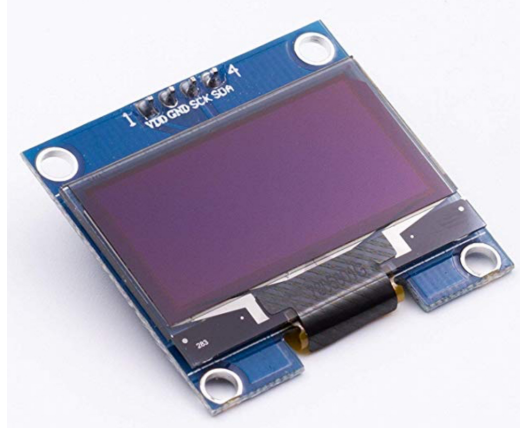


10 Application - OLED

Up to now we have used the simple 16 character by 2 line LCD display. But we have used it with a serial I2C bus. Now we will look at how to use a more sophisticated display, an OLED. Since the OLED works at the pixel level, not the character level, it can display graphics and text in various fonts.



Displays come in a huge number of varieties. Pixel sizes/resolution, colour/B&W, type of controller IC, etc etc. The one we will use is a 128 x 64 pixel B&W display with a SH1106 controller.

Library

There are two popular libraries for displays. One from the company Adafruit, another which is open source software. I prefer to use open source as it is not tied into any specific hardware on sale by the supplier.

The library is called U8g2lib.h (what a strange name!). It is specifically for monochrome displays and supports more than 20 different controllers and many sizes of display. The manual for U8g2lib is on line (github.com/olikraus/u8g2/wiki) and gives all the primitives that you can use to display text and graphics. It also lists all the fonts available.

The constructor we need to create an object for our OLED display is,

```
U8G2_SH1106_128X64_NONAME_1_HW_I2C oled(U8G2_R0);  
// create the right object "oled"  
// for our display type, SH1106 controller, 128x64 pixels, I2C interface
```

After this we can use many functions from the library, such as

```
oled.setFont(u8g2_font_5x8_tf);           // set font  
oled.setFontPosTop();                    // set x, y reference  
oled.setCursor(x, y);                    // position  
oled.print(m);                            // print the message
```

Example

Here is some simple code that illustrates how to use the library,

```
#include "U8g2lib.h"           // display library
#include "Wire.h"             // I2C library

U8G2_SH1106_128X64_NONAME_1_HW_I2C oled(U8G2_R0);
// create the right object "oled"
// for our display type, SH1106 controller, 128x64 pixels, I2C interface

void setup() {
  oled.begin();               // initialise the display
  dispMessage();             // update
}

void loop() {
  // nothing to do here
}

void dispMessage() {         // picture loop
  oled.firstPage();
  do {
    oled.setFont(u8g2_font_5x8_tf); // set font
    oled.setFontPosTop();          // set x, y reference
    oled.setCursor(0, 10);        // position
    oled.print("HELLO WORLD");    // print the message
  } while ( oled.nextPage() );
}
```

Let's run through it to understand what it does.

1. At the top we include the U8g2lib.h library and the Wire.h I2C library in our sketch
2. Next we create an object "oled" using the right constructor from the library
3. In setup() we initialise the display by calling a library function "begin()"
4. We then call our own function to display a message on the screen "dispMessage()"

Let's look at dispMessage() as it is crucial to using the library.

The library can handle more than one page of data stored in memory on the Arduino. We will use only one page. This is the code

```
void dispMessage() {                                // picture loop
  oled.firstPage();
  do {
    oled.setFont(u8g2_font_5x8_tf);                // set font
    oled.setFontPosTop();                          // set x, y reference
    oled.setCursor(0, 10);                         // position
    oled.print("HELLO WORLD");                     // print the message
  } while ( oled.nextPage() );
}
```

Explanation,

1. "oled.firstPage()" obviously selects the first page in memory to display
2. Then we have a "do" loop, which runs a series of instructions, and tests if there is a "next" page at the bottom (there isn't so it stops here)
3. Inside the "do" loop are the actual functions to display some text on the screen: set the font to use, set the reference position of x, y for the font (top, left), set the cursor position to x, y (0, 10) where the message will be displayed, and finally print the message "HELLO WORLD".

