```
/************************************************************

WHEN: 07-SEP-2020

WHAT: Magic eye emulator with OLED 1.3” 128×64 I2C Blue Display I2C address 0x3C

https://protosupplies.com/product/oled-1-3-128x64-i2c-blue-display/

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

DETAILS:

without RunningAvg library as per

https://www.powertronika.com/2020/04/reducing-noise-from-sensor-data.html

bitmap for central triangle, side with circle and rectangle

On Mega2056 SDA A20, SCL A21

On UNO SDA A4, SCL A5

Use analogReference() for signal 1-2.5 v

Analog input on A0

Works on direct input from radio receiver

/************************************************************/

#include <Arduino.h> #include <U8g2lib.h>

#ifdef U8X8_HAVE_HW_I2C #include <Wire.h> #endif

#define ARR_SIZE 5

// U8g2 Contructor List (Frame Buffer)
// The complete list is available here: https://github.com/olikraus/u8g2/wiki/u8g2setupcpp U8G2_SH1106_128X64_NONAME_F_HW_I2C u8g2(U8G2_R1, /* reset=*/
U8X8_PIN_NONE); int sum, a, y_max, delta_side,delta_y;
//delta_side,delta_y - for small side triangles when a>32; y_max - maximum height float p,p_avg,myRA[ARR_SIZE]; byte index;

#define owl_width 64 #define owl_height 36 static const unsigned char owl_bits[]
U8X8_PROGMEM = { 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x80, 0x01, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x80, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x80, 0x01,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x80, 0x03, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0xc0, 0x03, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xc0, 0x03, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0xc0, 0x07, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xe0, 0x07, 0x00, 0x00,
```

```
0x00, 0x00, 0x00, 0x00, 0xe0, 0x0f, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xf0, 0x0f,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xf0, 0x1f, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0xf8, 0x1f, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xf8, 0x3f, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0xfc, 0x7f, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xfc, 0x7f, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0xfe, 0xff, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xff, 0xff,
0x01, 0x00, 0x00, 0x00, 0x00, 0x80, 0xff, 0xff, 0x03, 0x00, 0x00, 0x00, 0x00, 0xc0,
0xff, 0xff, 0x07, 0x00, 0x00, 0x00, 0x00, 0xc0, 0xff, 0xff, 0x0f, 0x00, 0x00, 0x00,
0x00, 0xf0, 0xff, 0xff, 0x1f, 0x00, 0x00, 0x00, 0x00, 0xf8, 0xff, 0xff, 0x3f, 0x00,
0x00, 0x00, 0x00, 0x00, 0xfc, 0xff, 0xff, 0x7f, 0x00, 0x00, 0x00, 0x00, 0xfe, 0xff, 0xff,
0xff, 0x01, 0x00, 0x00, 0x80, 0xff, 0xff, 0xff, 0xff, 0x07, 0x00, 0x00, 0xe0, 0xff,
0xff, 0xff, 0xff, 0x1f, 0x00, 0x00, 0xf8, 0xff, 0xff, 0xff, 0xff, 0xff, 0x00, 0x00,
0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0x3f, 0xfc, 0xff, 0xff, 0xff };

void setup(void) { Serial.begin(115200); u8g2.begin(); Serial.print("Sketch: ");
Serial.println(FILE); Serial.print("Uploaded: "); Serial.println(DATE);
delta_side=12; delta_y=40; y_max=100; sum=0; index=0;
u8g2.setFont(u8g2_font_timR10_tr); u8g2.setFontDirection(2); }

void loop(void) { p=analogRead(A0); //averaging sum=sum-myRA[index]; myRA[index]=p;
sum+=p; index=(index+1)%ARR_SIZE; p_avg=sum/ARR_SIZE; //myRA.addValue(p);
//p_avg=myRA.getAverage(); a=min(map(p_avg,1,60,1,32),80); //drawScreen(a,p_avg,1);
 //print amplitude drawScreen(a,p_avg,0); //don't print amplitude }

void drawScreen(int a, int vu, boolean printVU) { int16_t delta, a_x,a_y,a1,r,r1;
u8g2.clearBuffer(); delta=min(32,a); a1=min(a,72);
a_y=min(map(a,4,32,18,32),50);//70 a_x=map(delta,4,32,16,10);
r=map(delta,4,32,11,8); r1=map(delta,4,32,6,0); if (a<6) u8g2.drawBox(29,0,7,y_max);
//central triangle else u8g2.drawTriangle(32,0,32-delta,y_max,32+delta,y_max);
if (a1>32) { u8g2.drawTriangle(32-delta_side, delta_y,0,y_max,0,y_max-(a1-32));
u8g2.drawTriangle(32+delta_side, delta_y,64,y_max,64,y_max-(a1-32)); }
//side triangles
u8g2.drawTriangle(28,10,a_x,a_y,a_x,10); u8g2.drawBox(0,10,a_x,a_y-10);
u8g2.drawDisc(r1+1,a_y-r1-2,r,U8G2_DRAW_LOWER_RIGHT|U8G2_DRAW_LOWER_LEFT);
u8g2.drawTriangle(36,10,64-a_x,a_y,64-a_x,10); u8g2.drawBox(64-a_x,10,a_x,a_y-10);
u8g2.drawDisc(64-r1-1,a_y-r1-2,r,U8G2_DRAW_LOWER_RIGHT|U8G2_DRAW_LOWER_LEFT);
 u8g2.setDrawColor(0); u8g2.drawBox(0,0,64,14); u8g2.setBitmapMode(1);
u8g2.setDrawColor(0); u8g2.drawXBMP(0,64,owl_width, owl_height, owl_bits);
u8g2.setDrawColor(1); if (printVU) { u8g2.setCursor(40, 110); u8g2.print(vu); }
u8g2.sendBuffer();
}
```