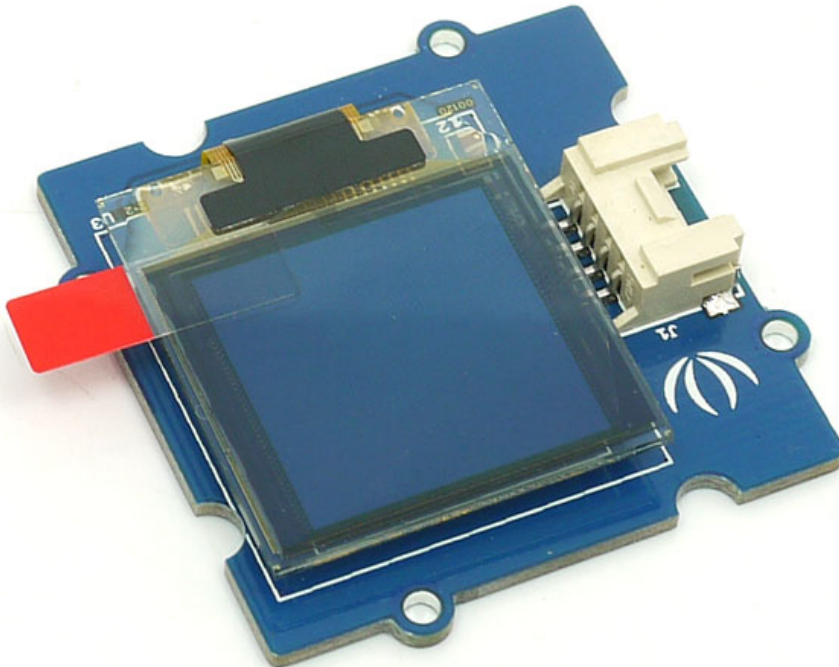


Grove - OLED Display 1.12"

Introduction



It is a 16 color grayscale 96x96 dot matrix OLED display module with Grove compatible 4pin I2C interface. This module is constructed with 96x96 dot matrix OLED module LY120 and SSD1327 driver IC. 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.

- Communicate Mode: I2C
- Grayscale Display: 16 Gray shades.
- Supports both Normal and Inverse Color Display.
- Supports Continuous Horizontal Scrolling.
- Grove compatible Interface

Specifications

Item	Value
Operating Voltage	3.3/5 V
Dot Matrix	96x96
Display Color	16 Grayscale
OLED Display	LY120-96096
Driver Chip	SSD1327Z
Dot Size	0.15(W)mm x 0.15(H)mm
Dot Pitch	0.75(W)mm x 0.175(H)mm
Operating Temperature	-40~70 oC

Tip

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

Platform Support

Arduino	Wio	BeagleBone	Raspberry Pi	LinkIt
				

Getting Started

Note

This chapter is based on Win10 and Arduino IDE 1.6.9

This is an easy-to-use module, what you need to do is connect the module to I2C port of a Base Shield. There're 4 pins, defined as below.

pin	Function	Note	Cable color
pin1	SCL	I2C Clock	YELLOW
pin2	SDA	I2C Data	WHITE
pin3	VCC	Power, 5V/3.3V	RED
pin4	GND	Ground	BLACK

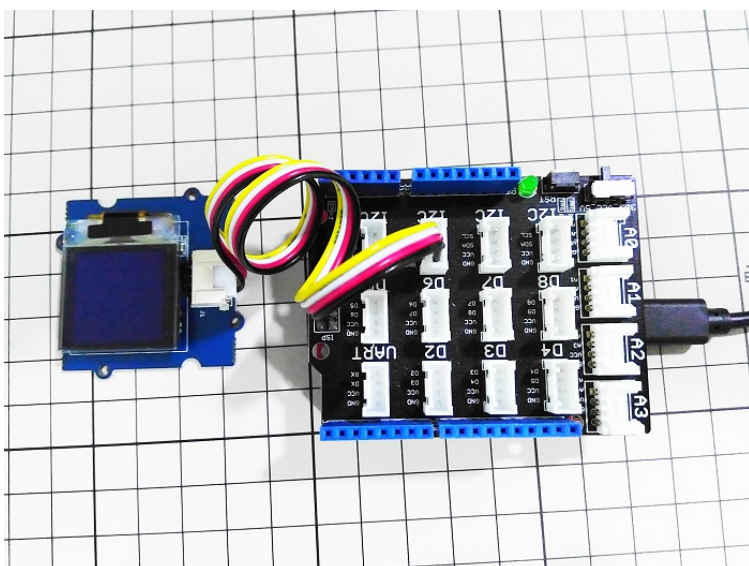
Here we will show you how this Grove - OLED Display works via a simple demo. First of all, you need to prepare the below stuffs:



Connection

Thanks to the benefit of Grove series modules, you don't need to make soldering or bread board, what you need to do is connect the modules to the right port of Base Shield. For this demo, we have only one Grove module.

