



Grove - OLED Display 1.12"

Release date : 9/20/2015

Version : 1.0

Wiki: [http://www.seeedstudio.com/wiki/Grove - OLED Display 128*64](http://www.seeedstudio.com/wiki/Grove_-_OLED_Display_128*64)

Bazaar: <http://www.seeedstudio.com/depot/Grove-OLED-Display-112-p-781.html>

Document Revision History

Revision	Date	Author	Description
1.0	Sep 21, 2015	Victor.He	Create file

Contents

Document Revision History	2
1. Introduction	2
2. Features	3
3. Specifications	4
4. Usage	5
4.1 With Arduino	5
4.2 With Beaglebone Green	10
5. Resources	17
6. Support	18

Disclaimer

For physical injuries and possessions loss caused by those reasons which are not related to product quality, such as operating without following manual guide, natural disasters or force majeure, we take no responsibility for that.

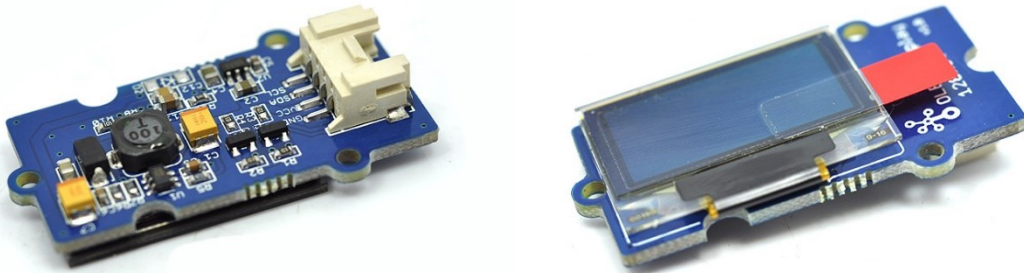
Under the supervision of Seeed Technology Inc., this manual has been compiled and published which covered the latest product description and specification. The content of this manual is subject to change without notice.

Copyright

The design of this product (including software) and its accessories is under tutelage of laws. Any action to violate relevant right of our product will be penalized through law. Please consciously observe relevant local laws in the use of this product.

1. Introduction

LED 128×64 Display module is an OLED monochrome 128×64dot matrix display module with Grove 4pin I2C Interface. Comparing to LCD, OLED screens are way 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.



2. Features

- Grove compatible interface
- Communicate Mode:I2C
- Low power consumption
- Display Color: White
- Wide range of operating temperature:-20°C~70°C

3. Specifications

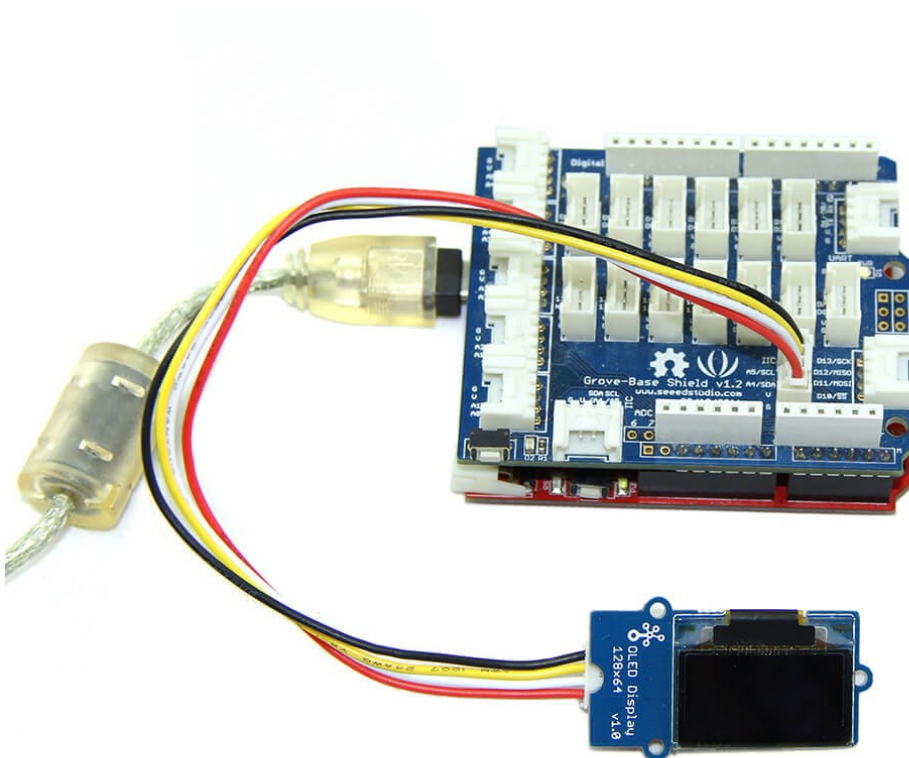
Items	Min	Norm	Max	Unit
Power Voltage (VCC)	3.3	5.0	5.5	V
Driver IC	SSD1308Z			-
Display Color	White			-
Dot Matrix	128×64			-
Panel Size	26.7(W)×19.26(H)			mm
Active Area	21.74(W)×11.175 (H)			mm
Dot Pitch	0.17(W)×0.175 (H)			mm
Dot Size	0.15(W)×0.15 (H)			mm
Wide range of operating temperature	-20~70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

