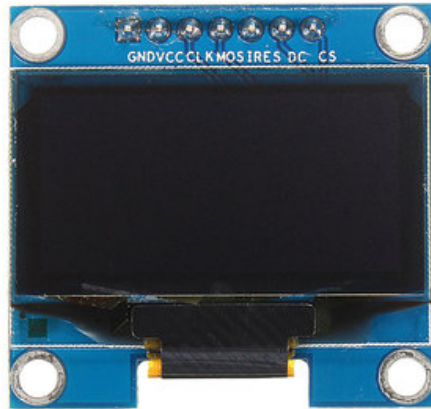


1.3Inch OLED Display Module for Arduino(TF051)



Specification:

- Needn't back light for the display unit can self-luminous
- High resolution:128*64
- Viewing angle:>160
- OLED screens inside the driver chips : SSH1106
- Supports many control chip : fully compatible with Arduino 51 series,msp430 series , stm32/2 CSR IC , ETC
- Ultra-low power consumption:full screen lit 0.08w
- Voltage: DC 3V-5V
- Work temp : -30°C-70°C
- SPI interface
- The color of word : blue

Pin Description

- VDD: 2.8-5.5V power supply
- SCK: CLK clock
- SDA: MOSI data
- RES: Reset
- DC: data / command
- CS: chip-select signal
- GND: Power ground

Example

Hardware connection

Arduino	OLED
5V	=====VCC
GND	=====GND
Digital 4	=====D0 (SCL)
Digital 5	=====D1 (SCK)
Digital 6	=====CS
Digital 7	=====DC

*****Code Begin*****

```
#include "U8glib.h"
```

```
U8GLIB_SH1106_128X64 u8g(4, 5, 6, 7); // SW SPI Com: SCK = 4, MOSI = 5, CS = 6, A0 = 7  
(new blue HalTec OLED)
```

```
void u8g_prepare(void) {  
    u8g.setFont(u8g_font_6x10);  
    u8g.setFontRefHeightExtendedText();  
    u8g.setDefaultForegroundColor();  
    u8g.setFontPosTop();  
}
```

```
void u8g_box_frame(uint8_t a) {  
    u8g.drawStr( 0, 0, "drawBox");  
    u8g.drawBox(5,10,20,10);  
    u8g.drawBox(10+a,15,30,7);  
    u8g.drawStr( 0, 30, "drawFrame");  
}
```

```

    u8g.drawFrame(5,10+30,20,10);
    u8g.drawFrame(10+a,15+30,30,7);
}

void u8g_disc_circle(uint8_t a) {
    u8g.drawStr( 0, 0, "drawDisc");
    u8g.drawDisc(10,18,9);
    u8g.drawDisc(24+a,16,7);
    u8g.drawStr( 0, 30, "drawCircle");
    u8g.drawCircle(10,18+30,9);
    u8g.drawCircle(24+a,16+30,7);
}

void u8g_r_frame(uint8_t a) {
    u8g.drawStr( 0, 0, "drawRFrame/Box");
    u8g.drawRFrame(5, 10,40,30, a+1);
    u8g.drawRBox(50, 10,25,40, a+1);
}

void u8g_string(uint8_t a) {
    u8g.drawStr(30+a,31, " 0");
    u8g.drawStr90(30,31+a, " 90");
    u8g.drawStr180(30-a,31, " 180");
    u8g.drawStr270(30,31-a, " 270");
}

void u8g_line(uint8_t a) {
    u8g.drawStr( 0, 0, "drawLine");
    u8g.drawLine(7+a, 10, 40, 55);
    u8g.drawLine(7+a*2, 10, 60, 55);
    u8g.drawLine(7+a*3, 10, 80, 55);
    u8g.drawLine(7+a*4, 10, 100, 55);
}

void u8g_triangle(uint8_t a) {
    uint16_t offset = a;
    u8g.drawStr( 0, 0, "drawTriangle");
    u8g.drawTriangle(14,7, 45,30, 10,40);
    u8g.drawTriangle(14+offset,7-offset, 45+offset,30-offset, 57+offset,10-offset);
    u8g.drawTriangle(57+offset*2,10, 45+offset*2,30, 86+offset*2,53);
    u8g.drawTriangle(10+offset,40+offset, 45+offset,30+offset, 86+offset,53+offset);
}

