

LAPORAN PRAKTIKUM

TEKNIK ANTARMUKA

MODUL I

SERIAL KOMUNIKASI PADA ARDUINO DAN SPI (*SERIAL PERIPHERAL INTERFACE*)

Nama : Syarif Hidayat

NIM : D400220086

Kelas : D

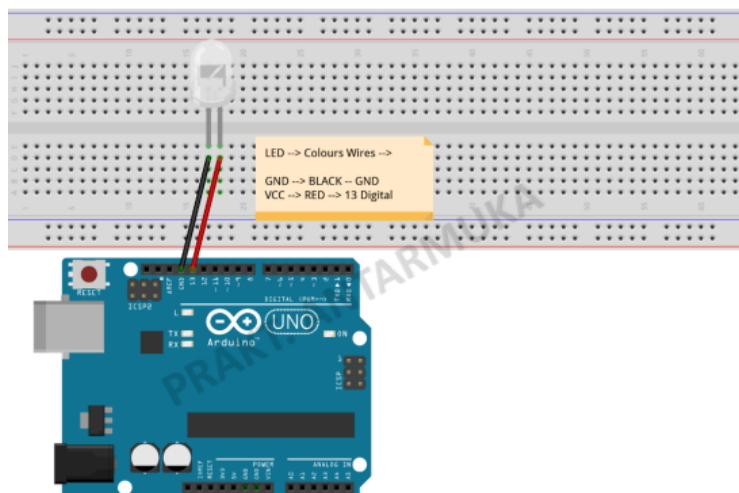
Tanggal Praktikum : 1 Oktober 2024

Nilai :

A. Percobaan 1 : Serial Komunikasi Arduino Dengan PC/Komputer

A.1 Hasil Percobaan

A.1.1 Gambar Rangkaian



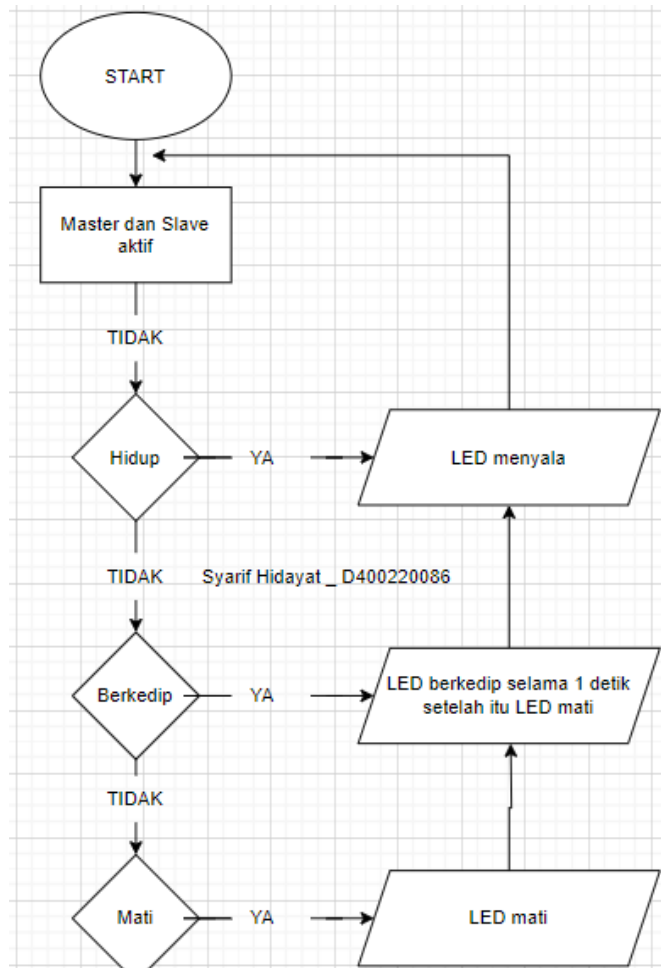
A.1.2 Script

```

1 // Syarif Hidayat_D400220086_Kelas A //
2 const int led = 13;
3 String code;
4 void setup() {
5     Serial.begin(9600);
6     pinMode(led, OUTPUT);
7 }
8 void loop() {
9     if (Serial.available()) {
10         code = Serial.readStringUntil('\n');
11         code.trim();
12     } if (code == "hidup") {
13         digitalWrite(led, HIGH);
14         Serial.println("LED HIDUP");
15         code = "";
16     }
17     else if (code == "mati") {
18         digitalWrite(led, LOW);
19         Serial.println("LED MATI");
20         code = "";
21     }
22     // Syarif Hidayat_D400220086_Kelas A //
23     else if (code == "kedip") {
24         Serial.println("LED BERKEDIP");
25         while (true) {
26             digitalWrite(led, HIGH);
27             delay(1000);
28             digitalWrite(led, LOW);
29             delay(1000);
30             if (Serial.available()) {
31                 code = Serial.readStringUntil('\n');
32                 code.trim();
33                 break;
34             }
35         }
36     }
37     delay(100);
38 }

```

A.1.3 Flowchart



A.1.4 Output



A.2 Pembahasan

A.2.1 Pertanyaan

Bagaimana lampu LED pada percobaan ini dapat menyala sesuai dengan perintah yang diberikan. Jelaskan secara rinci!