- **Grove - OLED Display 1.12``** is an **I2C** module, we connect it to **I2C** port at this demo.



Download the library

We provide an Arduino Library for this Grove - OLED Display 1.12``, click on the below button to download it.

[Download Library for OLED 1.12``](#)

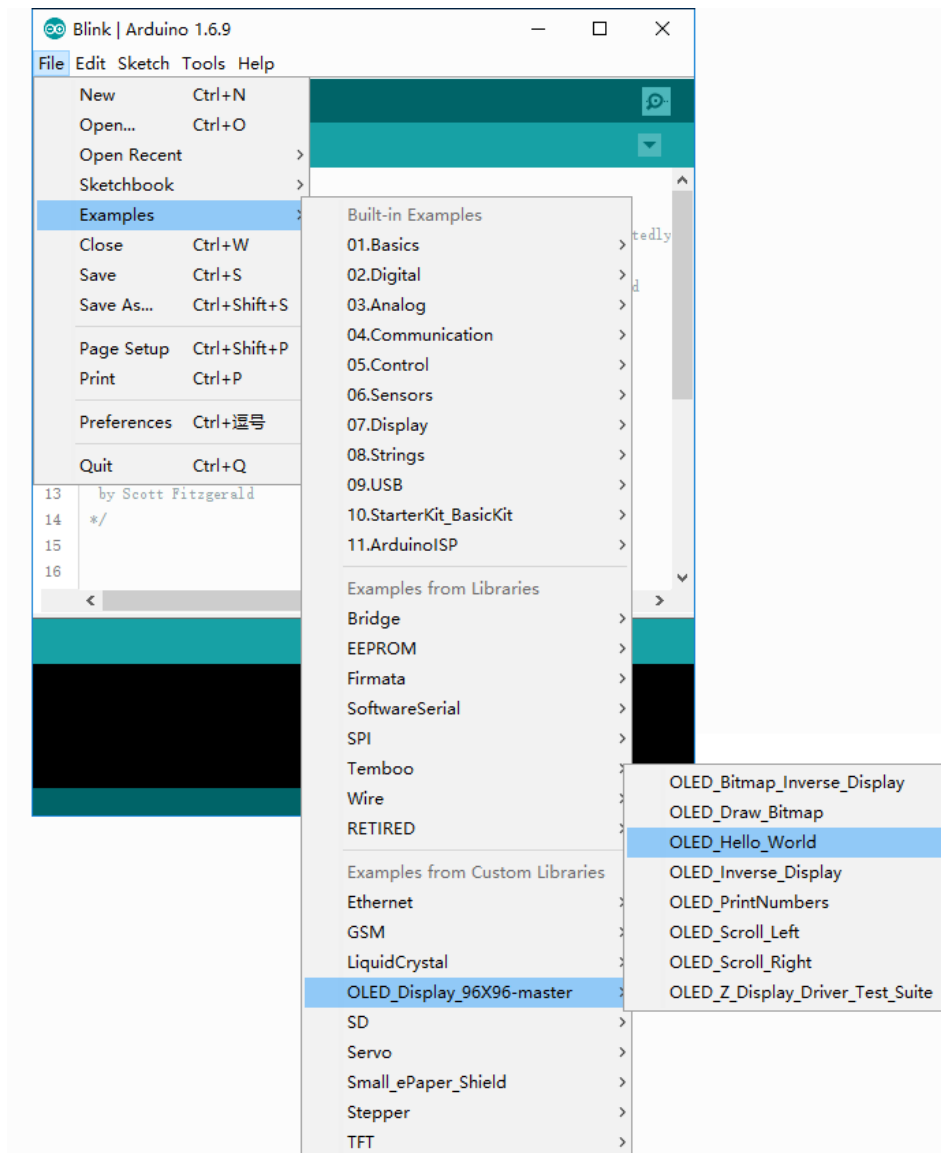
Unzip the file and put to libraries folder of your Arduino IDE. There're many examples in this library, which is consist of

