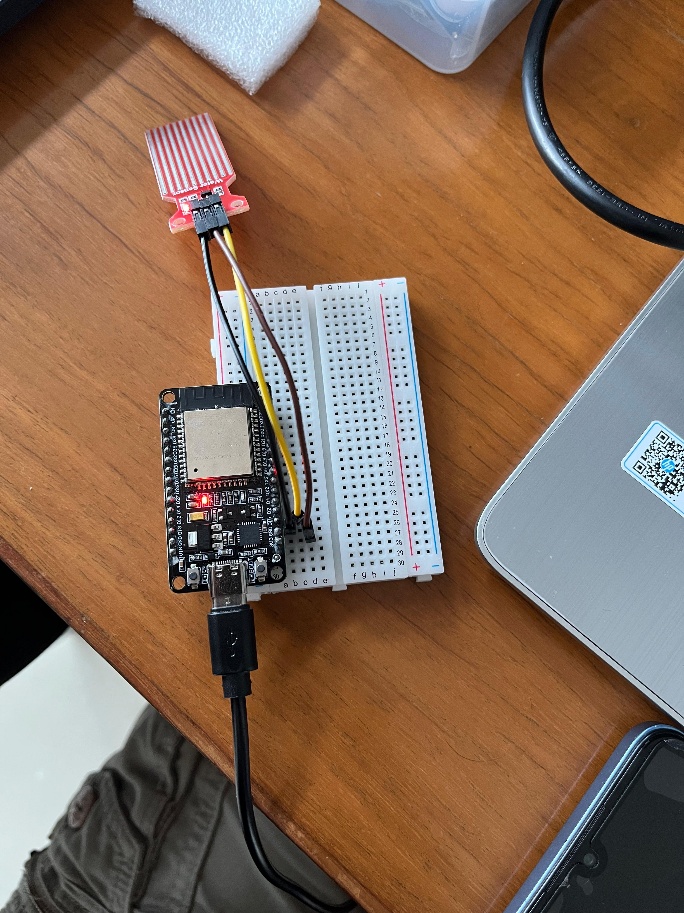
**Detektor Ketinggian Air - Pengumpulan Data**



**1. Pendahuluan**

Dalam proyek ini, saya mengembangkan sebuah Objek Pintar yang menggunakan sensor ketinggian air untuk mendeteksi ketinggian air dan mengirimkan data ke platform ThingSpeak untuk pemantauan secara real-time. Proyek ini memanfaatkan mikrokontroler ESP32, yang terhubung ke jaringan Wi-Fi dan mengirimkan data sensor ke ThingSpeak melalui API HTTP. Tujuan utama dari proyek ini adalah untuk memantau ketinggian air menggunakan sensor dan memvisualisasikan data di platform ThingSpeak.

**2. Konfigurasi Antara WOKWI dan ThingSpeak**

Data dalam proyek ini dikirimkan menggunakan API HTTP melalui platform ThingSpeak. Meskipun MQTT adalah alternatif lain untuk komunikasi data IoT, API HTTP dipilih karena kesederhanaannya dalam mengirimkan data langsung melalui URL.

**Langkah-langkah Konfigurasi:**

1. **Wokwi:** Di Wokwi, saya menggunakan perangkat ESP32 yang disimulasikan untuk menghubungkan proyek ke jaringan Wi-Fi. Wokwi memungkinkan pengujian virtual baik untuk kode maupun perangkat keras sebelum implementasi fisik.
2. **ThingSpeak:** Saya membuat akun di ThingSpeak dan menyiapkan saluran baru untuk pengiriman data. Setelah saluran dibuat, saya memperoleh Channel ID dan API Key yang digunakan untuk mengirimkan data dari perangkat ke ThingSpeak.
3. **Pengiriman Data ke ThingSpeak:** Saya menggunakan URL API ThingSpeak untuk mengirimkan data dari ESP32:

https://api.thingspeak.com/update?api\_key=<Your\_API\_Key>&field1=<value>

URL ini digunakan dalam kode untuk mentransmisikan data sensor ke saluran ThingSpeak yang telah dibuat.

**3. Modifikasi pada Objek Pintar untuk Menghubungkan dan Mengirim Data ke ThingSpeak**

Untuk memastikan perangkat dapat terhubung ke ThingSpeak dan mengirimkan data, beberapa modifikasi dilakukan pada Objek Pintar (ESP32), termasuk yang berikut:

1. **Koneksi Wi-Fi:** Kode dimulai dengan menghubungkan ESP32 ke jaringan Wi-Fi menggunakan SSID dan password yang telah ditentukan. Setelah koneksi berhasil, perangkat melanjutkan untuk mengirimkan data ke ThingSpeak.

