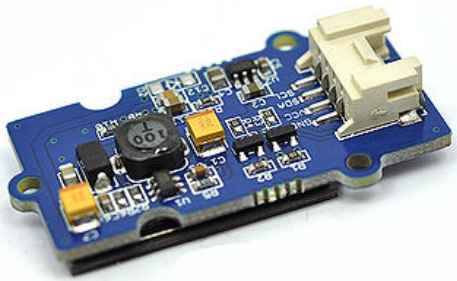
Grove - OLED Display 0.96 inch



Grove - OLED Display 0.96" module is an OLED monochrome 128×64dot matrix display module with Grove 4pin I2C Interface.Comparing to LCD, OLED screens are more competitive, which has a number of advantages such as high brightness, self-emission, high contrast ratio, slim / thin outline, wide viewing angle, wide temperature range, and low power consumption. It has bigger screen so that it can display more contents than the OLED 96×96.

Table of contents

Version

Product Version	Changes	Released Date
Grove - OLED Display 0.96 inch V1.1	Initial	Oct 2015

Features

- Grove compatible interface
- Communicate Mode:I2C
- Low power consumption
- Display Color: White
- Wide range of operating temperature:-20℃~70℃
- I2C Address 0x3C

Note
If you want to use multiplue I2C devices, please refer to Software I2C.

Warning
Please notice: heavy impact or stress on the OLED will cause the breakdown of screen.

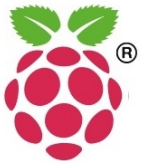
Tip
More details about Grove modules please refer to Grove System

Specifications

Items	Min	Norm	Max	Unit
Power Voltage (VCC)	3.3	5.0	5.5	V
Driver IC	-	SSD1308Z	-	-
Display Color	-	White	-	-
Dot Matrix	-	128×64	-	-
Panel Size	-	26.7(W)×19.26(H)	-	mm
Active Area	-	21.74(W)×11.175 (H)	-	mm
Dot Pitch	-	0.17(W)×0.175 (H)	-	mm
Dot Size	-	0.15(W)×0.15 (H)	-	mm
Wide range of operating temperature	-	-20~70	-	℃

Tip
More details about Grove modules please refer to Grove System

Platforms Supported

Arduino	Raspberry Pi	
		

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 Arduino before the start.

Play With Arduino

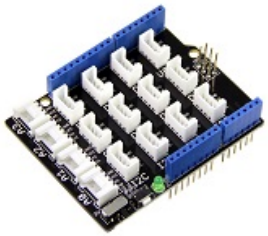
Demonstration

The OLED128*64 uses all the pins of SSD1308 chip, the default original point is on the top left corner. You can also change the original point by adjusting the program and in order to display

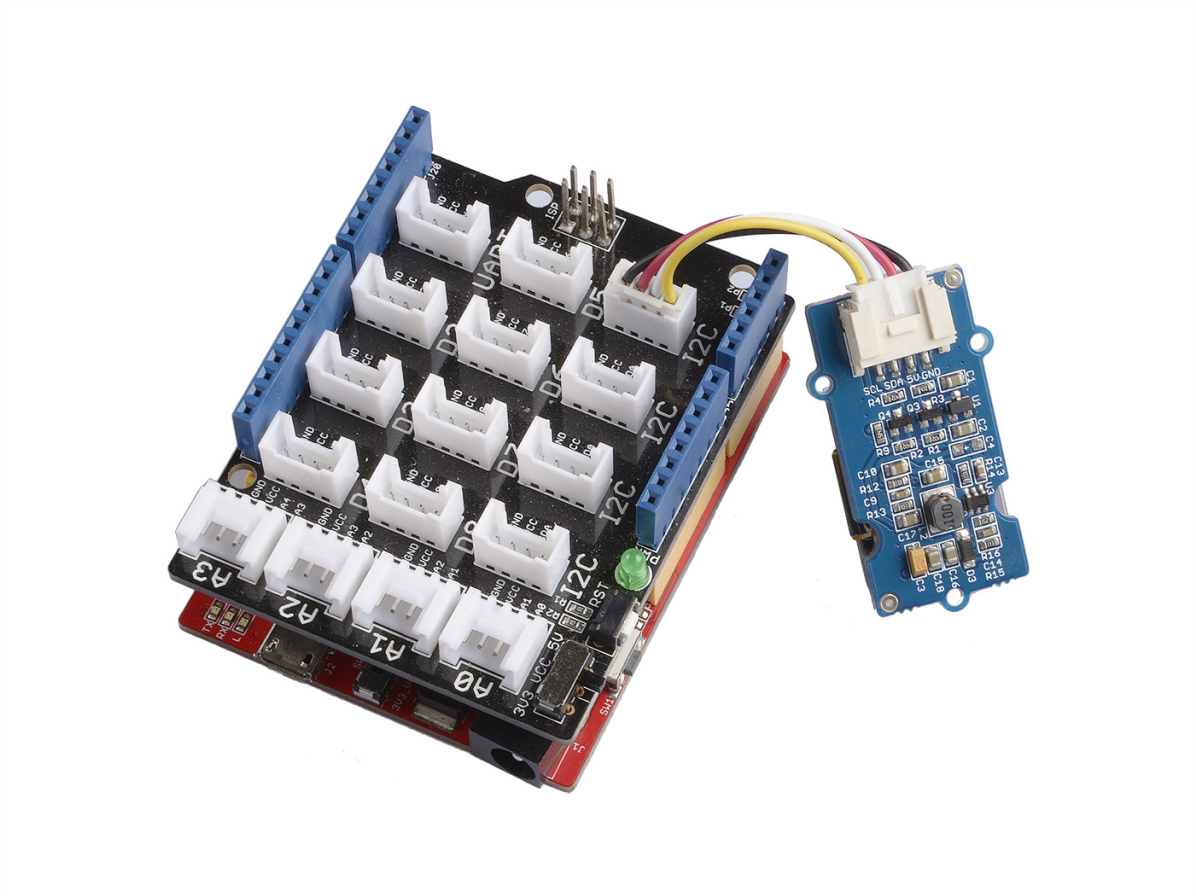
your desired patterns. For more details, please refer SSD1308_1.0.pdf and LY190-128064.pdf.

Hardware

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

Seeeduino V4	Base Shield	Grove - OLED Display 0.96inch
		
Get ONE Now	Get ONE Now	Get ONE Now

- **Step 2.** Plug the Grove OLED Display 128*64 onto the I2C port on 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 - OLED Display 0.96 inch to Seeeduino as below.

