IOT NOISE MONITOR HARDWARE AND SOFWARE DOCUMENTATION

In this project, we design and implement an IOT **Noise Monitoring System** using an **ESP32** microcontroller, **sensor (LDR module)**, and an **LCD display** to measure and display noise levels in real-time. The system also integrates with **Blynk IoT** to enable remote monitoring via a mobile application.

The ESP32 reads analog data from the sound sensor, processes it to determine the noise level in **decibels (dB)**, and displays the result on an **I2C LCD screen**. Additionally, the noise data is sent to Blynk, where users can monitor noise levels remotely. The system also provides an alert mechanism when noise levels exceed a predefined threshold.

This project demonstrates the use of:

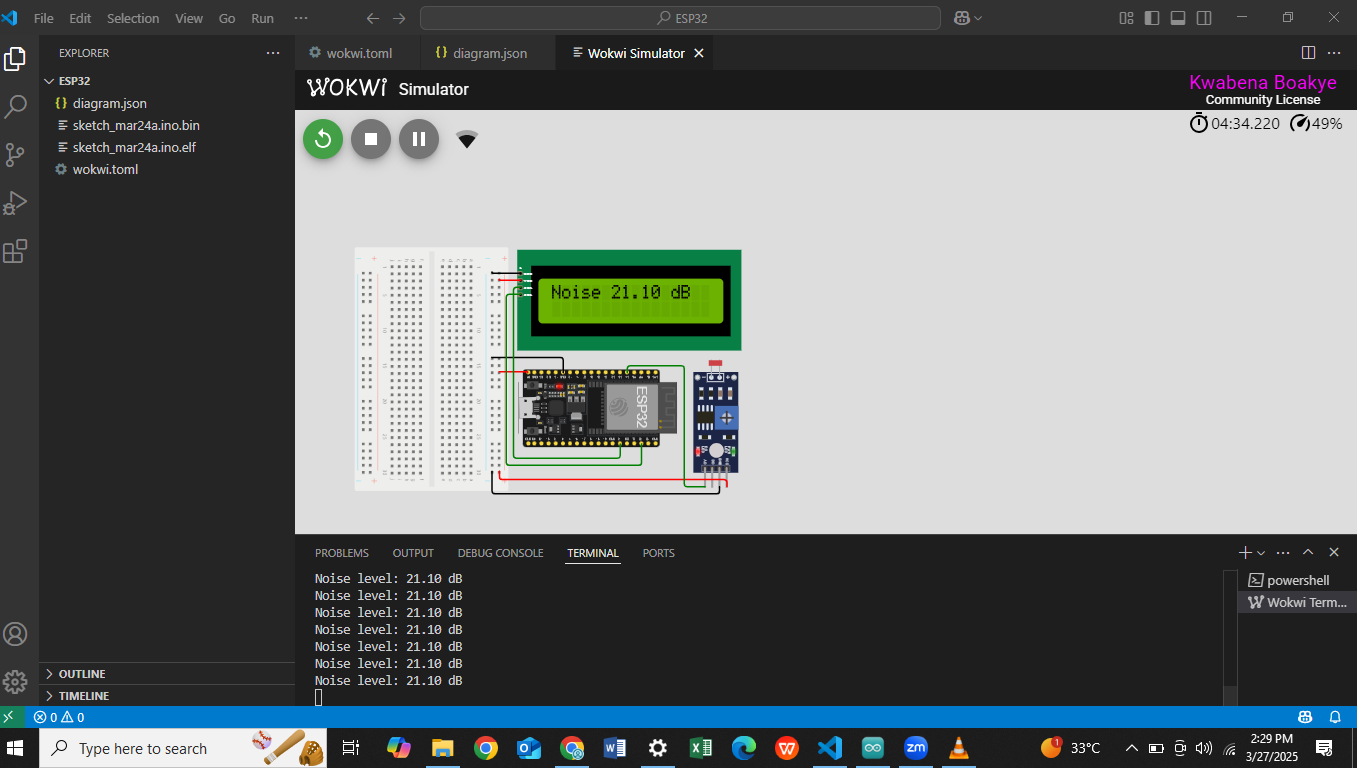
**ESP32 for IoT connectivity and data processing**

**Blynk IoT for cloud-based monitoring**

**I2C LCD for real-time display**

**Light Dependent Resistor(LDR) -To model the output of a sound sensor**

**COMPONENT DESCRIPTION**



The ESP32 is a microcontroller with built-in WiFi and Bluetooth capabilities, making it ideal for IoT applications. It features multiple ADC (Analog-to-Digital Converter) pins, allowing it to read analog sensor data . In this project, it serves as the core processing unit, handling sensor data, displaying results on an LCD, and transmitting noise level readings to the Blynk IoT platform.

The I2C LCD module is a display unit used for real-time data visualization. Unlike traditional LCDs that require multiple digital pins for communication, this module uses the I2C protocol, reducing wiring complexity and making it easier to interface with the microcontroller .The LCD displays measured noise levels in decibels (dB), allowing users to view real-time readings directly from the device. The **LiquidCrystal\_I2C** library is used to control the display, enabling smooth and efficient updates.

