

LAPORAN PRAKTIKUM

TEKNIK ANTARMUKA

MODUL VI

SERIAL KOMUNIKASI ARDUINO DENGAN RASBERRY PI

Nama : Syarif Hidayat

NIM : D400220086

Kelas : A

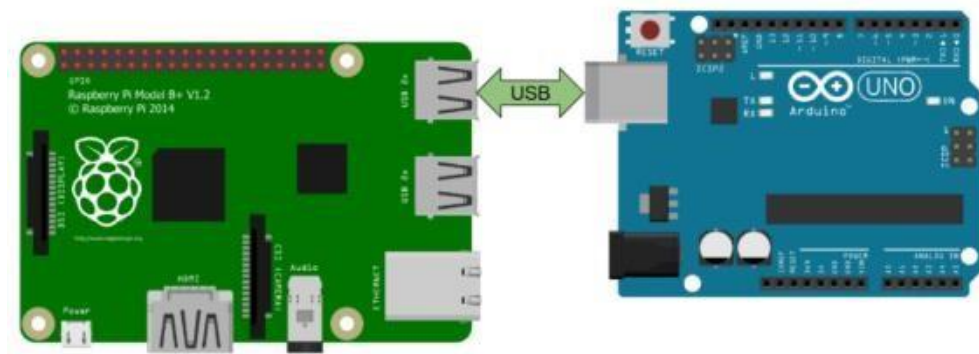
Tanggal Praktikum : 19 – November - 2024

Nilai :

A. Percobaan 1 : Serial Komunikasi Arduino dengan Raspberry Pi

A.1 Hasil Percobaan

A.1.1 Gambar Rangkaian



A.1.2 Script

- Arduino

```
1 void setup() {  
2   Serial.begin(9600); // Inisialisasi komunikasi serial  
3 } // Syarif Hidayat_D400220086  
4 void loop() {  
5   Serial.println("Hello from Arduino!"); // Kirim pesan setiap 1 detik  
6   delay(1000);  
7 }
```

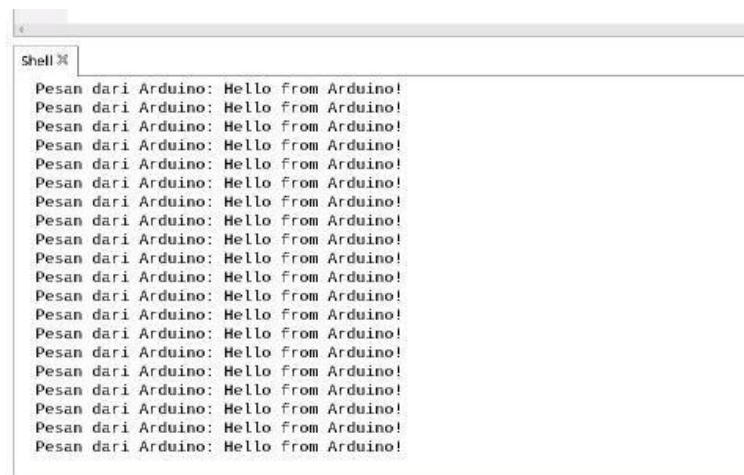
- Thony Python

```

1 #Syarif Hidayat_D400220080
2 import serial
3
4 # Tentukan port serial dan baud rate
5 arduino = serial.Serial('/dev/ttyUSB0', 9600)
6 print("Menunggu data dari Arduino...")
7
8 while True:
9     try:
10 # Membaca data dari Arduino
11
12         data = arduino.readline().decode('utf-8').strip()
13
14         print(f"Pesan dari Arduino: {data}")
15     except KeyboardInterrupt:
16         print("Program dihentikan.")
17         break

```

A.1.3 Foto Hasil Praktikum



```

Shell
Pesan dari Arduino: Hello from Arduino!
Pesan dari Arduino: Hello from Arduino!
Pesan dari Arduino: Hello from Arduino!
Pesan dari Arduino: Hello from Arduino!
Pesan dari Arduino: Hello from Arduino!
Pesan dari Arduino: Hello from Arduino!
Pesan dari Arduino: Hello from Arduino!
Pesan dari Arduino: Hello from Arduino!
Pesan dari Arduino: Hello from Arduino!
Pesan dari Arduino: Hello from Arduino!
Pesan dari Arduino: Hello from Arduino!
Pesan dari Arduino: Hello from Arduino!
Pesan dari Arduino: Hello from Arduino!
Pesan dari Arduino: Hello from Arduino!
Pesan dari Arduino: Hello from Arduino!
Pesan dari Arduino: Hello from Arduino!

