0.96” OLED Display Module

BMD31M090
Arduino Library V1.0.1 Description

Revision: V1.10 Date: December 26, 2023

www.bestmodulescorp.com

Contents

Introduction

The Best Modules BMD31M090 is a 0.96” OLED display module, which uses the I²C communication mode. This document provides the description of the BMD31M090 Arduino Lib functions and how to install the Arduino Lib. This example demonstrates the display function of OLED.

Arduino Library Functions

Arduino Lib Name: BMD31M090		Lib Version: V1.0.1
Constructors & Initialisation		
1	BMD31M090(uint8_t width, uint8_t height, TwoWire *theWire=&Wire)	
	Description	Constructor, uses Display width,height and wire interface
	Parameter	width: pixel display width height: pixel display height * theWire: wire parameter
	Return Value	—
	Note	—
2	void begin(uint8_t i2c_addr=BMD31M090_DEVICEADDR0, uint32_t clkFrequency=BMD31M090_CLKFREQ)	
	Description	Module initialisation, I²C device address and I²C communication rate configuration
	Parameter	i2c_addr: I²C device address, default: 0x3C clkFrequency: I²C communication rate, default: 400kHz
	Return Value	void
	Note	—
Performance Functions		
3	void clearDisplay(void)	
	Description	Clear the display buffer contents (Set all pixels to off)
	Parameter	void
	Return Value	void
	Note	Use with display()
4	void display(void)	
	Description	Display current buffer contents
	Parameter	void
	Return Value	void
	Note	—
5	void drawPixel(uint8_t x, uint8_t y, uint8_t pixelColor)	
	Description	Draw a pixel
	Parameter	x: x-coordinate, range: 0~127 y: y-coordinate, range: 0~63 pixelColor: pixel color 0 (pixelColor_BLACK): black pixel 1 (pixelColor_WHITE): white pixel 2 (pixelColor_INVERSE): inverse pixel color
	Return Value	void
	Note	Use with display()

6	void drawLine(uint8_t x_Start, uint8_t y_Start, uint8_t x_End, uint8_t y_End, uint8_t pixelColor)	
	Description	Draw a line
	Parameter	x_Start: x-coordinate of the start of the line y_Start: y-coordinate of the start of the line x_End: x-coordinate of the end of the line y_End: y-coordinate of the end of the line pixelColor: pixel color 0 (pixelColor_BLACK): black pixel 1 (pixelColor_WHITE): white pixel 2 (pixelColor_INVERSE): inverse pixel color
	Return Value	void
	Note	Use with display()
7	void drawFastHLine(uint8_t x, uint8_t y, uint8_t width, uint8_t pixelColor)	
	Description	Draw a horizontal line
	Parameter	x: x-coordinate, range: 0~127 y: y-coordinate, range: 0~63 width: line width, width(w)≤128 pixelColor: pixel color 0 (pixelColor_BLACK): black pixel 1 (pixelColor_WHITE): white pixel 2 (pixelColor_INVERSE): inverse pixel color
	Return Value	void
	Note	Use with display()
8	void drawFastVLine(uint8_t x, uint8_t y, uint8_t height, uint8_t pixelColor)	
	Description	Draw a vertical line
	Parameter	x: x-coordinate, range: 0~127 y: y-coordinate, range: 0~63 height: line height, height (h)≤64 pixelColor: pixel color 0 (pixelColor_BLACK): black pixel 1 (pixelColor_WHITE): white pixel 2 (pixelColor_INVERSE): inverse pixel color
	Return Value	void
	Note	Use with display()
9	void drawChar(uint8_t x, uint8_t row, uint8_t chr)	
	Description	Write characters
	Parameter	x: x-coordinate, range: 0~127 row: character row, range: 0~7 0 (displayROW0): Row 0 1 (displayROW1): Row 1 2 (displayROW2): Row 2 3 (displayROW3): Row 3 4 (displayROW4): Row 4 5 (displayROW5): Row 5 6 (displayROW6): Row 6 7 (displayROW7): Row 7 chr: characters on the font table
	Return Value	void
	Note	—