Seeeduino	Grove - OLED Display 0.96 inch
5V	Red
GND	Black
SDA	White
SCL	Yellow

Software

- **Step 1.** Download the U8G2 128*64 library from Github.
- **Step 2.** Refer How to install library to install library for Arduino.
- **Step 3.** Copy the code into Arduino IDE and upload. If you do not know how to upload the code, please check how to upload code.

```
#include <Arduino.h>
#include <U8g2lib.h>

#ifdef U8X8_HAVE_HW_SPI
#include <SPI.h>
#endif
#ifdef U8X8_HAVE_HW_I2C
#include <Wire.h>
#endif

U8G2_SSD1306_128X64_ALT0_F_HW_I2C u8g2(U8G2_R0, /* reset=*/ U8X8_PIN_NON
```

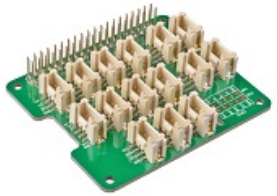
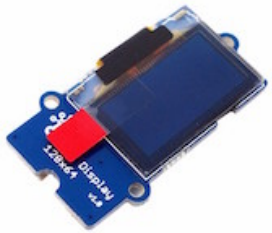
```
12
13 // U8G2_SSD1306_128X64_NONAME_F_SW_I2C u8g2(U8G2_R0, /* clock=*/ SCL, /* d
14
15 void setup(void) {
16   u8g2.begin();
17 }
18
19 void loop(void) {
20   u8g2.clearBuffer();                    // clear the internal memory
21   u8g2.setFont(u8g2_font_ncenB08_tr);    // choose a suitable font
22   u8g2.drawStr(0,10,"Hello World!");     // write something to the intern
23   u8g2.sendBuffer();                     // transfer internal memory to t
24   delay(1000);
25 }
```

- **Step 3.** We can see "Hello World!" on screen.

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

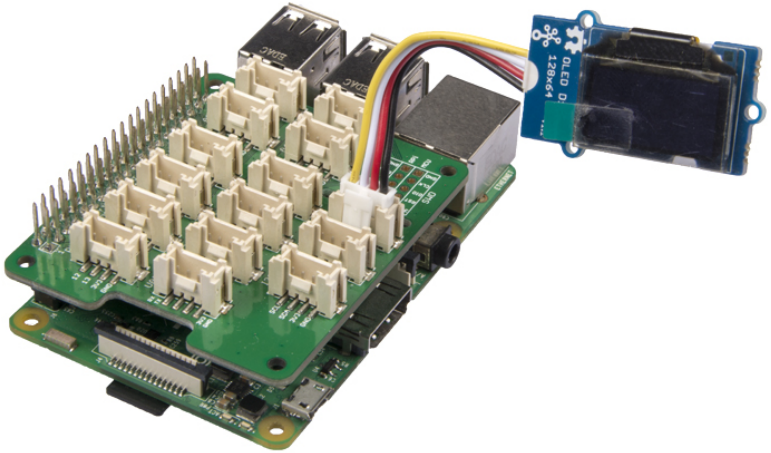
Hardware

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

Raspberry pi	Grove Base Hat for RasPi	Grove - OLED Display 0.96 inch
		
Get ONE Now	Get ONE Now	Get ONE Now

- **Step 2**. Plug the Grove Base Hat into Raspberry.
- **Step 3**. Connect the OLED display to the **I^{2}C** port of the Base Hat.

- **Step 4**. Connect the Raspberry Pi to PC through USB cable.



Software

- **Step 1**. Follow Setting Software to configure the development environment.
- **Step 2**. Download the source file by cloning the grove.py library.

```
```

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

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

```
cd grove.py/grove
python grove_oled_display_128x64.py
```

