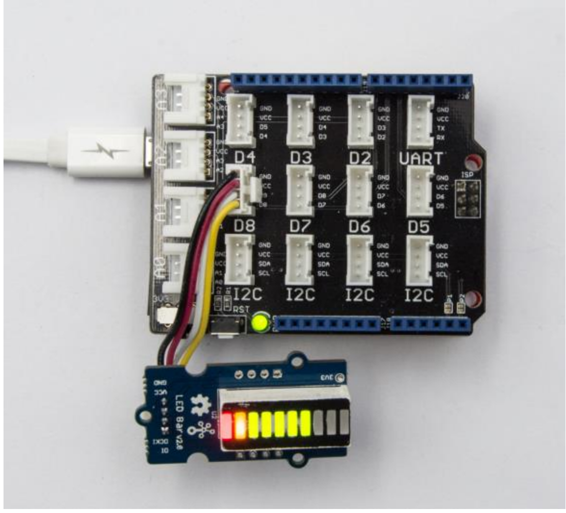
Play With Arduino

Hardware

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

Seeeduino V4.2	Base Shield	Grove-LED Bar

- **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.



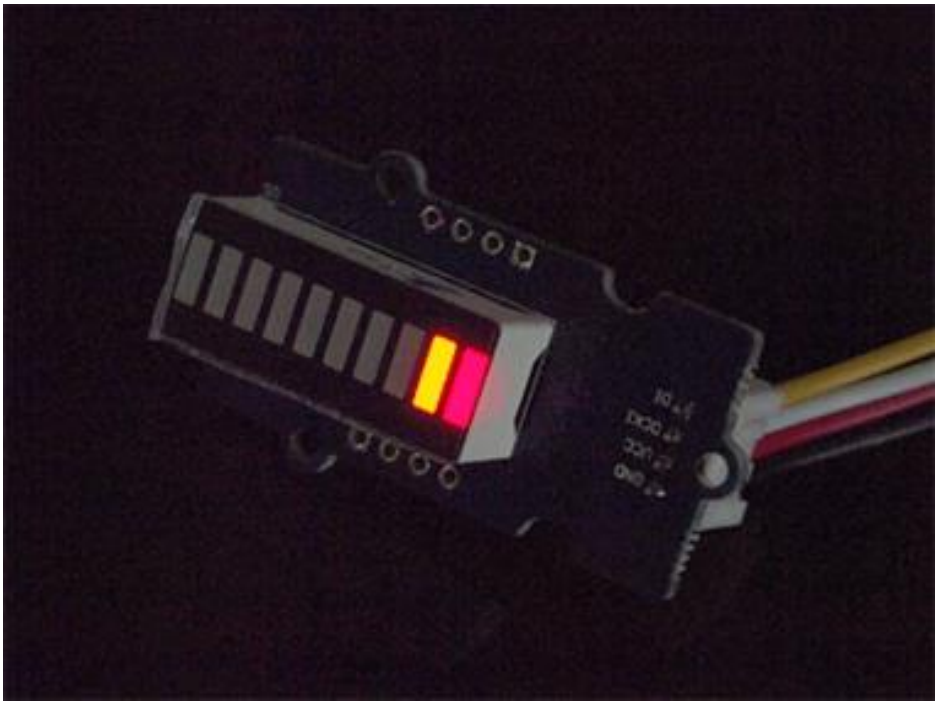
Note
If we don't have Grove Base Shield, We also can directly connect Grove-LED Bar to Seeeduino as below.

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**.
- **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:



