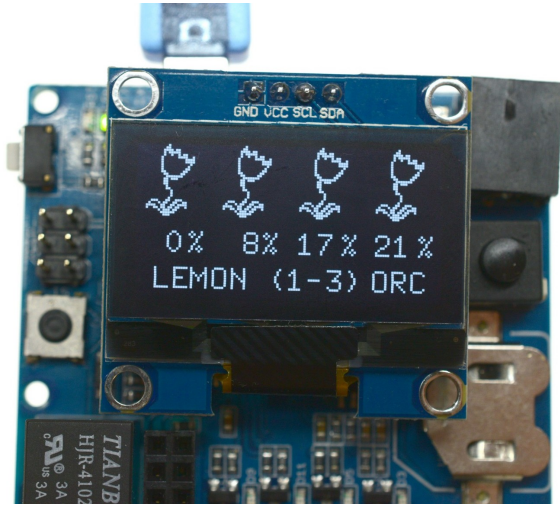
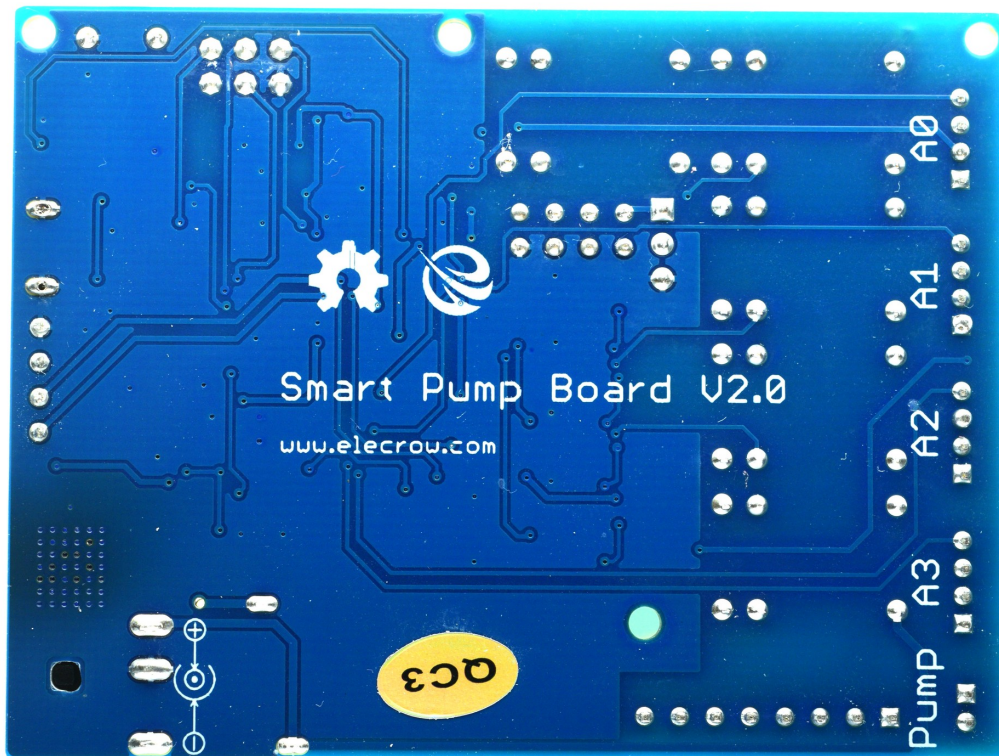
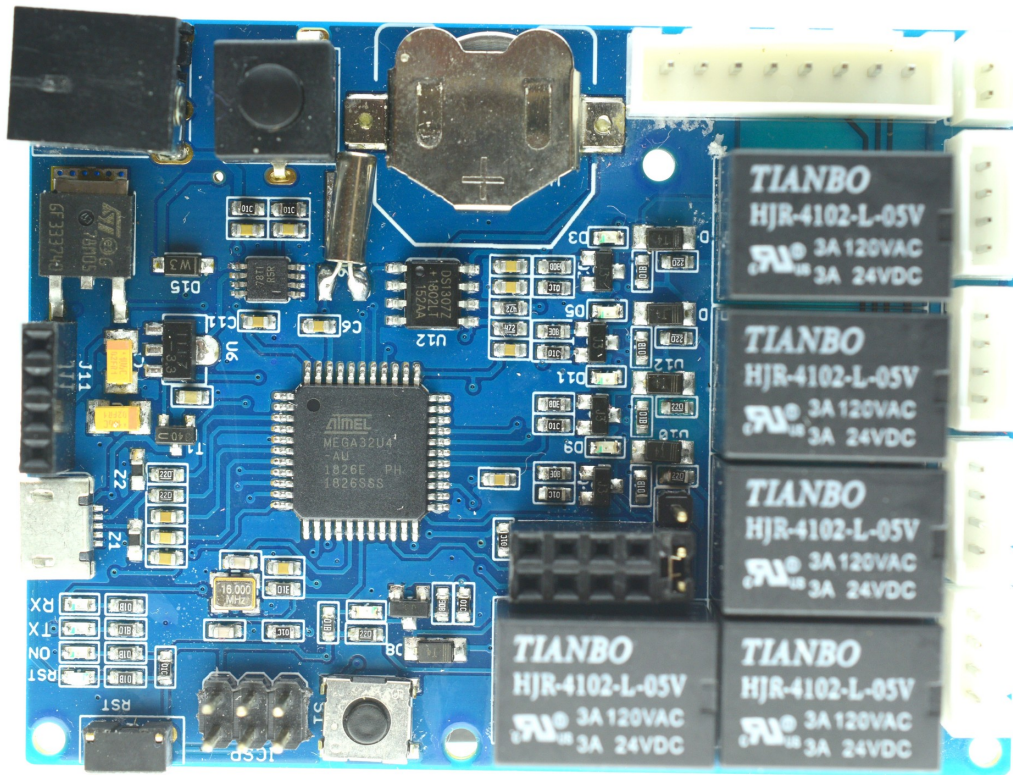