```

A.2 Pembahasan A.2.1 Sub-pertanyaan

Bagaimana komunikasi serial memungkinkan Arduino mengirim data string ke Raspberry Pi, dan apa pengaruh baud rate terhadap keberhasilan komunikasi? Jelaskan bagaimana Python membaca data dari serial port dan mengatasi jika data yang diterima tidak sesuai format.

A.2.2 Analisis

Pada Percobaan Pertama di Modul ke 6 Yaitu tentang komunikasi satu arah dari arduino ke Raspy dengan menggunakan kabel usb untuk pengiriman data di progam arduino cukup mudah dikarenakan kita seperti menampilkan nilai di serial monitor untuk syarat komunikasi ini adalah dengan mensamakan baud rate pengiriman dan penerimaan data.Untuk di Scrip Python menggunakan bahasa mikro python dengan aturan pemrograman arduino berbeda jauh dikarenakan batas fungsi di arduino menggunakan kurung kurawal sedangkan di thony menggunakan Tab baris , lalu kita juga

membutuhkan library serial untuk membaca data dan menerima data yang dikirim lalu kita baca lewat port raspy yang biasanya bernama /dev/ttyUSB0 atau sesuai dengan tipe raspy yang digunakan.

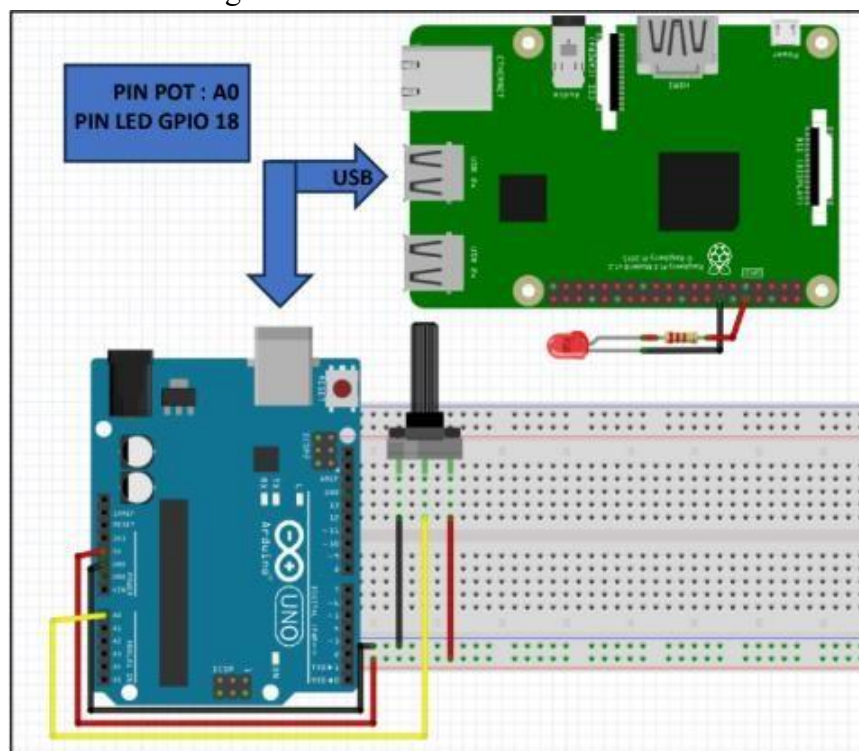
Unutk cara kerja program di arduino yaitu mengirim string ke raspy lalu di raspy akan menerima string tersebut lalu string itu di ubah agar bisa ditampilkan di raspy dengan perintah utf-8 , raspy akan membaca data tersebut sampai ada baris baru baru dia akan berhenti membaca dan menampilkan hasil pembacaan ini akan terus berulang terus sampai tombol interupt di tekan maka program akan berhenti.

Jika Baud rate anantara raspy dan arduino tidak sama maka data yang dikirim akan tidak sama dengan data yang dikirim atau data akan rusak.di raspy data akan dibaca perbaris samapi ada baris baru jika data yang dikirim tidak sesuai maka tidak akan bisa ditampilkan walaupun bisa nanti tampilan data akan eror.