4. Usage

4.1 With Arduino

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 consult [File:SSD1308_1.0.pdf](#) and [File:LY190-128064.pdf](#).

- Here we demonstrate how to display "Seedstudio" on the screen.
- Plug the Grove - OLED Display 128*64 onto the I2C port on Grove - Base Shield, and then plug the Base Shield onto Arduino;
- Error creating thumbnail: Invalid thumbnail parameters.



- Download the library [File:OLED_Display128X64 Library](#);
- Unzip it into the libraries file of Arduino IDE by the path: `..\arduino-1.0\libraries`.
- Open the code directly by the path: File -> Example
->OLED_Display12864->OLED_Bitmap_Inverse_Display.

```
#include <Wire.h>
#include <SeedOLED.h>
```



```
#include <avr/pgmspace.h>

static unsigned char SeedLogo[] PROGMEM ={
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x80, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x80,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00,
  0x60, 0xf0, 0xc0, 0x00, 0x00, 0x00, 0xfc, 0xff, 0x87, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x03,
  0xff, 0xfc, 0x00, 0x00, 0x00, 0x80, 0xf0, 0x20, 0x00, 0x00, 0x80,
  0xc0, 0xc0, 0x60, 0xe0, 0xc0,
  0xc0, 0x00, 0x00, 0x00, 0xc0, 0xc0, 0xc0, 0x60, 0xe0, 0xc0, 0xc0,
  0x80, 0x00, 0x00, 0x80, 0xc0,
  0xc0, 0xe0, 0x60, 0xc0, 0xc0, 0x80, 0x00, 0x00, 0x00, 0xc0, 0xc0,
  0xc0, 0x60, 0xe0, 0xc0, 0xc0,