10	void drawString(uint8_t x, uint8_t row, uint8_t *str)	
	Description	Write strings
	Parameter	x: x-coordinate, range: 0~127 row: string row, range: 0~7 0 (displayROW0): Row 0 1 (displayROW1): Row 1 2 (displayROW2): Row 2 3 (displayROW3): Row 3 4 (displayROW4): Row 4 5 (displayROW5): Row 5 6 (displayROW6): Row 6 7 (displayROW7): Row 7 *str: strings on the font table
	Return Value	void
	Note	—
11	void drawNum(uint8_t x, uint8_t row, uint32_t num, uint8_t numLen)	
	Description	Write integer numbers
	Parameter	x: x-coordinate, range: 0~127 row: row, range: 0~7 0 (displayROW0): Row 0 1 (displayROW1): Row 1 2 (displayROW2): Row 2 3 (displayROW3): Row 3 4 (displayROW4): Row 4 5 (displayROW5): Row 5 6 (displayROW6): Row 6 7 (displayROW7): Row 7 Num: number on the font table numLen: length of number
	Return Value	void
	Note	—
12	void drawBitmap(int8_t x, int8_t y, const uint8_t *Bitmap, uint8_t w, uint8_t h, uint8_t pixelColor)	
	Description	Draw bitmaps
	Parameter	x: x-coordinate, range: 0~127 y: y-coordinate, range: 0~63 *Bitmap: bitmap name w: bitmap width h: bitmap height pixelColor: pixel color 0 (pixelColor_BLACK): black pixel 1 (pixelColor_WHITE): white pixel 2 (pixelColor_INVERSE): inverse pixel color
	Return Value	void
	Note	Use with display()

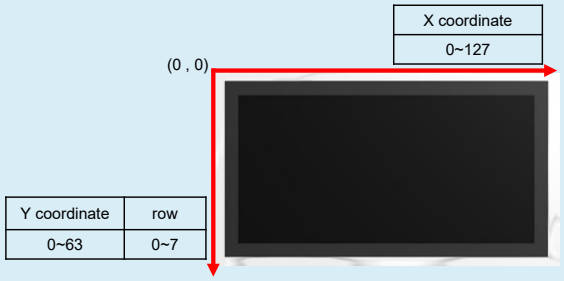
13	void startScrollRight(uint8_t startRow, uint8_t endRow, uint8_t scrollSpeed, uint8_t scrollVDirection= SCROLLV_NONE)	
	Description	Start scrolling to the right
	Parameter	startRow: start row address, range: 0~7 endRow: end row address, range: 0~7 scrollSpeed: scroll speed 0x07 (SCROLL_2FRAMES): Scroll 2 frames 0x04 (SCROLL_3FRAMES): Scroll 3 frames 0x05 (SCROLL_4FRAMES): Scroll 4 frames 0x00 (SCROLL_5FRAMES): Scroll 5 frames 0x06 (SCROLL_25FRAMES): Scroll 25 frames 0x01 (SCROLL_64FRAMES): Scroll 64 frames 0x02 (SCROLL_128FRAMES): Scroll 128 frames 0x03 (SCROLL_256FRAMES): Scroll 256 frames scrollVDirection: scroll diagonally 0x00 (SCROLLV_NONE): Non-scrolling 0x01 (SCROLLV_TOP): Scroll up 0x3F (SCROLLV_BOTTOM): Scroll down
	Return Value	void
	Note	—
14	void startScrollLeft(uint8_t startRow, uint8_t endRow, uint8_t scrollSpeed, uint8_t scrollVDirection= SCROLLV_NONE)	
	Description	Start scrolling to the left
	Parameter	startRow: start row address, range: 0~7 endRow: end row address, range: 0~7 scrollSpeed: scroll speed 0x07 (SCROLL_2FRAMES): Scroll 2 frames 0x04 (SCROLL_3FRAMES): Scroll 3 frames 0x05 (SCROLL_4FRAMES): Scroll 4 frames 0x00 (SCROLL_5FRAMES): Scroll 5 frames 0x06 (SCROLL_25FRAMES): Scroll 25 frames 0x01 (SCROLL_64FRAMES): Scroll 64 frames 0x02 (SCROLL_128FRAMES): Scroll 128 frames 0x03 (SCROLL_256FRAMES): Scroll 256 frames scrollVDirection: diagonal scrolling direction 0x00 (SCROLLV_NONE): Non-scrolling 0x01 (SCROLLV_TOP): Scroll up 0x3F (SCROLLV_BOTTOM): Scroll down
	Return Value	void
	Note	—
15	void stopScroll(void)	
	Description	Stop scrolling
	Parameter	void
	Return Value	void
	Note	—
Parameter Configuration		
16	void setFont(const unsigned char* font)	
	Description	Set font format
	Parameter	Font: font format FontTable_6X8: Font format is 6×8 FontTable_8X16: Font format is 8×16 FontTable_16X32: Font format is 16×32 FontTable_32X64: Font format is 32×64
	Return Value	void
	Note	—