The oled.h header

Chunks of code like

```
oled.setFont(u8g2_font_5x8_tf);                    // set font
oled.setFontPosTop();                              // set x, y reference
oled.setCursor(0, 10);                             // position
oled.print("HELLO WORLD");                          // print the message
```

often repeat, with "HELLO WORLD" replaced by other x, y & messages from your application. To handle this I have created a header file with a number of useful functions. The header file also replaces the need to enter the initialisation code, #include "U8g2lib.h" etc and it creates the "oled" object for you.

The contents of oled.h are below, study them to see the functions it provides. These functions are designed for radio applications, but other can easily be added, for example for word processing or games. To use it simply put at the top of your sketch,

```
#include "Oled.h"                                  // include header
```

```

// Oled.h
// 21-2-19
// defines oled pins, creates "oled" object for 128x64 SH1106 display
// fonts github.com/olikraus/u8g2/wiki

// HEADERS & LIBRARIES
#include "U8g2lib.h"
#include "Wire.h"

// OBJECTS
// oled obj, SH1106 controller, 128X64, one line cache, HW I2C and normal
orientation R0
U8G2_SH1106_128X64_NONAME_1_HW_I2C oled(U8G2_R0);

//FUNCTIONS

// BAR
// display bar at x, y, height, length (0-100 pixels)
void dispBar(u8g2_uint_t x, u8g2_uint_t y, byte h, byte l) {
    byte n;

    oled.drawFrame(x, y, 100, h+1);
    for (n = 0; n < l; n++) {
        oled.drawLine(x + n, y, x + n, y + h);
    }
}

// FREQ
// display freq at x, y, of f (Hz) plus cf (cHz), to decimal places (max 3)
void dispFreq(u8g2_uint_t x, u8g2_uint_t y, double f, double cf, byte d) {
    oled.setFont(u8g2_font_10x20_tf); // font
    oled.setFontPosTop(); // origin top

    f = f / 1000.0;
    cf = cf / 100000.0;

    oled.setCursor(x, y);
    oled.print(f + cf, d);
    oled.print("kHz");
}

// display freq small at x, y, of f (Hz) plus cf (cHz), to decimal places (max 3)
void dispFreqS(u8g2_uint_t x, u8g2_uint_t y, double f, double cf, byte d) {
    oled.setFont(u8g2_font_5x8_tf); // font
    oled.setFontPosTop(); // origin top

    f = f / 1000.0;
    cf = cf / 100000.0;

```

```

oled.setCursor(x, y);
oled.print(f + cf, d);
oled.print("kHz");
}

// STEP
// display step at x, y, value s)tep (Hz)
void dispStep(u8g2_uint_t x, uint16_t y, uint16_t s) {
    oled.setFont(u8g2_font_7x13_tf); // font
    oled.setFontPosTop();

    oled.setCursor(x, y);
    switch (s) // display freqStep
    {
        case 10:
            oled.print(" 10Hz");
            break;
        case 100:
            oled.print("100Hz");
            break;
        case 1000:
            oled.print(" 1kHz");
            break;
        case 10000:
            oled.print(" 10kHz");
            break;
        case 100000:
            oled.print("100kHz");
            break;
        case 1000000:
            oled.print(" 1MHz");
            break;
        case 10000000:
            oled.print("10MHz");
            break;
    }
}

//MSG
// display small message at at x, y, *m)essage
void dispMsgS(u8g2_uint_t x, u8g2_uint_t y, char *m) {
    // sets font, cursor position and displays message
    oled.setFont(u8g2_font_5x8_tf); // font
    oled.setFontPosTop();
    oled.setCursor(x, y);
    oled.print(m);
}