Arduino Automatic Smart Plant Watering Kit 2.0



<https://www.elecrow.com/arduino-automatic-smart-plant-watering-kit.html> (69\$)

<https://ru.aliexpress.com/item/Elecrow-Arduino/32950752157.html> (55\$)



Board: [Arduino Leonardo](#)

Code: https://www.elecrow.com/download/watering_kit.zip

Libs:

- <https://github.com/adafruit/RTCLib/tree/master/examples>
- <https://github.com/olikraus/u8glib>

Modified Watering kit firmware

```
#include <Wire.h>
#include "U8glib.h"
U8GLIB_SSD1306_128X64 u8g(U8G_I2C_OPT_NONE);    // I2C
#include "Wire.h"
#include "RTCLib.h"
RTC_DS1307 RTC;

// set all moisture sensors PIN ID
int moisture1 = A0;
int moisture2 = A1;
int moisture3 = A2;
int moisture4 = A3;

// declare moisture values
int moisture1_value = 0 ;
int moisture2_value = 0;
int moisture3_value = 0;
int moisture4_value = 0;

// set water relays
int relay1 = 6;
int relay2 = 8;
int relay3 = 9;
int relay4 = 10;

// set water pump
int pump = 4;

// set button
int button = 12;

//pump state  1:open  0:close
int pump_state_flag = 0;

//relay1 state  1:open  0:close
int relay1_state_flag = 0;

//relay2 state  1:open  0:close
int relay2_state_flag = 0;

//relay3 state  1:open  0:close
int relay3_state_flag = 0;

//relay4 state  1:open  0:close
int relay4_state_flag = 0;

static unsigned long currentMillis_send = 0;
static unsigned long Lasttime_send = 0;

char daysOfTheWeek[7][12] = {"Sun", "Mon", "Tue", "Wed", "Thu", "Fri", "Sat",};
unsigned long nowtime;
unsigned long endtime;
unsigned long nowtimeNext;
unsigned long nowtime1;
unsigned long endtime1;
unsigned long nowtimeNext1;
unsigned long nowtime2;
unsigned long endtime2;
unsigned long nowtimeNext2;
unsigned long nowtime3;
unsigned long endtime3;
unsigned long nowtimeNext3;
```

```

// good flower
unsigned char bitmap_good[] U8G_PROGMEM = {

    0x00, 0x42, 0x4C, 0x00, 0x00, 0xE6, 0x6E, 0x00, 0x00, 0xAE, 0x7B, 0x00, 0x00, 0x3A, 0x51, 0x00,
    0x00, 0x12, 0x40, 0x00, 0x00, 0x02, 0x40, 0x00, 0x00, 0x06, 0x40, 0x00, 0x00, 0x06, 0x40, 0x00,
    0x00, 0x04, 0x60, 0x00, 0x00, 0x0C, 0x20, 0x00, 0x00, 0x08, 0x30, 0x00, 0x00, 0x18, 0x18, 0x00,
    0x00, 0xE0, 0x0F, 0x00, 0x00, 0x80, 0x01, 0x00, 0x00, 0x00, 0x01, 0x00, 0x00, 0x00, 0x01, 0x00,
    0x00, 0x00, 0x01, 0x00, 0x00, 0x00, 0x00, 0x01, 0x00, 0x00, 0x02, 0xC1, 0x00, 0x00, 0x61, 0x00,
    0x00, 0x1C, 0x79, 0x00, 0x00, 0x34, 0x29, 0x00, 0x00, 0x28, 0x35, 0x00, 0x00, 0x48, 0x17, 0x00,
    0x00, 0xD8, 0x1B, 0x00, 0x00, 0x90, 0x1B, 0x00, 0x00, 0xB0, 0x09, 0x00, 0x00, 0xA0, 0x05, 0x00,
    0x00, 0xE0, 0x07, 0x00, 0x00, 0xC0, 0x03, 0x00
};

// bad flower
unsigned char bitmap_bad[] U8G_PROGMEM = {
    0x00, 0x80, 0x00, 0x00, 0x00, 0xC0, 0x00, 0x00, 0x00, 0xE0, 0x0D, 0x00, 0x00, 0xA0, 0x0F, 0x00,
    0x00, 0x20, 0x60, 0x00, 0x00, 0x10, 0x78, 0x02, 0x00, 0x10, 0xC0, 0x03, 0x00, 0x10, 0xC0, 0x03,
    0x00, 0x10, 0x00, 0x01, 0x00, 0x10, 0x80, 0x00, 0x00, 0x10, 0xC0, 0x00, 0x00, 0x30, 0x60, 0x00,
    0x00, 0x60, 0x30, 0x00, 0x00, 0xC0, 0x1F, 0x00, 0x00, 0x60, 0x07, 0x00, 0x00, 0x60, 0x00, 0x00,
    0x00, 0x60, 0x00, 0x00, 0x00, 0x40, 0x00, 0x00, 0xC0, 0x00, 0x00, 0x00, 0x00, 0x01, 0x00,
    0x00, 0x00, 0x01, 0x00, 0x00, 0x00, 0x01, 0x00, 0x00, 0x80, 0x00, 0x00, 0x00, 0xC7, 0x1C, 0x00,
