

MODUL 5

Komunikasi Kabel Menggunakan CAN Bus dan Ethernet

1. TUJUAN

1. Memahami konsep dasar komunikasi kabel dan perbedaan antara berbagai jenis kabel yang digunakan dalam sistem komunikasi.
2. Mengkonfigurasi dan melakukan pengujian komunikasi kabel menggunakan Arduino IDE.

2. ALAT DAN BAHAN PRAKTIKUM

- Arduino IDE
- Kabel LAN
- Kabel Jumper
- Project Board
- Module W5500 Lite
- Module MCP2515

3. DASAR TEORI

3.1 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.

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

Gambar 3.1 Perbedaan Protokol

3.1 MCP2515

Modul MCP2515 berfungsi sebagai antarmuka yang menerjemahkan data antara mikrokontroler dan jaringan CAN menggunakan protokol komunikasi serial SPI. Komponen utamanya adalah chip MCP2515 yang mengelola komunikasi CAN dan chip TJA1050 sebagai transceiver untuk mengubah sinyal digital menjadi sinyal yang sesuai untuk jaringan CAN

3.1 W550 Lite

Module **W5500 Lite** merupakan modul **Ethernet berbasis chip WIZnet W5500** yang berfungsi untuk memberikan konektivitas jaringan kabel (LAN) pada mikrokontroler. Chip W5500 sendiri adalah *embedded Ethernet controller* yang sudah dilengkapi dengan *hardware TCP/IP stack*, sehingga seluruh proses komunikasi jaringan seperti TCP, UDP, IPv4, ICMP, ARP, IGMP, hingga PPPoE dapat dilakukan langsung oleh chip tanpa membebani kerja mikrokontroler. Dengan begitu, mikrokontroler hanya perlu berkomunikasi menggunakan perintah sederhana melalui antarmuka SPI (Serial Peripheral Interface) untuk mengirim atau menerima data dari jaringan.

Fungsi utama modul ini adalah untuk menghubungkan mikrokontroler dengan jaringan internet melalui kabel LAN, memungkinkan komunikasi data berbasis protokol TCP/IP, serta menjadikan sistem mikrokontroler dapat berperan sebagai perangkat IoT berbasis LAN. Modul ini juga memungkinkan mikrokontroler berfungsi sebagai *client* maupun *server* dalam jaringan lokal atau internet. W5500 Lite memiliki ukuran yang ringkas, konektor RJ45 dengan transformator magnetik

Modul Praktikum

bawaan, serta pin header untuk koneksi SPI dan pin daya 3.3V hingga 5V. Modul ini juga dilengkapi LED indikator seperti LINK dan ACT untuk menunjukkan status koneksi dan aktivitas data.

Pin	Nama	Fungsi
1	MISO	Data keluar dari W5500 ke mikrokontroler
2	MOSI	Data masuk dari mikrokontroler ke W5500
3	SCK	Clock SPI
4	CS (SS)	Chip Select – aktif rendah
5	RST	Reset modul
6	INT	Interrupt output (opsional)
7	3V3 / 5V	Daya modul
8	GND	Ground

Gambar 3.2 konfigurasi pin umum

Adapun Cara kerja W5500 yaitu sebagai berikut :

1. Mikrokontroler berkomunikasi dengan W5500 menggunakan SPI interface.
2. Mikrokontroler mengirimkan perintah untuk membuka socket TCP/UDP.
3. W5500 menangani seluruh proses protokol jaringan di hardware (seperti handshake, packet segmentation, checksum, dll).
4. Mikrokontroler hanya bertugas mengirim dan menerima data melalui socket, tanpa perlu menghitung atau mengatur header jaringan.

Kelebihan Modul W5500 Lite:

1. Hardware TCP/IP stack → lebih cepat dan ringan dibanding software stack.
2. Konsumsi daya rendah (sekitar 180 mA saat aktif).
3. Kecepatan tinggi – mendukung 10/100 Mbps Ethernet.
4. Komunikasi SPI cepat hingga 80 MHz.
5. Stabil untuk aplikasi IoT industri (tidak mudah disconnect seperti WiFi).
6. Kompatibel dengan berbagai platform seperti Arduino, STM32, ESP32, dan Raspberry Pi.