Following is the grove_oled_display_128x64.py code.

```
import time

from grove.i2c import Bus

_COMMAND_MODE = 0x80
_DATA_MODE = 0x40
_NORMAL_DISPLAY = 0xA6

_DISPLAY_OFF = 0xAE
_DISPLAY_ON = 0xAF
_INVERSE_DISPLAY = 0xA7
_SET_BRIGHTNESS = 0x81


BasicFont = [[0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00],
             [0x00, 0x00, 0x5F, 0x00, 0x00, 0x00, 0x00, 0x00],
             [0x00, 0x00, 0x07, 0x00, 0x07, 0x00, 0x00, 0x00],
             [0x00, 0x14, 0x7F, 0x14, 0x7F, 0x14, 0x00, 0x00],
             [0x00, 0x24, 0x2A, 0x7F, 0x2A, 0x12, 0x00, 0x00],
             [0x00, 0x23, 0x13, 0x08, 0x64, 0x62, 0x00, 0x00],
             [0x00, 0x36, 0x49, 0x55, 0x22, 0x50, 0x00, 0x00],
```

```
                [0x00, 0x00, 0x05, 0x03, 0x00, 0x00, 0x00, 0x00],
                [0x00, 0x1C, 0x22, 0x41, 0x00, 0x00, 0x00, 0x00],
                [0x00, 0x41, 0x22, 0x1C, 0x00, 0x00, 0x00, 0x00],
                [0x00, 0x08, 0x2A, 0x1C, 0x2A, 0x08, 0x00, 0x00],
                [0x00, 0x08, 0x08, 0x3E, 0x08, 0x08, 0x00, 0x00],
                [0x00, 0xA0, 0x60, 0x00, 0x00, 0x00, 0x00, 0x00],
                [0x00, 0x08, 0x08, 0x08, 0x08, 0x08, 0x00, 0x00],
                [0x00, 0x60, 0x60, 0x00, 0x00, 0x00, 0x00, 0x00],
                [0x00, 0x20, 0x10, 0x08, 0x04, 0x02, 0x00, 0x00],
                [0x00, 0x3E, 0x51, 0x49, 0x45, 0x3E, 0x00, 0x00],
                [0x00, 0x00, 0x42, 0x7F, 0x40, 0x00, 0x00, 0x00],
                [0x00, 0x62, 0x51, 0x49, 0x49, 0x46, 0x00, 0x00],
                [0x00, 0x22, 0x41, 0x49, 0x49, 0x36, 0x00, 0x00],
                [0x00, 0x18, 0x14, 0x12, 0x7F, 0x10, 0x00, 0x00],
                [0x00, 0x27, 0x45, 0x45, 0x45, 0x39, 0x00, 0x00],
                [0x00, 0x3C, 0x4A, 0x49, 0x49, 0x30, 0x00, 0x00],
                [0x00, 0x01, 0x71, 0x09, 0x05, 0x03, 0x00, 0x00],
                [0x00, 0x36, 0x49, 0x49, 0x49, 0x36, 0x00, 0x00],
                [0x00, 0x06, 0x49, 0x49, 0x29, 0x1E, 0x00, 0x00],
                [0x00, 0x00, 0x36, 0x36, 0x00, 0x00, 0x00, 0x00],
                [0x00, 0x00, 0xAC, 0x6C, 0x00, 0x00, 0x00, 0x00],
                [0x00, 0x08, 0x14, 0x22, 0x41, 0x00, 0x00, 0x00],
                [0x00, 0x14, 0x14, 0x14, 0x14, 0x14, 0x00, 0x00],
                [0x00, 0x41, 0x22, 0x14, 0x08, 0x00, 0x00, 0x00],
                [0x00, 0x02, 0x01, 0x51, 0x09, 0x06, 0x00, 0x00],
                [0x00, 0x32, 0x49, 0x79, 0x41, 0x3E, 0x00, 0x00],
                [0x00, 0x7E, 0x09, 0x09, 0x09, 0x7E, 0x00, 0x00],
                [0x00, 0x7F, 0x49, 0x49, 0x49, 0x36, 0x00, 0x00],
                [0x00, 0x3E, 0x41, 0x41, 0x41, 0x22, 0x00, 0x00],
                [0x00, 0x7F, 0x41, 0x41, 0x22, 0x1C, 0x00, 0x00],
                [0x00, 0x7F, 0x49, 0x49, 0x49, 0x41, 0x00, 0x00],
                [0x00, 0x7F, 0x09, 0x09, 0x09, 0x01, 0x00, 0x00],
                [0x00, 0x3E, 0x41, 0x41, 0x51, 0x72, 0x00, 0x00],
                [0x00, 0x7F, 0x08, 0x08, 0x08, 0x7F, 0x00, 0x00],
                [0x00, 0x41, 0x7F, 0x41, 0x00, 0x00, 0x00, 0x00],
```

```
57                [0x00, 0x20, 0x40, 0x41, 0x3F, 0x01, 0x00, 0x00],
58                [0x00, 0x7F, 0x08, 0x14, 0x22, 0x41, 0x00, 0x00],
59                [0x00, 0x7F, 0x40, 0x40, 0x40, 0x40, 0x00, 0x00],
60                [0x00, 0x7F, 0x02, 0x0C, 0x02, 0x7F, 0x00, 0x00],
61                [0x00, 0x7F, 0x04, 0x08, 0x10, 0x7F, 0x00, 0x00],
62                [0x00, 0x3E, 0x41, 0x41, 0x41, 0x3E, 0x00, 0x00],
63                [0x00, 0x7F, 0x09, 0x09, 0x09, 0x06, 0x00, 0x00],
64                [0x00, 0x3E, 0x41, 0x51, 0x21, 0x5E, 0x00, 0x00],
65                [0x00, 0x7F, 0x09, 0x19, 0x29, 0x46, 0x00, 0x00],
66                [0x00, 0x26, 0x49, 0x49, 0x49, 0x32, 0x00, 0x00],
67                [0x00, 0x01, 0x01, 0x7F, 0x01, 0x01, 0x00, 0x00],
68                [0x00, 0x3F, 0x40, 0x40, 0x40, 0x3F, 0x00, 0x00],
69                [0x00, 0x1F, 0x20, 0x40, 0x20, 0x1F, 0x00, 0x00],
70                [0x00, 0x3F, 0x40, 0x38, 0x40, 0x3F, 0x00, 0x00],
71                [0x00, 0x63, 0x14, 0x08, 0x14, 0x63, 0x00, 0x00],
72                [0x00, 0x03, 0x04, 0x78, 0x04, 0x03, 0x00, 0x00],
73                [0x00, 0x61, 0x51, 0x49, 0x45, 0x43, 0x00, 0x00],
74                [0x00, 0x7F, 0x41, 0x41, 0x00, 0x00, 0x00, 0x00],
75                [0x00, 0x02, 0x04, 0x08, 0x10, 0x20, 0x00, 0x00],
76                [0x00, 0x41, 0x41, 0x7F, 0x00, 0x00, 0x00, 0x00],
77                [0x00, 0x04, 0x02, 0x01, 0x02, 0x04, 0x00, 0x00],
78                [0x00, 0x80, 0x80, 0x80, 0x80, 0x80, 0x00, 0x00],
79                [0x00, 0x01, 0x02, 0x04, 0x00, 0x00, 0x00, 0x00],
80                [0x00, 0x20, 0x54, 0x54, 0x54, 0x78, 0x00, 0x00],
81                [0x00, 0x7F, 0x48, 0x44, 0x44, 0x38, 0x00, 0x00],
82                [0x00, 0x38, 0x44, 0x44, 0x28, 0x00, 0x00, 0x00],
83                [0x00, 0x38, 0x44, 0x44, 0x48, 0x7F, 0x00, 0x00],
84                [0x00, 0x38, 0x54, 0x54, 0x54, 0x18, 0x00, 0x00],
85                [0x00, 0x08, 0x7E, 0x09, 0x02, 0x00, 0x00, 0x00],
86                [0x00, 0x18, 0xA4, 0xA4, 0xA4, 0x7C, 0x00, 0x00],
87                [0x00, 0x7F, 0x08, 0x04, 0x04, 0x78, 0x00, 0x00],
88                [0x00, 0x00, 0x7D, 0x00, 0x00, 0x00, 0x00, 0x00],
89                [0x00, 0x80, 0x84, 0x7D, 0x00, 0x00, 0x00, 0x00],
90                [0x00, 0x7F, 0x10, 0x28, 0x44, 0x00, 0x00, 0x00],
91                [0x00, 0x41, 0x7F, 0x40, 0x00, 0x00, 0x00, 0x00],
```

```
92                [0x00, 0x7C, 0x04, 0x18, 0x04, 0x78, 0x00, 0x00],
93                [0x00, 0x7C, 0x08, 0x04, 0x7C, 0x00, 0x00, 0x00],
94                [0x00, 0x38, 0x44, 0x44, 0x38, 0x00, 0x00, 0x00],
95                [0x00, 0xFC, 0x24, 0x24, 0x18, 0x00, 0x00, 0x00],
96                [0x00, 0x18, 0x24, 0x24, 0xFC, 0x00, 0x00, 0x00],
97                [0x00, 0x00, 0x7C, 0x08, 0x04, 0x00, 0x00, 0x00],
98                [0x00, 0x48, 0x54, 0x54, 0x24, 0x00, 0x00, 0x00],
99                [0x00, 0x04, 0x7F, 0x44, 0x00, 0x00, 0x00, 0x00],
100               [0x00, 0x3C, 0x40, 0x40, 0x7C, 0x00, 0x00, 0x00],
101               [0x00, 0x1C, 0x20, 0x40, 0x20, 0x1C, 0x00, 0x00],
102               [0x00, 0x3C, 0x40, 0x30, 0x40, 0x3C, 0x00, 0x00],
103               [0x00, 0x44, 0x28, 0x10, 0x28, 0x44, 0x00, 0x00],
104               [0x00, 0x1C, 0xA0, 0xA0, 0x7C, 0x00, 0x00, 0x00],
105               [0x00, 0x44, 0x64, 0x54, 0x4C, 0x44, 0x00, 0x00],
106               [0x00, 0x08, 0x36, 0x41, 0x00, 0x00, 0x00, 0x00],
107               [0x00, 0x00, 0x7F, 0x00, 0x00, 0x00, 0x00, 0x00],
108               [0x00, 0x41, 0x36, 0x08, 0x00, 0x00, 0x00, 0x00],
109               [0x00, 0x02, 0x01, 0x01, 0x02, 0x01, 0x00, 0x00],
110               [0x00, 0x02, 0x05, 0x05, 0x02, 0x00, 0x00, 0x00]]
111
112
113 class GroveOledDisplay128x64(object):
114     HORIZONTAL = 0x00
115     VERTICAL = 0x01
116     PAGE = 0x02
117
118     def __init__(self, bus=None, address=0x3C):
119         self.bus = Bus(bus)
120         self.address = address
121
122         self.off()
123         self.inverse = False
124         self.mode = self.HORIZONTAL
125
126         self.clear()
```

```
127             self.on()
128
129         def on(self):
130             self.send_command(_DISPLAY_ON)
131
132         def off(self):
133             self.send_command(_DISPLAY_OFF)
134
135         def send_command(self, command):
136             self.bus.write_byte_data(self.address, _COMMAND_MODE, command)
137
138         def send_data(self, data):
139             self.bus.write_byte_data(self.address, _DATA_MODE, data)
140
141         def send_commands(self, commands):
142             for c in commands:
143                 self.send_command(c)
144
145         def clear(self):
146             self.off()
147             for i in range(8):
148                 self.set_cursor(i, 0)
149                 self.puts(' ' * 16)
150
151             self.on()
152             self.set_cursor(0, 0)
153
154         @property
155         def inverse(self):
156             return self._inverse
157
158         @inverse.setter
159         def inverse(self, enable):
160             self.send_command(_INVERSE_DISPLAY if enable else _NORMAL_DISP]
161             self._inverse = enable
```

```
162
163     @property
164     def mode(self):
165         return self._mode
166
167     @mode.setter
168     def mode(self, mode):
169         self.send_command(0x20)
170         self.send_command(mode)
171         self._mode = mode
172
173     def set_cursor(self, row, column):
174         self.send_command(0xB0 + row)
175         self.send_command(0x00 + (8*column & 0x0F))
176         self.send_command(0x10 + ((8*column>>4)&0x0F))
177
178     def putc(self, c):
179         C_add = ord(c)
180         if C_add < 32 or C_add > 127:     # Ignore non-printable ASCII
181             c = ' '
182             C_add = ord(c)
183
184         for i in range(0, 8):
185             self.send_data(BasicFont[C_add-32][i])
186
187     def puts(self, text):
188         for c in text:
189             self.putc(c)
190
191     def show_image(self, image):
192         from PIL import Image
193         import numpy as np
194
195         im = Image.open(image)
196
```

```
        bw = im.convert('1')
        pixels = np.array(bw.getdata())
        page_size = 128 * 8

        self.set_cursor(0, 0)
        for page in range(8):
            start = page_size * page
            end = start + page_size

            for i in range(start, start + 128):
                data = np.packbits(pixels[i:end:128][::-1])[0]
                self.send_data(data)


def main():
    display = GroveOledDisplay128x64()

    display.set_cursor(0, 0)
    display.puts('hello')
    display.set_cursor(1, 4)
    display.puts('world')

if __name__ == "__main__":
    main()
```