    0x80, 0x68, 0x66, 0x00, 0xC0, 0x33, 0x7B, 0x00, 0x40, 0xB6, 0x4D, 0x00, 0x00, 0xE8, 0x06, 0x00,
    0x00, 0xF0, 0x03, 0x00, 0x00, 0xE0, 0x00, 0x00
};

// Logo
unsigned char bitmap_logo[] U8G_PROGMEM ={
    0x30, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x30, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x06, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x38, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x06, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x38, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x06, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x38, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x07, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x38, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x07, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x38, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x07, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x18, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x07, 0x00, 0x00, 0x06, 0x00, 0x00, 0x00, 0x00,
    0x18, 0x00, 0x00, 0x03, 0x0e, 0x00, 0xf8, 0xff, 0xc3, 0x00, 0x07, 0x0c,
    0x00, 0x00, 0x00, 0x18, 0x00, 0x80, 0x03, 0x0e, 0x30, 0xfc, 0xff, 0xc1,
    0x00, 0x07, 0x0e, 0x00, 0x00, 0x00, 0x1c, 0x00, 0x80, 0x03, 0x0e, 0x30,
    0x80, 0x03, 0xe0, 0x00, 0x07, 0x0e, 0x00, 0x00, 0x1c, 0x00, 0x80,
    0x03, 0x06, 0x38, 0x80, 0x01, 0xe0, 0x00, 0x07, 0x0e, 0x00, 0x00, 0x00,
    0x1c, 0x00, 0x80, 0x01, 0x07, 0x38, 0xc0, 0x01, 0x60, 0x00, 0x07, 0x06,
    0x00, 0x00, 0x00, 0x1c, 0x00, 0x80, 0x01, 0x07, 0x3c, 0xc0, 0x01, 0x70,
    0x00, 0x07, 0x06, 0x00, 0x00, 0x03, 0x1c, 0x00, 0x80, 0x01, 0x03, 0x1c,
    0xc0, 0x00, 0x30, 0x00, 0x07, 0x06, 0x08, 0x80, 0x02, 0x0c, 0x00, 0xc0,
    0x01, 0x03, 0x1e, 0xc0, 0x00, 0x30, 0x80, 0x07, 0x07, 0x00, 0x40, 0x00,
    0x0c, 0x00, 0xc0, 0x81, 0x03, 0x1e, 0xe0, 0x00, 0x30, 0x80, 0x03, 0x07,
    0x00, 0x40, 0x00, 0x0c, 0x00, 0xc0, 0x81, 0x03, 0x1f, 0xe0, 0x00, 0x30,
    0xe0, 0x03, 0x07, 0x04, 0x40, 0x70, 0x0e, 0x00, 0xc0, 0x80, 0x83, 0x1b,
    0xe0, 0x00, 0x30, 0xf0, 0x03, 0x03, 0x24, 0x43, 0x88, 0x0e, 0x00, 0xc0,
    0x80, 0xc3, 0x19, 0xe0, 0x40, 0x30, 0x3c, 0x03, 0x03, 0xa4, 0xc4, 0x84,
    0x0e, 0xe0, 0xc7, 0x80, 0xe3, 0x18, 0xe0, 0x60, 0xf0, 0x1f, 0x03, 0x03,
    0xa4, 0x44, 0x84, 0xfe, 0xff, 0xc7, 0x00, 0x7f, 0x38, 0xe0, 0x78, 0xf0,
    0x87, 0x03, 0x03, 0x64, 0x44, 0x44, 0xff, 0xff, 0xc3, 0x00, 0x3f, 0x78,
    0xe0, 0x3f, 0xe0, 0x81, 0x03, 0x43, 0x24, 0x68, 0x38, 0x7f, 0x00, 0xc0,
    0x00, 0x0e, 0x70, 0xc0, 0x0f, 0x00, 0x80, 0x01, 0x03, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x80, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x80, 0x01, 0x02,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0xc0, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0xc0, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0xc0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x04, 0xe0, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0c,
    0x60, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x1c, 0x70, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x78, 0x3f, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xe0, 0x1f, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x80,
    0x07, 0x00, 0x00, 0x00, 0x00, 0x00
}