B. Percobaan 2 : Serial Komunikasi Satu Arah Arduino dengan Raspberry Pi

B.1 Hasil Percobaan

B.1.1 Gambar Rangkaian



B.1.2 Script

- Arduino

- Thony Python

```
#Syarif Hidayat_D400220086
from gpiozero import PWMLED
import serial
import time

# Inisialisasi LED menggunakan GPIO Zero
led = PWMLED(18) # Pin GPIO 18 digunakan untuk LED

# Inisialisasi komunikasi serial dengan Arduino
arduino = serial.Serial('/dev/ttyUSB0', 9600) #
Ganti '/dev/ttyUSB0' sesuai port Arduino Anda

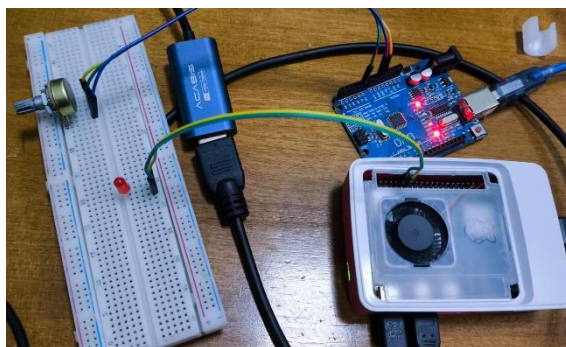
try:
    while True:# Membaca data dari Arduino
        data = arduino.readline().decode('utf-8').strip()
        pot_value = int(data) # Konversi data string ke integer
        # Mengatur kecerahan LED (0.0 - 1.0) berdasarkan nilai potensiometer (0-1023)
        brightness = pot_value / 1023
        led.value = brightness # Set kecerahan LED

        print(f"Potensiometer: {pot_value},Kecerahan LED: {brightness:.2f}")

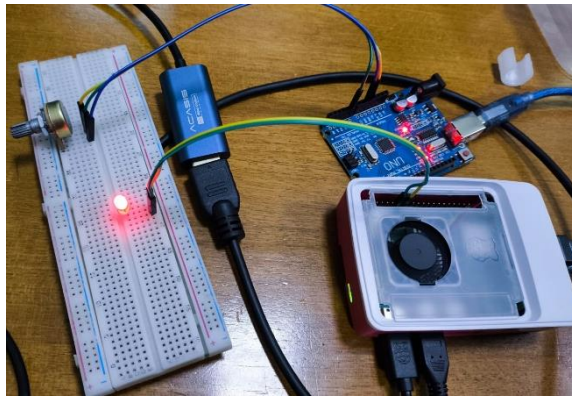
        time.sleep(0.1) # Tunggu sebelum membacaulang
except KeyboardInterrupt:
    print("Program dihentikan.")
finally:
    led.off() # Matikan LED saat program selesai
```

B.1.3 Foto Hasil Praktikum

- Mati

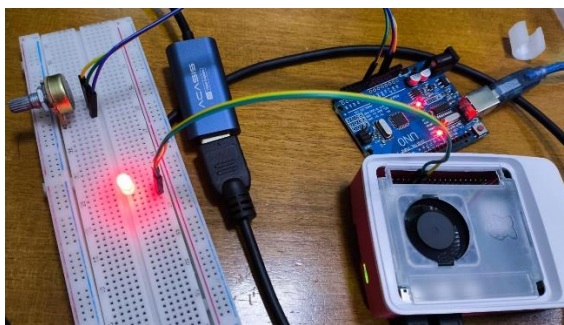
[illegible]

- Redup



```
Shell
Potensiometer: 215, Kecerahan LED: 0.21
Potensiometer: 215, Kecerahan LED: 0.21
Potensiometer: 214, Kecerahan LED: 0.21
Potensiometer: 216, Kecerahan LED: 0.21
Potensiometer: 215, Kecerahan LED: 0.21
Potensiometer: 215, Kecerahan LED: 0.21
Potensiometer: 215, Kecerahan LED: 0.21
Potensiometer: 215, Kecerahan LED: 0.21
Potensiometer: 215, Kecerahan LED: 0.21
Potensiometer: 215, Kecerahan LED: 0.21
Potensiometer: 215, Kecerahan LED: 0.21
Potensiometer: 215, Kecerahan LED: 0.21
Potensiometer: 215, Kecerahan LED: 0.21
Potensiometer: 215, Kecerahan LED: 0.21
Potensiometer: 215, Kecerahan LED: 0.21
Potensiometer: 215, Kecerahan LED: 0.21
Potensiometer: 215, Kecerahan LED: 0.21
Potensiometer: 215, Kecerahan LED: 0.21
Potensiometer: 215, Kecerahan LED: 0.21
Potensiometer: 215, Kecerahan LED: 0.21
Potensiometer: 215, Kecerahan LED: 0.21
Potensiometer: 215, Kecerahan LED: 0.21
Potensiometer: 215, Kecerahan LED: 0.21
```