Blynk IoT is a cloud-based platform that enables remote monitoring and control of IoT devices via a mobile application. It provides an interface for visualizing real-time sensor data, setting up alerts, and logging historical data. In this project, the ESP32 transmits noise level readings to the Blynk cloud, allowing users to monitor noise levels remotely from their smartphones. Additionally, Blynk was configured to send notifications if noise levels exceed a predefined threshold, making it a useful tool for real-time monitoring and alerts.

The LDR module although traditionally used to detect light intensity was used in this project to stand in as a sound sensor.This is because the output of the LDR module is analogous to that of a sound sensor and as such provides an appropriate model of a sound sensor.



#define BLYNK\_TEMPLATE\_ID "TMPL2e-cxSI\_P"

#define BLYNK\_TEMPLATE\_NAME "ESP32 NOISE MONITOR"

#define BLYNK\_AUTH\_TOKEN "O4fTK5kYgP4z79ndGLhH-v\_a11vHw-lM"

These define the project template ID and name in Blynk.

BLYNK\_AUTH\_TOKEN is a unique authentication token from the Blynk app, allowing your ESP32 to communicate with Blynk's cloud services.

Wire.h: Enables I2C communication, which is required for the LCD.

LiquidCrystal\_I2C.h: Provides functions to control the LCD over I2C.

WiFi.h: Allows the ESP32 to connect to WiFi networks.

BlynkSimpleEsp32.h: Enables communication with the Blynk IoT platform over WiFi.

Define WiFi Credentials

char ssid[] = "Wokwi-GUEST";

char pass[] = "";

Ssid is the WiFi network name (Wokwi-GUEST is the name of the network being used.

pass is the password, which is empty for Wokwi’s guest network.

LiquidCrystal\_I2C lcd(0x27, 16, 2);

#define SOUND\_SENSOR 34

The LCD is created with I2C address 0x27.

SOUND\_SENSOR is connected to GPIO34, which is an ADC (Analog-to-Digital Converter) pin on ESP32.



Serial.begin(115200);

Initializes serial communication at a baud rate of 115200 for debugging.

WiFi.begin(ssid, pass);

This is what initializes wifi on the ESP32

7. Initialize LCD

lcd.begin();

This initialiazes the LCD

lcd.backlight();

This turns on the LCD backlight

lcd.setCursor(0, 0);

lcd.print("Noise Level:");

pinMode(SOUND\_SENSOR, INPUT);

lcd.setCursor(0, 0); moves the cursor to the first row, first column.

lcd.print("Noise Level:"); displays "Noise Level:" on the LCD.

pinMode(SOUND\_SENSOR, INPUT); sets the sensor pin as an input.

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

Blynk.begin() connects the ESP32 to the Blynk IoT platform.

It uses WiFi credentials (ssid, pass) and Blynk authentication token.



int sensorValue = analogRead(SOUND\_SENSOR);

float voltage = (sensorValue / 4095.0) \* 3.3; // Convert ADC value to voltage

float noise\_level = 20 \* log10(voltage / 3.3) + 50; // Convert voltage to dB

analogRead(SOUND\_SENSOR); reads the analog value from the LDR-based sound sensor.

The ESP32 ADC (Analog-to-Digital Converter) outputs values between 0 and 4095.

The voltage is calculated using voltage = (sensorValue / 4095.0) \* 3.3;.

The noise level is estimated in decibels (dB) using 20 \* log10(voltage / 3.3) + 50.

This formula is an approximation

lcd.setCursor(6, 0);

lcd.print(" ");

lcd.setCursor(6, 0);

lcd.print(noise\_level);

lcd.print(" dB");

lcd.setCursor(6, 0); moves the cursor where the noise level will be displayed.

lcd.print(" "); clears the previous value to prevent overlapping numbers.

lcd.print(noise\_level); prints the new noise level in decibels.

lcd.print(" dB"); adds the unit label.

12. Send Data to Serial Monitor

Serial.print("Noise level: ");

Serial.print(noise\_level);

Serial.println(" dB");

Sends the noise level to the serial monitor for debugging.

Blynk.run();

Blynk.virtualWrite(V0, noise\_level);

Blynk.run(); keeps the Blynk connection active.

Blynk.virtualWrite(V0, noise\_level); sends the noise level to Virtual Pin V0 in the Blynk app.

if (noise\_level > 40) {

lcd.setCursor(0, 1);

lcd.print("High noise level!");

Blynk.logEvent("high\_noise\_level", "Warning! High Noise Level Detected");

}

else {

lcd.setCursor(0, 1);

lcd.print(" "); // Clear the warning message

}

If the noise level exceeds 40 dB, the LCD displays a warning:High noise level!

Blynk.logEvent("high\_noise\_level", "Warning! High Noise Level Detected");

Sends an event notification to the Blynk app.

If notifications are enabled, it alerts the user.

If the noise level is below 40 dB, the message is cleared using lcd.print(" ");.

delay(1000);

Waits 1 second before reading the sensor again to prevent excessive updates.