17	void setPixelRow(uint8_t x, uint8_t row)	
	Description	Set pixel row
	Parameter	x: x-coordinate, range: 0~127 row: row, range: 0~7
	Return Value	void
	Note	—
18	void dim(bool dim)	
	Description	Set screen brightness
	Parameter	dim: brightness selection true: Dim false: Normal brightness
	Return Value	void
	Note	—
19	void invertDisplay(bool i)	
	Description	Invert display
	Parameter	I: invert or not true: black-on-white false: white-on-black
	Return Value	void
	Note	—

Note: OLED display, the vertical direction adopts different expressions according to different functions

(1) Function#5, #6, #7, #8, #12, the vertical direction is “Y coordinate”, value ranges from 0 to 63

(2) Function #9, #10, #11, #13, #14, #17, the vertical direction is “row”, value ranges from 0 to 7

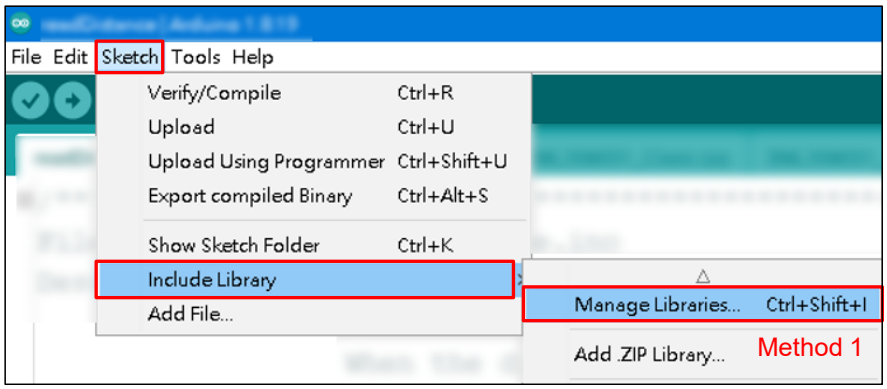


Arduino Lib Download and Installation

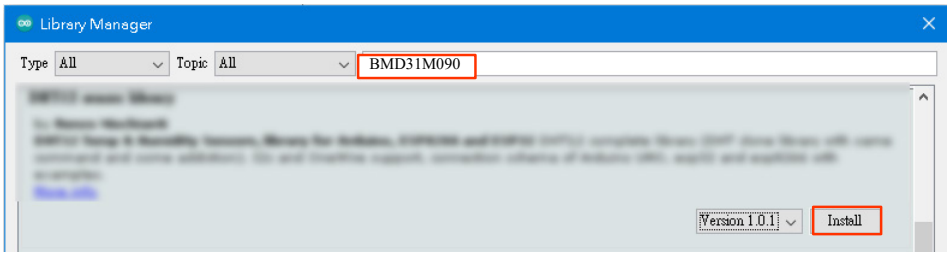
BMD31M090 Library: Refer to the following two methods to install the BMD31M090 Arduino Library.