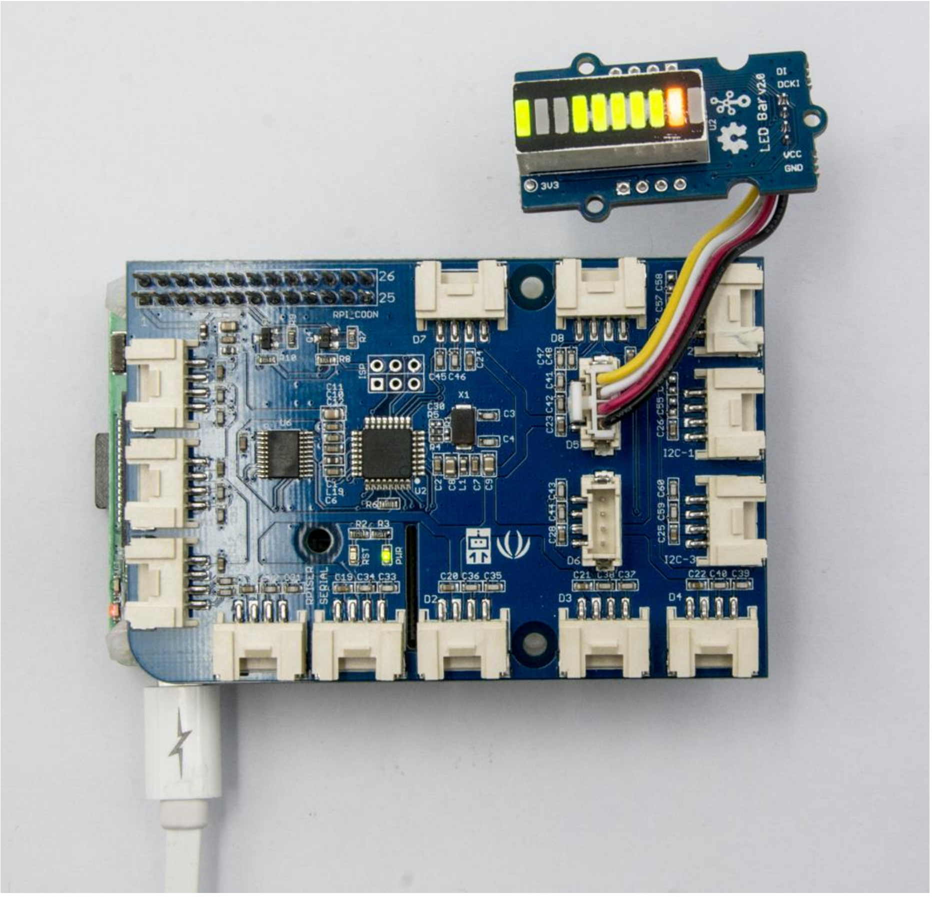
Play With Raspberry Pi

Hardware

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

Raspberry pi	GrovePi_Plus	Grove-LED Bar

- **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.

```
cd ~
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.

```
import time
import grovepi
import random

# Connect the Grove LED Bar to digital port D5
# DI,DCKI,VCC,GND
ledbar = 5

grovepi.pinMode(ledbar,"OUTPUT")
time.sleep(1)
i = 0

# LED Bar methods
# grovepi.ledBar_init(pin,orientation)
# grovepi.ledBar_orientation(pin,orientation)
# grovepi.ledBar_setLevel(pin,level)
# grovepi.ledBar_setLed(pin,led,state)
# grovepi.ledBar_toggleLed(pin,led)
# grovepi.ledBar_setBits(pin,state)
# grovepi.ledBar_getBits(pin)

    while True:
        try:
        print "Test 1) Initialise - red to green"
        # ledbar_init(pin,orientation)
        # orientation: (0 = red to green, 1 = green to red)
        grovepi.ledBar_init(ledbar, 0)
        time.sleep(.5)


        print "Test 2) Set level"
        # ledbar_setLevel(pin,level)
        # level: (0-10)
        for i in range(0,11):
            grovepi.ledBar_setLevel(ledbar, i)
            time.sleep(.2)
        time.sleep(.3)

        grovepi.ledBar_setLevel(ledbar, 8)
        time.sleep(.5)

        grovepi.ledBar_setLevel(ledbar, 2)
        time.sleep(.5)

        grovepi.ledBar_setLevel(ledbar, 5)
        time.sleep(.5)


        print "Test 3) Switch on/off a single LED"
        # ledbar_setLed(pin,led,state)
        # led: which led (1-10)
        # state: off or on (0,1)
        grovepi.ledBar_setLed(ledbar, 10, 1)
        time.sleep(.5)

        grovepi.ledBar_setLed(ledbar, 9, 1)
        time.sleep(.5)

        grovepi.ledBar_setLed(ledbar, 8, 1)
        time.sleep(.5)

        grovepi.ledBar_setLed(ledbar, 1, 0)
        time.sleep(.5)

        grovepi.ledBar_setLed(ledbar, 2, 0)
        time.sleep(.5)

        grovepi.ledBar_setLed(ledbar, 3, 0)
        time.sleep(.5)


        print "Test 4) Toggle a single LED"
        # flip a single led - if it is currently on, it will become off
and vice versa
        # ledbar_toggleLed(ledbar, led)
        grovepi.ledBar_toggleLed(ledbar, 1)
        time.sleep(.5)

        grovepi.ledBar_toggleLed(ledbar, 2)
        time.sleep(.5)

        grovepi.ledBar_toggleLed(ledbar, 9)
        time.sleep(.5)

        grovepi.ledBar_toggleLed(ledbar, 10)
        time.sleep(.5)


        print "Test 5) Set state - control all leds with 10 bits"
        # ledbar_setBits(ledbar, state)
        # state: (0-1023) or (0x00-0x3FF) or (0b0000000000-0b1111111111)
or (int('0000000000',2)-int('1111111111',2))
        for i in range(0,32):
            grovepi.ledBar_setBits(ledbar, i)
            time.sleep(.2)
        time.sleep(.3)


        print "Test 6) Get current state"
        # state = ledbar_getBits(ledbar)
        # state: (0-1023) a bit for each of the 10 LEDs
        state = grovepi.ledBar_getBits(ledbar)
        print "with first 5 leds lit, the state should be 31 or 0x1F"
        print state

