

Show Code

```
// Look at the thresh1,thresh2, thresh3 values for creating levels  
// there is 2 buttons for making a standby ( on/off) as well as a baseline value for shifting the values down  
to a baseline.
```

```
#include <Arduino.h>
```

```
#include <U8x8lib.h>
```

```
#ifndef U8X8_HAVE_HW_SPI
```

```
#include <SPI.h>
```

```
#endif
```

```
#ifndef U8X8_HAVE_HW_I2C
```

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

```
#endif
```

```
U8X8_SSD1306_128X64_NONAME_SW_I2C u8x8(/* clock=*/ SCL, /* data=*/ SDA, /* reset=*/  
U8X8_PIN_NONE);
```

```
#include <Adafruit_NeoPixel.h>
```

```
#define PIN 8 // the data pin for the serial leds
```

```
#define NUMPIXELS 2 // number of pixels in circuit.
```

```
Adafruit_NeoPixel pixels(NUMPIXELS, PIN, NEO_GRB + NEO_KHZ800);
```

```
#define DELAYVAL 10 // pause time
```

```
byte btn1 = 6; // first button (momentary)
```

```
byte btn2 = 7; // second button (toggle)
```

```
byte buttonState1 = 0;
```

```
byte buttonState2 = 0;
```

```

int inputPin = A2; // hook up the GSR to A2
unsigned long gsr = 0;
const int numReadings = 50;
int readings[numReadings]; // the readings from the analog input
int ndex = 0; // the index
int total = 0; // the total
unsigned long average = 0;
int thresh1 = 10;
int thresh2 = 600;
int thresh3 = 500;
int baselineVal = 0;

// these are function declares, if you get the error " 'xxxx was not declared in this scope' put the function
// you made,
//below like the others.
void setupGSR();
unsigned long runGSR();

void setup(){
  pinMode(btn1, INPUT);
  pinMode(btn2, INPUT);
  Serial.begin(9600);
  setupGSR();
  pixels.begin();

  u8x8.begin();
  u8x8.setPowerSave(0);

```

```
}
```

```
void loop() {
```

```
  //Serial.print(0);
```

```
  buttonState1 = digitalRead(btn1);
```

```
  buttonState2 = digitalRead(btn2);
```

```
  gsr = runGSR();
```

```
  if (buttonState2 == HIGH ) { // on/off button
```

```
    if (buttonState1 == HIGH ) { // creating a baseline offset for the gsr value
```

```
      baselineVal = gsr;
```

```
    }
```

```
  // use this to test your values
```

```
  int diffgsrval = gsr - baselineVal;
```

```
  Serial.print(0);
```

```
  Serial.print(",");
```

```
  Serial.print(gsr);
```

```
  Serial.print(",");
```

```
  Serial.print(diffgsrval);
```

```
  Serial.print(",");
```

```
  Serial.println(700);
```

```
  // sound that is optional.
```

```
  tone(0,(gsr)/2, 200);
```

```
  delay(10); //timing for sound
```

```
pixels.clear();
```

```
int mapgsr = map(diffgsrval,0,90,0,255); // map the diffgsrval values to your liking.
```

```
if ( mapgsr < 0 ) {  
    mapgsr = 0;  
}
```

```
// The Threshold values can be what ever point that looks like its  
// a transition to another state. relaxed, stressed, excited, etc.
```

```
if (mapgsr < thresh1) {  
    pixels.setPixelColor(0, pixels.Color(0,0,201));  
    pixels.setPixelColor(1, pixels.Color(0,0,0));
```

```
    } else {  
//  
        pixels.setPixelColor(0, pixels.Color(204,0,0));  
        pixels.setPixelColor(1, pixels.Color(204,0,0));  
    }
```

```
if (mapgsr > thresh2) {  
    pixels.setPixelColor(0, pixels.Color(0,204,141));  
    pixels.setPixelColor(1, pixels.Color(0,129,201));  
}
```

```
} else {  
    pixels.clear(); // Set pixel to 'off'
```

```
}
```

```
pixels.show(); // updat pixels
```

```
delay(DELAYVAL);
```

```
} // end of loop
```

Ic2 add on Code

```
/*
```

```
GraphicsTest.ino
```

```
Some graphics/text output for U8x8 API
```

```
Universal 8bit Graphics Library (https://github.com/olikraus/u8g2/)
```

```
cxvb b
```

```
Copyright (c) 2016, olikraus@gmail.com
```

```
All rights reserved.
```

Redistribution and use in source and binary forms, with or without modification,
are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list
of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this
list of conditions and the following disclaimer in the documentation and/or other
materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

*/

```
#include <Arduino.h>
```

```
#include <U8x8lib.h>
```

```
byte btn2 = 7; // first button (momentary)
```

```
byte buttonState2 = 0;
```

```
#ifdef U8X8_HAVE_HW_SPI
```

```
#include <SPI.h>
```

```
#endif
```

```
#ifdef U8X8_HAVE_HW_I2C
```

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

```
#endif
```

```
U8X8_SSD1306_128X64_NONAME_SW_I2C u8x8(/* clock= */ SCL, /* data= */ SDA, /* reset= */ U8X8_PIN_NONE);
```

```
// End of constructor list
```

```
/*
```

```
    This example will probably not work with the SSD1606, because of the  
    internal buffer swapping
```

```
*/
```

```
void setup(void)
```

```
{
```

```
    /* U8g2 Project: KS0108 Test Board */
```

```
    //pinMode(7, OUTPUT);
```

```
    //digitalWrite(7, 0);
```

```
    /* U8g2 Project: Pax Instruments Shield: Enable Backlight */
```

```
    //pinMode(7, OUTPUT);
```

```
    //digitalWrite(7, 0);
```

```
    u8x8.begin();
```

```
    //u8x8.setFlipMode(1);
```

```
    pinMode(btn2, INPUT);
```

```
}
```

```
void pre(void)
```

```
{
```

```
    u8x8.setFont(u8x8_font_amstrad_cpc_extended_f );
```

```
    u8x8.clear();
```

```
    u8x8.inverse();
```

```
    u8x8.print("Feeling Stressed");
```

```
    u8x8.setFont(u8x8_font_chroma48medium8_r);
```

```
u8x8.noInverse();  
u8x8.setCursor(0,1);  
delay(300);  
}
```

```
void loop(void)  
{  
    buttonState2 = digitalRead(btn2);  
  
    if (buttonState2 == HIGH ) {  
        int i;  
        uint8_t c, r, d;  
        pre();  
        u8x8.setCursor(0,2);  
        u8x8.print("Meditation");  
        delay(1000);  
        u8x8.setCursor(0,4);  
        u8x8.print("IS THE");  
        delay(1000);  
        u8x8.setCursor(0,6);  
        u8x8.print("Answer");  
        delay(1000);  
  
    } else {  
        int i;  
        uint8_t c, r, d;  
        pre();  
        u8x8.drawString(0, 2, "Go Outside");  
        u8x8.setFont(u8x8_font_open_iconic_weather_4x4);  
        for(c = 0; c < 6; c++)  
        {
```



```
u8x8.drawGlyph(0, 4, '@'+c);
```

```
delay(200);
```

```
}
```

```
}
```

```
}
```