```

```

// display message at at x, y, *m)essage
void dispMsg(u8g2_uint_t x, u8g2_uint_t y, char *m) {
    // sets font, cursor position and displays message
    oled.setFont(u8g2_font_7x13_tf); // font
    oled.setFontPosTop();
    oled.setCursor(x, y);
    oled.print(m);
}

// display large message at at x, y, *m)essage
void dispMsgL(u8g2_uint_t x, u8g2_uint_t y, char *m) {
    oled.setFont(u8g2_font_10x20_tf); // font
    oled.setFontPosTop();
    oled.setCursor(x, y);
    oled.print(m);
}

// display ultra large message at at x, y, *m)essage
void dispMsgUL(u8g2_uint_t x, u8g2_uint_t y, char *m) {
    oled.setFont(u8g2_font_logisoso30_tf); // font
    oled.setFontPosTop();
    oled.setCursor(x, y);
    oled.print(m);
}

// NUM
// display number at x, y, n)umber, d)ecimal places
void dispNum(u8g2_uint_t x, u8g2_uint_t y, double n, byte d) {
    oled.setFont(u8g2_font_7x13_tf); // font
    oled.setFontPosTop();
    oled.setCursor(x, y);
    oled.print(n, d);
}

// display number large at x, y, n)umber, d)ecimal places
void dispNumL(u8g2_uint_t x, u8g2_uint_t y, double n, byte d) {
    oled.setFont(u8g2_font_10x20_tf); // font
    oled.setFontPosTop();
    oled.setCursor(x, y);
    oled.print(n, d);
}

// display number ultra large at x, y, n)umber, d)ecimal places
void dispNumUL(u8g2_uint_t x, u8g2_uint_t y, double n, byte d) {
    oled.setFont(u8g2_font_logisoso24_tf); // font
    oled.setFontPosTop();
    oled.setCursor(x, y);
    oled.print(n, d);
}

```

```

// DATE
// display date "May 21 June 2018"
void dispDate(u8g2_uint_t x, u8g2_uint_t y, byte dw, byte da, byte mo, byte yr) {
    oled.setFont(u8g2_font_7x13_tf); // font
    oled.setFontPosTop();
    oled.setCursor(x, y);

    switch (dw) {
        case 1:
            oled.print("Mon");
            break;
        case 2:
            oled.print("Tue");
            break;
        case 3:
            oled.print("Wed");
            break;
        case 4:
            oled.print("Thu");
            break;
        case 5:
            oled.print("Fri");
            break;
        case 6:
            oled.print("Sat");
            break;
        case 7:
            oled.print("Sun");
            break;
    }

    oled.print(" ");
    oled.print(da);

    oled.print(" ");
    switch (mo)
    {
        case 1:
            oled.print("Jan");
            break;
        case 2:
            oled.print("Feb");
            break;
        case 3:
            oled.print("Mar");
            break;
        case 4:
            oled.print("Apr");

```

```

        break;
    case 5:
        oled.print("May");
        break;
    case 6:
        oled.print("Jun");
        break;
    case 7:
        oled.print("Jul");
        break;
    case 8:
        oled.print("Aug");
        break;
    case 9:
        oled.print("Sep");
        break;
    case 10:
        oled.print("Oct");
        break;
    case 11:
        oled.print("Nov");
        break;
    case 12:
        oled.print("Dec");
        break;
}
oled.print(" ");
oled.print("20");
oled.print(yr);
}

// TIME
// display time HH:MM:SS at x, y)
void dispTime(u8g2_uint_t x, u8g2_uint_t y, byte h, byte m, byte s) {
    oled.setFont(u8g2_font_7x13_tf); // fix font for now
    oled.setFontPosTop();
    oled.setCursor(x, y);
    if (h < 10)
        oled.print("0");
    oled.print(h);
    oled.print(":");
    if (m < 10)
        oled.print("0");
    oled.print(m);
    oled.print(":");
    if (s < 10)
        oled.print("0");
    oled.print(s);
}

```



```
// display time HH:MM:SS at x, y)
void dispTimeL(u8g2_uint_t x, u8g2_uint_t y, byte h, byte m, byte s) {
    oled.setFont(u8g2_font_10x20_tf); // font
    oled.setFontPosTop();
    oled.setCursor(x, y);
    if (h < 10)
        oled.print("0");
    oled.print(h);
    oled.print(":");
    if (m < 10)
        oled.print("0");
    oled.print(m);
    oled.print(":");
    if (s < 10)
        oled.print("0");
    oled.print(s);
}
```