        # bitwise shift five bits to the left
        state = state << 5
        # the state should now be 992 or 0x3E0
        # when saved the last 5 LEDs will be lit instead of the first 5
LEDs
        time.sleep(.5)


        print "Test 7) Set state - save the state we just modified"
        # ledbar_setBits(ledbar, state)
        # state: (0-1023) a bit for each of the 10 LEDs
        grovepi.ledBar_setBits(ledbar, state)
        time.sleep(.5)


        print "Test 8) Swap orientation - green to red - current state is preserved"
        # ledbar_orientation(pin,orientation)
        # orientation: (0 = red to green, 1 = green to red)
        # when you reverse the led bar orientation, all methods know how to handle the new LED index
        # green to red
        grovepi.ledBar_orientation(ledbar, 1)
        time.sleep(.5)

        # red to green
        grovepi.ledBar_orientation(ledbar, 0)
        time.sleep(.5)

        # green to red
        grovepi.ledBar_orientation(ledbar, 1)
        time.sleep(.5)


        print "Test 9) Set level, again"
        # ledbar_setLevel(pin,level)
        # level: (0-10)
        # note the red LED is now at index 10 instead of 1
        for i in range(0,11):
            grovepi.ledBar_setLevel(ledbar, i)
            time.sleep(.2)
        time.sleep(.3)


        print "Test 10) Set a single LED, again"
        # ledbar_setLed(pin,led,state)
        # led: which led (1-10)
        # state: off or on (0,1)
        grovepi.ledBar_setLed(ledbar, 1, 0)
        time.sleep(.5)

        grovepi.ledBar_setLed(ledbar, 3, 0)
        time.sleep(.5)

        grovepi.ledBar_setLed(ledbar, 5, 0)
        time.sleep(.5)


        print "Test 11) Toggle a single LED, again"
        # ledbar_toggleLed(ledbar, led)
        grovepi.ledBar_toggleLed(ledbar, 2)
        time.sleep(.5)

```

```python
        grovepi.ledBar_toggleLed(ledbar, 4)
        time.sleep(.5)


        print "Test 12) Get state"
        # state = ledbar_getBits(ledbar)
        # state: (0-1023) a bit for each of the 10 LEDs
        state = grovepi.ledBar_getBits(ledbar)

        # the last 5 LEDs are lit, so the state should be 992 or 0x3E0

        # bitwise shift five bits to the right
        state = state >> 5
        # the state should now be 31 or 0x1F


        print "Test 13) Set state, again"
        # ledbar_setBits(ledbar, state)
        # state: (0-1023) a bit for each of the 10 LEDs
        grovepi.ledBar_setBits(ledbar, state)
        time.sleep(.5)


        print "Test 14) Step"
        # step through all 10 LEDs
        for i in range(0,11):
            grovepi.ledBar_setLevel(ledbar, i)
            time.sleep(.2)
        time.sleep(.3)


        print "Test 15) Bounce"
        # switch on the first two LEDs
        grovepi.ledBar_setLevel(ledbar, 2)

        # get the current state (which is 0x3)
        state = grovepi.ledBar_getBits(ledbar)

        # bounce to the right
        for i in range(0,9):
            # bit shift left and update
            state <<= 1;
            grovepi.ledBar_setBits(ledbar, state)
            time.sleep(.2)

        # bounce to the left
        for i in range(0,9):
            # bit shift right and update
            state >>= 1;
            grovepi.ledBar_setBits(ledbar, state)
            time.sleep(.2)
        time.sleep(.3)


        print "Test 16) Random"
        for i in range(0,21):
            state = random.randint(0,1023)
            grovepi.ledBar_setBits(ledbar, state)
            time.sleep(.2)
        time.sleep(.3)


        print "Test 17) Invert"
        # set every 2nd LED on - 341 or 0x155
        state = 341
        for i in range(0,5):
            grovepi.ledBar_setBits(ledbar, state)
            time.sleep(.2)

            # bitwise XOR all 10 LEDs on with the current state
            state = 0x3FF ^ state

            grovepi.ledBar_setBits(ledbar, state)
            time.sleep(.2)
        time.sleep(.3)


        print "Test 18) Walk through all possible combinations"
        for i in range(0,1024):
            grovepi.ledBar_setBits(ledbar, i)
            time.sleep(.1)
        time.sleep(.4)

    except KeyboardInterrupt:
        grovepi.ledBar_setBits(ledbar, 0)
        break
    except IOError:
        print "Error"
```

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

```
sudo python grove_ledbar.py
```

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

Projects

Grove LED Bar v2.0: Calliope Mini is equipped with two Grove connectors. In this project, I want to explore, how to talk to these Seeed Grove parts.

Grove LED Bar Controller with the Bean+: Learn the basics of using popular Grove components with the new LightBlue Bean+ to get started with building your own projects!

Tech Support

Please submit any technical issue into our forum.