- Terang



```
Shell
Potensiometer: 1022, Kecerahan LED: 1.00
Potensiometer: 1023, Kecerahan LED: 1.00
Potensiometer: 1023, Kecerahan LED: 1.00
Potensiometer: 1022, Kecerahan LED: 1.00
Potensiometer: 1023, Kecerahan LED: 1.00
Potensiometer: 1023, Kecerahan LED: 1.00
Potensiometer: 1022, Kecerahan LED: 1.00
Potensiometer: 1023, Kecerahan LED: 1.00
Potensiometer: 1023, Kecerahan LED: 1.00
Potensiometer: 1022, Kecerahan LED: 1.00
Potensiometer: 1023, Kecerahan LED: 1.00
Potensiometer: 1022, Kecerahan LED: 1.00
Potensiometer: 1023, Kecerahan LED: 1.00
Potensiometer: 1022, Kecerahan LED: 1.00
Potensiometer: 1023, Kecerahan LED: 1.00
Potensiometer: 1023, Kecerahan LED: 1.00
Potensiometer: 1022, Kecerahan LED: 1.00
Potensiometer: 1023, Kecerahan LED: 1.00
Potensiometer: 1023, Kecerahan LED: 1.00
Potensiometer: 1022, Kecerahan LED: 1.00
Potensiometer: 1023, Kecerahan LED: 1.00
Potensiometer: 1023, Kecerahan LED: 1.00
```

B.2 Pembahasan B.2.1 Sub-pertanyaan

Bagaimana nilai analog dari potensiometer pada Arduino diubah menjadi data digital, dikirim ke Raspberry Pi, dan digunakan untuk mengatur kecerahan LED? Jelaskan potensi masalah seperti noise pada pembacaan potensiometer dan cara menjaga kestabilan kecerahan LED.

B.2.2 Analisis

Pada Percobaan Ke 2 Yaitu hampir sama dengan percobaan pertama di percobaan yang kedua ini mengontrol lampu dengan potensio Arduino digunakan untuk membaca nilai potensio lalu nilai hasil pembacaannya di kirim ke raspy lalu raspy akan membaca nilai yang dikirim dan menghidupkan lampu sesuai nilai yang dikirim.

Di scrip arduino hanya ada penambahan untuk pembacaan nilai potensio di script thony kita mengambil fungsi PWMLED dari library gpiozero lalu ada library serial untuk menerima data yang dikirim dan library time untuk menggunakan time slep atau seperti delay.

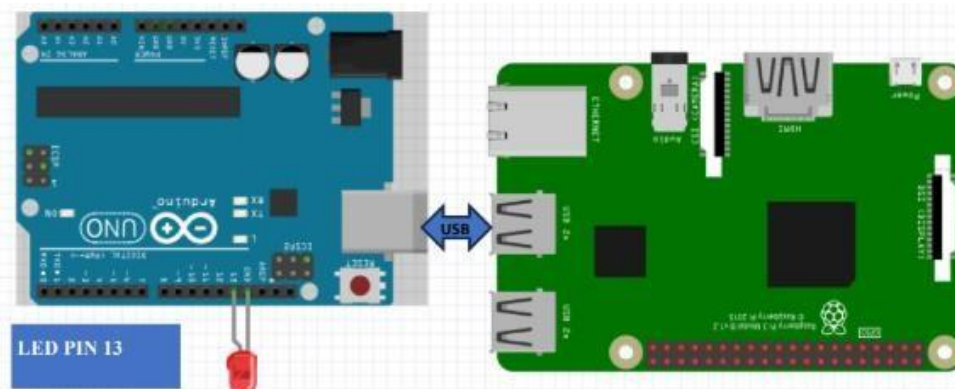
Nilai yang sudah diterima akan dikonvert ke integer setelah menjadi integer maka akan dibagi dengan 1023 agar nilai yang dikirim untuk menghidupkan led antara nilai 0.0 sampai 1.0.

Kenapa Nilai Yang awalnya 0 – 1023 ke 0.0 – 1.0 dikarenakan rasberry.pi tidak bisa membaca nilai analog 0 – 1023 oleh karena itu rasberiy menggunakan rentang nilai 0.0 – 1.0 untuk nilai yang dapat mengontrol nilai PWM lampu.Untuk Mengaktifkan Lampu di Rasbery pi kita menggunakan gpiozero dengan mengambil fungsi PWM led untuk mengontrol lampu .Fungsi yang berada pada fungsi try akan dieksekusi terus menerus program akan berhenti ketika tombol interrupt ditekan dikarenakan fungsi except keybord interrupt maka akan menampilkan program dihentikan setelah itu masuk ke finally atau selesai untuk mematikan lampu.

