

Komunikasi kabel menggunakan Ethernet, CAN & RS485

7.1 Tujuan Praktikum

Setelah mempraktekkan topik ini, praktikan diharapkan dapat :

1. Memahami konsep dasar komunikasi kabel dan perbedaan antara berbagai jenis kabel yang digunakan dalam sistem komunikasi.
2. Mengimplementasikan komunikasi data menggunakan protokol RS485, serta memahami cara kerja modul RS485.
3. Mengkonfigurasi dan melakukan pengujian komunikasi kabel menggunakan Arduino IDE.

a. Alat Dan Bahan

1. Arduino Uno
2. Modul RS485 (MAX485)
3. Kabel Jumper
4. Laptop
5. Kabel Type-D
6. Arduino IDE

7.2 Dasar Teori

7.2.1 Komunikasi Kabel

Komunikasi kabel adalah bentuk komunikasi di mana informasi dikirimkan melalui kabel melalui sinyal listrik, suatu jenis telekomunikasi. Beberapa contoh komunikasi kabel adalah jaringan telepon, komunikasi serat optik, jaringan telepon umum (PSTN), dll. Kejelasan dan stabilitas komunikasi kabel membuat kabel tetap relevan di era komunikasi nirkabel.

Komunikasi kabel bekerja dengan mengirimkan data melalui jalur fisik, seperti kabel tembaga atau serat optik, dari perangkat pengirim (transmitter) ke perangkat penerima (receiver). Data digital diubah menjadi sinyal listrik atau cahaya yang kemudian dikirim sebagai pulsa biner (0 dan 1) melalui kabel.

Jenis kabel, seperti twisted pair atau serat optik, membantu menjaga kualitas sinyal dan mengurangi interferensi. Protokol komunikasi, seperti Ethernet, RS485, atau CAN, mengatur aturan transmisi, seperti arah data (half-duplex atau full-

Modul Praktikum

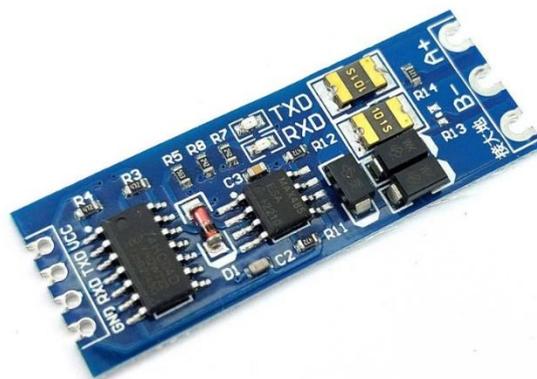
duplex), kecepatan, dan pengendalian kesalahan. Sesampainya di receiver, sinyal diubah kembali menjadi data digital yang bisa dipahami. Proses ini menghasilkan komunikasi data yang cepat, stabil, dan andal, menjadikannya ideal untuk aplikasi industri.

Adapun jenis-jenis kabel yaitu sebagai berikut :

1. Kabel Twisted Pair
2. Kabel Coaxial
3. Kabel serat Optik
4. Kabel USB
5. Kabel Ethernet

Protokol	Kecepatan Maksimum	Jarak Maksimum	Topologi Jaringan	Aplikasi
Ethernet	10 Mbps – 100 Gbps	100 m (tembaga), lebih jauh dengan serat optik	Bintang, bus	Jaringan komputer, IoT
CAN	1 Mbps (CAN klasik), 5 Mbps (CAN FD)	40 – 500 m	Bus	Otomotif, mesin industri
RS485	10 Mbps (jarak pendek)	Hingga 1.200 m	Bus	Kontrol industri, otomasi

7.2.2 RS485



Gambar 1 1 modul RS485

RS485 adalah protokol komunikasi serial yang dirancang untuk transmisi data jarak jauh dengan daya tahan tinggi terhadap interferensi elektromagnetik. RS485 menggunakan arsitektur multipoint yang memungkinkan hingga 32 perangkat untuk terhubung pada satu bus, menjadikannya ideal untuk aplikasi di

