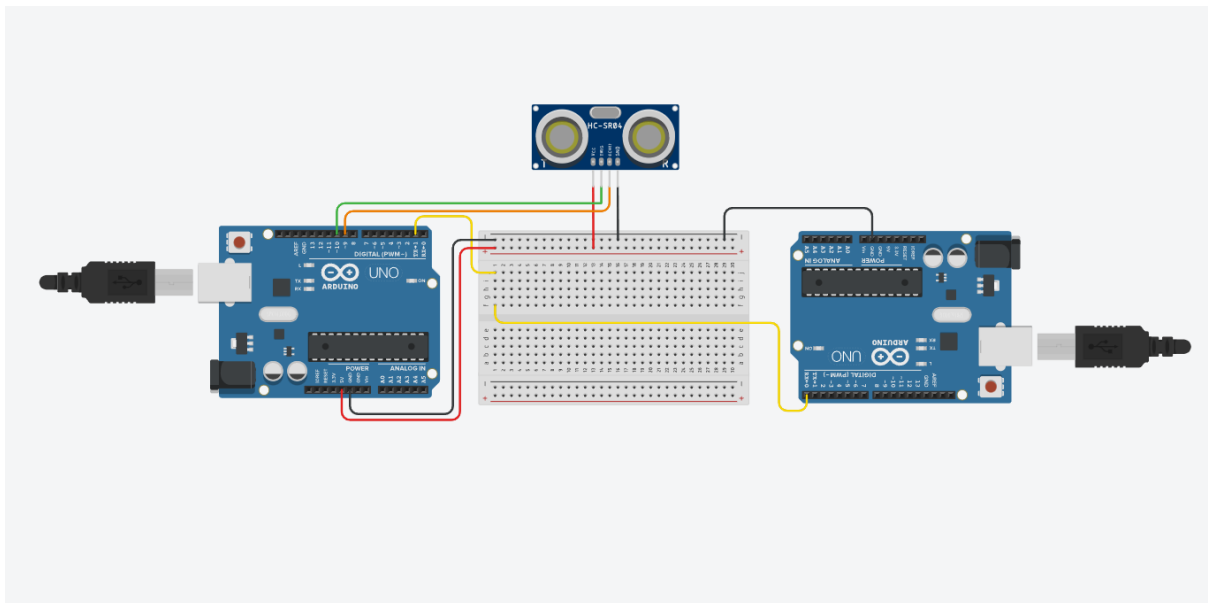


Pengiriman Hasil Jarak

Untuk memperluas pemahaman tentang materi sebelumnya, kita akan membahas tentang penggunaan Komunikasi Serial dan Sensor Ultrasonik dalam konteks praktis. Kali ini, kita akan mengirimkan hasil pengukuran jarak yang diperoleh dari sensor ultrasonik yang terpasang pada Arduino Transfer ke Arduino Receiver. Ini berarti bahwa Arduino Transfer akan mengukur jarak menggunakan sensor ultrasonik, kemudian hasilnya akan dikirimkan ke Arduino Receiver.

Dengan cara ini, kita dapat memanfaatkan data jarak yang terukur untuk mengontrol perangkat lain, misalnya, untuk menggerakkan servo motor berdasarkan jarak yang diukur. Dengan demikian, kita dapat melihat bagaimana dua Arduino dapat bekerja bersama-sama dengan menggunakan Komunikasi Serial dan Sensor Ultrasonik untuk menciptakan aplikasi yang lebih kompleks dan bermanfaat. Penjelasan Kode Arduino Transfer:



Kode Arduino Transfer :

```
String kirim;
const int TRIGPIN = 10;
const int ECHOPIN = 9;
long timer;
int jarak;

void setup()
{
    Serial.begin(9600);
    pinMode(ECHOPIN, INPUT);
    pinMode(TRIGPIN, OUTPUT);
}

void loop()
{
    String hasil = String(distance());
    Serial.print(hasil);
    Serial.println("");
    delay(1000);
    if (Serial.available()) {
        kirim = Serial.readString();
        Serial.print(kirim);
        Serial.println("");
        delay(100);
    }
}

int distance() {
    digitalWrite(TRIGPIN, LOW);
    delayMicroseconds(2);
    digitalWrite(TRIGPIN, HIGH);
    delayMicroseconds(10);
    digitalWrite(TRIGPIN, LOW);

    timer = pulseIn(ECHOPIN, HIGH);
    jarak = 0.0343 * (timer / 2);
    delay(1000);

    return jarak;
}
```