C. Percobaan 3 : Serial Komunikasi Dua Arah Arduino dengan Raspberry Pi

C.1 Hasil Percobaan

C.1.1 Gambar Rangkaian



C.1.2 Script

- Arduino

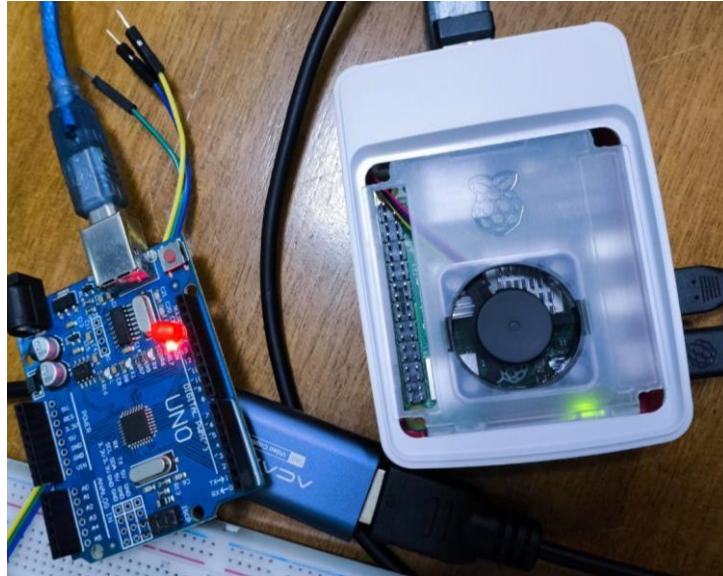
```
1 void setup() {
2     Serial.begin(9600);
3     pinMode(13, OUTPUT);
4 } // Syarif Hidayat_D400220086
5 void loop() {
6     if (Serial.available() > 0) {
7         String command = Serial.readString();
8         if (command == "ON\n") {
9             digitalWrite(13, HIGH); // Nyalakan LED
10            Serial.println("LED ON");
11        } else if (command == "OFF\n") {
12            digitalWrite(13, LOW); // Matikan LED
13            Serial.println("LED OFF");
14        } else { // Syarif Hidayat_D400220086
15            Serial.println("Perintah tidak dikenali");
16        }
17    }
18 }
```

- Thony Python

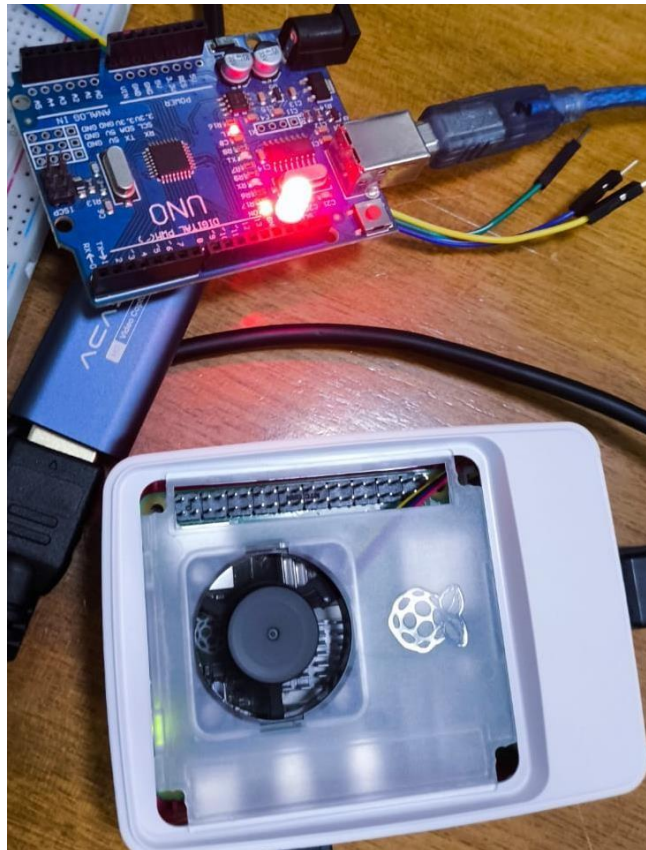
```
1 #Mahesa Arya_D400220087
2 import serial
3
4 arduino = serial.Serial('/dev/ttyUSB0', 9600)
5 while True:
6     command = input("Enter command (ON/OFF): ")
7     arduino.write(command.encode())
8
9     response = arduino.readline().decode('utf-8').strip()
10
11     print(f"Arduino Response: {response}")
```

