

Laporan Instrumentasi Geofisika Lanjut

Pembacaan Sensor Getar dan Penyimpanan Data via SPIFFS dengan Microcontroler
ESP32



Avif Maulana Azis

140710180048

DEPARTEMEN GEOFISIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS PADJADJARAN
TAHUN 2021

I. Pendahuluan

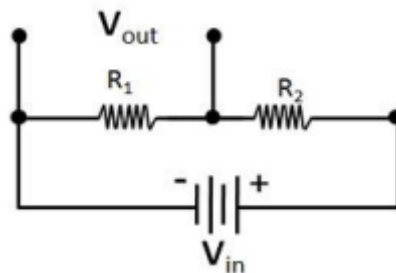
Laporan ini akan menjelaskan tentang pembuatan rangkaian alat sensor getaran menggunakan microcontroller ESP32, Sensor piezoelektrik, sensor geophone, serta beberapa komponen elektronik lain seperti kabel, resistor, potensiometer.

Dalam pembacaan nilai sensornya, digunakan sistem pembagi tegangan, dimana tegangan masuknya bisa diatur besarnya dengan nilai maksimumnya 5V. Kemudian dialirkan ke rangkaian paralel lalu rangkaian paralel tersebut dihubungkan ke sensor getar. Dimana pada masing-masing cabang paralel akan dihubungkan dengan ADC sehingga nilai tegangannya bisa terbaca. Maka nilai yang dicari adalah nilai differensial dari kedua cabang tersebut baik saat terjadi getaran maupun saat tidak terjadi getaran.

Setelah nilai differensialnya telah didapatkan, maka nilai tersebut akan disimpan dalam sebuah file txt di flash memori yang ada dalam ESP32.

A. Pembagi Tegangan

Rangkaian pembagi tegangan merupakan rangkaian elektronika sederhana yang dapat mengubah tegangan besar menjadi tegangan yang lebih kecil dengan jumlah satu output atau lebih di suatu rangkaian elektronik. Bentuk dari rangkaian pembagi tegangan adalah seperti ini :



Gambar 1. Rangkaian pembagi tegangan.

Dari rangkaian tersebut dapat dilihat bahwa tegangan masuk terbagi menjadi dua bagian yang masing-masing sebanding dengan nilai resistor yang dikenai oleh tegangan tersebut. Dari pemahaman tersebut dapat diambil sebuah rumus untuk mengetahui nilai V_{out} yakni sebagai berikut.

$$V_{out} = V_{in} \times (R_2 / (R_1 + R_2))$$

B. Penyimpanan flash memori ESP32

Dalam percobaan ini, library yang digunakan dalam penyimpanan flash adalah SPIFFS. SPIFFS merupakan sistem pengaturan file original dalam yang ideal dalam penyimpanan dalam RAM. Namun untuk saat ini pengoperasiannya tidak lagi mendapat pembaharuan, jadi kemungkinan akan tergantikan dengan sistem file lain seperti LittleFS yang lebih cepat dan support sistem direktori pada umumnya. Kemudian nama dari filenya terbatas di 32 char.