```

```
}
```

```
void u8g_ascii_1() {  
    char s[2] = " ";  
    uint8_t x, y;  
    u8g.drawStr( 0, 0, "ASCII page 1");  
    for( y = 0; y < 6; y++ ) {  
        for( x = 0; x < 16; x++ ) {  
            s[0] = y*16 + x + 32;  
            u8g.drawStr(x*7, y*10+10, s);  
        }  
    }  
}
```

```
void u8g_ascii_2() {  
    char s[2] = " ";  
    uint8_t x, y;  
    u8g.drawStr( 0, 0, "ASCII page 2");  
    for( y = 0; y < 6; y++ ) {  
        for( x = 0; x < 16; x++ ) {  
            s[0] = y*16 + x + 160;  
            u8g.drawStr(x*7, y*10+10, s);  
        }  
    }  
}
```

```
void u8g_extra_page(uint8_t a)  
{  
    if ( u8g.getMode() == U8G_MODE_HICOLOR || u8g.getMode() == U8G_MODE_R3G3B2) {  
        /* draw background (area is 128x128) */  
        u8g_uint_t r, g, b;  
        b = a << 5;  
        for( g = 0; g < 64; g++ )  
        {  
            for( r = 0; r < 64; r++ )  
            {  
                u8g.setRGB(r<<2, g<<2, b );  
                u8g.drawPixel(g, r);  
            }  
        }  
        u8g.setRGB(255,255,255);  
    }  
}
```

```

    u8g.drawStr( 66, 0, "Color Page");
}
else if ( u8g.getMode() == U8G_MODE_GRAY2BIT )
{
    u8g.drawStr( 66, 0, "Gray Level");
    u8g.setColorIndex(1);
    u8g.drawBox(0, 4, 64, 32);
    u8g.drawBox(70, 20, 4, 12);
    u8g.setColorIndex(2);
    u8g.drawBox(0+1*a, 4+1*a, 64-2*a, 32-2*a);
    u8g.drawBox(74, 20, 4, 12);
    u8g.setColorIndex(3);
    u8g.drawBox(0+2*a, 4+2*a, 64-4*a, 32-4*a);
    u8g.drawBox(78, 20, 4, 12);
}
else
{
    u8g.drawStr( 0, 12, "setScale2x2");
    u8g.setScale2x2();
    u8g.drawStr( 0, 6+a, "setScale2x2");
    u8g.undoScale();
}
}

```

```
uint8_t draw_state = 0;
```

```

void draw(void) {
    u8g_prepare();
    switch(draw_state >> 3) {
        case 0: u8g_box_frame(draw_state&7); break;
        case 1: u8g_disc_circle(draw_state&7); break;
        case 2: u8g_r_frame(draw_state&7); break;
        case 3: u8g_string(draw_state&7); break;
        case 4: u8g_line(draw_state&7); break;
        case 5: u8g_triangle(draw_state&7); break;
        case 6: u8g_ascii_1(); break;
        case 7: u8g_ascii_2(); break;
        case 8: u8g_extra_page(draw_state&7); break;
    }
}

```

```
void setup(void) {

    // flip screen, if required
    //u8g.setRot180();

    pinMode(13, OUTPUT);
    digitalWrite(13, HIGH);
}

void loop(void) {

    // picture loop
    u8g.firstPage();
    do {
        draw();
    } while( u8g.nextPage() );

    // increase the state
    draw_state++;
    if ( draw_state >= 9*8 )
        draw_state = 0;

    // rebuild the picture after some delay
    //delay(150);

}
*****Code End*****
```