A.2.2 Analisis

Pada Percobaan kali ini menggunakan Serial komunikasi yang terdapat pada laptop / PC. Pertama mendeklarasikan pada pin 13 dengan menggunakan tipe data const int yang merupakan kombinasi 2 buah tipe data. Const merupakan tipe data yang nilainya tidak dapat diubah – ubah dan int adalah tipe data yang hanya bisa berisi bilangan bulat saja . kemudian mendeklarasikan variabel code dengan menggunakan tipe data dan string yang digunakan untuk menampung teks atau sebuah karakter, variabel code ini nantinya akan digunakan untuk menampung data yang diterima dari serial monitor dari arduino . Serial.begin(9600) merupakan komunikasi Serial dengan perangkat luar dengan menggunakan USB, begin(9600) menginisialisasi komunikasi dengan serial yang memiliki kecepatan pembacaan yaitu 9600 bit per detik. pinMode merupakan fungsi untuk mengatur mode pin, pada pinMode mode nya dapat diatur sebagai

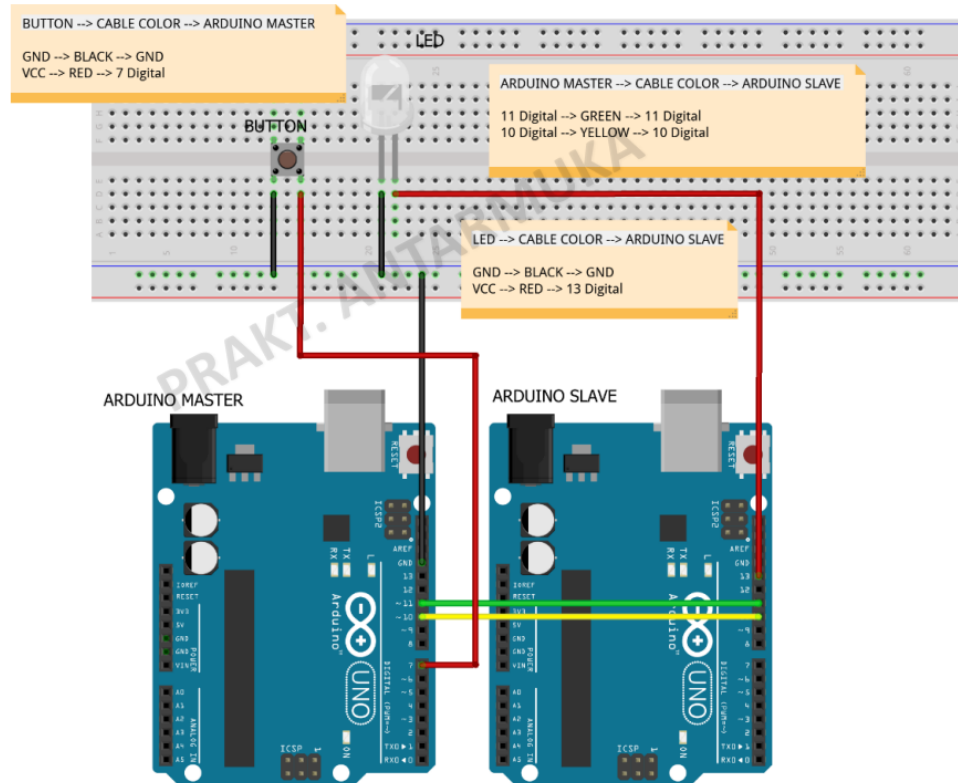
INPUT , INPUT_PULLUP,OUTPUT dll , tapi pada percobaan kali pinMode diatur sebagai OUTPUT berarti pin ini digunakan sebagai pin keluaran atau tidak bisa digunakan sebagai penerima data inputan. Kemudian Serial.available() digunakan untuk memeriksa apakah ada data yang terbaca dari Serial komunikasi arduino atau tidak. Serial.readStringUntil('\n') digunakan untuk membaca isi serial monitor lalu \n untuk berpindah baris , lalu hasil pembacaan dari Serial.readStringUntil akan disimpan di variable code yang bertipe data string. Kemudian melakukan pengkodisian dengan menggunakan syntax if jika data yang terbaca adalah "hidup" maka LED akan menyala dengan menggunakan syntax digitalWrite, merupakan syntax yang digunakan untuk mengeksekusi semua pin digital bisa digunakan untuk LOW dan juga HIGH jika if pada syntax menerima data "mati" maka LED akan mati dan jika if pada syntax data yang diterima adalah "kedip" maka LED akan menyala selama 1 detik kemudian LED akan mati. Hasil kerja pada program ketika sudah diupload dan kemudian membukan serial monitor pada Arduino dan mengirimkan kata dari serial monitor maka program akan langsung mendeteksi kata apa yang diterima dari serial monitor dan kata yang diterima akan ditampung pada syntax Serial.readStringUntil. Jika kata yang diterima dari serial monitor

adalah "hidup" maka LED akan hidup dan akan menampilkan tulisan LED HIDUP juga pada serial monitor jika kata yang diterima dari serial monitor adalah "mati" maka LED akan mati dan menampilkan tulisan LED MATI pada serial monitor dan jika kata yang diterima dari serial monitor adalah kata "kedip" maka LED akan menyala selama 1 detik dan kemudian LED akan mati dan juga LED akan menampilkan tulisan LED BERKEDIP pada serial monitor. Kemudian program secara otomatis akan berjalan berulang-ulang atau melooping sampai hardware dimatikan.