WiFi.begin(ssid, password); while (WiFi.status() != WL\_CONNECTED) { delay(1000); Serial.println("Connecting to WiFi..."); } Serial.println("Connected to WiFi");

1. **Inisialisasi ThingSpeak:** Setelah berhasil terhubung ke Wi-Fi, saya menginisialisasi ThingSpeak untuk memulai komunikasi antara perangkat dan ThingSpeak menggunakan ThingSpeak.begin(client);
2. **Pengiriman Data ke ThingSpeak:** Setiap 5 detik, sensor ketinggian air membaca ketinggian air (HIGH atau LOW), dan data ini dikirimkan ke ThingSpeak melalui API menggunakan perintah berikut:

ThingSpeak.setField(1, val); // Mengirimkan nilai sensor ke field 1 int responseCode = ThingSpeak.writeFields(channelNumber, apiKey);

Data ditransmisikan setiap interval 5 detik (setiap kali sensor dibaca).

1. **Kontrol Buzzer:** Sistem ini juga dilengkapi dengan buzzer yang aktif saat ketinggian air rendah. Buzzer akan mati saat ketinggian air tinggi:

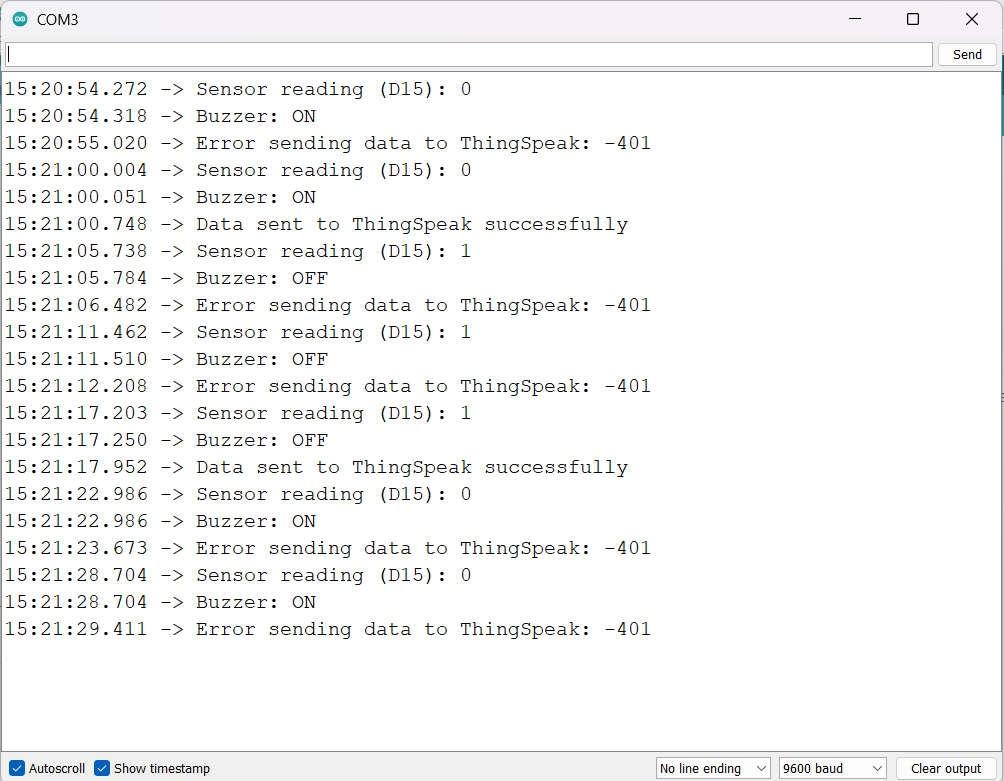
if (val == LOW) { digitalWrite(bz, LOW); // Buzzer ON Serial.println("Buzzer: ON"); } else { digitalWrite(bz, HIGH); // Buzzer OFF Serial.println("Buzzer: OFF"); }