3.2 Kabel LAN



Gambar 3. 2 kabel LAN

Kabel LAN (Local Area Network) adalah jenis kabel jaringan yang digunakan untuk menghubungkan berbagai perangkat seperti komputer, router, switch, dan printer dalam satu area terbatas, misalnya di rumah, sekolah, atau kantor. Melalui kabel ini, data dapat ditransmisikan dari satu perangkat ke perangkat lain dengan cepat dan stabil, sehingga memungkinkan pertukaran informasi serta akses bersama terhadap sumber daya seperti file, internet, dan perangkat keras. Kabel LAN berfungsi sebagai media transmisi fisik yang membawa sinyal data dalam bentuk arus listrik atau cahaya, tergantung pada jenis kabel yang digunakan.

4. PERCOBAAN

4.1 Percobaan MCP2515

Gunakan kode dibawah untuk kedua arduino UNO

```

1  #include <mcp_can.h>
2  #include <SPI.h>
3
4  #define CAN_CS 10
5  #define CAN_INT 2
6
7  MCP_CAN CAN(CAN_CS);
8
9  void setup() {
10     Serial.begin(9600);
11     while (!Serial);
12
13     if (CAN.begin(MCP_ANY, CAN_500KBPS, MCP_8MHZ) == CAN_OK)
14         Serial.println("CAN BUS init OK!");
15     else
16         Serial.println("CAN BUS init FAIL!");
17
18     CAN.setMode(MCP_NORMAL);
19     pinMode(CAN_INT, INPUT);
20 }
21
22 void loop() {
23     if (!digitalRead(CAN_INT)) {
24         long unsigned int rxId;
25         unsigned char len = 0;
26         unsigned char rxBuf[8];
27         CAN.readMsgBuf(&rxId, &len, rxBuf);
28
29         Serial.print("Terima: ");
30         for (byte i = 0; i < len; i++) Serial.write(rxBuf[i]);
31         Serial.println();
32     }
33
34     if (Serial.available()) {
35         String msg = Serial.readStringUntil('\n');
36         byte buf[8];
37         byte len = msg.length();
38         if (len > 8) len = 8;
39         msg.getBytes(buf, len + 1);
40         CAN.sendMsgBuf(0x10, 0, len, buf);
41     }
42 }
43 |


```

Arduino UNO	MCP2515		MCP2515 Pertama	MCP2515 Kedua
5 V	5 V		CAN-H	CAN-H
GND	GND		CAN-L	CAN-L
D10	CS		GND	GND
D12	SO			
D11	SI		Short Pin J1	Short Pin J1
D13	SCK			
D2	INT			

Modul Praktikum

4.2 Percobaan Ethernet

1. Buka Software Arduino IDE
2. Ketikkan kode berikut:



```
sketch_oct16a | Arduino IDE 2.3.4
File Edit Sketch Tools Help
Arduino Uno
sketch_oct16a.ino
1 #include <SPI.h>
2 #include <Ethernet.h>
3
4 // MAC address
5 byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
6
7 // Konfigutasi IP Statik
8 IPAddress ip(192, 168, 137, 177); // Statik IP nya arduino
9 IPAddress gateway(192, 168, 137, 1); // Statik IP yg ditaruh dalam laptop
10 IPAddress subnet(255, 255, 255, 0);
11 IPAddress dns(8, 8, 8, 8);
12
13 EthernetServer server(80);
14
15 const int VRx = A0;
16 const int VRy = A1;
17 const int SW = 2;
18
19 // kalau mau di balik nilai axis nya (inverted), uncomment 2 kode dibawah
20 // VRx = 1023 - VRx;
21 // VRy = 1023 - VRy;
22
23 void setup() {
24   Serial.begin(115200);
25   pinMode(10, OUTPUT);
26   digitalWrite(10, HIGH);
27   pinMode(SW, INPUT_PULLUP);
28
29   Ethernet.begin(mac, ip, dns, gateway, subnet);
30   server.begin();
31
32   Serial.print("Server started at: ");
33   Serial.println(Ethernet.localIP());
```