  0x80, 0x00, 0x00, 0x80, 0xc0, 0xc0, 0xe0, 0xe0, 0xc0, 0xc0, 0xf8,
  0xf8, 0x00, 0x00, 0x00, 0x00,
  0x00, 0xc0, 0xc0, 0xe0, 0x60, 0xc0, 0xc0, 0x80, 0x00, 0xc0, 0xf0,
  0xf0, 0xf0, 0xc0, 0x00, 0xc0,
  0xc0, 0x00, 0x00, 0x00, 0x00, 0xc0, 0xc0, 0x00, 0x00, 0x80, 0xc0,
  0xc0, 0xe0, 0xe0, 0xc0, 0xc0,
  0xf8, 0xf8, 0x00, 0xd8, 0xd8, 0x00, 0x00, 0x80, 0xc0, 0xc0, 0xe0,
  0x60, 0xc0, 0xc0, 0x80, 0x00,
  0x00, 0x03, 0x0f, 0x1e, 0x3c, 0x70, 0xe3, 0xcf, 0x9f, 0x30, 0x00,
  0x00, 0x00, 0x00, 0x70, 0xbf,
  0xcf, 0xe3, 0x70, 0x78, 0x3e, 0x0f, 0x03, 0x00, 0x00, 0x00, 0x33,
  0x77, 0x66, 0x66, 0x66, 0x6c,
  0x7d, 0x18, 0x00, 0x1f, 0x3f, 0x76, 0x66, 0x66, 0x66, 0x76, 0x37,
  0x07, 0x00, 0x0f, 0x3f, 0x7f,
  0x66, 0x66, 0x66, 0x66, 0x77, 0x27, 0x07, 0x00, 0x1f, 0x3f, 0x76,
  0x66, 0x66, 0x66, 0x76, 0x37,
```



```
};

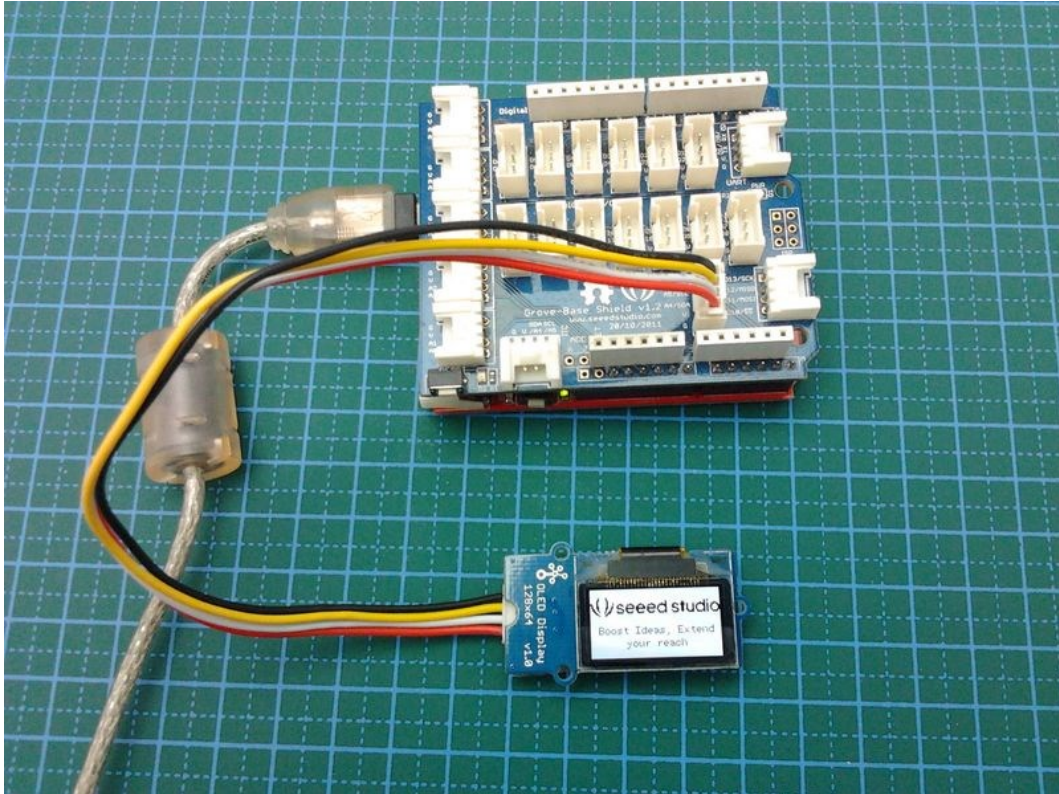
void setup()
{
  Wire.begin();
  SeeedOled.init(); //initialize SEEED OLED display
  DDRB |= 0x21;      //digital pin 8, LED glow indicates OLED properly
  Connected .
  PORTB |= 0x21;

  SeeedOled.setInverseDisplay(); // Set Display to inverse mode
  SeeedOled.clearDisplay();      // clear the screen and set
  start position to top left corner
  SeeedOled.drawBitmap(SeeedLogo,1024); // 1024 = 128 Pixels * 64
  Pixels / 8
}

void loop()
{
}
}
```

- Upload the code. Please click here if you do not know how to upload.