Modul Praktikum

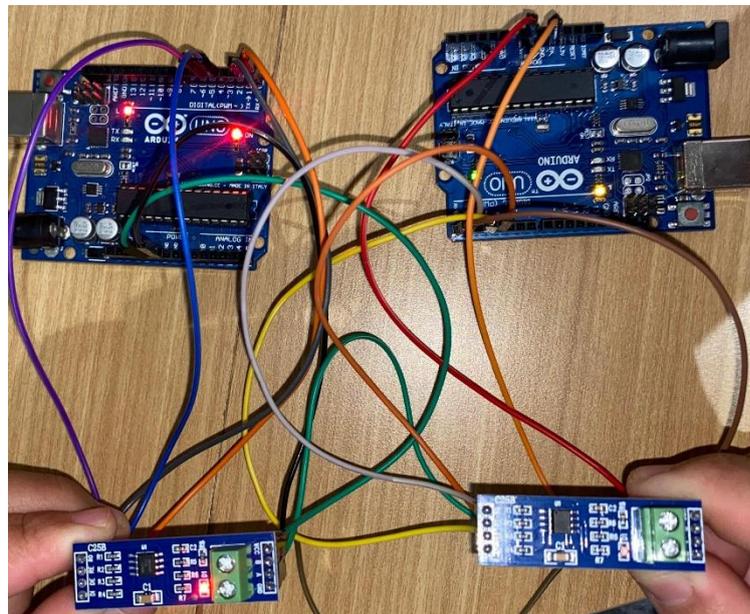
lingkungan industri.

Modul RS485 to TTL ini dapat digunakan untuk menghubungkan board Arduino atau mikrokontroler lain nya ke RS-485. Komunikasi serial menggunakan RS-485 dapat menempuh jarak lebih jauh dibanding RS232 atau TTL, dan mendukung untuk beberapa unit dalam satu bus yang sama.

- Power supply range : 3.3V - 5.5V
- Signal yang kompatible : 3.3V dan 5V
- Jarak komunikasi : 800 meter
- Anti interference
- Auto Flow Control
- 2-way converter
- 110 - 128000 bps
- RXD dan TXD indicator

7.3 Praktikum

1. Langkah pertama buatlah rangkaian seperti dengan gambar di bawah ini, untuk wrenya ikuti pada table berikut :



Modul Praktikum

RS-485	Arduino Uno (pengirim)	RS-485	Arduino Uno (penerima)
RO	TX	RO	TX
RE	RX	RE	RX
DE	PIN 4	DE	PIN 4
DI	PIN 5	DI	PIN 5
VCC	5V	VCC	5V
B	sambungkan ke	B	
A	sambungkan ke	A	
GND	GND	GND	GND

2. Buka Arduino IDE

3. Ketikkan kode berikut :

```
1  #include <SoftwareSerial.h>
2
3  // Pin komunikasi serial modul RS485
4  #define Pino_RS485_RX 10 // DI dari Modul RS485 (Transmitter)
5  #define Pino_RS485_TX 11 // DO dari Modul RS485 (Receiver)
6
7  // Pin kontrol transmisi/terima (RE, DE)
8  #define SSerialTxControl 4
9  #define SSerialTxControl 5
10
11 #define RS485Transmit HIGH
12 #define RS485Receieve LOW
13
14 // Membuat serial software untuk koneksi dengan modul RS485
15 SoftwareSerial RS485Serial(Pino_RS485_RX, Pino_RS485_TX);
16
17 // Menyimpan data yang diterima dari serial
18 String inputString = "";
19
20 unsigned long waktuSebelumnya = 0; // Waktu cetak terakhir
21
22 bool isTransmitter = true; // Variabel untuk menunjukkan apakah ini kode transmitter atau receiver
23
24 void setup() {
25     // Inisialisasi serial pada Arduino
26     Serial.begin(115200);
27
28     if (isTransmitter) {
29         // Pengaturan khusus transmitter
30         Serial.println("Modul Transmitter");
```