Method 1: Search for installation

Search for installation: Arduino IDE → Sketch → Include Library → Manage Libraries...→ Search BMD31M090 → Install



Search for Installation Step 1

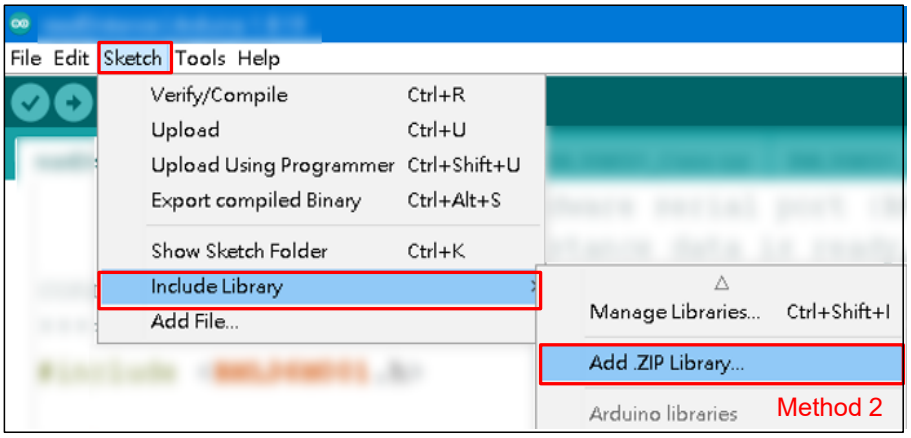


Search for Installation Step 2

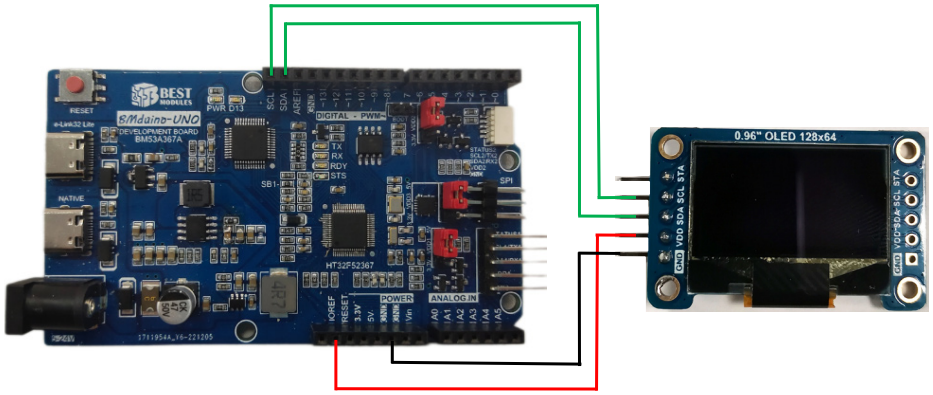
Method 2: Download the .ZIP library before adding it

Download the Arduino example (BMD31M090 Library) under the DOCUMENTS menu from the Best Modules website (https://www.bestmodulescorp.com/ bmd31m090.html)

Add .ZIP library: Arduino IDE → Sketch → Include Library → Add .ZIP Library...



Arduino Example



Physical Connection Diagram

Example: display

Function: The functions provided in the BMD31M090 will be used and displayed on the OLED module.