It seems nothing happened in terminal, however you can find the most famous sentence in the cyber world if you check your oled.😄

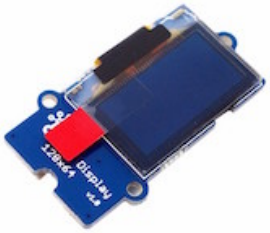
Note
If you use the I2C tool to scan the I2C address of the grove module, you may find two or more address. 0x04 is the adrress of the *Grove Base Hat for Raspberry Pi*.

Play With Raspberry Pi (with GrovePi_Plus)

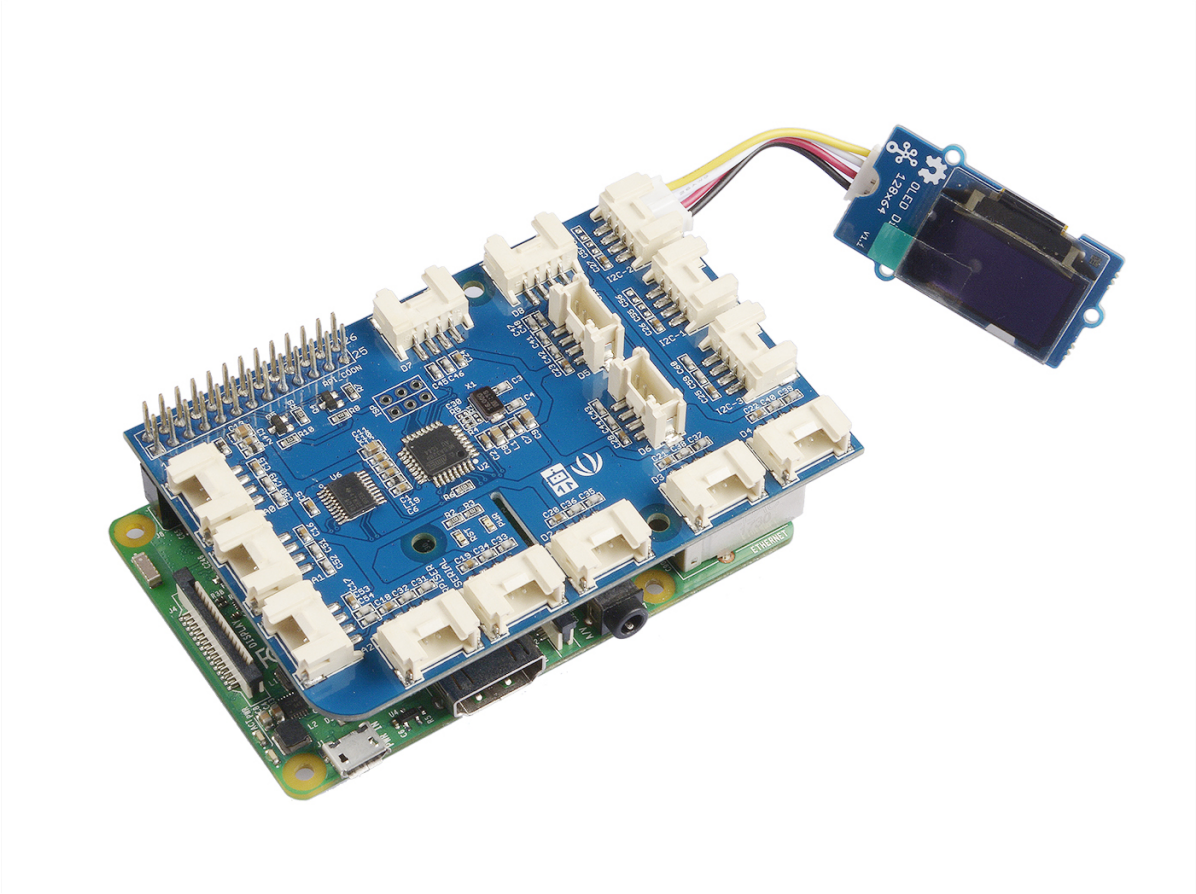
The python script for BeagleBone Green below also works for Raspberry Pi, But the I2C bus is different. On Raspberry Pi ⅔, the I2C bus is 1. The script with changing `bus=2` to `bus=1` will work on Pi.

Hardware

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

Raspberry pi	GrovePi_Plus	Grove - OLED Display 0.96 inch
		
Get One Now	Get One Now	Get One Now

- **Step 2.** Plug the GrovePi_Plus into Raspberry.
- **Step 3.** Connect Grove - OLED Display 0.96 inch ranger to **I2C-2** 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.** Navigate to the demos' directory:

```
```