C.1.3 Foto Hasil Praktikum

- Mati



- Hidup



C.2 Pembahasan C.2.1 Sub-pertanyaan

Bagaimana Arduino memproses perintah "ON" atau "OFF" dari Raspberry Pi, dan bagaimana Python memastikan input perintah valid? Analisis potensi konflik saat kedua perangkat mengirim data bersamaan dan cara menjaga sinkronisasi.

C.2.2 Analisis

Pada Percobaan 3 Yaitu tentang komunikasi dua Arah yaitu antara raspy dengan arduino , cara kerja arduino disini untuk membaca karakter string yang dikirim dari Rasberypi Setelah Menerima Data String, Arduino akan memberikan Feddback Sesuai dengan String Yang Dikirimkan , feddback berupa data string yang nantinya akan di tampilkan di shel raspy.

jika string yang terbaca adalah ON maka Menghidupkan Lampu lalu mengirim fedback ke raspy dengan string LED ON yang nantinya akan ditampilkan di shel Raspy. jika string yang terbaca adalah OFF maka mematikan Lampu lalu mengirim fedback ke raspy dengan string LED OFF yang nantinya akan ditampilkan di shel Raspy. jika string yang terbaca adalah Selain ON / OFF maka Memberikan fedback ke raspy dengan string Perintah Tiidak Dikenali yang nantinya akan ditampilkan di shel Raspy.

D. Foto Kelompok

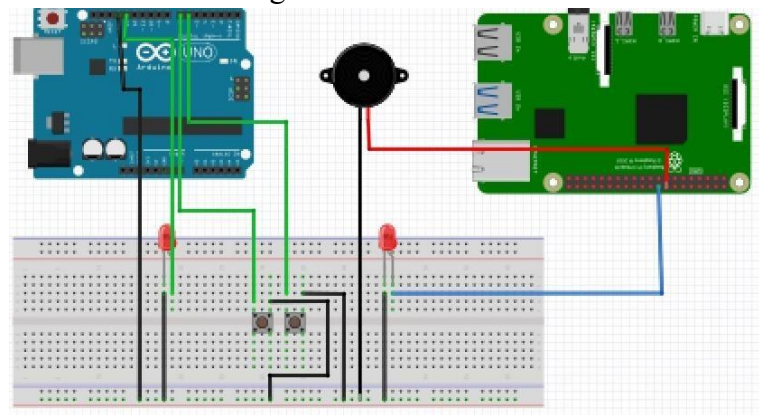


E. Pertanyaan

Buatlah program komunikasi dua arah antara Arduino dan Raspberry Pi, di mana pada Arduino terdapat 2 tombol dan 1 LED, sedangkan pada Raspberry Pi terdapat 1 LED dan 1 buzzer. Tombol pertama pada Arduino digunakan untuk mengontrol LED di Raspberry Pi, dan tombol kedua untuk mengontrol buzzer di Raspberry Pi. Sementara itu, LED pada Arduino dikendalikan tingkat kecerahannya oleh Raspberry Pi melalui komunikasi serial. Jika tugas sudah jadi, silahkan direkam dan dijelaskan secara program, cara kerja, dan hasil.

Jawab :

- Gambar Rangkaian



- Script
- Arduino

```

1 const int Tombol_1 = 7;
2 const int Tombol_2 = 6;
3 const int PInLED = 13;
4 void setup() { // Syarif Hidayat_D400220086
5     Serial.begin(9600);
6     pinMode(Tombol_1, INPUT_PULLUP);
7     pinMode(Tombol_2, INPUT_PULLUP);
8     pinMode(PInLED, OUTPUT);
9 } // Syarif Hidayat_D400220086
10 void loop() {
11     if (Serial.available() > 0) {
12         int pwm_value = Serial.parseInt();
13         if (pwm_value >= 0 && pwm_value <= 255) {
14             analogWrite(PInLED, pwm_value);
15             Serial.println(pwm_value);
16         } else { // Syarif Hidayat_D400220086
17             Serial.println("Invalid PWM value");
18         }
19     } // Syarif Hidayat_D400220086
20     if (digitalRead(Tombol_1) == LOW && digitalRead(Tombol_2) == HIGH) {
21         Serial.println("2");
22     } // Syarif Hidayat_D400220086
23     else if (digitalRead(Tombol_2) == LOW && digitalRead(Tombol_1) == HIGH) {
24         Serial.println("4");
25     } // Syarif Hidayat_D400220086
26     else if (digitalRead(Tombol_1) == HIGH && digitalRead(Tombol_2) == HIGH) {
27         Serial.println("1");
28     } // Syarif Hidayat_D400220086
29     delay(500); // Syarif Hidayat_D400220086
30 }