- OLED_Bitmap_Inverse_Display
- OLED_Draw_Bitmap
- OLED_Hello_World
- OLED_Inverse_Display
- OLED_PrintNumbers
- OLED_Scroll_Left
- OLED_Scroll_Right
- OLED_Z_Display_Driver_Test_Suite

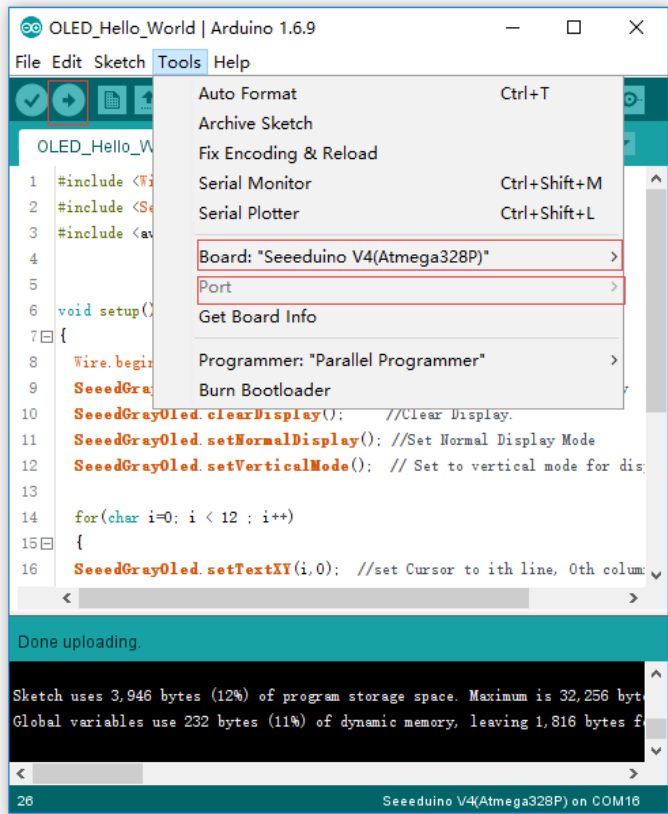
Upload example to an Arduino

Now let's try upload **OLED_Hello_World** to Seeeduino V4. Open your Arduino IDE, click on

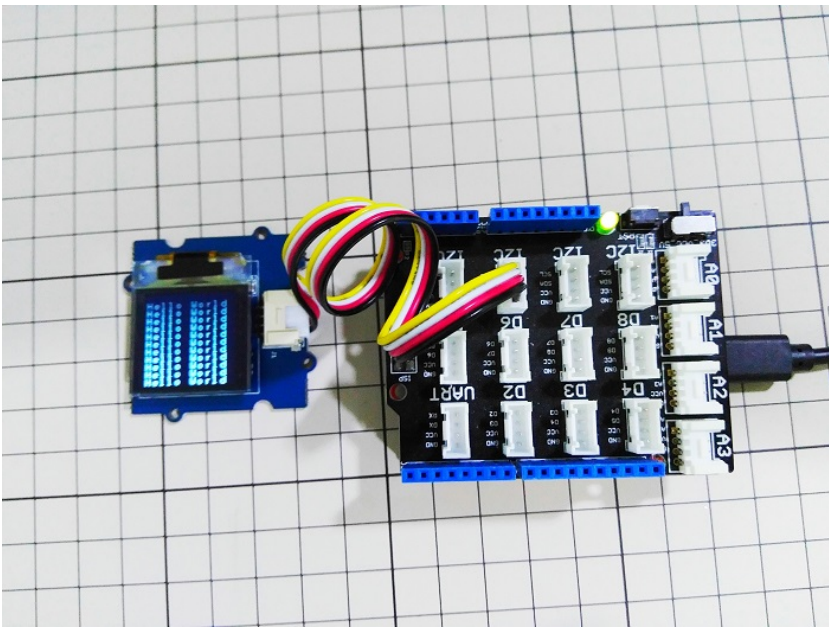
File > Example > OLED_Display_96x96-master > OLED_Hello_World



The code is open, select the right board and right COM Port, then click on Upload button which will take few seconds.



If the code is uploaded correctly, take look at your display, something was printed on it.



Then please try the other examples to see what will happen.