```

```

};

unsigned char bitmap_T[] U8G_PROGMEM = {
    0xF7, 0x01, 0x1D, 0x03, 0x0B, 0x02, 0x0C, 0x02, 0x0C, 0x00, 0x0C, 0x00, 0x0C, 0x00, 0x08, 0x02,
    0x18, 0x03, 0xF0, 0x01
};

unsigned char bitmap_H[] U8G_PROGMEM = {
    0x00, 0x00, 0x80, 0x01, 0xC0, 0x03, 0xE0, 0x07, 0xF0, 0x0F, 0xF8, 0x1F, 0xF8, 0x1F, 0xFC, 0x3F,
    0xFC, 0x3F, 0xFE, 0x7F, 0xEE, 0x7F, 0xB3, 0xF7, 0xBB, 0xFB, 0xBB, 0xFD, 0xBB, 0xFD, 0xC7, 0xFE,
    0x7F, 0xC3, 0x3F, 0xDD, 0xBF, 0xFD, 0xDF, 0xDD, 0xEE, 0x5B, 0xFE, 0x7F, 0xFC, 0x3F, 0xF8, 0x1F,
    0xE0, 0x07, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00
};

void setup()
{
    Wire.begin();
    RTC.begin();
    Serial.begin(9600);
    // declare relay as output
    pinMode(relay1, OUTPUT);
    pinMode(relay2, OUTPUT);
    pinMode(relay3, OUTPUT);
    pinMode(relay4, OUTPUT);
    // declare pump as output
    pinMode(pump, OUTPUT);
    // declare switch as input
    pinMode(button, INPUT);
    //pinMode(ROTARY_ANGLE_SENSOR, INPUT);
    // water_flowler();
    u8g.firstPage();
    do
    {
        draw_ad();
    } while ( u8g.nextPage() );
    delay(1000);
}

void loop()
{
    // read the value from the moisture sensors:
    read_value();
    water_flowler();
    int button_state = digitalRead(button);
    if (button_state == 1)
    {
        //read_value();
        u8g.firstPage();
        do
        {
            drawTH();
            drawflower();
        } while ( u8g.nextPage() );
        delay(500);
    }
    else
    {
        u8g.firstPage();
        do
        {
            drawtime();
            u8g.drawStr(18, 55 , "liutyi.info");
        } while (u8g.nextPage());
        delay(500);
    }
}

//Set moisture value

```

```

void read_value()
{
    float value1 = analogRead(A0);
    moisture1_value = (value1 * 120) / 1023; delay(20);
    float value2 = analogRead(A1);
    moisture2_value = (value2 * 120) / 1023; delay(20);
    float value3 = analogRead(A2);
    moisture3_value = (value3 * 120) / 1023; delay(20);
    float value4 = analogRead(A3);
    moisture4_value = (value4 * 120) / 1023; delay(20);
}

void water_flower()
{
    if (moisture1_value < 30)
    {
        digitalWrite(relay1, HIGH);
        relay1_state_flag = 1;
        delay(50);
        if (pump_state_flag == 0)
        {
            digitalWrite(pump, HIGH);
            pump_state_flag = 1;
            delay(50);
        }
    }
    else if (moisture1_value > 55)
    {
        digitalWrite(relay1, LOW);
        relay1_state_flag = 0;
        delay(50);
        if ((relay1_state_flag == 0) && (relay2_state_flag == 0) && (relay3_state_flag == 0) && (relay4_state_flag
== 0))
        {
            digitalWrite(pump, LOW);
            pump_state_flag = 0;
            delay(50);
        }
    }

    if (moisture2_value < 30)
    {
        digitalWrite(relay2, HIGH);
        relay2_state_flag = 1;
        delay(50);
        if (pump_state_flag == 0)
        {
            digitalWrite(pump, HIGH);
            pump_state_flag = 1;
            delay(50);
        }
    }
    else if (moisture2_value > 55)
    {
        digitalWrite(relay2, LOW);
        relay2_state_flag = 0;
        delay(50);
        if ((relay1_state_flag == 0) && (relay2_state_flag == 0) && (relay3_state_flag == 0) && (relay4_state_flag
== 0))
        {
            digitalWrite(pump, LOW);
            pump_state_flag = 0;
            delay(50);
        }
    }

    if (moisture3_value < 30)
    {
        digitalWrite(relay3, HIGH);
        relay3_state_flag = 1;
        delay(50);
    }
}