You can see:



You can open other examples in the same menu bar. The key code here is the main instruction program extracted by the example, hope it can be helpful for you to understand clearly. For more details, please check the Demo code Resources linked.

Note:This module is similar to [Grove - OLED Display 96*96](#) in its usage and some library functions, you can make an gray 128*64 image arbitrary that can be shown on OLED 128*64. Please consult the latter one's **Usage** part of [WIKI](#) page for detailed instructions.

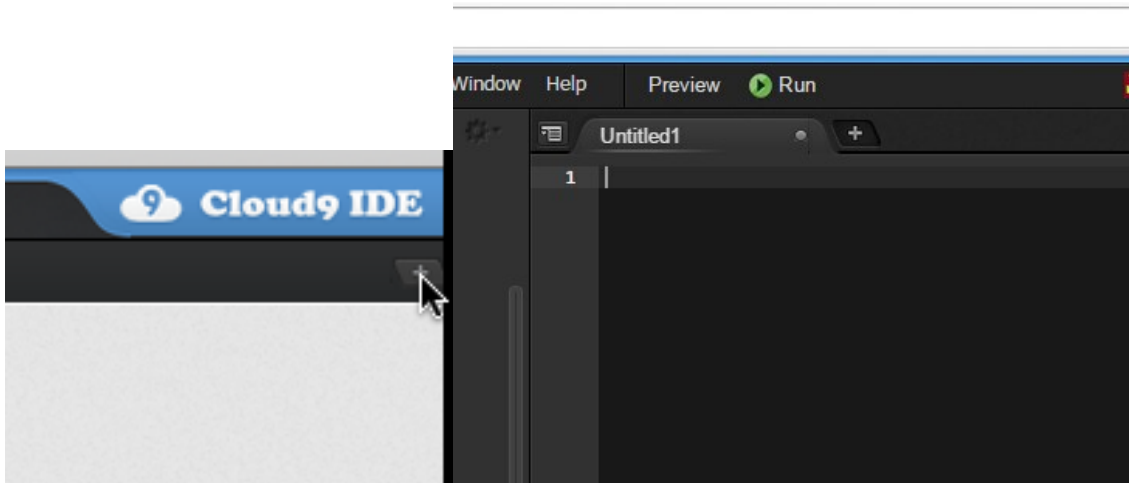
4.2 With Beaglebone Green

To begin editing programs that live on BBG, you can use the Cloud9 IDE.

As a simple exercise to become familiar with Cloud9 IDE, creating a simple application to blink one of the 4 user programmable LEDs on the BeagleBone is a good start.

If this is your first time to use Cloud9 IDE, please follow this [link](#).

Step1: Click the "+" in the top-right to create a new file.