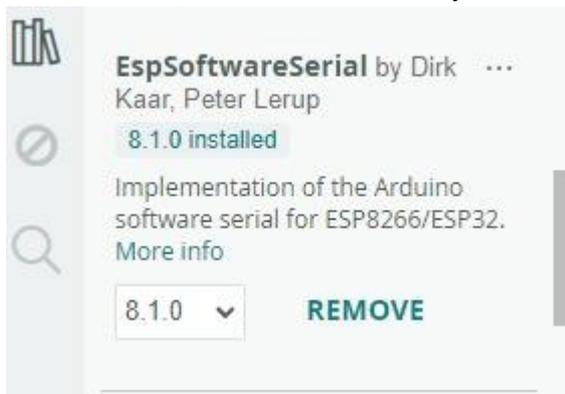
Modul Praktikum

```
30 Serial.println("Modul Transmitter");
31 Serial.println("Masukkan nilai di Serial untuk mengirim data...");
32 pinMode(SSerialTxControl, OUTPUT);
33 RS485Serial.begin(4800);
34 } else {
35 // Pengaturan khusus receiver
36 Serial.println("Modul Receiver");
37 Serial.println("Menunggu data...");
38 pinMode(SSerialTxControl, OUTPUT);
39 digitalWrite(SSerialTxControl, RS485Receive); // Set ke mode penerimaan
40 RS485Serial.begin(4800);
41 }
42 }
43
44 void loop() {
45 if (isTransmitter) {
46 // Logika transmitter
47 while (Serial.available() > 0) {
48 char c = Serial.read();
49
50 // Cek apakah karakter dapat dicetak (mengabaikan karakter kontrol)
51 if (isPrintable(c)) {
52 inputString += c; // Gabungkan karakter untuk membentuk string lengkap
53 }
54 }
55
56 // Memanggil fungsi untuk mengirim data jika ada sesuatu di inputString
57 if (inputString.length() > 0) {
58 kirimData(inputString);
59 inputString = ""; // Menghapus string setelah mengirim data
60 }
61 } else {
62 // Logika receiver
63 if (RS485Serial.available()) {
64 while (RS485Serial.available()) {
65 // Terima data dan buat string
66 char inChar = (char)RS485Serial.read();
67 inputString += inChar;
68
69 // Jika menemukan karakter baris baru, proses string
70 if (inChar == '\n') {
71 prosesBuffer(inputString);
72 inputString = ""; // Menghapus string untuk menerima data baru
73 }
74 }
75 }
76 }
77 }
78
79 void kirimData(String data) {
80 // Aktifkan modul untuk transmisi
81 digitalWrite(SSerialTxControl, RS485Transmit);
82
83 // Kirim data melalui serial RS485
84 RS485Serial.println(data);
85
86 // Tunggu hingga semua data terkirim
87 delay(10); // Tunggu sebentar untuk memastikan transmisi selesai
88 RS485Serial.flush();
```

Modul Praktikum

```
88   RS485Serial.flush();
89
90   // Nonaktifkan modul untuk transmisi (set ke penerimaan)
91   digitalWrite(SSerialTxControl, RS485Receive);
92
93   // Umpan balik ke Serial Monitor
94   Serial.println("Data terkirim: " + data);
95 }
96
97 void prosesBuffer(String buffer) {
98   // Cek apakah buffer memiliki format yang diharapkan /003011X
99   if (buffer.startsWith("/")) {
100    // Hapus karakter awal "/"
101    String data = buffer.substring(1);
102
103    // Cek apakah buffer memiliki panjang yang diharapkan 8 karakter
104    if (data.length() == 8) {
105      // Format ke output 003011R
106      String hasil = data.substring(0, 6) + "R";
107
108      // Cetak hasil ke monitor serial
109      Serial.println(hasil);
110    }
111  }
112 }
```

4. Save File, lalu download library berikut sesuai dengan gambar di bawah ini :



5. Kemudian sambungkan Arduino uno ke laptop.
6. Pilih Board : “Arduino uno “ dan port nya sesuai dengan yang digunakan.
7. Setelah itu Verify dan Upload code yang telah diketik ke Arduino uno yang telah kalian rangkai.
8. Maka hasilnya akan seperti ini, pada serial monitor :

Modul Praktikum

Output Serial Monitor X

Message (Enter to send message to 'Arduino Uno' on 'COM10')

```
Modul Tran..  
Modul Transmitter  
Masukkan nilai di Serial untuk mengirim data...  
Data terkirim: 1  
Data terkirim: 2  
Data terkirim: 1  
Data terkirim: 2  
Data terkirim: 1  
Data terkirim: 2  
Data terkirim: 1  
Data terkirim: 23
```