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

- **Step 3.** To see the code

```
nano Grove-OLED_Display_0.96inch.py   # "Ctrl+x" to exit #
```

```
import time
import smbus2 as smbus

_COMMAND_MODE = 0x80
_DATA_MODE = 0x40
_NORMAL_DISPLAY = 0xA6

_DISPLAY_OFF = 0xAE
_DISPLAY_ON = 0xAF
_INVERSE_DISPLAY = 0xA7
_SET_BRIGHTNESS = 0x81


BasicFont = [[0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00],
             [0x00, 0x00, 0x5F, 0x00, 0x00, 0x00, 0x00, 0x00],
             [0x00, 0x00, 0x07, 0x00, 0x07, 0x00, 0x00, 0x00],
             [0x00, 0x14, 0x7F, 0x14, 0x7F, 0x14, 0x00, 0x00],
             [0x00, 0x24, 0x2A, 0x7F, 0x2A, 0x12, 0x00, 0x00],
             [0x00, 0x23, 0x13, 0x08, 0x64, 0x62, 0x00, 0x00],
             [0x00, 0x36, 0x49, 0x55, 0x22, 0x50, 0x00, 0x00],
             [0x00, 0x00, 0x05, 0x03, 0x00, 0x00, 0x00, 0x00],
             [0x00, 0x1C, 0x22, 0x41, 0x00, 0x00, 0x00, 0x00],
             [0x00, 0x41, 0x22, 0x1C, 0x00, 0x00, 0x00, 0x00],
             [0x00, 0x08, 0x2A, 0x1C, 0x2A, 0x08, 0x00, 0x00],
             [0x00, 0x08, 0x08, 0x3E, 0x08, 0x08, 0x00, 0x00],
```

```
26                [0x00, 0xA0, 0x60, 0x00, 0x00, 0x00, 0x00, 0x00],
27                [0x00, 0x08, 0x08, 0x08, 0x08, 0x08, 0x00, 0x00],
28                [0x00, 0x60, 0x60, 0x00, 0x00, 0x00, 0x00, 0x00],
29                [0x00, 0x20, 0x10, 0x08, 0x04, 0x02, 0x00, 0x00],
30                [0x00, 0x3E, 0x51, 0x49, 0x45, 0x3E, 0x00, 0x00],
31                [0x00, 0x00, 0x42, 0x7F, 0x40, 0x00, 0x00, 0x00],
32                [0x00, 0x62, 0x51, 0x49, 0x49, 0x46, 0x00, 0x00],
33                [0x00, 0x22, 0x41, 0x49, 0x49, 0x36, 0x00, 0x00],
34                [0x00, 0x18, 0x14, 0x12, 0x7F, 0x10, 0x00, 0x00],
35                [0x00, 0x27, 0x45, 0x45, 0x45, 0x39, 0x00, 0x00],
36                [0x00, 0x3C, 0x4A, 0x49, 0x49, 0x30, 0x00, 0x00],
37                [0x00, 0x01, 0x71, 0x09, 0x05, 0x03, 0x00, 0x00],
38                [0x00, 0x36, 0x49, 0x49, 0x49, 0x36, 0x00, 0x00],
39                [0x00, 0x06, 0x49, 0x49, 0x29, 0x1E, 0x00, 0x00],
40                [0x00, 0x00, 0x36, 0x36, 0x00, 0x00, 0x00, 0x00],
41                [0x00, 0x00, 0xAC, 0x6C, 0x00, 0x00, 0x00, 0x00],
42                [0x00, 0x08, 0x14, 0x22, 0x41, 0x00, 0x00, 0x00],
43                [0x00, 0x14, 0x14, 0x14, 0x14, 0x14, 0x00, 0x00],
44                [0x00, 0x41, 0x22, 0x14, 0x08, 0x00, 0x00, 0x00],
45                [0x00, 0x02, 0x01, 0x51, 0x09, 0x06, 0x00, 0x00],
46                [0x00, 0x32, 0x49, 0x79, 0x41, 0x3E, 0x00, 0x00],
47                [0x00, 0x7E, 0x09, 0x09, 0x09, 0x7E, 0x00, 0x00],
48                [0x00, 0x7F, 0x49, 0x49, 0x49, 0x36, 0x00, 0x00],
49                [0x00, 0x3E, 0x41, 0x41, 0x41, 0x22, 0x00, 0x00],
50                [0x00, 0x7F, 0x41, 0x41, 0x22, 0x1C, 0x00, 0x00],
51                [0x00, 0x7F, 0x49, 0x49, 0x49, 0x41, 0x00, 0x00],
52                [0x00, 0x7F, 0x09, 0x09, 0x09, 0x01, 0x00, 0x00],
53                [0x00, 0x3E, 0x41, 0x41, 0x51, 0x72, 0x00, 0x00],
54                [0x00, 0x7F, 0x08, 0x08, 0x08, 0x7F, 0x00, 0x00],
55                [0x00, 0x41, 0x7F, 0x41, 0x00, 0x00, 0x00, 0x00],
56                [0x00, 0x20, 0x40, 0x41, 0x3F, 0x01, 0x00, 0x00],
57                [0x00, 0x7F, 0x08, 0x14, 0x22, 0x41, 0x00, 0x00],
58                [0x00, 0x7F, 0x40, 0x40, 0x40, 0x40, 0x00, 0x00],
59                [0x00, 0x7F, 0x02, 0x0C, 0x02, 0x7F, 0x00, 0x00],
60                [0x00, 0x7F, 0x04, 0x08, 0x10, 0x7F, 0x00, 0x00],
```

```
61                [0x00, 0x3E, 0x41, 0x41, 0x41, 0x3E, 0x00, 0x00],
62                [0x00, 0x7F, 0x09, 0x09, 0x09, 0x06, 0x00, 0x00],
63                [0x00, 0x3E, 0x41, 0x51, 0x21, 0x5E, 0x00, 0x00],
64                [0x00, 0x7F, 0x09, 0x19, 0x29, 0x46, 0x00, 0x00],
65                [0x00, 0x26, 0x49, 0x49, 0x49, 0x32, 0x00, 0x00],
66                [0x00, 0x01, 0x01, 0x7F, 0x01, 0x01, 0x00, 0x00],
67                [0x00, 0x3F, 0x40, 0x40, 0x40, 0x3F, 0x00, 0x00],
68                [0x00, 0x1F, 0x20, 0x40, 0x20, 0x1F, 0x00, 0x00],
69                [0x00, 0x3F, 0x40, 0x38, 0x40, 0x3F, 0x00, 0x00],
70                [0x00, 0x63, 0x14, 0x08, 0x14, 0x63, 0x00, 0x00],
71                [0x00, 0x03, 0x04, 0x78, 0x04, 0x03, 0x00, 0x00],
72                [0x00, 0x61, 0x51, 0x49, 0x45, 0x43, 0x00, 0x00],
73                [0x00, 0x7F, 0x41, 0x41, 0x00, 0x00, 0x00, 0x00],
74                [0x00, 0x02, 0x04, 0x08, 0x10, 0x20, 0x00, 0x00],
75                [0x00, 0x41, 0x41, 0x7F, 0x00, 0x00, 0x00, 0x00],
76                [0x00, 0x04, 0x02, 0x01, 0x02, 0x04, 0x00, 0x00],
77                [0x00, 0x80, 0x80, 0x80, 0x80, 0x80, 0x00, 0x00],
78                [0x00, 0x01, 0x02, 0x04, 0x00, 0x00, 0x00, 0x00],
79                [0x00, 0x20, 0x54, 0x54, 0x54, 0x78, 0x00, 0x00],
80                [0x00, 0x7F, 0x48, 0x44, 0x44, 0x38, 0x00, 0x00],
81                [0x00, 0x38, 0x44, 0x44, 0x28, 0x00, 0x00, 0x00],
82                [0x00, 0x38, 0x44, 0x44, 0x48, 0x7F, 0x00, 0x00],
83                [0x00, 0x38, 0x54, 0x54, 0x54, 0x18, 0x00, 0x00],
84                [0x00, 0x08, 0x7E, 0x09, 0x02, 0x00, 0x00, 0x00],
85                [0x00, 0x18, 0xA4, 0xA4, 0xA4, 0x7C, 0x00, 0x00],
86                [0x00, 0x7F, 0x08, 0x04, 0x04, 0x78, 0x00, 0x00],
87                [0x00, 0x00, 0x7D, 0x00, 0x00, 0x00, 0x00, 0x00],
88                [0x00, 0x80, 0x84, 0x7D, 0x00, 0x00, 0x00, 0x00],
89                [0x00, 0x7F, 0x10, 0x28, 0x44, 0x00, 0x00, 0x00],
90                [0x00, 0x41, 0x7F, 0x40, 0x00, 0x00, 0x00, 0x00],
91                [0x00, 0x7C, 0x04, 0x18, 0x04, 0x78, 0x00, 0x00],
92                [0x00, 0x7C, 0x08, 0x04, 0x7C, 0x00, 0x00, 0x00],
93                [0x00, 0x38, 0x44, 0x44, 0x38, 0x00, 0x00, 0x00],
94                [0x00, 0xFC, 0x24, 0x24, 0x18, 0x00, 0x00, 0x00],
95                [0x00, 0x18, 0x24, 0x24, 0xFC, 0x00, 0x00, 0x00],
```

```
 96                [0x00, 0x00, 0x7C, 0x08, 0x04, 0x00, 0x00, 0x00],
 97                [0x00, 0x48, 0x54, 0x54, 0x24, 0x00, 0x00, 0x00],
 98                [0x00, 0x04, 0x7F, 0x44, 0x00, 0x00, 0x00, 0x00],
 99                [0x00, 0x3C, 0x40, 0x40, 0x7C, 0x00, 0x00, 0x00],
100                [0x00, 0x1C, 0x20, 0x40, 0x20, 0x1C, 0x00, 0x00],
101                [0x00, 0x3C, 0x40, 0x30, 0x40, 0x3C, 0x00, 0x00],
102                [0x00, 0x44, 0x28, 0x10, 0x28, 0x44, 0x00, 0x00],
103                [0x00, 0x1C, 0xA0, 0xA0, 0x7C, 0x00, 0x00, 0x00],
104                [0x00, 0x44, 0x64, 0x54, 0x4C, 0x44, 0x00, 0x00],
105                [0x00, 0x08, 0x36, 0x41, 0x00, 0x00, 0x00, 0x00],
106                [0x00, 0x00, 0x7F, 0x00, 0x00, 0x00, 0x00, 0x00],
107                [0x00, 0x41, 0x36, 0x08, 0x00, 0x00, 0x00, 0x00],
108                [0x00, 0x02, 0x01, 0x01, 0x02, 0x01, 0x00, 0x00],
109                [0x00, 0x02, 0x05, 0x05, 0x02, 0x00, 0x00, 0x00]]
110
111
112  class GroveOledDisplay128x64(object):
113      HORIZONTAL = 0x00
114      VERTICAL = 0x01
115      PAGE = 0x02
116
117      def __init__(self, bus=2, address=0x3C):
118          self.bus = smbus.SMBus(bus)
119          self.address = address
120
121          self.off()
122          self.inverse = False
123          self.mode = self.HORIZONTAL
124
125          self.clear()
126          self.on()
127
128      def on(self):
129          self.send_command(_DISPLAY_ON)
130
```

```
131     def off(self):
132         self.send_command(_DISPLAY_OFF)
133 
134     def send_command(self, command):
135         self.bus.write_byte_data(self.address, _COMMAND_MODE, command)
136 
137     def send_data(self, data):
138         self.bus.write_byte_data(self.address, _DATA_MODE, data)
139 
140     def send_commands(self, commands):
141         for c in commands:
142             self.send_command(c)
143 
144     def clear(self):
145         self.off()
146         for i in range(8):
147             self.set_cursor(i, 0)
148             self.puts(' ' * 16)
149 
150         self.on()
151         self.set_cursor(0, 0)
152 
153     @property
154     def inverse(self):
155         return self._inverse
156 
157     @inverse.setter
158     def inverse(self, enable):
159         self.send_command(_INVERSE_DISPLAY if enable else _NORMAL_DISPl
160         self._inverse = enable
161 
162     @property
163     def mode(self):
164         return self._mode
165 
```

```
    @mode.setter
    def mode(self, mode):
        self.send_command(0x20)
        self.send_command(mode)
        self._mode = mode

    def set_cursor(self, row, column):
        self.send_command(0xB0 + row)
        self.send_command(0x00 + (8*column & 0x0F))
        self.send_command(0x10 + ((8*column>>4)&0x0F))

    def putc(self, c):
        C_add = ord(c)
        if C_add < 32 or C_add > 127:     # Ignore non-printable ASCII
            c = ' '
            C_add = ord(c)

        for i in range(0, 8):
            self.send_data(BasicFont[C_add-32][i])

    def puts(self, text):
        for c in text:
            self.putc(c)

    def show_image(self, image):
        from PIL import Image
        import numpy as np

        im = Image.open(image)

        bw = im.convert('1')
        pixels = np.array(bw.getdata())
        page_size = 128 * 8

        self.set_cursor(0, 0)
```

```
        for page in range(8):
            start = page_size * page
            end = start + page_size

            for i in range(start, start + 128):
                data = np.packbits(pixels[i:end:128][::-1])[0]
                self.send_data(data)


if __name__ == "__main__":
    display = GroveOledDisplay128x64(bus=1)

    display.set_cursor(0, 0)
    display.puts('hello')
    display.set_cursor(1, 4)
    display.puts('world')
```