1. Open the example: File → Examples → Select Lib (BMD31M090) → Select example
2. Example Description:

a. Create object & initialise object

```
#include "BMD31M090.h"
#include "Bitmap.h"
#define BMD31M090_WIDTH   128          // Set the display width of
                                       // the module in pixels
#define BMD31M090_HEIGHT  64           // Set the display height of
                                       // the module in pixels
#define BMD31M090_ADDRESS 0x3C         // Set the I²C address of
                                       // the module
uint8_t t = ' ';
BMD31M090  BMD31(BMD31M090_WIDTH, BMD31M090_HEIGHT, &Wire);
                                       // Create object
void setup()
{
  Serial.begin(115200);                // Configure the serial monitor
  Serial.println("BMD31M090 0.96\" OLED Module Sketch");
  BMD31.begin(BMD31M090_ADDRESS);      // Module initialisation
  delay(100);                 // Delay 100ms after module initialisation
  test_drawString_6x8();      // Test function: with the drawing font 6×8,
                              // draw the string "Hello world!"
  test_drawString_8x16();     // Test function: with the drawing font 8×16,
                              // draw the string "Hello world!"
  test_drawString_drawChar_drawNum();// Test function: drawString,
                                     // drawChar and drawNum
  test_drawPixel();                  // Test function: drawPixel
  test_drawFastHLine_drawFastVLine();// Test function: drawFastHLine
                                     // and drawFastVLine
 test_drawBitmap();           // Test function: draw a bitmap to display
                              // the BestModule_LOGO and
                              // the NameBestModule_LOGO
```

```
  test_variousScroll();  // Test function: scroll function can scroll
                         // in various directions
  test_invertDisplay();  // Test function: invert display
                         // and restore display
  test_dim();            // Test function: dimming display
}
```

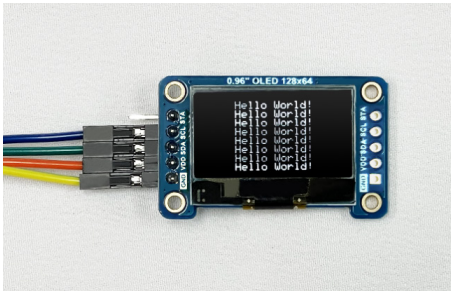
b. loop

```
void loop()
{

}
```

c. test_drawString_6x8()

```
void test_drawString_6x8(void)
{
  BMD31.clearDisplay();
  BMD31.display();
  uint8_t col, row;
  BMD31.setFont(FontTable_6X8); // Set the font format to 6×8
                                //(Default font format: 8×16)
  col = (128 - (6 * sizeof(“Hello World!”)))/2;
  for (row=0; row<8; row++)
  {
    BMD31.drawString(col, row, (u8*)“Hello World!”);
  }
  delay(500);
}
```

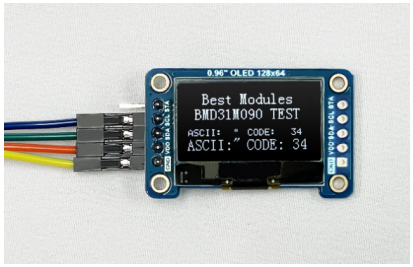


d. test_drawString_8x16()

```
void test_drawString_8x16(void)
{
  BMD31.clearDisplay();
  BMD31.display();
  uint8_t col, row;
  BMD31.setFont(FontTable_8X16);// Set the font format to 8×16
  col = (128 - (8 * sizeof(“Hello World!”)))/2;
  for (row=0; row<8; row+=2)
  {
    BMD31.drawString(col, row, (u8*)“Hello World!”);
  }
  delay(500);
}
```

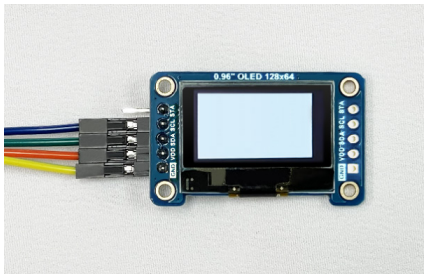
e. test_drawString_drawChar_drawNum()

```
void test_drawString_drawChar_drawNum(void)
{
  BMD31.clearDisplay();
  BMD31.display();
  BMD31.drawString(16, displayROW0, (u8*)“Best Modules”);
  BMD31.drawString(8,  displayROW2, (u8*)“BMD31M090 TEST”);
// Displays the ‘ ’ (0x20/32) to ‘~’ (0x7E/126) ASCII values
// for words 6×8 and 8×16 in sequence
  do
  {
    BMD31.setFont(FontTable_6X8);
    BMD31.drawString(0, displayROW5, (u8*)“ASCII:”);
    BMD31.drawString(63, displayROW5, (u8*)“CODE:”);
    BMD31.drawChar(48, displayROW5, t);
    BMD31.drawNum(103, displayROW5, t, 3);
    BMD31.setFont(FontTable_8X16);
    BMD31.drawString(0, displayROW6, (u8*)“ASCII:”);
    BMD31.drawString(63, displayROW6, (u8*)“CODE:”);
    BMD31.drawChar(48, displayROW6, t);
    BMD31.drawNum(103, displayROW6, t, 3);
    delay(10);
  } while(++t < ‘~’);
  t=‘ ’;
}
```



