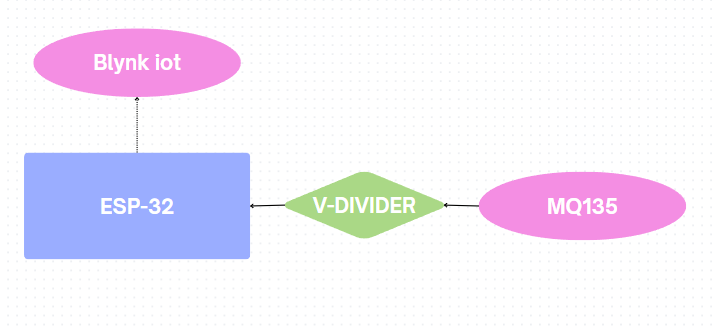
**Air Quality Sensor**

**BLOCK DIAGRAM**



**CODE**

#define BLYNK\_TEMPLATE\_ID "TMPL32FBoJGiG"

#define BLYNK\_TEMPLATE\_NAME "ESP32 AQI"

#define BLYNK\_AUTH\_TOKEN "Ht\_3AEamBch8u6ehSUu63xSNjhSjpiBU"

#include <WiFi.h>

#include <BlynkSimpleEsp32.h>

// WiFi credentials

char ssid[] = "POCO X6 Pro 5G\_PeCSNsn\_MI";

char pass[] = "12345678";

#define MQ135\_PIN 34 // Analog pin

BlynkTimer timer;

// Constants for MQ135 calculations

float RL = 10000.0; // Load resistance (10k Ohm)

float R0 = 22000.0; // Calibrated sensor resistance in clean air

float a = 116.6020682; // Curve fit constant

float b = -2.769034857; // Curve fit constant

void sendSensor() {

int rawADC = analogRead(MQ135\_PIN);

float voltage = rawADC \* (3.3 / 4095.0);

float RS = (3.3 - voltage) \* RL / voltage;

float ratio = RS / R0;

float ppm = a \* pow(ratio, b);

// Determine air quality status based on raw ADC value

String airStatus;

if (rawADC < 300) airStatus = "Excellent";

else if (rawADC < 800) airStatus = "Good";

else if (rawADC < 1500) airStatus = "Moderate";

else if (rawADC < 2500) airStatus = "Unhealthy";

else airStatus = "Hazardous";

// Print data to Serial

Serial.print("Raw ADC: "); Serial.print(rawADC);

Serial.print(" | Voltage: "); Serial.print(voltage, 3); Serial.print(" V");

Serial.print(" | RS: "); Serial.print(RS, 1); Serial.print(" Ω");

Serial.print(" | Ratio: "); Serial.print(ratio, 2);

Serial.print(" | PPM: "); Serial.print(ppm, 1);

Serial.print(" | Status: "); Serial.println(airStatus);

// Send data to Blynk virtual pins

Blynk.virtualWrite(V2, ppm); // Send ppm value

Blynk.virtualWrite(V1, airStatus); // Send status string based on raw ADC

Blynk.virtualWrite(V0, rawADC); // Send raw ADC value

}

void setup() {

Serial.begin(9600);

analogReadResolution(12); // ESP32 ADC resolution (0-4095)

delay(2000); // Stabilize sensor power-up

Blynk.begin(BLYNK\_AUTH\_TOKEN, ssid, pass);

timer.setInterval(2000L, sendSensor); // Send sensor data every 2 seconds

}

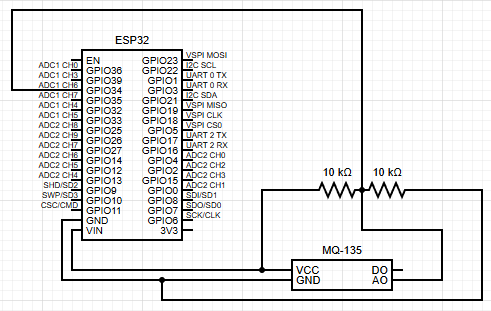
void loop() {

Blynk.run();

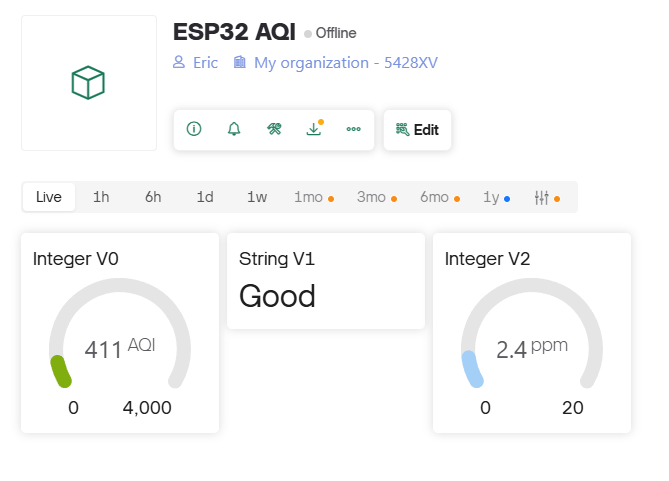
timer.run();

}

**CIRCUIT DIAGRAM**



**BLYNK DASHBOARD**

****