- **Step 4:** Install smbus2 library `sudo pip install smbus2`
- **Step 5:** Run the code. We'll find that the Grove - OLED outputs "Hello World".

```
sudo python Grove-OLED_Display_0.96inch.py
```

Play With Beaglebone Green

To begin editing programs that live on BBG, you can use the Cloud9 IDE and refer Beaglebone Green Wiki.

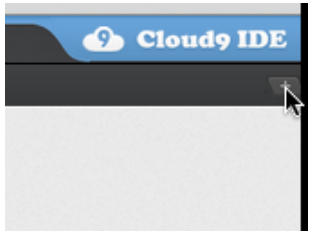
Here are the steps how to display "Hello World" on OLED.

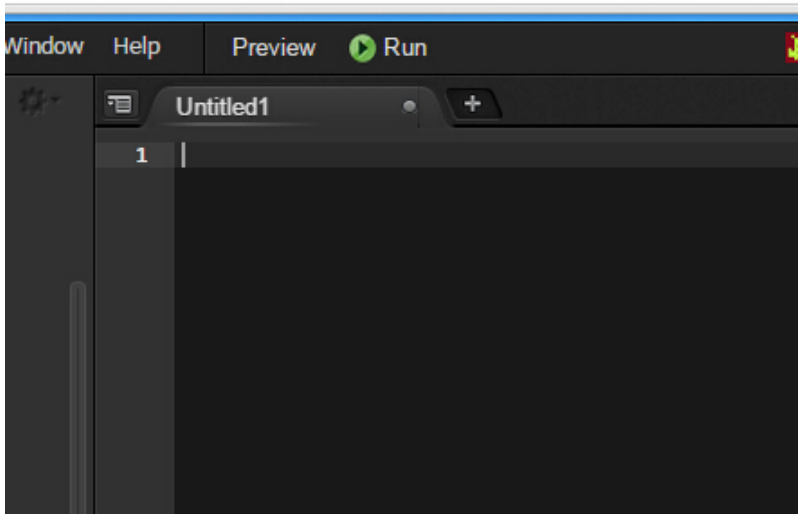
Hardware

- **Step 1.** Connect Grove - OLED to Grove I2C socket on BBG.
- **Step 2.** Connect Seeeduino to PC via a USB cable.