B. Percobaan 2 : Serial Komunikasi Satu Arah Arduino Dengan Arduino

B.1 Hasil Percobaan

B.1.1 Gambar Rangkaian



B.1.2 Script

- Master

```
1 // Syarif Hidayat_D400220086_Kelas A //
2 #include <SoftwareSerial.h>
3 const int button = 7;
4 int code;
5 SoftwareSerial Txserial(11, 10);
6 void setup() {
7   pinMode(7, INPUT_PULLUP);
8   Txserial.begin(9600);
9   Serial.begin(9600);
10 }
11 void loop() {
12   code = digitalRead(button);
13   if (code == LOW) {
14     Txserial.write("1");
15     Serial.println("LED HIDUP");
16   } else if (code == HIGH) {
17     Txserial.write("0");
18     Serial.println("LED MATI");
19   }
20 // Syarif Hidayat_D400220086_Kelas A //
21   delay(500);
22 }
```

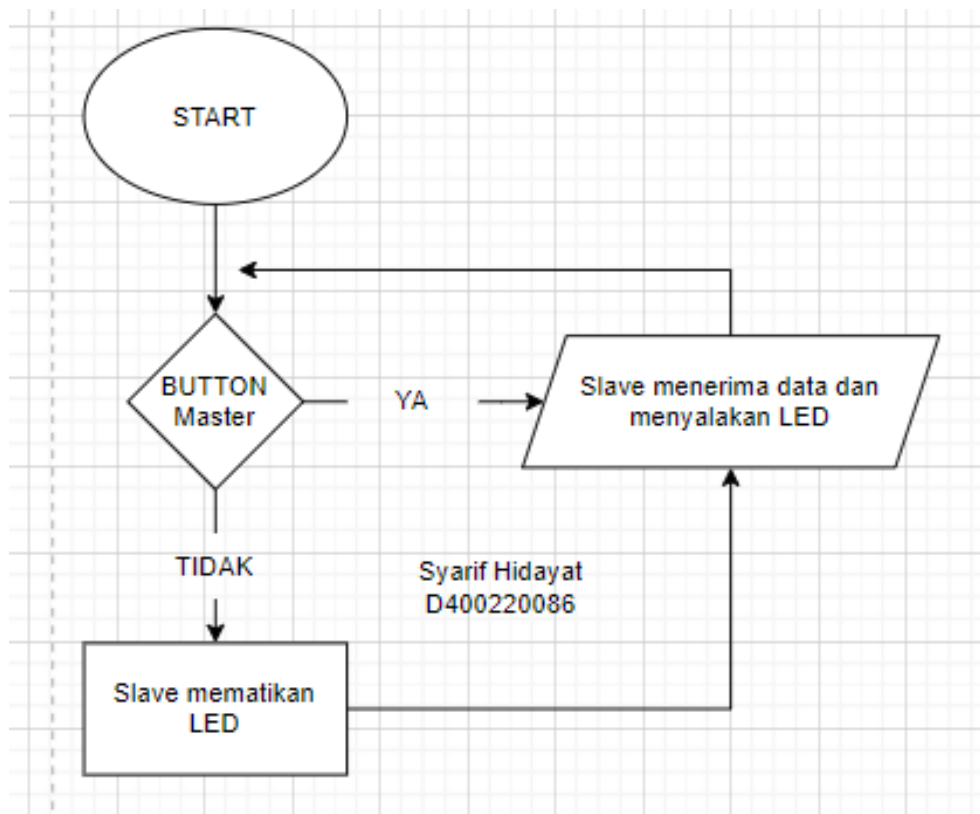
- Slave

```

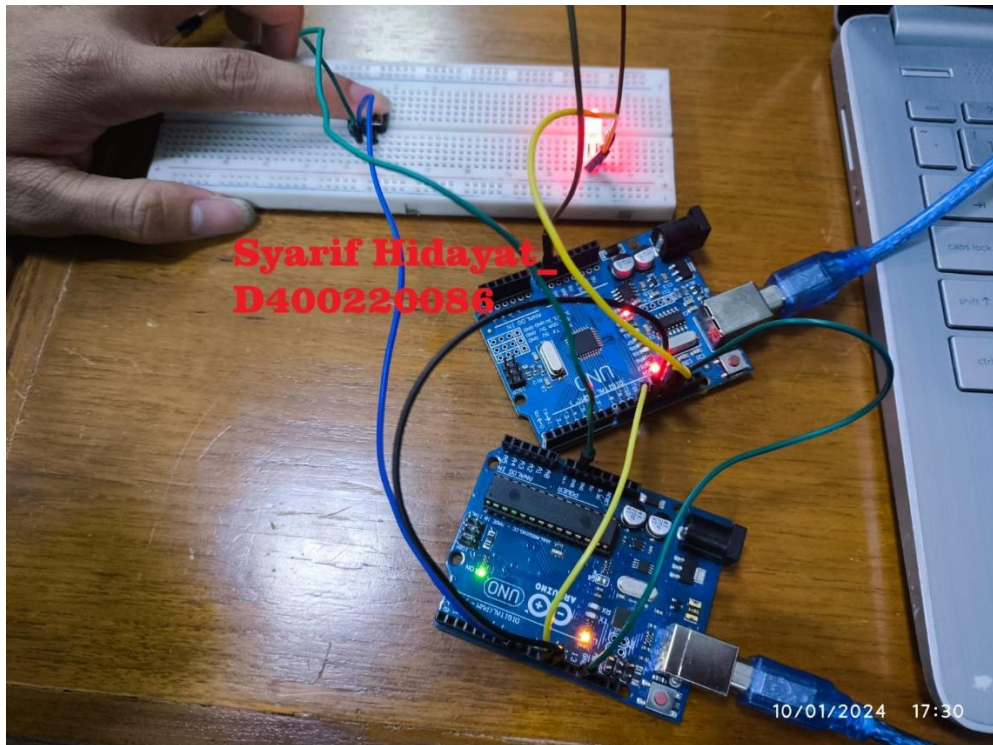
1 // Syarif Hidayat_D400220086_Kelas A //
2 #include <SoftwareSerial.h>
3 SoftwareSerial Rxserial(10, 11);
4 char val;
5 const int led = 13;
6 void setup() {
7     Rxserial.begin(9600);
8     pinMode(13, OUTPUT);
9 }
10 void loop() {
11     if (Rxserial.available() > 0) {
12         val = Rxserial.read();
13     }
14     if (val == '1') {
15         Serial.println(val);
16         digitalWrite(led, HIGH);
17     }
18     // Syarif Hidayat_D400220086_Kelas A //
19     else if (val == '0') {
20         Serial.println(val);
21         digitalWrite(led, LOW);
22     }
23 }