Penjelasan :

1. Deklarasi Variabel dan Konstanta :

Dilakukan deklarasi variabel dan konstanta yang diperlukan untuk program. Variabel yang dideklarasikan antara lain ` kirim ` (untuk menyimpan data yang akan dikirim melalui komunikasi serial), ` TRIGPIN ` dan ` ECHOPIN ` (untuk menentukan pin yang digunakan untuk sensor ultrasonik), ` timer ` (untuk menyimpan waktu yang diukur oleh sensor ultrasonik), dan ` jarak ` (untuk menyimpan hasil jarak yang diukur).

2. Setup :

Pada fungsi setup(), dilakukan inisialisasi komunikasi serial dengan kecepatan 9600 baud rate menggunakan ` Serial.begin(9600) `. Selain itu, pin ECHOPIN diatur sebagai input (untuk menerima sinyal ultrasonik) dan pin TRIGPIN diatur sebagai output (untuk mengirimkan sinyal triger ke sensor ultrasonik).

3. Loop :

Pada fungsi loop(), dilakukan pengukuran jarak menggunakan sensor ultrasonik dengan memanggil fungsi ` distance() `. Hasil pengukuran tersebut dikonversi menjadi String menggunakan ` String(hasil) ` dan dikirimkan melalui komunikasi serial dengan ` Serial.print(hasil) ` dan ` Serial.println("") ` (untuk memberikan baris baru setelah setiap data). Terdapat juga pengecekan apakah ada data yang tersedia di serial dengan ` Serial.available() `. Jika ada data yang tersedia, data tersebut dibaca menggunakan ` Serial.readString() ` dan kemudian diprint ke serial monitor.

4. Fungsi distance() :

Fungsi ini bertugas untuk melakukan pengukuran jarak menggunakan sensor ultrasonik. Pertama-tama, sinyal triger dikirimkan ke sensor ultrasonik dengan mengatur pin TRIGPIN menjadi LOW, kemudian HIGH, dan kembali ke LOW. Waktu yang diperlukan untuk sinyal echo kembali ke sensor diukur menggunakan ` pulseIn() ` dan disimpan dalam variabel ` timer `. Selanjutnya, waktu yang diukur dikonversi menjadi jarak dalam satuan sentimeter dan kemudian dikembalikan sebagai nilai integer.

Kode Arduino Receiver:

```
String terima;

void setup()
{
    Serial.begin(9600);
    Serial.println("Arduino Reciver");
}

void loop()
{
    if (Serial.available()) {
        terima = Serial.readString();
        Serial.println(terima);
        delay(100);
    }
}
```

Penjelasan :

1. Setup :

Pada kode tersebut, dalam `setup()`, inisialisasi komunikasi serial dilakukan dengan memanggil fungsi `Serial.begin(9600)`. Ini menetapkan kecepatan baudrate (9600 bits per detik) untuk komunikasi serial. Selain itu, menggunakan fungsi `Serial.println()`, pesan "Arduino Receiver" akan dicetak ke Serial Monitor, memberi tahu pengguna bahwa perangkat Arduino telah siap sebagai penerima.

2. Loop :

Dalam `loop()`, perangkat akan terus memeriksa apakah data tersedia untuk dibaca di port serial menggunakan fungsi `Serial.available()`. Jika data tersedia untuk dibaca, maka perangkat akan membaca string yang diterima menggunakan fungsi `Serial.readString()`, kemudian menyimpannya dalam variabel `terima`. Selanjutnya, string yang diterima akan dicetak ke Serial Monitor menggunakan fungsi `Serial.println()`. `Delay(100)` digunakan untuk memberikan jeda sejenak setelah mencetak data, memastikan tampilan di Serial Monitor dapat terbaca dengan baik.