Software

- **Step 1:** Click the "+" in the top-right to create a new file.





- **Step 2:** Copy and paste the following code into the new tab

```
import time
import smbus2 as smbus

_COMMAND_MODE = 0x80
_DATA_MODE = 0x40
_NORMAL_DISPLAY = 0xA6

_DISPLAY_OFF = 0xAE
_DISPLAY_ON = 0xAF
_INVERSE_DISPLAY = 0xA7
_SET_BRIGHTNESS = 0x81


BasicFont = [[0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00],
             [0x00, 0x00, 0x5F, 0x00, 0x00, 0x00, 0x00, 0x00],
             [0x00, 0x00, 0x07, 0x00, 0x07, 0x00, 0x00, 0x00],
             [0x00, 0x14, 0x7F, 0x14, 0x7F, 0x14, 0x00, 0x00],
             [0x00, 0x24, 0x2A, 0x7F, 0x2A, 0x12, 0x00, 0x00],
             [0x00, 0x23, 0x13, 0x08, 0x64, 0x62, 0x00, 0x00],
             [0x00, 0x36, 0x49, 0x55, 0x22, 0x50, 0x00, 0x00],
             [0x00, 0x00, 0x05, 0x03, 0x00, 0x00, 0x00, 0x00],
             [0x00, 0x1C, 0x22, 0x41, 0x00, 0x00, 0x00, 0x00],
             [0x00, 0x41, 0x22, 0x1C, 0x00, 0x00, 0x00, 0x00],
             [0x00, 0x08, 0x2A, 0x1C, 0x2A, 0x08, 0x00, 0x00],
             [0x00, 0x08, 0x08, 0x3E, 0x08, 0x08, 0x00, 0x00],
             [0x00, 0xA0, 0x60, 0x00, 0x00, 0x00, 0x00, 0x00],
             [0x00, 0x08, 0x08, 0x08, 0x08, 0x08, 0x00, 0x00],
             [0x00, 0x60, 0x60, 0x00, 0x00, 0x00, 0x00, 0x00],
             [0x00, 0x20, 0x10, 0x08, 0x04, 0x02, 0x00, 0x00],
             [0x00, 0x3E, 0x51, 0x49, 0x45, 0x3E, 0x00, 0x00],
             [0x00, 0x00, 0x42, 0x7F, 0x40, 0x00, 0x00, 0x00],
             [0x00, 0x62, 0x51, 0x49, 0x49, 0x46, 0x00, 0x00],
```

```
33                [0x00, 0x22, 0x41, 0x49, 0x49, 0x36, 0x00, 0x00],
34                [0x00, 0x18, 0x14, 0x12, 0x7F, 0x10, 0x00, 0x00],
35                [0x00, 0x27, 0x45, 0x45, 0x45, 0x39, 0x00, 0x00],
36                [0x00, 0x3C, 0x4A, 0x49, 0x49, 0x30, 0x00, 0x00],
37                [0x00, 0x01, 0x71, 0x09, 0x05, 0x03, 0x00, 0x00],
38                [0x00, 0x36, 0x49, 0x49, 0x49, 0x36, 0x00, 0x00],
39                [0x00, 0x06, 0x49, 0x49, 0x29, 0x1E, 0x00, 0x00],
40                [0x00, 0x00, 0x36, 0x36, 0x00, 0x00, 0x00, 0x00],
41                [0x00, 0x00, 0xAC, 0x6C, 0x00, 0x00, 0x00, 0x00],
42                [0x00, 0x08, 0x14, 0x22, 0x41, 0x00, 0x00, 0x00],
43                [0x00, 0x14, 0x14, 0x14, 0x14, 0x14, 0x00, 0x00],
44                [0x00, 0x41, 0x22, 0x14, 0x08, 0x00, 0x00, 0x00],
45                [0x00, 0x02, 0x01, 0x51, 0x09, 0x06, 0x00, 0x00],
46                [0x00, 0x32, 0x49, 0x79, 0x41, 0x3E, 0x00, 0x00],
47                [0x00, 0x7E, 0x09, 0x09, 0x09, 0x7E, 0x00, 0x00],
48                [0x00, 0x7F, 0x49, 0x49, 0x49, 0x36, 0x00, 0x00],
49                [0x00, 0x3E, 0x41, 0x41, 0x41, 0x22, 0x00, 0x00],
50                [0x00, 0x7F, 0x41, 0x41, 0x22, 0x1C, 0x00, 0x00],
51                [0x00, 0x7F, 0x49, 0x49, 0x49, 0x41, 0x00, 0x00],
52                [0x00, 0x7F, 0x09, 0x09, 0x09, 0x01, 0x00, 0x00],
53                [0x00, 0x3E, 0x41, 0x41, 0x51, 0x72, 0x00, 0x00],
54                [0x00, 0x7F, 0x08, 0x08, 0x08, 0x7F, 0x00, 0x00],
55                [0x00, 0x41, 0x7F, 0x41, 0x00, 0x00, 0x00, 0x00],
56                [0x00, 0x20, 0x40, 0x41, 0x3F, 0x01, 0x00, 0x00],
57                [0x00, 0x7F, 0x08, 0x14, 0x22, 0x41, 0x00, 0x00],
58                [0x00, 0x7F, 0x40, 0x40, 0x40, 0x40, 0x00, 0x00],
59                [0x00, 0x7F, 0x02, 0x0C, 0x02, 0x7F, 0x00, 0x00],
60                [0x00, 0x7F, 0x04, 0x08, 0x10, 0x7F, 0x00, 0x00],
61                [0x00, 0x3E, 0x41, 0x41, 0x41, 0x3E, 0x00, 0x00],
62                [0x00, 0x7F, 0x09, 0x09, 0x09, 0x06, 0x00, 0x00],
63                [0x00, 0x3E, 0x41, 0x51, 0x21, 0x5E, 0x00, 0x00],
64                [0x00, 0x7F, 0x09, 0x19, 0x29, 0x46, 0x00, 0x00],
65                [0x00, 0x26, 0x49, 0x49, 0x49, 0x32, 0x00, 0x00],
66                [0x00, 0x01, 0x01, 0x7F, 0x01, 0x01, 0x00, 0x00],
67                [0x00, 0x3F, 0x40, 0x40, 0x40, 0x3F, 0x00, 0x00],
```

```
68                [0x00, 0x1F, 0x20, 0x40, 0x20, 0x1F, 0x00, 0x00],
69                [0x00, 0x3F, 0x40, 0x38, 0x40, 0x3F, 0x00, 0x00],
70                [0x00, 0x63, 0x14, 0x08, 0x14, 0x63, 0x00, 0x00],
71                [0x00, 0x03, 0x04, 0x78, 0x04, 0x03, 0x00, 0x00],
72                [0x00, 0x61, 0x51, 0x49, 0x45, 0x43, 0x00, 0x00],
73                [0x00, 0x7F, 0x41, 0x41, 0x00, 0x00, 0x00, 0x00],
74                [0x00, 0x02, 0x04, 0x08, 0x10, 0x20, 0x00, 0x00],
75                [0x00, 0x41, 0x41, 0x7F, 0x00, 0x00, 0x00, 0x00],
76                [0x00, 0x04, 0x02, 0x01, 0x02, 0x04, 0x00, 0x00],
77                [0x00, 0x80, 0x80, 0x80, 0x80, 0x80, 0x00, 0x00],
78                [0x00, 0x01, 0x02, 0x04, 0x00, 0x00, 0x00, 0x00],
79                [0x00, 0x20, 0x54, 0x54, 0x54, 0x78, 0x00, 0x00],
80                [0x00, 0x7F, 0x48, 0x44, 0x44, 0x38, 0x00, 0x00],
81                [0x00, 0x38, 0x44, 0x44, 0x28, 0x00, 0x00, 0x00],
82                [0x00, 0x38, 0x44, 0x44, 0x48, 0x7F, 0x00, 0x00],
83                [0x00, 0x38, 0x54, 0x54, 0x54, 0x18, 0x00, 0x00],
84                [0x00, 0x08, 0x7E, 0x09, 0x02, 0x00, 0x00, 0x00],
85                [0x00, 0x18, 0xA4, 0xA4, 0xA4, 0x7C, 0x00, 0x00],