```

B.1.3 Flowchart



B.1.4 Output



B.2 Pembahasan

B.2.1 Pertanyaan

Apa fungsi dari INPUT_PULLUP pada percobaan ini, apa bedanya dengan deklarasi INPUT biasa? Jelaskan!

B.2.2 Analisis

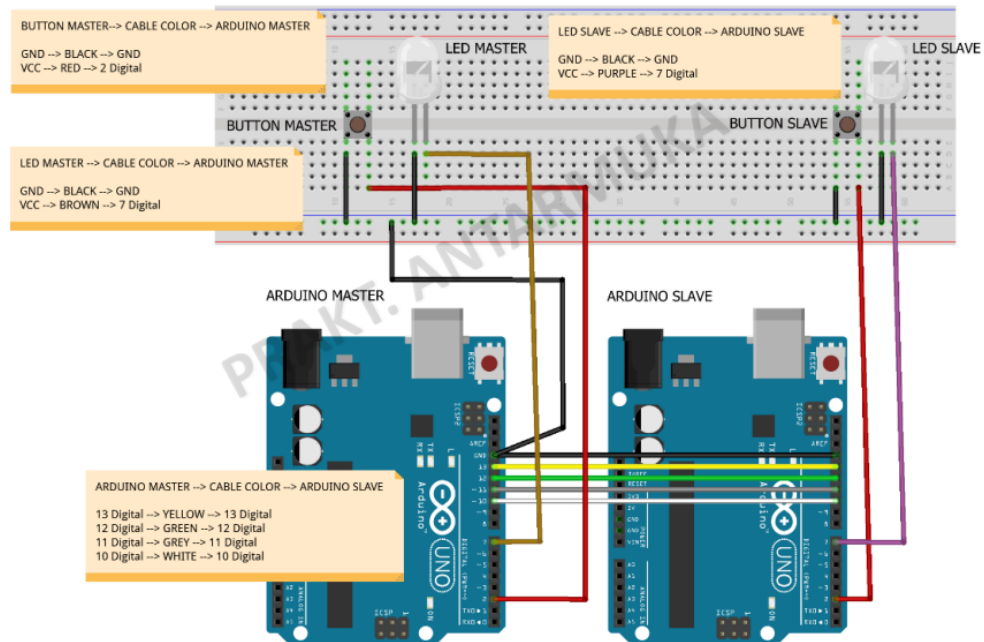
Pada percobaan kali ini melakukan percobaan serial komunikasi satu arah dengan menggunakan 2 buah Arduino yang nantinya salah satu Arduino(master) ini akan mengirim data ke arduino satunya(slave) dengan cara menghubungkan rx Arduino master ke tx Arduino slave dan tx Arduino master ke rx Arduino slave dan pada percobaan ini tentunya menggunakan 2 buah program yaitu master dan slave. Pada kedua program juga menggunakan library yang sama yaitu SoftwareSerial yang digunakan membuat pin digital pada Arduino yang tidak mendukung komunikasi serial agar bisa digunakan serial komunikasi antar Arduino. Pada program Arduino master mendeklarasikan Const int, Const merupakan tipe data yang nilainya tidak dapat diubah-ubah dan int adalah tipe data yang hanya bisa digunakan untuk bilangan bulat saja kemudian juga mendeklarasikan