```

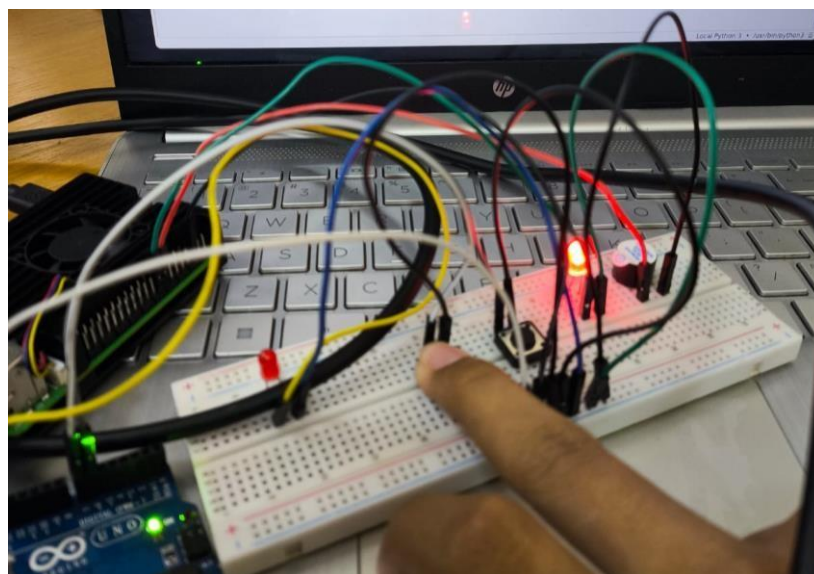
- Thony Python

```

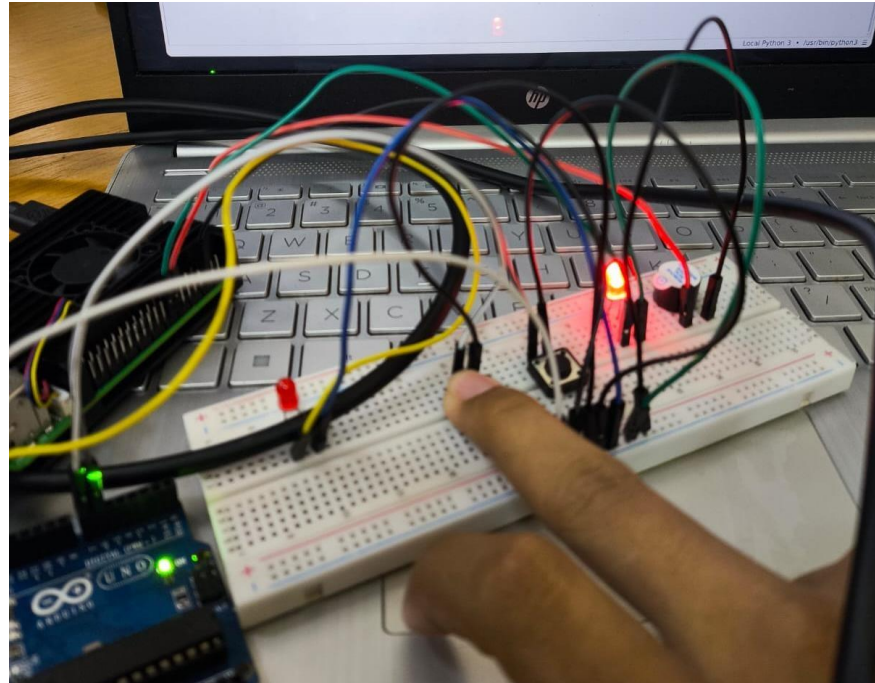
1 #Syarif Hidayat_D400220086
2 from gpiozero import PWMLED
3 import serial
4 import time
5 import threading
6
7 led = PWMLED(24)
8 Buzzer = PWMLED(23)
9 arduino = serial.Serial('/dev/ttyACM0', 9600, timeout=1) # Timeout added to avoid blocking
10
11 def menerima_data():
12     while True:
13         # Menerima data dari Arduino
14         data = arduino.readline().decode('utf-8').strip()
15         if data == "1":
16             led.value = 0
17             Buzzer.value = 0
18         elif data == "2":
19             led.value = 1
20         elif data == "4":
21             Buzzer.value = 1
22
23 def mengirim_data():
24     while True:
25         # Mengirim data dari Raspberry Pi ke Arduino
26         DataRaspy = input("Enter PWM value for LED (0-255): ")
27         if DataRaspy.isdigit():
28             pwm_value = int(DataRaspy)
29             if 0 <= pwm_value <= 255:
30                 arduino.write(str(pwm_value).encode()) # Send the PWM value to Arduino
31                 time.sleep(0.1) # Small delay to allow Arduino to process
32             else:
33                 print("Masukkan nilai antara 0 sampai 255.")
34         else:
35             print("masukkan nilai yang tepat antara 0 - 255.")
36
37 # Membuat thread untuk menerima dan mengirim data
38 menerima_thread = threading.Thread(target=menerima_data)
39 mengirim_thread = threading.Thread(target=mengirim_data)
40
41 # Memulai thread
42 menerima_thread.start()
43 mengirim_thread.start()
44
45 # Menunggu kedua thread selesai (tidak akan pernah selesai dalam kasus ini)
46 menerima_thread.join()
47 mengirim_thread.join()