Modul Praktikum

sketch_oct16a | Arduino IDE 2.3.4

File Edit Sketch Tools Help

Arduino Uno

```
sketch_oct16a.ino
33 Serial.println(Ethernet.localIP());
34 }
35
36 void loop() {
37   EthernetClient client = server.available();
38   if (client) {
39     String request = readHTTPRequest(client);
40     if (request.indexOf("GET /data") >= 0) {
41       sendData(client);
42     } else {
43       sendHTML(client);
44     }
45     client.stop();
46   }
47 }
48
49 String readHTTPRequest(EthernetClient &client) {
50   String request = "";
51   unsigned long timeout = millis();
52   while (client.connected() && millis() - timeout < 1000) {
53     if (client.available()) {
54       char c = client.read();
55       request += c;
56       if (c == '\n' && request.endsWith("\r\n\r\n")) break;
57     }
58   }
59   return request;
60 }
61
62 void sendData(EthernetClient &client) {
63   int x = analogRead(VRx);
64   int y = analogRead(VRy);
65   int sw = digitalRead(SW);
```

sketch_oct16a | Arduino IDE 2.3.4

File Edit Sketch Tools Help

Arduino Uno

```
sketch_oct16a.ino
65 int sw = digitalRead(SW);
66
67 client.println("HTTP/1.1 200 OK");
68 client.println("Content-Type: application/json");
69 client.println("Connection: close");
70 client.println();
71 client.print("{\"x\":");
72 client.print(x);
73 client.print(",\"y\":");
74 client.print(y);
75 client.print(",\"sw\":\"");
76 client.print(sw == LOW ? "Pressed" : "Released");
77 client.println("\");");
78 }
79
80 void sendHTML(EthernetClient &client) {
81   client.println("HTTP/1.1 200 OK");
82   client.println("Content-Type: text/html");
83   client.println("Connection: close");
84   client.println();
85   client.println("<!DOCTYPE html>");
86   client.println("<html><head><title>Joystick Monitor</title>");
87   client.println("<style>");
88   client.println("<body{font-family:Arial;text-align:center;background:#f0f0f0}>");
89   client.println("<card{background:white;border-radius:10px;padding:20px;display:inline-block;box-shadow:0 0 10px #aaa;margin-top:20px}>");
90   client.println("<#pad(position:relative;width:300px;height:300px;margin:20px auto;background:#ddd;border:2px solid #aaa;border-radius:10px}>");
91   client.println("<#dot(position:absolute;width:20px;height:20px;background:red;border-radius:50%;top:140px;left:140px;transition:top 0.1s,left 0.1s;background 0.1s}>");
92   client.println("</style>");
93   client.println("<script>");
94   client.println("<function updateData(){");
95   client.println("<fetch('/data').then(res=>res.json()).then(d=>{");
96   client.println("<document.getElementById('x').innerText=d.x;");
97   client.println("<document.getElementById('y').innerText=d.y;");
```