Step2: Copy and paste the following code into the new tab

```

from Adafruit_I2C import Adafruit_I2C
import time
import math

Oled = Adafruit_I2C(0x3c)
Command_Mode=0x80
Data_mode=0x40

grayH= 0xF0
grayL= 0x0F
Normal_Display_Cmd=0xA4

BasicFont = [[0 for x in xrange(8)] for x in xrange(10)]
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],  
[0x00,0x20,0x40,0x41,0x3F,0x01,0x00,0x00],  
[0x00,0x7F,0x08,0x14,0x22,0x41,0x00,0x00],  
[0x00,0x7F,0x40,0x40,0x40,0x40,0x00,0x00],  
[0x00,0x7F,0x02,0x0C,0x02,0x7F,0x00,0x00],  
[0x00,0x7F,0x04,0x08,0x10,0x7F,0x00,0x00],  
[0x00,0x3E,0x41,0x41,0x41,0x3E,0x00,0x00],  
[0x00,0x7F,0x09,0x09,0x09,0x06,0x00,0x00],  
[0x00,0x3E,0x41,0x51,0x21,0x5E,0x00,0x00],  
[0x00,0x7F,0x09,0x19,0x29,0x46,0x00,0x00],  
[0x00,0x26,0x49,0x49,0x49,0x32,0x00,0x00],  
[0x00,0x01,0x01,0x7F,0x01,0x01,0x00,0x00],  
[0x00,0x3F,0x40,0x40,0x40,0x3F,0x00,0x00],  
[0x00,0x1F,0x20,0x40,0x20,0x1F,0x00,0x00],  
[0x00,0x3F,0x40,0x38,0x40,0x3F,0x00,0x00],  
[0x00,0x63,0x14,0x08,0x14,0x63,0x00,0x00],  
[0x00,0x03,0x04,0x78,0x04,0x03,0x00,0x00],  
[0x00,0x61,0x51,0x49,0x45,0x43,0x00,0x00],  
[0x00,0x7F,0x41,0x41,0x00,0x00,0x00,0x00],
```

```

[0x00,0x02,0x04,0x08,0x10,0x20,0x00,0x00],
[0x00,0x41,0x41,0x7F,0x00,0x00,0x00,0x00],
[0x00,0x04,0x02,0x01,0x02,0x04,0x00,0x00],
[0x00,0x80,0x80,0x80,0x80,0x80,0x00,0x00],
[0x00,0x01,0x02,0x04,0x00,0x00,0x00,0x00],
[0x00,0x20,0x54,0x54,0x54,0x78,0x00,0x00],
[0x00,0x7F,0x48,0x44,0x44,0x38,0x00,0x00],
[0x00,0x38,0x44,0x44,0x28,0x00,0x00,0x00],
[0x00,0x38,0x44,0x44,0x48,0x7F,0x00,0x00],
[0x00,0x38,0x54,0x54,0x54,0x18,0x00,0x00],
[0x00,0x08,0x7E,0x09,0x02,0x00,0x00,0x00],
[0x00,0x18,0xA4,0xA4,0xA4,0x7C,0x00,0x00],
[0x00,0x7F,0x08,0x04,0x04,0x78,0x00,0x00],
[0x00,0x00,0x7D,0x00,0x00,0x00,0x00,0x00],
[0x00,0x80,0x84,0x7D,0x00,0x00,0x00,0x00],
[0x00,0x7F,0x10,0x28,0x44,0x00,0x00,0x00],
[0x00,0x41,0x7F,0x40,0x00,0x00,0x00,0x00],
[0x00,0x7C,0x04,0x18,0x04,0x78,0x00,0x00],
[0x00,0x7C,0x08,0x04,0x7C,0x00,0x00,0x00],
[0x00,0x38,0x44,0x44,0x38,0x00,0x00,0x00],
[0x00,0xFC,0x24,0x24,0x18,0x00,0x00,0x00],
[0x00,0x18,0x24,0x24,0xFC,0x00,0x00,0x00],
[0x00,0x00,0x7C,0x08,0x04,0x00,0x00,0x00],
[0x00,0x48,0x54,0x54,0x24,0x00,0x00,0x00],
[0x00,0x04,0x7F,0x44,0x00,0x00,0x00,0x00],
[0x00,0x3C,0x40,0x40,0x7C,0x00,0x00,0x00],
[0x00,0x1C,0x20,0x40,0x20,0x1C,0x00,0x00],
[0x00,0x3C,0x40,0x30,0x40,0x3C,0x00,0x00],
[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]]

```

```

def oled_init():
    sendCommand(0xFD) # Unlock OLED driver IC MCU interface from entering
command. i.e: Accept commands
    sendCommand(0x12)
    sendCommand(0xAE) # Set display off
    sendCommand(0xA8) # set multiplex ratio
    sendCommand(0x5F) # 96

