

# Entrega Avaliação dos Grupos

## Proposta realizada

O sensor de detecção de gases tóxicos para segurança do indivíduo.

## Materiais utilizados

Um Protoboard; Um ESP 32; Jumpers; Um LED vermelho; Um LED verde; Dois resistores de 220 Ohms; Um sensor MQ7; Um Arduino UNO;

## Código desenvolvido

```
*****
* 1. CREDENCIAIS BLYNK E WIFI
* Substitua pelos valores obtidos no Blynk Console
*****/
```

```
#define BLYNK_TEMPLATE_ID  "SEU_TEMPLATE_ID_AQUI"
#define BLYNK_DEVICE_NAME  "SEU_DEVICE_NAME_AQUI"
#define BLYNK_AUTH_TOKEN   "SEU_AUTH_TOKEN_AQUI"

char ssid[] = "NOME_DA_SUA_REDE_WIFI";      // Ex: "MinhaCasa"
char pass[] = "SENHA_DA_SUA_REDE_WIFI";      // Ex: "12345678"

*****
* 2. INCLUSÃO DE BIBLIOTECAS
*****/
```

```
#include <WiFi.h>
#include <BlynkSimpleEsp32.h>

// Objeto Timer do Blynk
BlynkTimer timer;

*****

```

```

* 3. DEFINIÇÃO DE PINOS E LÓGICA
*****/
```

```

// Pinos físicos do ESP32 para o hardware
const int MQ135_PIN = 34;      // Saída Analógica A0 do MQ-135
const int LED_VERDE = 2;       // LED Verde para AR BOM
const int LED_VERMELHO = 4;    // LED Vermelho para ALERTA/AR RUIM

// Limite (Threshold) para acionar o alerta (Ajuste este valor após testar!)
// Valor bruto de 0 a 4095. 2500 é um bom ponto de partida.
const int ALERTA_THRESHOLD = 2500;

// Função chamada pelo timer para ler o sensor e enviar dados
void sendSensorData()
{
    // 1. Leitura do valor analógico bruto do sensor
    int airQualityRaw = analogRead(MQ135_PIN);

    // 2. Envio da leitura para o pino virtual V0 do Blynk
    Blynk.virtualWrite(V0, airQualityRaw);

    // (Opcional) Envio do status (0 ou 1) para o pino virtual V1
    Blynk.virtualWrite(V1, (airQualityRaw > ALERTA_THRESHOLD) ? 1 : 0);

    // 3. Lógica de Controle dos LEDs
    if (airQualityRaw < ALERTA_THRESHOLD) {
        // QUALIDADE BOA: Liga o Verde, Desliga o Vermelho
        digitalWrite(LED_VERDE, HIGH);
        digitalWrite(LED_VERMELHO, LOW);
        Serial.print("Ar Bom (Verde): ");
    } else {
        // QUALIDADE RUIM: Liga o Vermelho, Desliga o Verde
        digitalWrite(LED_VERDE, LOW);
        digitalWrite(LED_VERMELHO, HIGH);
        Serial.print("Alerta de Poluição (Vermelho): ");
    }

    // (Opcional) Envia uma notificação push para o app Blynk
    Blynk.logEvent("alerta_poluicao", "Nível de gases alto detectado!");
}

// Imprime a leitura bruta no Monitor Serial para depuração
Serial.println(airQualityRaw);
}

```

```

*****
* 4. VOID SETUP - EXECUTADO UMA VEZ
*****
void setup()
{
    // Inicializa a comunicação serial para debug
    Serial.begin(115200);

    // Define os pinos dos LEDs como SAÍDAS
    pinMode(LED_VERDE, OUTPUT);
    pinMode(LED_VERMELHO, OUTPUT);

    // Conecta ao Blynk usando as credenciais e o WiFi
    Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);

    // Define a função 'sendSensorData' para ser chamada a cada 5 segundos (5000 ms)
    timer.setInterval(5000L, sendSensorData);
}

*****
* 5. VOID LOOP - EXECUTADO CONTINUAMENTE
*****
void loop()
{
    Blynk.run(); // Mantém a conexão com o Blynk ativa
    timer.run(); // Verifica se é hora de executar a função do timer
}

```

## Projeto funcionando

Veja o [vídeo completo no Youtube](#).

## Colaboradores

Luiz Henrique Crepaldi

Matheus Armelindo

Tarsis Lima

Thiago Plancke

Mateus Linardi

Kevin Walker

Emannuel Paulo