variabel code dengan menggunakan tipe data int yang merupakan bilangan bulat. Kemudian memanggil library SoftwareSerial yang di definisikan dengan nama Txserial yang memiliki parameter pin 11 sebagai tx dan pin 10 sebagai pin rx. Kemudian pinMode dengan mode INPUT_PULLUP digunakan untuk mendefinikan pin yang digunakan dengan pin yang digunakan sebagai INPUT PULLUP yang ketika tidak ada sinyal yang masuk maka pin ini akan bernilai HIGH dan jika ada sinyal ground atau logika rendah disambungkan kepin maka pin ini akan bernilai LOW. Perbedaannya dengan INPUT biasa jika INPUT biasa itu menggunakan resistor pull_up atau pull_down jika pin tidak ada sinyal yang masuk maka akan bernilai LOW dan jika ada sinyal yang masuk maka pin ini akan bernilai HIGH dengan menggunakan resistor ini maka pin ini dalam keadaan pasti. Txserial.begin(9600) digunakan untuk komunikasi serial rx tx dengan kecepatan 9600 bit per detik dan Serial.begin(9600) digunakan untuk menampilkan tulisan pada serial monitor dengan kecepatan 9600 bit per detik. Pada void loop digunakan untuk perulangan berkali-kali sampai sistem dimatikan. digitalRead digunakan untuk membaca nilai dari pin digital, nilai yang diterima dari pin digital akan ditampung pada variabel code jika sudah diterima pada variabel code maka akan melakukan pengkondisian jika tombol atau data yang diterima adalah LOW atau ground tersambung maka akan mengirimkan data dengan menggunakan Txserial. write dan ("1") merupakan objek data yang dikirimkan dan juga akan menampilkan LED HIDUP dan jika tombol ditekan maka akan mengirimkan data objek "0" dan juga menampilkan tulisan LED MATI pada serial monitor dan pada akhir program diberikan delay 500 milisecond untuk menjeda programnya. Pada program slave digunakan untuk menerima data yang diberikan dari Arduini masternya jika ada data yang diterima sesuai dengan program slave maka LED akan langsung di eksekusi jika data yang diterima adalah karakter "1" maka LED akan menyala dan jika data yang diterima ada karakter "0" maka LED akan mati.

C. Percobaan 3 : Serial Komunikasi Dua Arah Arduino Dengan Arduino Menggunakan SPI

C.1 Hasil Percobaan

C.1.1 Gambar Rangkaian



C.1.2 Script

- Master

```
1 // Syarif Hidayat_D400220086_Kelas A //
2 #include <SPI.h>
3 #define LED 7
4 #define ipbutton 2
5 int buttonvalue;
6 int x;
7 void setup(void) {
8   Serial.begin(9600);
9   pinMode(ipbutton, INPUT_PULLUP);
10  pinMode(LED, OUTPUT);
11  SPI.begin();
12  SPI.setClockDivider(SPI_CLOCK_DIV8);
13  digitalWrite(SS, HIGH);
14 }
15
16 void loop(void) {
17   byte Mastersend, Mastereceive;
18   buttonvalue = digitalRead(ipbutton);
19   if (buttonvalue == LOW)
20
21   {
22     x = 1;
23   }
24   else {
25     x = 0;
26   }
27   digitalWrite(SS, LOW);
28   Mastersend = x;
29   Mastereceive = SPI.transfer(Mastersend);
30   // Syarif Hidayat_D400220086_Kelas A //
31   if (Mastereceive == 1)
32   {
33     digitalWrite(LED, HIGH);
34     Serial.println("Master LED ON");
35   }
36   else {
37     digitalWrite(LED, LOW);
38     Serial.println("Master LED OFF");
39   }
40   delay(1000);
41 }
```