**4. Kode Program**

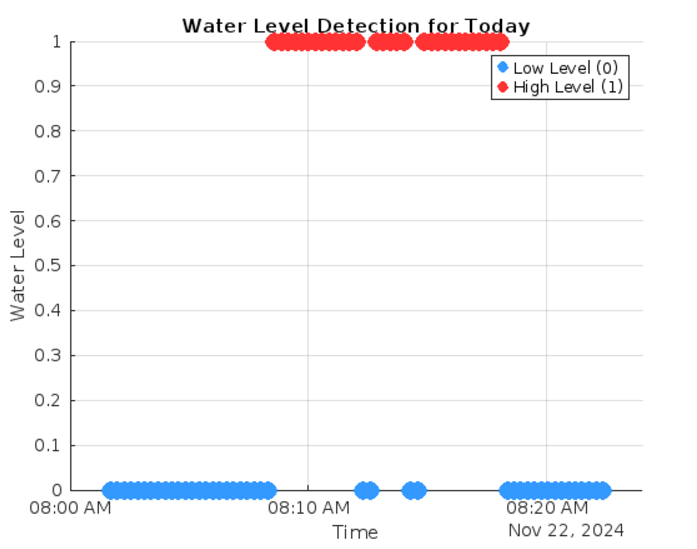
Berikut adalah kode lengkap untuk mendeteksi ketinggian air menggunakan sensor dan mengirimkan data ke ThingSpeak:

#include <WiFi.h>  
#include <HTTPClient.h>  
#include "ThingSpeak.h"  
//-- Kredensial Wi-Fi --  
const char\* ssid = "Woi"; // SSID Wi-Fi Anda  
const char\* password = "candra123"; // Password Wi-Fi Anda  
//-- Pengaturan ThingSpeak --  
unsigned long channelNumber = 2747170; // Ganti dengan nomor saluran ThingSpeak Anda  
const char\* apiKey = "HBT1SVX4RVBCG2PW"; // Ganti dengan API key ThingSpeak Anda  
const char\* server = "https://api.thingspeak.com/update?api\_key=";  
//-- Pengaturan pin --  
#define wlS 15 // Sensor ketinggian air di pin digital D15  
#define bz 19 // Buzzer di pin 19  
WiFiClient client;  
void setup() {  
//-- Monitor Serial --  
Serial.begin(9600); delay(1000); // Tunggu sejenak agar Serial monitor terbuka  
//-- Pengaturan pin --  
pinMode(wlS, INPUT); pinMode(bz, OUTPUT);  
//-- Koneksi Wi-Fi --  
WiFi.begin(ssid, password);  
while (WiFi.status() != WL\_CONNECTED) { delay(1000); Serial.println("Connecting to WiFi..."); }  
Serial.println("Connected to WiFi");  
//-- Inisialisasi ThingSpeak --  
ThingSpeak.begin(client);  
Serial.println("Setup complete. Starting loop...");  
}  
void loop() {  
int val = digitalRead(wlS); // Membaca nilai digital dari sensor ketinggian air  
Serial.print("Pembacaan sensor (D15): ");  
Serial.println(val); // Memperbarui status ketinggian air berdasarkan pembacaan sensor  
String nilai = val ? "HIGH" : "LOW";  
// Kontrol Buzzer dan output serial  
if (val == LOW) { digitalWrite(bz, LOW); // Nyalakan buzzer jika ketinggian air rendah  
Serial.println("Buzzer: ON"); } else { digitalWrite(bz, HIGH); // Matikan buzzer jika ketinggian air tinggi  
Serial.println("Buzzer: OFF"); }  
// Mengirimkan data ke ThingSpeak  
if (WiFi.status() == WL\_CONNECTED) {  
ThingSpeak.setField(1, val); // Mengirimkan ketinggian air sebagai 0 atau 1 ke Field 1  
int responseCode = ThingSpeak.writeFields(channelNumber, apiKey);  
if (responseCode == 200) { Serial.println("Data berhasil dikirim ke ThingSpeak"); }  
else { Serial.println("Kesalahan pengiriman data ke ThingSpeak: " + String(responseCode)); }  
} else { Serial.println("WiFi tidak terhubung"); }  
delay(5000); // ThingSpeak memungkinkan pembaruan setiap 15 detik  
}

**Output**

****

**Grafik Output**



**5. Kesimpulan**

Dalam proyek ini, saya berhasil menghubungkan Objek Pintar (sensor ketinggian air menggunakan ESP32) ke ThingSpeak untuk pengiriman data secara real-time. Penggunaan API HTTP memungkinkan perangkat untuk mengirimkan data ke ThingSpeak dengan mudah. Sebagai hasilnya, kita dapat memantau ketinggian air melalui dasbor ThingSpeak yang diperbarui pada interval waktu tertentu.