APIs of the library

Seeed Gray OLED library provides complete software interfaces to exercise the capabilities of SSD1327Z driver with a 96x96 gray OLED. Almost all useful features are implemented and all functions are in public scope. This makes Seeed Gray OLED Library extensible. Seeed Gray OLED library uses Arduino Wire library. Hence initialize wire library before initializing Seeed OLED library.

init()

Initializes the Seeed OLED frame and sets the display to Normal mode.

Example:

```
SeeedGrayOled.init(); //initialize SEEED Gray OLED display
```

clearDisplay()

Clears the whole screen. Should be used before starting a fresh start or after scroll deactivation. This function also sets the cursor to top left corner.

Example:

```
SeeedGrayOled.clearDisplay(); //clear the screen and set start position to top left corner
```

setNormalDisplay()

Configures the display to normal mode(non-inverse) mode.

Example:

```
SeeedGrayOled.setNormalDisplay();//Set display to normal mode (i.e non-inverse mode)
```

setContrastLevel(unsigned char ContrastLevel)

Set the contrast ratio of OLED display. ContrastLevel can be any number from 0 - 255. Example:

```
SeeedGrayOled.setContrastLevel(127); //Set display contrast ratio to half level( i.e 256/2 1 ).
```

setInverseDisplay()

Configures the display to inverse mode. Example:

```
SeedGrayOled.setInverseDisplay(); //Set display to inverse mode
```

setHorizontalMode()

Configures the display to horizontal addressing mode. Example:

```
SeedGrayOled.setHorizontalMode(); //Set addressing mode to Horizontal Mode
```

setVerticalMode()

Configures the display to vertical addressing mode. Texts are drawn in vertical mode. Please set the display to vertical mode before printing text. Example:

```
SeedGrayOled.setVerticalMode(); //Set addressing mode to Vertical Mode
```

setTextXY(X,Y)

Set the text's position (cursor) to Xth Text Row, Yth Text Column. 96x96 OLED is divided into 12 rows and 12 Columns of text. This row and column should not be confused with OLED Row and Column.

- X can be any number from 0 - 11.
- Y can be any number from 0 - 11.

Example:

```
SeedGrayOled.setTextXY(0,0); //Set the cursor to 0th Text Row, 0th Text Column
```

putChar(unsigned char c)

Print a character to OLED display starting from current address-pointer set by setTextXY(X,Y). This function is internally used by putString().

Example:

```
SeedGrayOled.putChar('S'); //Print the character S
```

putString(const char *string)

Print string to OLED display starting from current address-pointer set by setTextXY(X,Y) Example:

```
SeedGrayOled.putString("Hello World!"); //Print the String
```

putNumber(long n)

Print numbers to OLED display starting from current address-pointer set by `setTextXY(X,Y)`. Number can be any char,int or long datatype. It also takes care of -ve sign.

Example:

```
SeeedGrayOled.putNumber(-56123); //Print number -56123
```

`drawBitmap(unsigned char *bitmaparray, int bytes)`

Display a binary bitmap on the OLED matrix. The data is provided through a pointer to uni-dimensional array holding bitmap. The bitmap data is available in continuous rows of columns as like Horizontal Addressing mode. bytes is size of bitmap in bytes.

Example:

```
SeeedGrayOled.drawBitmap(SeeedLogo,96*96/8); // Draw binary Bitmap (96 pixels *96 pixels / 8)
```

bytes

`setHorizontalScrollProperties`

Set the properties of horizontal scroll.

- Direction can be any of `Scroll_Left` and `Scroll_Right`.
- `startRow` can be 0 - 127
- `endRow` can be 0 - 127. It should be greater than `startRow`
- `startColumn` can be 0 - 63
- `endColumn` can be 0 - 63. It should be greater than `startRow`
- `scrollSpeed` can be any of defines:`Scroll_2Frames`, `Scroll_3Frames`, `Scroll_4Frames`, `Scroll_5Frames`, `Scroll_25Frames`,`Scroll_64Frames`, `Scroll_128Frames`,`Scroll_256Frames`.

Example:

```
SeeedGrayOled.setHorizontalScrollProperties(Scroll_Left,72,95,0,47,Scroll_5Frames); //Set the
```

properties of Horizontal Scroll

`activateScroll()`

Enable scrolling. This should be used only after setting horizontal scroll properties.

Example:

```
SeedGrayOled.activateScroll(); //Enable scrolling.
```

deactivateScroll()

Disable scrolling. This should be used after activateScroll(); Example:

```
SeedGrayOled.activateScroll(); //Disable scrolling.
```