II. Kode Arduino dan Maknanya

```
#include "SPIFFS.h"

int freq = 2048;
int channel = 0;
int resolution = 12;
int PinAO = 12;

void setup() {
    Serial.begin(115200);

    Serial.println("Starting ...");
    if (SPIFFS.begin(true)) {
        Serial.println("SPIFFS mounted!");
    } else {
        Serial.println("Problem to mount the SPIFFS");
    }
    delay(3000);

    ledcSetup(channel, freq, resolution);
    ledcAttachPin(PinAO, channel);
    ledcWriteTone(channel, 2048);
}

void loop() {

    File SPIFFS_File = SPIFFS.open("/test.txt", FILE_WRITE);

    if (SPIFFS_File) {
        Serial.println("File created");
        for(int i = 0; i <= 256; i++){
            int sensorValue= analogRead(33);
            int sensorValue2= analogRead(32);
            float voltage= sensorValue* 1;// (5.0/4095);
            float voltage2= sensorValue2* 1;// (5.0/4095);
            float voltagep= voltage-voltage2;
            Serial.println(voltagep);
            delay(100);

            if (SPIFFS_File.println(voltagep)) {

                Serial.print("Line ");
                Serial.print(i);
                Serial.print("Added to the file.");
                Serial.println();
                delay(100);}
            else {
                Serial.println("Error while adding a line the to the file");
            }
        }
        else {
            Serial.println("Error while creating the file.");
        }

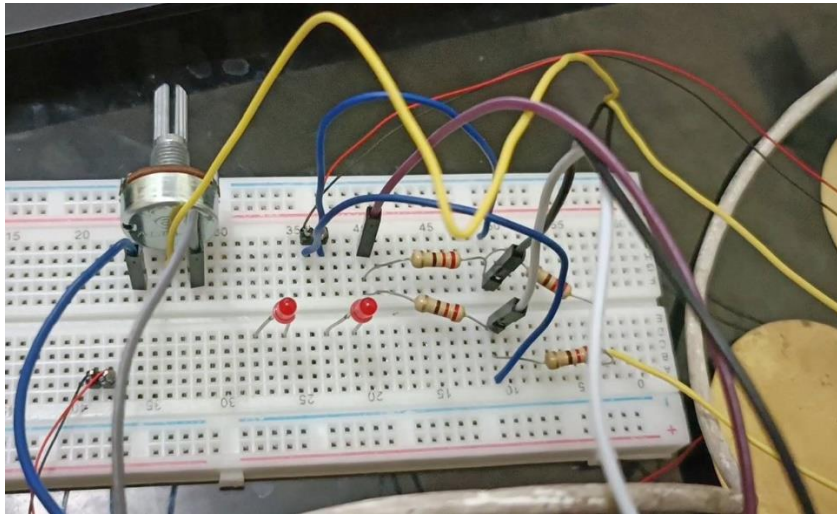
        SPIFFS_File.close();

        Serial.println("Now ... Reading from the file.");
        File SPIFFS_File2 = SPIFFS.open("/test.txt");
        while ((SPIFFS_File2.available())){
            Serial.write((SPIFFS_File2.read()));
            delay(100);
        }
        SPIFFS_File2.close();
    }
```

Penjelasan

- Baris 1 → Memasukan library SPIFFS
- Baris 2-5 → Deklarasi variable yang bertipe integer
- Baris 6 → Memulai fungsi setup
- Baris 7 → Memulai serial dengan 115200 Hz
- Baris 8 → Print serial "starting..."
- Baris 9-10 → Fungsi IF, yaitu jika SPIFFS mulai, maka print "SPIFFS Mounted"
- Baris 11-13 → Fungsi Else, yaitu jika SPIFFSnya tidak mulai, maka print "Problem to mount the SPIFFS."
- Baris 14 → Fungsi delay 3 detik.
- Baris 15 → ledcSetup untuk mengatur konfigurasi awal dalam mengeluarkan output analog, sebagai ganti analogWrite.
- Baris 16 → ledAttachPin untuk mengatur Pin tertentu yang dihubungkan dengan channel tertentu.
- Baris 17 → ledWriteTone untuk mengatur keluaran sinyal analog (tegangan keluar) dan channelnya.
- Baris 18 → Mengakhiri fungsi Setup
- Baris 19 → Mulai Fungsi loop (yang berulang).
- Baris 20 → Deklarasi untuk membuat file melalui SPIFFS dengan fungsi SPIFFS.open(path,mode)
- Baris 21-22 → Saat file berhasil dibuat, maka print "File created" dalam serial monitor.
- Baris 23 → Mengawali fungsi For sampai sekian perulangan pengambilan
- Baris 24-28 → Deklarasi variable yang akan digunakan untuk pembacaan sensor. Yaitu analog read pembagi tegangan 1, analog read pembagi tegangan 2, dan differensial antara keduanya.
- Baris 29 → Print hasil differensial dalam monitor serial
- Baris 30 → Delay 100 ms
- Baris 31-36 → Fungsi If yaitu saat data hasil differensial telah di print dalam file melalui SPIFFS_open, maka print "Line i added to the file".
- Baris 37-39 → Fungsi else 9 (pesan error) apabila data tidak ditulis di file.
- Baris 40 → Akhir dari fungsi else dan for
- Baris 41-43 → Akhir dari fungsi IF pembuatan file dan memulai fungsi else saat file tidak terbuat dan print "error while creating the file" dalam serial monitor.
- Baris 44 → SPIFFS_File.close() untuk menutup filenya
- Baris 45 → Print "Now...Reading from the file." Di serial monitor
- Baris 46 → Membuka file yang telah ditulis tadi
- Baris 47-50 → Memulai fungsi while (selama) filenya available, maka menulis data yang ada dalam file tersebut di serial monitor dan delay 100 ms setiap pembacaannya.
- Baris 51 → Menutup filenya.

