

Komunikasi UART

Komunikasi serial UART (Universal Asynchronous Receiver/Transmitter) pada Arduino Uno melibatkan penggunaan dua pin, yaitu TX (Transmit) dan RX (Receive). Arduino Uno memiliki satu port serial UART bawaan yang dapat digunakan untuk berkomunikasi dengan perangkat lain, seperti sensor, modul, atau perangkat mikrokontroler lainnya.

Seperti pada gambar rangkaian, terdapat dua buah Arduino Uno, Dimana Arduino yang menghubungkan Pin Tx (Transfer) ke Rx (Receiver) disebut Arduino Transfer, sedangkan Arduino yang menghubungkan pin Rx (Receiver) ke Pin Tx (Transfer) disebut Arduino Receiver.

Arduino Pengirim (Transfer):

```
String kirim;

void setup() {
    Serial.begin(9600);
}

void loop() {
    if (Serial.available()) {
        kirim = Serial.readString(); // Membaca string dari port serial
        Serial.print(kirim); // Mencetak string yang diterima ke Serial
        Monitor
        Serial.println(""); // Mencetak newline untuk pemisah
        delay(100); // Memberi sedikit waktu untuk memproses data
    }
}
```

Penjelasan:

1. `Serial.begin(9600);`: Menginisialisasi komunikasi serial dengan baud rate 9600 bps.
2. `if (Serial.available());`: Memeriksa apakah ada data yang tersedia di port serial.
3. `kirim = Serial.readString();`: Membaca string yang dikirim dari Arduino penerima.
4. `Serial.print(kirim);`: Mencetak string yang diterima ke Serial Monitor.
5. `Serial.println("");`: Mencetak newline untuk pemisah antar data.
6. `delay(100);`: Memberikan sedikit waktu untuk memproses data sebelum melanjutkan ke iterasi selanjutnya.

Arduino Penerima (Receiver):

```
String terima;

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

void loop() {
  if (Serial.available()) {
    terima = Serial.readString(); // Membaca string dari port serial
    Serial.println(terima); // Mencetak string yang diterima ke Serial
    // Monitor
    delay(100); // Memberi sedikit waktu untuk memproses data
  }
}
```

Penjelasan:

1. `Serial.begin(9600);`: Menginisialisasi komunikasi serial dengan baud rate 9600 bps.
2. `Serial.println("Arduino Receiver");`: Mencetak pesan di Serial Monitor untuk menandai bahwa Arduino berada dalam mode penerima.
3. `if (Serial.available());`: Memeriksa apakah ada data yang tersedia di port serial.
4. `terima = Serial.readString();`: Membaca string yang dikirim dari Arduino pengirim.
5. `Serial.println(terima);`: Mencetak string yang diterima ke Serial Monitor.
6. `delay(100);`: Memberikan sedikit waktu untuk memproses data sebelum melanjutkan ke iterasi selanjutnya.