```

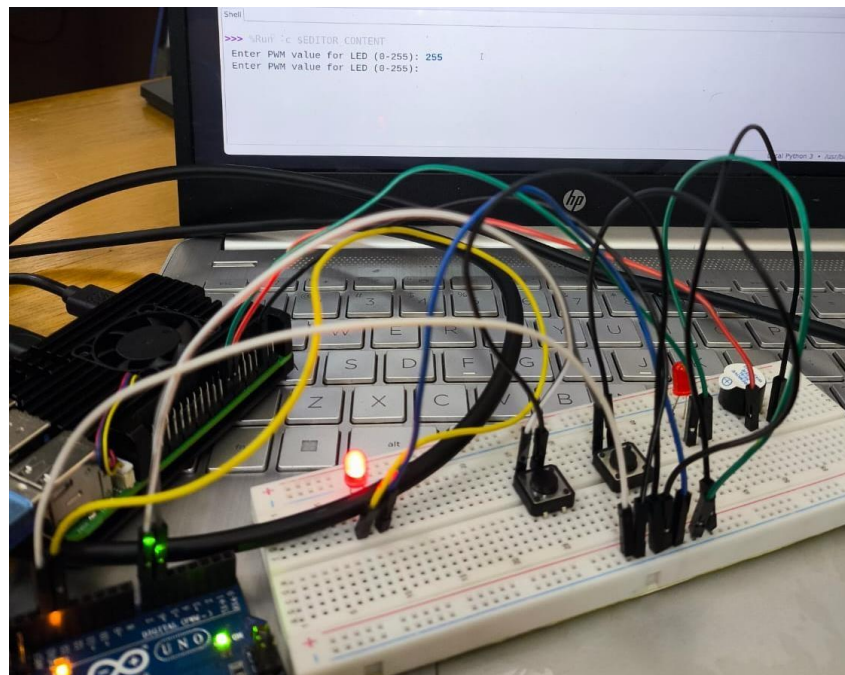
- Hasil
- Button 1 Arduino ditekan



- Button 2 Arduino ditekan



- Kontrol kecerahan LED Arduino



F. Kesimpulan

- Pada Praktikum Kali ini Bertujuan Untuk Komunikasi Antara Arduino Dan Rasbery Pi Menggunakan Kabel USB.
- Jika Ingin Mengirim Dari arduino ke rasbery tinggal gunakan perintah `Serial.println` , In digunakan untuk menghentikan pembacaan raspy.
- Jika ingin Mengirim Dari Raspy ke Arduino tinggal menggunakan library dan gunakan fungsi `arduino.write`.
- Jika Ingin Menerima Data Dari Raspy ke arduino kita harus mengecek dulu dengan `serial.available` jika ada data di serial baru masuk ke fungsi `Serial.readString` maka akan membaca data string tersebut.
- Jika Ingin Membaca Data Dari Arduino Ke Raspy kita harus menggunakan fungsi `arduino.readline`.

Telah diperiksa oleh,

Muhammad Abdurrohlim

NIM. D400200088

(.....)

Dedy Ary Prasetya.S.T.,M.Eng

NIDN. 615117504

(.....)