III. Foto Rangkaian



Gambar 2. Rangkaian alat.

IV. Hasil serta Analisa



Gambar3. Hasil Respon Piezoelectric.



Gambar 4. Hasil Respon Geophone

```
08:48:12.041 -> Starting ...
08:48:12.135 -> SPIFFS mounted!
08:48:15.186 -> File created
08:48:15.186 -> -466.00
08:48:15.279 -> Line 0 to the file.
08:48:15.373 -> -209.00
08:48:15.467 -> Line 1 to the file.
08:48:15.560 -> -192.00
08:48:15.654 -> Line 2 to the file.
08:48:15.794 -> -214.00
08:48:15.893 -> Line 3 to the file.
08:48:15.980 -> -197.00
08:48:16.074 -> Line 4 to the file.
08:48:16.168 -> -208.00
08:48:16.262 -> Line 5 to the file.
08:48:16.355 -> -221.00
08:48:16.496 -> Line 6 to the file.
08:48:16.590 -> -207.00
08:48:16.683 -> Line 7 to the file.
08:48:16.777 -> -215.00
08:48:16.871 -> Line 8 to the file.
08:48:16.965 -> -208.00
08:48:17.058 -> Line 9 to the file.
08:48:17.199 -> -205.00
08:48:17.293 -> Line 10 to the file.
08:48:17.386 -> Now ... Reading from the file.
08:48:17.386 -> -466.00
08:48:18.276 -> -209.00
08:48:19.165 -> -192.00
08:48:20.102 -> -214.00
```

Gambar 5. Hasil Penyimpanan file dalam SPIFFS

ANALISA

❖ Analisa SPIFFS dan LITTLEFS

Dalam penyimpanan data internal di ESP32 ini saya masih menggunakan Filesystem SPIFFS karena baru mempelajari hal tersebut. Penyimpanan melalui SPIFFS juga cukup mudah dipelajari bagi pemula seperti saya. Namun setelah saya cari tahu lebih lanjut, sebaiknya menggunakan LittleFS lebih menguntungkan. Karena library ini support ke banyak microcontroller, support sistem direktori seperti file manager umumnya dan nama file yang dibuatnya bisa lebih fleksibel karena jumlah karakter di nama filenya tidak dibatasi.

❖ Pembagi Tegangan

Pada dasarnya rangkaian pembagi tegangan atau voltage divider berfungsi untuk memberikan tegangan referensi dari sumber tegangan yang lebih besar nilainya. Rangkaian voltage divider ini akan terasa fungsinya apabila kita merangkai sebuah sensor getar / apapun yang multifungsi atau bisa ditempatkan di beberapa kondisi. Contoh, Saat respon getaran natural tanah sangat kecil amplitudanya, dengan menaikkan tegangan alat kita bisa menaikkan juga amplitude respon getarannya. Kemudian kebalikannya apabila saat merekam data getaran terlalu banyak noise yang terekam, maka kita bisa coba kecilkan tegangannya agar respon getar yang diinginkan (selain noise) dapat terlihat. Hal ini bisa dilakukan dengan mengaplikasikan voltage divider yang dikombinasikan dengan suatu switch pada alat.

❖ Perbedaan Piezoelektrik dan Geophone SM24

Hasil respon getar dari sensor piezoelektrik dan geophone terlihat sangat berbeda. Respon dari getaran yang ditangkap geophone lebih terlihat dan jelas dibanding respon getaran yang ditangkap piezoelektrik. Hal ini mungkin dikarenakan kondisi sensor piezoelektrik yang tidak sempurna. Kemudian terlihat juga bahwa sensitivitas kedua alat ini berbeda.

❖ Potensiometer

Dalam rangkaian ini saya menggunakan potensiometer untuk membantu mengatur tegangan yang masuk dalam ujicoba.

Referensi

<https://arduino-esp8266.readthedocs.io/en/latest/filesystem.html#spiffs-file-system-limitations>

<https://arduino-esp8266.readthedocs.io/en/latest/filesystem.html>