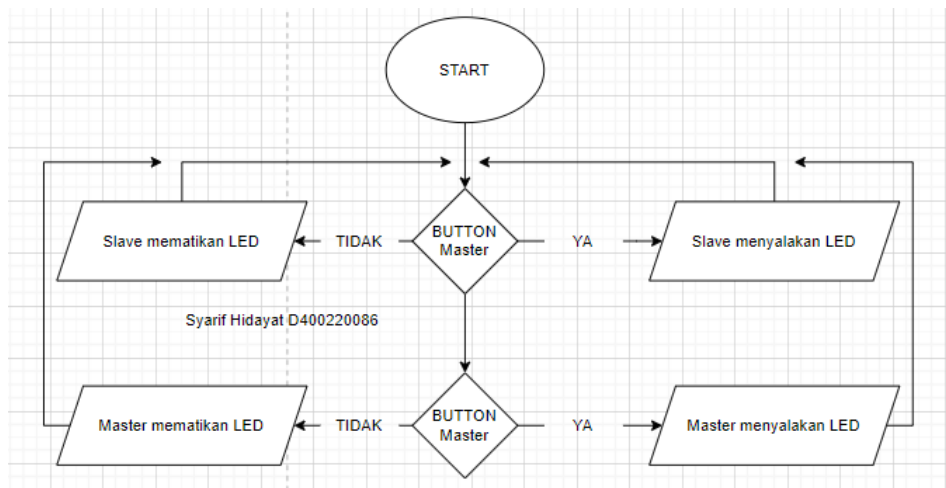
- Slave

```

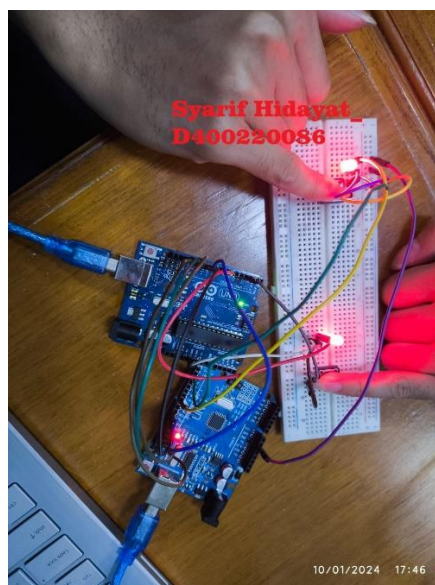
1 // Syarif Hidayat_D400220086_Kelas A //
2 #include <SPI.h>
3 #define LEDpin 7
4 #define buttonpin 2
5 volatile boolean received;
6 volatile byte Slaverceived, Slavesend;
7 int buttonvalue;
8 int x;
9 void setup() {
10   Serial.begin(9600);
11   pinMode(buttonpin, INPUT_PULLUP);
12   pinMode(LEDpin, OUTPUT);
13   pinMode(MISO, OUTPUT);
14   SPCR |= _BV(SPE);
15   received = false;
16   SPI.attachInterrupt();
17 }
18 ISR(SPI_STC_vect)
19 {
20   Slaverceived = SPDR;
21   received = true;
22 }
23 void loop() {
24   if (received) {
25     {
26       if (Slaverceived == 1) {
27         digitalWrite(LEDpin, HIGH);
28         Serial.println("Slave LED ON");
29       } else {
30         digitalWrite(LEDpin, LOW);
31         Serial.println("Slave LED OFF");
32       }
33     }
34     // Syarif Hidayat_D400220086_Kelas A //
35     buttonvalue = digitalRead(buttonpin);
36     if (buttonvalue == LOW)
37     {
38       x = 1;
39     } else {
40       x = 0;
41     }
42     Slavesend = x;
43     SPDR = Slavesend;
44     delay(1000);
45   }
46 }

```

C.1.3 Flowchart



C.1.4 Output



C.2 Pembahasan

C.2.1 Pertanyaan

Bagaimana cara transaksi data pada arduino ketika menggunakan mode SPI (*Serial Peripheral Interface*)? Jika mengutip dari internet ataupun jurnal, berikan sitasi pada akhir jawaban.

C.2.2 Analisis

Percobaan kali ini serial komunikasi dua arah dengan menggunakan SPI, 2 buah Arduino, 2 buah LED dan 2 buah button. Pada kali ini menggunakan library SPI yang digunakan untuk serial komunikasi 2 arah, (SPI.begin) digunakan untuk memulai komunikasi SPI, `SPI.setClockDivider(SPI_CLOCK_DIV8)` digunakan untuk mengatur kecepatan pembacaan SPI, `digitalWrite(SS, HIGH);` Mengaktifkan pin SS (Slave Select) sebagai HIGH. Jika sinyal yang diterima dari button LOW maka akan menyimpan data pada variabel $x = 1$ dan jika tidak LOW maka akan menyimpan data pada variabel $x = 0$. `digitalWrite(SS, LOW)` digunakan pin SS menjadi LOW, menandakan awalan komunikasi SPI dengan slave. `SPI.transfer(Mastersend)` merupakan metode yang digunakan untuk mengirim data melalui SP secara bersamaan dan menerima response dari slave kemudian menyimpannya di `Masterreceive`, metode ini digunakan untuk mengirim data melalui jalur MOSI dan secara bersamaan menerima data dari jalur MISO setelah mengirim data metode ini akan mengembalikan data yang diterima dari slave. Kemudian memeriksa data yang diterima dari slave jika data yang diterima dari slave adalah 1 maka akan menghidupkan LED dan menampilkan tulisan Master LED On jika data yang diterima dari slave bukan 1 maka LED tidak akan menyala dan menampilkan tulisan Master LED Off pada akhir program akan memberikan delay selama 1 detik. Pada program slave menggunakan `SPCR |= _BV(SPE)` yang merupakan operasi byte yang digunakan untuk mengaktifkan modul SPInya dan digunakan untuk mengatur kecepatan transmisi, interupsi dll, BV digunakan untuk menghasilkan nilai biner yang berada pada posisi bit yang ditentukan, SPE merupakan konstanta bit yang menyala pada modul SPI. `BV(SPE)` menghasilkan bilangan biner

dengan bit ke-6 yang diaktifkan (1) dan bit non-aktif (0). `SPI.attachInterrupt()` digunakan untuk mengaktifkan fungsi interrupt ketika menerima data dari SPI. `ISR(SPI_STC_vect)` fungsi ini nantinya akan dipanggil ketika ada data yang diterima dari SPI kemudian data yang diterima akan disimpan pada variabel `Slavereceived` dan variabel `received` diubah menjadi true menandakan ada data yang diterima, kemudian mengecek jika ada yang diterima dan data yang diterima dari master itu adalah 1 maka akan menyalakan LED dan jika data yang diterima dari master bukan 1 maka LED tidak akan menyala. Pada pengecekan tombol berikutnya digunakan untuk mengirimkan data tombol pada slave ke master yang nantinya digunakan untuk menyalakan LED pada master dengan cara menyimpan data pada variabel `x`, data yang sudah disimpan pada variabel `x` ditampung lagi pada variabel `Slavesend` kemudian data yang sudah disimpan akan di kirim ke master melalui SPI dan melakukan pertukaran data.