86                [0x00, 0x7F, 0x08, 0x04, 0x04, 0x78, 0x00, 0x00],
87                [0x00, 0x00, 0x7D, 0x00, 0x00, 0x00, 0x00, 0x00],
88                [0x00, 0x80, 0x84, 0x7D, 0x00, 0x00, 0x00, 0x00],
89                [0x00, 0x7F, 0x10, 0x28, 0x44, 0x00, 0x00, 0x00],
90                [0x00, 0x41, 0x7F, 0x40, 0x00, 0x00, 0x00, 0x00],
91                [0x00, 0x7C, 0x04, 0x18, 0x04, 0x78, 0x00, 0x00],
92                [0x00, 0x7C, 0x08, 0x04, 0x7C, 0x00, 0x00, 0x00],
93                [0x00, 0x38, 0x44, 0x44, 0x38, 0x00, 0x00, 0x00],
94                [0x00, 0xFC, 0x24, 0x24, 0x18, 0x00, 0x00, 0x00],
95                [0x00, 0x18, 0x24, 0x24, 0xFC, 0x00, 0x00, 0x00],
96                [0x00, 0x00, 0x7C, 0x08, 0x04, 0x00, 0x00, 0x00],
97                [0x00, 0x48, 0x54, 0x54, 0x24, 0x00, 0x00, 0x00],
98                [0x00, 0x04, 0x7F, 0x44, 0x00, 0x00, 0x00, 0x00],
99                [0x00, 0x3C, 0x40, 0x40, 0x7C, 0x00, 0x00, 0x00],
100               [0x00, 0x1C, 0x20, 0x40, 0x20, 0x1C, 0x00, 0x00],
101               [0x00, 0x3C, 0x40, 0x30, 0x40, 0x3C, 0x00, 0x00],
102               [0x00, 0x44, 0x28, 0x10, 0x28, 0x44, 0x00, 0x00],
```

```
            [0x00, 0x1C, 0xA0, 0xA0, 0x7C, 0x00, 0x00, 0x00],
            [0x00, 0x44, 0x64, 0x54, 0x4C, 0x44, 0x00, 0x00],
            [0x00, 0x08, 0x36, 0x41, 0x00, 0x00, 0x00, 0x00],
            [0x00, 0x00, 0x7F, 0x00, 0x00, 0x00, 0x00, 0x00],
            [0x00, 0x41, 0x36, 0x08, 0x00, 0x00, 0x00, 0x00],
            [0x00, 0x02, 0x01, 0x01, 0x02, 0x01, 0x00, 0x00],
            [0x00, 0x02, 0x05, 0x05, 0x02, 0x00, 0x00, 0x00]]


class GroveOledDisplay128x64(object):
    HORIZONTAL = 0x00
    VERTICAL = 0x01
    PAGE = 0x02

    def __init__(self, bus=2, address=0x3C):
        self.bus = smbus.SMBus(bus)
        self.address = address

        self.off()
        self.inverse = False
        self.mode = self.HORIZONTAL

        self.clear()
        self.on()

    def on(self):
        self.send_command(_DISPLAY_ON)

    def off(self):
        self.send_command(_DISPLAY_OFF)

    def send_command(self, command):
        self.bus.write_byte_data(self.address, _COMMAND_MODE, command)

    def send_data(self, data):
```

```
138         self.bus.write_byte_data(self.address, _DATA_MODE, data)
139
140     def send_commands(self, commands):
141         for c in commands:
142             self.send_command(c)
143
144     def clear(self):
145         self.off()
146         for i in range(8):
147             self.set_cursor(i, 0)
148             self.puts(' ' * 16)
149
150         self.on()
151         self.set_cursor(0, 0)
152
153     @property
154     def inverse(self):
155         return self._inverse
156
157     @inverse.setter
158     def inverse(self, enable):
159         self.send_command(_INVERSE_DISPLAY if enable else _NORMAL_DISPl
160         self._inverse = enable
161
162     @property
163     def mode(self):
164         return self._mode
165
166     @mode.setter
167     def mode(self, mode):
168         self.send_command(0x20)
169         self.send_command(mode)
170         self._mode = mode
171
172     def set_cursor(self, row, column):
```

```
173         self.send_command(0xB0 + row)
174         self.send_command(0x00 + (8*column & 0x0F))
175         self.send_command(0x10 + ((8*column>>4)&0x0F))
176 
177     def putc(self, c):
178         C_add = ord(c)
179         if C_add < 32 or C_add > 127:     # Ignore non-printable ASCII
180             c = ' '
181             C_add = ord(c)
182 
183         for i in range(0, 8):
184             self.send_data(BasicFont[C_add-32][i])
185 
186     def puts(self, text):
187         for c in text:
188             self.putc(c)
189 
190     def show_image(self, image):
191         from PIL import Image
192         import numpy as np
193 
194         im = Image.open(image)
195 
196         bw = im.convert('1')
197         pixels = np.array(bw.getdata())
198         page_size = 128 * 8
199 
200         self.set_cursor(0, 0)
201         for page in range(8):
202             start = page_size * page
203             end = start + page_size
204 
205             for i in range(start, start + 128):
206                 data = np.packbits(pixels[i:end:128][::-1])[0]
207                 self.send_data(data)
```

```
208
209
210 if __name__ == "__main__":
211     display = GroveOledDisplay128x64(bus=2)
212
213     display.set_cursor(0, 0)
214     display.puts('hello')
215     display.set_cursor(1, 4)
216     display.puts('world')
```

- **Step 3:** Save the file by clicking the disk icon with with the .py extension.
- **Step 4:** Install smbus2 library `sudo pip3 install smbus2`
- **Step 5:** Run the code. We'll find that the Grove - OLED outputs "Hello World".

```
sudo python3 Grove-OLED_Display_0.96inch.py
```

Schematic Online Viewer

Resources

- **[PDF]** Grove-OLED128x64 Schematic
- **[PDF]** Grove-OLED128x64 PCB

- **[Wiki]** Beaglebone Green Wiki
- **[Eagle]** Grove-OLED128x64
- **[Library]** GitHub Library for OLED
- **[Datasheet]** Resources of SSD1308_1.0.pdf
- **[Datasheet]** Resources of LY190-128064.pdf

Tech Support

Please submit any technical issue into our forum.