```



```

sendCommand(0xA1) # set display start line
sendCommand(0x00)
sendCommand(0xA2) # set display offset
sendCommand(0x60)
sendCommand(0xA0) # set remap
sendCommand(0x46)
sendCommand(0xAB) # set vdd internal
sendCommand(0x01)
sendCommand(0x81) # set contrast
sendCommand(0x53) # 100 nit
sendCommand(0xB1) # Set Phase Length
sendCommand(0x51)
sendCommand(0xB3) # Set Display Clock Divide Ratio/Oscillator
Frequency
sendCommand(0x01)
sendCommand(0xB9)
sendCommand(0xBC) # set pre_charge voltage/VCOMH
sendCommand(0x08) # (0x08);
sendCommand(0xBE) # set VCOMH
sendCommand(0x07) # (0x07);
sendCommand(0xB6) # Set second pre-charge period
sendCommand(0x01)
sendCommand(0xD5) # enable second precharge and external vs1
sendCommand(0x62) # (0x62);
sendCommand(0xA4) # Set Normal Display Mode
sendCommand(0x2E) # Deactivate Scroll
sendCommand(0xAF) # Switch on display
time.sleep(0.1)
# delay(100);

# Row Address
sendCommand(0x75) # Set Row Address
sendCommand(0x00) # Start 0
sendCommand(0x5f) # End 95

# Column Address
sendCommand(0x15) # Set Column Address
sendCommand(0x08) # Start from 8th Column of driver IC. This is 0th
Column for OLED
sendCommand(0x37) # End at (8 + 47)th column. Each Column has 2
pixels(segments)

# Init gray level for text. Default:Brightest White

```

```

grayH= 0xF0
grayL= 0x0F

def sendCommand(byte):
    Oled.write8(Command_Mode,byte)

def sendData(byte):
    Oled.write8(Data_mode,byte)

def multi_comm(commands):
    for c in commands:
        sendCommand(c)

def oled_clearDisplay():
    for j in range (0,48):
        for i in range (0,96):
            sendData(0x00)

def oled_setNormalDisplay():
    sendCommand(Normal_Display_Cmd)

def oled_setVerticalMode():
    sendCommand(0xA0)    # remap to
    sendCommand(0x46)    # Vertical mode

def oled_setTextXY(Row,Column):
    sendCommand(0x15)      # Set Column Address
    sendCommand(0x08+(Column*4)) # Start Column: Start from 8
    sendCommand(0x37)      # End Column
    # Row Address
    sendCommand(0x75)      # Set Row Address
    sendCommand(0x00+(Row*8)) # Start Row
    sendCommand(0x07+(Row*8)) # End Row

def oled_putChar(C):
    C_add=ord(C)
    if C_add<32 or C_add>127:    # Ignore non-printable ASCII characters
        C=' '
        C_add=ord(C)

    for i in range(0,8,2):
        for j in range(0,8):
            c=0x00
            bit1=((BasicFont[C_add-32][i])>>j)&0x01

```

```
        bit2=((BasicFont[C_add-32][i+1])>>j)&0x01
        if bit1:
            c=c|grayH
        else:
            c=c|0x00
        if bit2:
            c=c|grayL
        else:
            c=c|0x00
        sendData(c)

def oled_putString(String):
    for i in range(len(String)):
        oled_putChar(String[i])

if __name__=="__main__":
    oled_init()
    oled_setNormalDisplay()
    oled_setTextXY(0,0)
    oled_putString("Hello")
    time.sleep(10)
    #Oled.write8(Command_Mode,0xFD)
    #sendCommand(0xFD)
    print 'hello world'
```

Step3: Save the file by clicking the disk icon with with the .py extension.

Step4: Connect Grove - OLED to Grove I2C socket on BBG.

Step5: Run the code. You'll find that the Grove - OLED outputs "Hello World".

5. Resources

[Seeed 128x64 OLED library for Arduino 1.0 and Arduino 0023](#)

[github repository for OLED Display 128*64](#)

[Grove-OLED128x64 Schematic.pdf](#)

[Resources of SSD1308 1.0.pdf](#)

[Resources of LY190-128064.pdf](#)

6. Support

If you have questions or other better design ideas, welcome to discuss with us in our [forum](#) or [wish](#) to discuss.