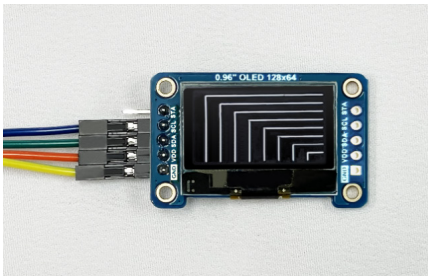
f. test_drawPixel()

```
void test_drawPixel(void)
{
  BMD31.clearDisplay();
  for (uint8_t col=0; col<128; col++)
  {
    for(uint8_t row=0; row<64; row++)
    {
      BMD31.drawPixel(col, row, pixelColor_WHITE);
    }
  }
  BMD31.display();
  delay(500);
  for (uint8_t col=0; col<128; col++)
  {
    for(uint8_t row=0; row<32; row++)
    {
      BMD31.drawPixel(col, row, pixelColor_BLACK);
    }
  }
  BMD31.display();
  delay(500);
  for (uint8_t col=0; col<128; col++)
  {
    for(uint8_t row=0; row<64; row++)
    {
      BMD31.drawPixel(col, row, pixelColor_INVERSE);
    }
  }
  BMD31.display();
  delay(500);
}
```



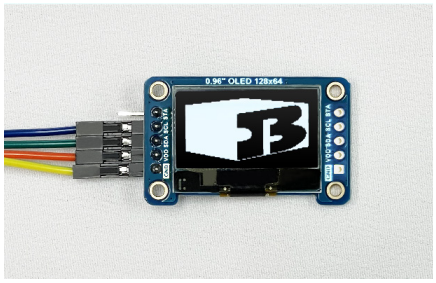
g. test_drawFastHLine_drawFastVLine()

```
void test_drawFastHLine_drawFastVLine(void)
{
  int8_t col, row;
  BMD31.clearDisplay();
  col = 112;
  for (row=0; row<64; row+=8)
  {
    BMD31.drawFastHLine(0,   row, col,pixelColor_WHITE);
    BMD31.drawFastVLine(col, row, 64 - row, pixelColor_WHITE);
    col -= 14;
  }
  BMD31.display();
  delay(500);
  BMD31.clearDisplay();
  col = 112;
  for (row=56; row>=0; row-=8)
  {
    BMD31.drawFastHLine(col, row, 128 - col, pixelColor_WHITE);
    BMD31.drawFastVLine(col, row, 64 - row,  pixelColor_WHITE);
    col -= 14;
  }
  BMD31.display();
  delay(500);
}
```