D. Foto Kelompok



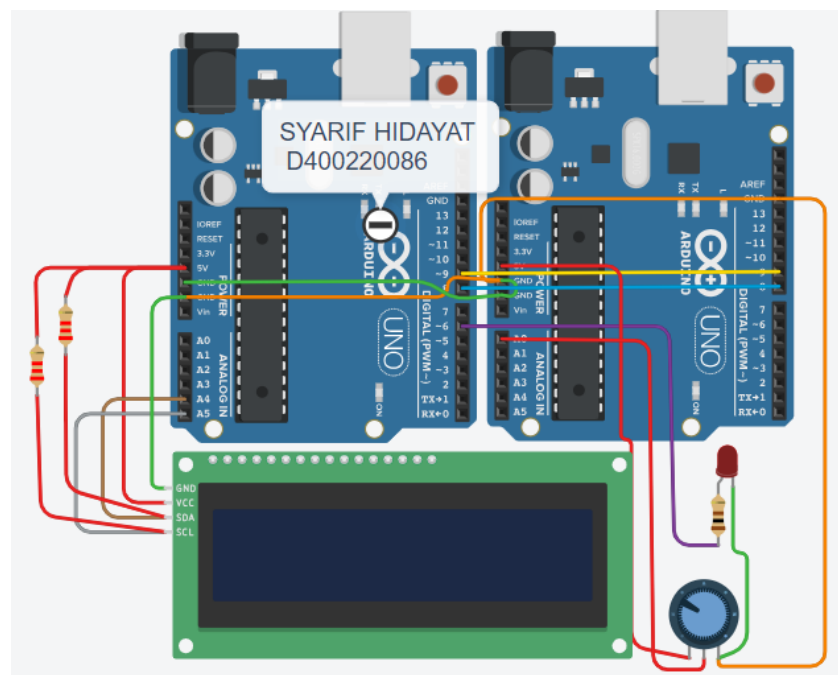
E. Pertanyaan

1. Buatlah sebuah sistem menggunakan koneksi Rx dan Tx Arduino dengan Arduino dengan ketentuan sebagai berikut :
 - Gunakan Potensiometer sebagai input dan LED sebagai output.
 - Arduino Master digunakan sebagai input dari Potensiometer.

- Arduino Slave digunakan sebagai output dari LED sekaligus menampilkan nilai potensiometer yang diterima ke LCD I2C.
- Ketika nilai dari Potensiometer mencapai $1, x \text{ k}\Omega$, maka lampu LED pada arduino slave akan menyala. Nilai dari x merupakan angka digit terakhir NIM.
- Nilai dari Potensiometer dapat dilihat pada serial monitor.
- Pastikan nama dari variable berbeda dari tiap praktikan, jika ada kesamaan dari keseluruhan nama variable, maka akan mengacu pada pengurangan point laporan.
- Dalam pengerjaan bisa menggunakan simulasi ataupun hardware secara langsung.

Jawab :

- Gambar Rangkaian



- Script

- Master

```

1  #include <SoftwareSerial.h>
2  const int PuterMemet = A0;
3  int SV_memet;
4  SoftwareSerial Txserial(9, 8);
5  // Syarif Hidayat - D400220086
6  void setup() {
7      pinMode(PuterMemet, INPUT);
8      Txserial.begin(9600);
9      Serial.begin(9600);
10 }
11
12 void loop() {
13     Baca();
14     delay(150);
15 }
16 // Syarif Hidayat - D400220086
17 void Baca() {
18     SV_memet = analogRead(PuterMemet);
19     Txserial.print(SV_memet);
20     Txserial.print("\n");
21     Serial.print("SV_memet = ");
22     Serial.println(SV_memet);
23 }
24