```

```

    if (pump_state_flag == 0)
    {
        digitalWrite(pump, HIGH);
        pump_state_flag = 1;
        delay(50);
    }
}
else if (moisture3_value > 55)
{
    digitalWrite(relay3, LOW);
    relay3_state_flag = 0;
    delay(50);
    if ((relay1_state_flag == 0) && (relay2_state_flag == 0) && (relay3_state_flag == 0) && (relay4_state_flag
== 0))
    {
        digitalWrite(pump, LOW);
        pump_state_flag = 0;
        delay(50);
    }
}

if (moisture4_value < 30)
{
    digitalWrite(relay4, HIGH);
    relay4_state_flag = 1;
    delay(50);
    if (pump_state_flag == 0)
    {
        digitalWrite(pump, HIGH);
        pump_state_flag = 1;
        delay(50);
    }
}
else if (moisture4_value > 55)
{
    digitalWrite(relay4, LOW);
    relay4_state_flag = 0;
    delay(50);
    if ((relay1_state_flag == 0) && (relay2_state_flag == 0) && (relay3_state_flag == 0) && (relay4_state_flag
== 0))
    {
        digitalWrite(pump, LOW);
        pump_state_flag = 0;
        delay(50);
    }
}
}

void draw_ad(void){
    u8g.setFont(u8g_font_gdr9r);
    u8g.drawStr(5,55 , "https://liutyi.info");
    u8g.drawXBMP(0, 0,120,34, bitmap_logo);
}

void drawtime(void)
{
    int x = 5;
    float i = 25.00;
    float j = 54;
    DateTime now = RTC.now();
    Serial.print(now.year(), DEC);
    if (! RTC.isrunning())
    {
        u8g.setFont(u8g_font_6x10);
        u8g.setPrintPos(5, 20);
        u8g.print("RTC is NOT running!");
        RTC.adjust(DateTime(__DATE__, __TIME__));
    }
}

```

```

else
{
    u8g.setFont(u8g_font_7x13);
    u8g.setPrintPos(x, 11);
    u8g.print(now.year(), DEC);
    u8g.setPrintPos(x + 80, 11);
    u8g.print(daysOfTheWeek[now.dayOfTheWeek()]);
    u8g.setPrintPos(x + 28, 11);
    u8g.print("/");
    u8g.setPrintPos(x + 33, 11);
    u8g.print(now.month(), DEC);
    if (now.month() < 10)
        x -= 7;
    u8g.setPrintPos(x + 47, 11);
    u8g.print("/");
    u8g.setPrintPos(x + 53, 11);
    u8g.print(now.day(), DEC);
    u8g.setFont(u8g_font_8x13);
    int x = 35;
    u8g.setPrintPos(x, 33);
    u8g.print(now.hour(), DEC);
    if (now.hour() < 10)
        x -= 7;
    u8g.setPrintPos(x + 15, 33);
    u8g.print(":");
    u8g.setPrintPos(x + 21, 33);
    u8g.print(now.minute(), DEC);
    if (now.minute() < 10)
        x -= 7;
    u8g.setPrintPos(x + 36, 33);
    u8g.print(":");
    u8g.setPrintPos(x + 42, 33);
    u8g.print(now.second(), DEC);
}
}

void drawLogo(uint8_t d)
{
    u8g.setFont(u8g_font_gdr25r);
    u8g.drawStr(8 + d, 30 + d, "l");
    u8g.setFont(u8g_font_gdr25r);
    u8g.drawStr(30 + d, 30 + d, "i");
    u8g.setFont(u8g_font_gdr25r);
    u8g.drawStr(40 + d, 30 + d, "u");
    u8g.setFont(u8g_font_gdr25r);
    u8g.drawStr(55 + d, 30 + d, "t");
    u8g.setFont(u8g_font_gdr25r);
    u8g.drawStr(70 + d, 30 + d, "y");
    u8g.setFont(u8g_font_gdr25r);
    u8g.drawStr(85 + d, 30 + d, "i");
    u8g.setFont(u8g_font_gdr25r);
    u8g.drawStr(100 + d, 30 + d, ".");
}

//Style the flowers    bitmap_bad: bad flowers    bitmap_good:good flowers
void drawflower(void)
{
    if (moisture1_value < 30)
    {
        u8g.drawXBMP(0, 0, 32, 30, bitmap_bad);
    }
    else
    {
        u8g.drawXBMP(0, 0, 32, 30, bitmap_good);
    }
    if (moisture2_value < 30)
    {
        u8g.drawXBMP(32, 0, 32, 30, bitmap_bad);
    }
    else