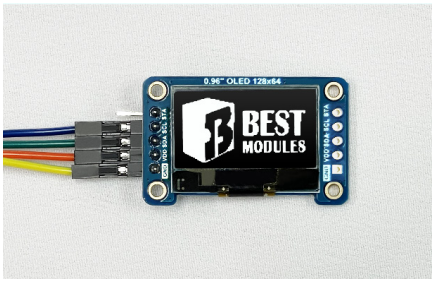


h. test_drawBitmap()

```
void test_drawBitmap(void)
{
  BMD31.clearDisplay();
  BMD31.drawBitmap(0, 0, BestModule_LOGO, 128, 64, pixelColor_WHITE);
  BMD31.display();
  delay(300);
  BMD31.drawBitmap(0, 0, BestModule_LOGO, 128, 64, pixelColor_BLACK);
  BMD31.display();
  delay(300);
  BMD31.drawBitmap(0, 0, BestModule_LOGO, 128, 64,
                   pixelColor_INVERSE);
  BMD31.display();
  delay(300);
  BMD31.clearDisplay();
  BMD31.drawBitmap(0, 0, BestModule_LOGOandName, 128, 64,
                   pixelColor_WHITE);
  BMD31.display();
  delay(300);
  BMD31.drawBitmap(0, 0, BestModule_LOGOandName, 128, 64,
                   pixelColor_BLACK);
  BMD31.display();
  delay(300);
  BMD31.drawBitmap(0, 0, BestModule_LOGOandName, 128, 64,
                   pixelColor_INVERSE);
  BMD31.display();
  delay(300);
}
```



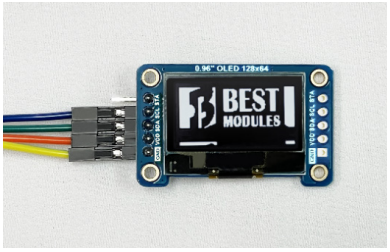
BestModule_LOGO



BestModule_LOGOandName

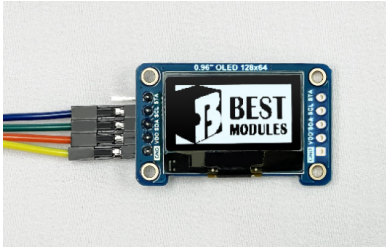
i. test_variousScroll()

```
void test_variousScroll(void)
{
// Scrolling  function test
  uint8_t startRow = displayROW0;
  uint8_t endRow = displayROW7;
  BMD31.startScrollRight(startRow, endRow, SCROLL_2FRAMES);
  delay(500);
  BMD31.startScrollRight(startRow, endRow, SCROLL_2FRAMES,
                         SCROLLV_TOP);
  delay(500);
  BMD31.startScrollRight(startRow, endRow, SCROLL_2FRAMES,
                         SCROLLV_BOTTOM);
  delay(500);
  BMD31.startScrollLeft(startRow, endRow, SCROLL_2FRAMES);
  delay(500);
  BMD31.startScrollLeft(startRow, endRow, SCROLL_2FRAMES,
                        SCROLLV_TOP);
  delay(500);
  BMD31.startScrollLeft(startRow, endRow, SCROLL_2FRAMES,
                        SCROLLV_BOTTOM);
  delay(500);
  BMD31.stopScroll();
}
```

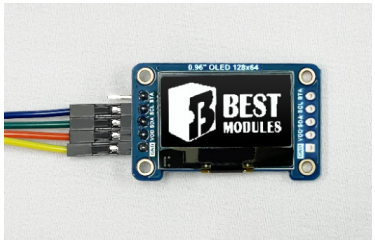


j. test_invertDisplay()

```
void test_invertDisplay(void)
{
  BMD31.invertDisplay(TRUE);  // Invert Display Mode: black-on-white
  delay(500);
  BMD31.invertDisplay(FALSE); // Normal Display Mode: white-on-black
  delay(500);
  BMD31.invertDisplay(TRUE);  // Invert Display Mode: black-on-white
  delay(500);
  BMD31.invertDisplay(FALSE); // Normal Display Mode: white-on-black
  delay(500);
}
```



BMD31.invertDisplay(TRUE)-black-on-white



BMD31.invertDisplay(FALSE)-white-on-black

k. test_dim()

```
void test_dim(void)
{
  BMD31.dim(TRUE);  // Dim Mode: contrast - 0x00
  delay(500);
  BMD31.dim(FALSE); // Normal Brightness Mode: contrast - 0xCF
  delay(500);
  BMD31.dim(TRUE);  // Dim Mode: contrast - 0x00
  delay(500);
  BMD31.dim(FALSE); // Normal Brightness Mode: contrast - 0xCF
  delay(500);
}
```

3. Open the serial monitor and set the baud rate to be 115200. The serial monitor will display sensor as follows: