

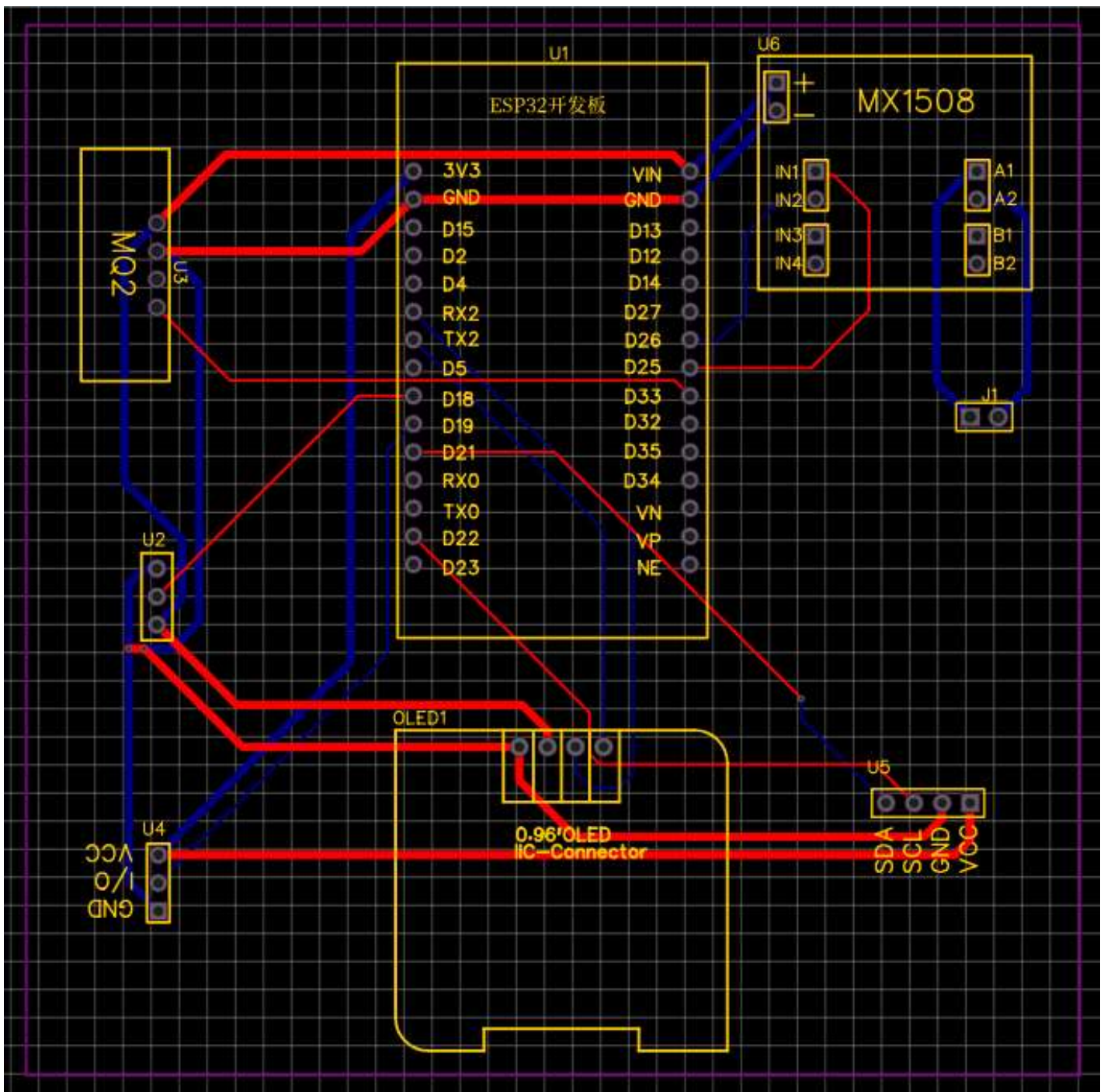
AUTODESK
Instructables

DIY ESP32 Blinker: Designing and Building a Temperature Alarm Device Using ESP32

By [ickey](#) in [CircuitsArduino](#)
Published Nov 19th, 2024

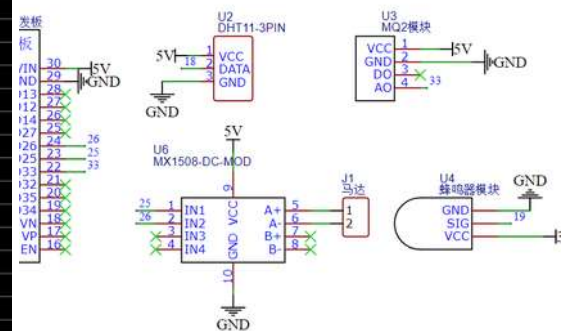
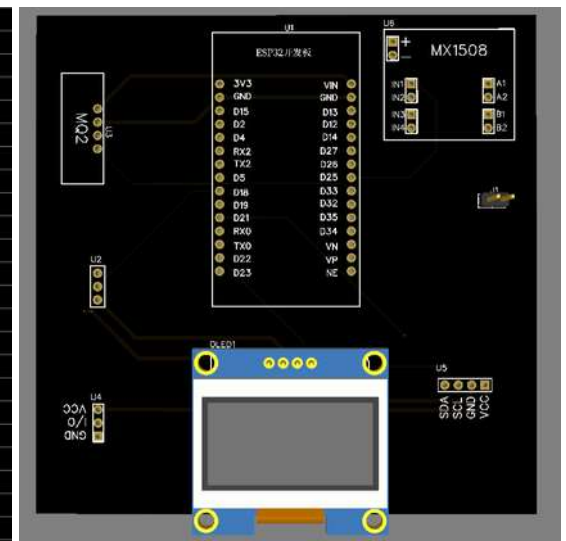
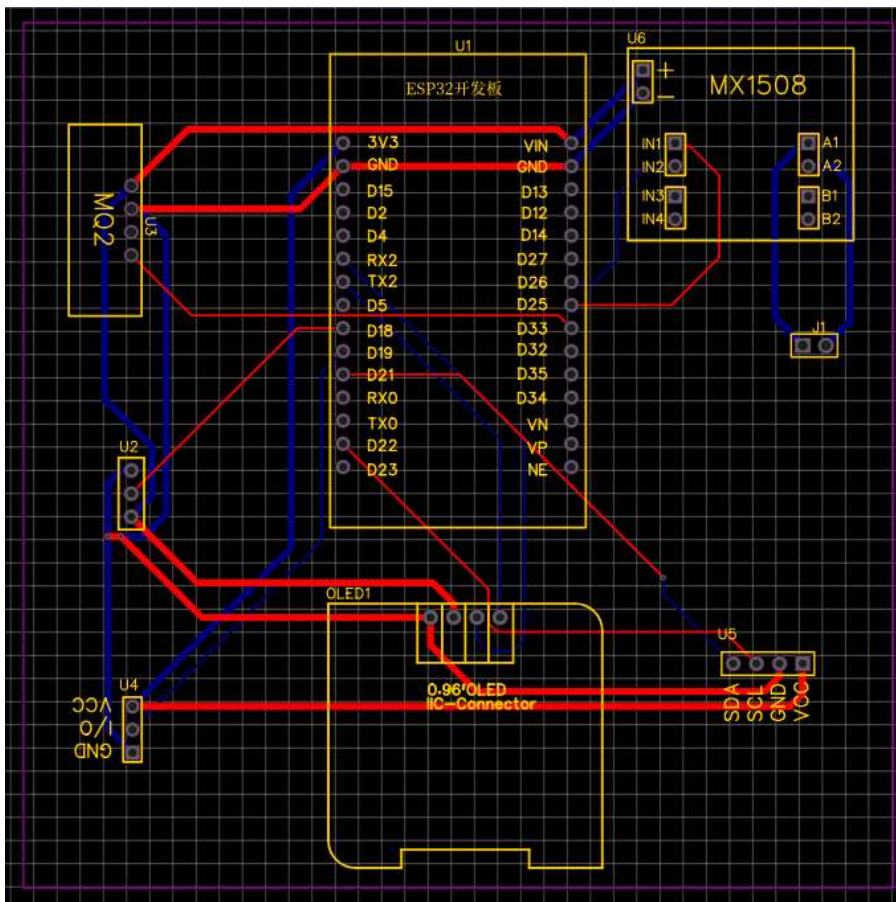


Introduction: DIY ESP32 Blinker: Designing and Building a Temperature Alarm Device Using ESP32



DIY ESP32 Blinker: Designing and Building a Temperature Alarm Device Using ESP32

Step 1:



1. Hardware Materials:

1. 1 * ESP32 development board
2. 1 * DHT11 temperature and humidity module
3. 1 * MQ-2 smoke module
4. 1 * MX1508 motor driver module
5. 1 * DC motor
6. 1 * OLED display module [white] [0.96 inches]
7. 1 * Buzzer module
8. 1 * MLX90614 infrared temperature measurement module
9. 1 * MICRO data cable [0.5m]
10. 1 * Battery [USB-A/microUSB]
11. Other potential components required (breadboard, resistors, capacitors, wires/DuPont wires, etc.)
12. Possible tools needed (multimeter, soldering iron and solder, pliers/scissors, wire strippers/lighter, regular tape/electrical tape/glue gun, etc.)

1. Functionality:

Step1: The [ESP32](#) development board reads temperature and humidity using the DHT11 sensor module, displaying the environmental temperature and humidity on the OLED screen and mobile app. When the temperature exceeds a certain level, the fan will be turned on.

Step2: The [ESP32](#) development board tracks smoke gas concentration using the MQ-2 smoke sensor module. When the concentration surpasses a predetermined level, the buzzer sounds an alarm. Both the current smoke concentration and the alarm status are displayed on the OLED screen and accessible via the mobile app.

Step3: The [ESP32](#) development board measures both object and ambient temperatures using the MLX90614 infrared thermometer module. If the body temperature exceeds a specified threshold, warning messages will be displayed on the OLED screen and the mobile app

Step4: Users can set the smoke concentration threshold via the mobile app.

Step5: Users can configure the environmental temperature threshold through the mobile app.

Step6: Users can set the human body temperature threshold via the mobile app.

For any details not specifically covered, please consult the requirements and concept images/videos for guidance.

1. **post-update circuit diagram**
2. **Schematic diagram:**

Step 2:

```

Arduino.h>
INKER_WIFI                                     //定义连接方式为wifi
Blinker.h>                                   //导入Blinker库文件
dht11.h"                                     //导入DHT11温湿度库文件
Wire.h>
Adafruit_MLX90614.h>
U8g2lib.h>

t11Pin          18                          //接dht11的I/O引脚
zzerPin         19                          //接蜂鸣器的I/O引脚
torIN1Pin       25                          //接电机驱动模块的IN1
torIN2Pin       26                          //接电机驱动模块的IN2
ogPin           33                          //接烟雾的AO引脚

edSdaPin        17                          //接oled的SDA引脚
edSclPin        16                          //接oled的SCL引脚

iggerValue = 1200;                          //烟雾的触发值（高于）
riggerValue = 38.0;                         //人体温度的触发值（高于）
atureTriggerValue = 30;                     //温度的触发值（高于）

```

```

float = "OK";

zerOn          LOW                          //打开蜂鸣器
zerOff         HIGH                         //关闭蜂鸣器

DHT11D_PIN100_H5 1000

= "864131F6B02";                          //MLX90614的地址
= "canlang85";                             //需要注册的WiFi名称
= "12345678";                              //需要注册的WiFi密码

是
on<NumberTemperature("num-temperature");    //温度元件对象
on<NumberHumidity("num-humidity");          //湿度元件对象
on<NumberSng("num-sng");                    //数字元件对象
on<NumberHeartRate("num-rv");               //人体心率元件对象
on<NumberSp2("num-sw");                     //数字元件对象

on<SliderSng("ran-sng");                     //数字滑块

on<SliderTemperature("ran-temperature");     //温度滑块
on<SliderSp2("ran-rvst");                    //人体心率滑块
on<ButtonUpdate("btn-update");               //手动更新数据元件对象

{
  //初始化MLX90614模块
  MLX90614 mlx = Adafruit_MLX90614();
  //初始化MLX90614模块的引脚
  K12804_NONAME_0_B0_I2C vsg2(vsg2_00, oledSclPin, oledSdaPin, VCC_5V_0V0E);

  turnValue = 0;                            //温度值
  pValue = 0;                                //湿度值
  sw = 0;                                     //数字值
  lsw = 0.00;                                //人体心率值
  lsw = 0.00;                                //数字值

  n1 {
    (3, 255);
    (2, 150);

  on {
    (5, 0);
    (2, 0);

  sngCallback(int32_t value) {
    on<数字滑块>(" ", value);
    rValue = value;
    int(r("设置数字滑块为 "));
    int(sngTriggerValue);

  TemperatureCallback(int32_t value) {
    on<温度滑块>(" ", value);
    reTriggerValue = value;
    int(r("设置温度滑块为 "));
    int(temperatureTriggerValue);
  }

```

1. complete Arduino code

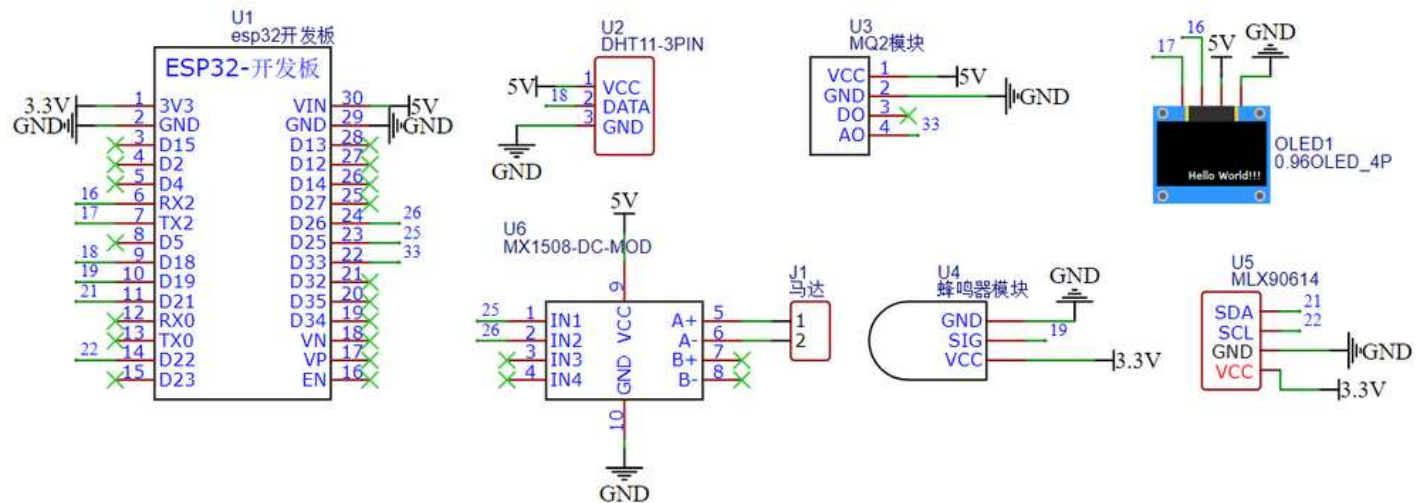
Below is the detailed process of the development code.

Step 3:



1. Effect Demonstration
2. Blinker APP Mobile Interface

Step 4:



1. Video [link](#)

Tag: Temperature Alarm Device, ESP32 Development Board, Electronic Components