```

```

{
    u8g.drawXBMP(32, 0, 32, 30, bitmap_good);
}
if (moisture3_value < 30)
{
    u8g.drawXBMP(64, 0, 32, 30, bitmap_bad);
}
else
{
    u8g.drawXBMP(64, 0, 32, 30, bitmap_good);
}
if (moisture4_value < 30)
{
    u8g.drawXBMP(96, 0, 32, 30, bitmap_bad);
}
else
{
    u8g.drawXBMP(96, 0, 32, 30, bitmap_good);
}
}

```

```

void drawTH(void)
{
    int A = 0;
    int B = 0;
    int C = 64;
    int D = 96;
    char moisture1_value_temp[5] = {0};
    char moisture2_value_temp[5] = {0};
    char moisture3_value_temp[5] = {0};
    char moisture4_value_temp[5] = {0};
    read_value();
    itoa(moisture1_value, moisture1_value_temp, 10);
    itoa(moisture2_value, moisture2_value_temp, 10);
    itoa(moisture3_value, moisture3_value_temp, 10);
    itoa(moisture4_value, moisture4_value_temp, 10);
    u8g.setFont(u8g_font_8x13);
    u8g.setPrintPos(9, 60);
    u8g.print("LEMON (1-3)");
    if (moisture1_value < 10)
    {
        //u8g.setPrintPos(A + 14, 45 );
        u8g.drawStr(A + 14, 45, moisture1_value_temp);
    }
    else if (moisture1_value < 100)
    {
        //u8g.setPrintPos(A + 6, 45);
        u8g.drawStr(A + 6, 45, moisture1_value_temp);
    }
    else
    {
        //u8g.setPrintPos(A + 2, 45 );
        moisture1_value = 100;
        itoa(moisture1_value, moisture1_value_temp, 10);
        u8g.drawStr(A + 2, 45, moisture1_value_temp);
    }
    //u8g.print(moisture1_value);
    u8g.setPrintPos(A + 23, 45 );
    u8g.print("%");
    //u8g.setPrintPos(41, 60 );
    //u8g.print("A1");
    if (moisture2_value < 10)
    {
        //u8g.setPrintPos(B + 46, 45 );
        u8g.drawStr(B + 46, 45, moisture2_value_temp);
    }
    else if (moisture2_value < 100)
    {
        //u8g.setPrintPos(B + 37, 45);
    }
}

```

```

    u8g.drawStr(B + 37, 45, moisture2_value_temp);
}
else
{
    //u8g.setPrintPos(B + 32, 45);
    moisture2_value = 100;
    itoa(moisture2_value, moisture2_value_temp, 10);
    u8g.drawStr(B + 32, 45, moisture2_value_temp);
}
// u8g.print(moisture2_value);
u8g.setPrintPos(B + 54, 45);
u8g.print("%");
//u8g.setPrintPos(73, 60);
//u8g.print("A2");
if (moisture3_value < 10)
{
    //u8g.setPrintPos(C + 14, 45 );
    u8g.drawStr(C + 14, 45, moisture3_value_temp);
}
else if (moisture3_value < 100)
{
    // u8g.setPrintPos(C + 5, 45);
    u8g.drawStr(C + 5, 45, moisture3_value_temp);
}
else
{
    // u8g.setPrintPos(C + 2, 45);
    moisture3_value = 100;
    itoa(moisture3_value, moisture3_value_temp, 10);
    u8g.drawStr(C + 2, 45, moisture3_value_temp);
}
//u8g.print(moisture3_value);
u8g.setPrintPos(C + 23, 45);
u8g.print("%");
u8g.setPrintPos(100, 60);
u8g.print("ORC");
if (moisture4_value < 10)
{
    //u8g.setPrintPos(D + 14, 45 );
    u8g.drawStr(D + 14, 45, moisture4_value_temp);
}
else if (moisture4_value < 100)
{
    // u8g.setPrintPos(D + 5, 45);
    u8g.drawStr(D + 5, 45, moisture4_value_temp);
}
else
{
    //u8g.setPrintPos(D + 2, 45);
    moisture4_value = 100;
    itoa(moisture4_value, moisture4_value_temp, 10);
    u8g.drawStr(D + 2, 45, moisture4_value_temp);
}
//u8g.print(moisture4_value);
u8g.setPrintPos(D + 23, 45);
u8g.print("%");
}

```