```

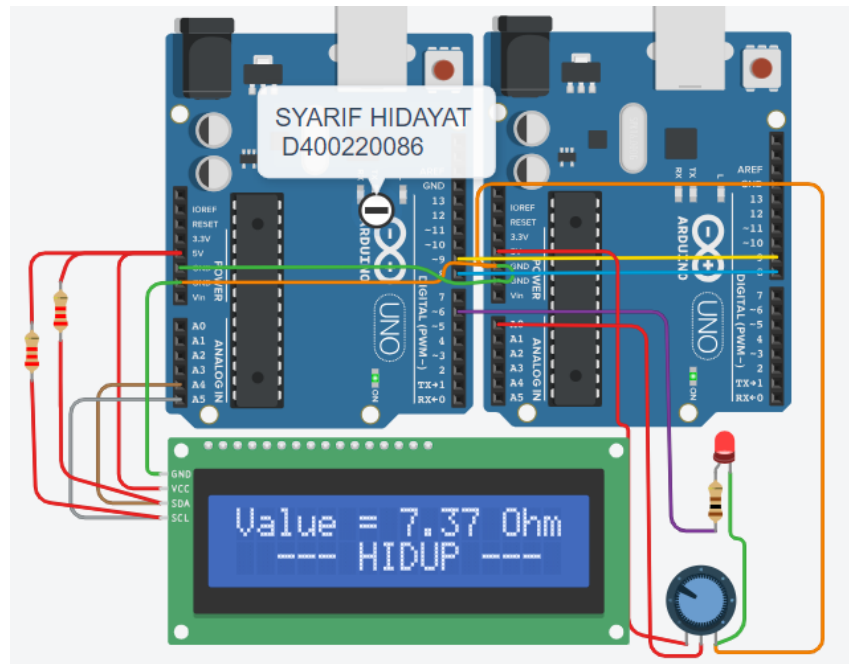
- Slave

```

1  #include <SoftwareSerial.h>
2  #include <Wire.h>
3  #include <LiquidCrystal_I2C.h>
4
5  SoftwareSerial Rxserial(8, 9);
6  LiquidCrystal_I2C Memet(0x27, 16, 2);
7
8  int MemetDiterima;
9  float Convert_Memet;
10 const int LED = 6;
11 // Syarif Hidayat - D400220086
12 void setup() {
13     Rxserial.begin(9600);
14     Serial.begin(9600);
15     pinMode(LED, OUTPUT);
16
17     Memet.init();
18     Memet.backlight();
19     Memet.clear();
20     Memet.setCursor(0, 0);
21     Memet.print("-----DONE-----");
22 }
23 void loop() {
24     Terima_memet();
25     delay(300);
26     if (Convert_Memet >= 1.6) {
27         digitalWrite(LED, HIGH);
28         Memet.clear();
29         Memet.setCursor(0, 0);
30         Memet.print("Con_Mmt = ");
31
32         Memet.setCursor(10, 0);
33         Memet.print(Convert_Memet);
34         Memet.setCursor(2, 1);
35         Memet.print("---- HIDUP ----");
36     }
37     else {
38         digitalWrite(LED, LOW);
39         Memet.clear();
40         Memet.setCursor(0, 0);
41         Memet.print("Con_Mmt = ");
42         Memet.setCursor(10, 0);
43         Memet.print(Convert_Memet);
44         Memet.setCursor(2, 1);
45         Memet.print("---- MATI ----");
46     }
47 } // Syarif Hidayat - D400220086
48 void Terima_memet() {
49     if (Rxserial.available()) {
50         String received = Rxserial.readStringUntil('\n');
51         MemetDiterima = received.toInt();
52         Serial.print("Data diterima: ");
53         Serial.print(MemetDiterima);
54         Serial.print(" Convert = ");
55         Convert_Memet = MemetDiterima / 100.0;
56         Serial.println(Convert_Memet);
57     }
58 }

```

- Hasil



LED ON

```
Data diterima: 737    Convert = 7.37
--- HIDUP ---
Data diterima: 737    Convert = 7.37
--- HIDUP ---
Data diterima: 737    Convert = 7.37
--- HIDUP ---
Data diterima: 737    Convert = 7.37
--- HIDUP ---
Data diterima: 737    Convert = 7.37
--- HIDUP ---
Data diterima: 737    Convert = 7.37
--- HIDUP ---
Data diterima: 737    Convert = 7.37
--- HIDUP ---
```



LED OFF

```
Data diterima: 21     Convert = 0.21
--- MATI ---
Data diterima: 21     Convert = 0.21
--- MATI ---
Data diterima: 21     Convert = 0.21
--- MATI ---
Data diterima: 21     Convert = 0.21
--- MATI ---
Data diterima: 21     Convert = 0.21
--- MATI ---
Data diterima: 21     Convert = 0.21
--- MATI ---
Data diterima: 21     Convert = 0.21
--- MATI ---
```



F. Kesimpulan

- Pada praktikum kali ini mempelajari serial komunikasi satu arah dan serial komunikasi dua arah.
- Pada praktikum kali ini serial komunikasi satu arah yang menggunakan RX dan TX harus menggunakan baud rate yang sama

pada kedua program jika pada master menggunakan baudrate 9600 maka pada slave juga harus menggunakan 9600

- Serial komunikasi 2 arah dengan menggunakan SPI pada program harus mencantumkan library SPI dan pada hardware yang digunakan ada 4 pin yang harus disambungkan yaitu MISO, MOSI, SS dan SCK
- Serial.komunikasi satu arah yang menggunakan RX dan TX, jika menggunakan pin digital yang tidak mendukung serial komunikasi harus menggunakan library softwareserial agar pin digital yang tidak bisa digunakan komonukasi bisa digunakan untuk komunikasi
- Perbandingan dari semua serial komunikasi, SPI memiliki respon yang cepat dari pada serial komunikasi yang lainnya.
- Komunikasi SPI menggunakan empat jalur utama yaitu MOSI (untuk mengirim data dari master), MISO (untuk menerima data dari slave), SCLK (clock yang mengatur sinkronisasi), dan SS (untuk memilih slave yang akan berkomunikasi)
- SPI adalah komunikasi full-duplex, yang berarti master dan slave dapat mengirim dan menerima data secara bersamaan dalam satu siklus clock.

Telah diperiksa oleh,

Muhammad Abdurrohman

NIM. D400210088

Dedi Ary Prasetya S.T.,M.Eng

NIDN. 615117504

\

(.....)

(.....)