Modul Praktikum

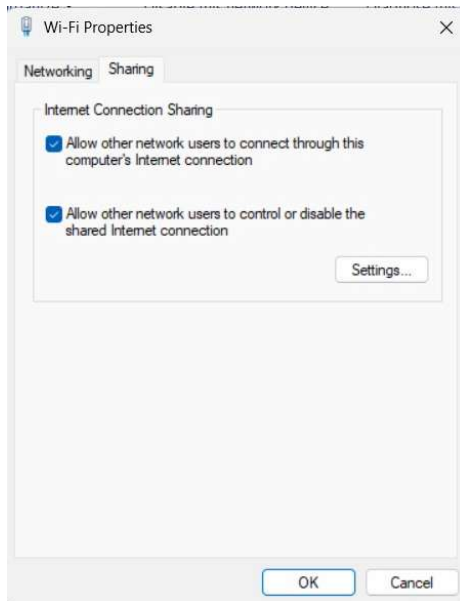
```

97     client.println(F("document.getElementById('y').innerText=d.y;"));
98     client.println(F("document.getElementById('sw').innerText=d.sw;"));
99     client.println(F("let xPos = (d.x/1023)*280;"));
100    client.println(F("let yPos = (d.y/1023)*280;"));
101    client.println(F("let dot = document.getElementById('dot');"));
102    client.println(F("dot.style.left = xPos + 'px';"));
103    client.println(F("dot.style.top = yPos + 'px';"));
104    client.println(F("dot.style.background = (d.sw=='Pressed' ? 'green' : 'red');"));
105    client.println(F("}"));
106    client.println(F(""));
107    client.println(F("setInterval(updateData,100);"));
108    client.println(F("</script>"));
109    client.println(F("</head><body onload='updateData()'>"));
110    client.println(F("<h2>Arduino Joystick Web Server</h2>"));
111    client.println(F("<div class='card'>"));
112    client.println(F("<p><b>VRx:</b> <span id='x'>---</span></p>"));
113    client.println(F("<p><b>VRy:</b> <span id='y'>---</span></p>"));
114    client.println(F("<p><b>SW:</b> <span id='sw'>---</span></p>"));
115    client.println(F("</div>"));
116    client.println(F("<div id='pad'><div id='dot'></div></div>"));
117    client.println(F("</body></html>"));
118 }

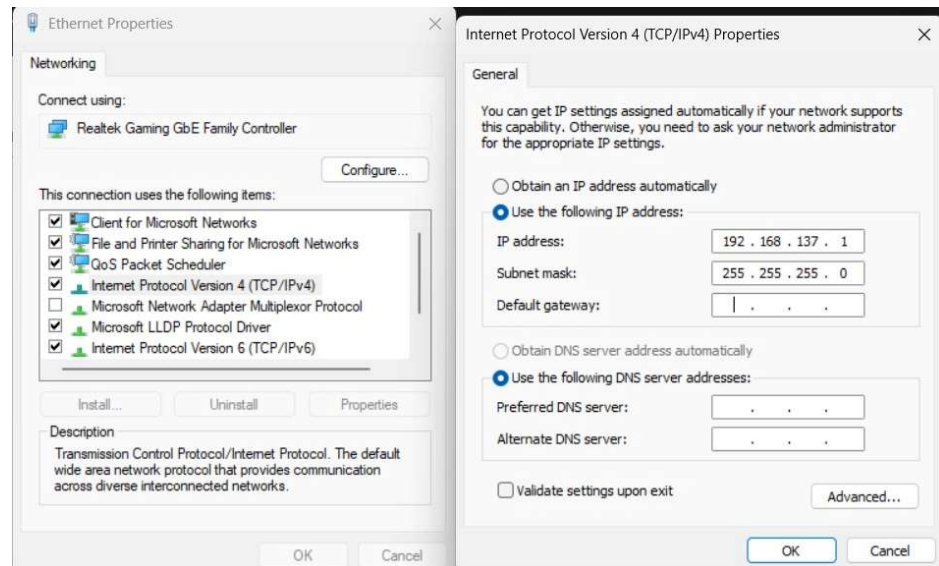
```

Arduino	W5500 Lite Module	Joystick KY-023
3.3V	V	-
GND	GND	-
9	RST	-
10	CS	-
11	MOSI	-
12	MISO	-
13	SCK	-
5V	-	5V
GND	-	GND
A0	-	VRx
A1	-	Vry
2	-	SW

3. Buka View Network Connection
4. Klik kanan Wifi >sharing>ceklis



5. Klik kanan ethernet>properti>masukan ip>ok



6. Buka browser>masukan alamat 192.168.137.177

5. Kesimpulan

Module W5500 Lite merupakan modul Ethernet yang berfungsi untuk memberikan konektivitas jaringan kabel (LAN) pada mikrokontroler. Modul ini menggunakan chip WIZnet W5500 yang telah dilengkapi dengan *hardware TCP/IP stack*, sehingga seluruh proses komunikasi jaringan seperti TCP, UDP, IPv4, ICMP, ARP, IGMP, hingga PPPoE dapat dilakukan langsung oleh chip tanpa membebani kerja mikrokontroler. Dengan komunikasi melalui antarmuka SPI, mikrokontroler dapat mengirim dan menerima data dengan mudah dan efisien.

W5500 Lite memiliki keunggulan berupa kecepatan transfer data tinggi hingga 100 Mbps, konsumsi daya rendah, serta kestabilan koneksi yang baik. Modul ini juga kompatibel dengan berbagai platform seperti Arduino, ESP32, dan STM32. Dengan kemampuan tersebut, W5500 Lite sangat cocok digunakan dalam sistem Internet of Things (IoT) berbasis LAN, otomasi industri, serta aplikasi jaringan lokal yang memerlukan koneksi cepat, stabil, dan andal.

6. Latihan

1. Mengapa W5500 Lite cocok digunakan dalam sistem Internet of Things (IoT) berbasis LAN? Jelaskan dengan contoh penerapannya!
2. Jelaskan apa yang terjadi ketika alamat IP yang dikonfigurasi dimasukkan ke dalam browser, dan bagaimana hal tersebut menunjukkan bahwa komunikasi antara Arduino dan jaringan berhasil!
3. Bagaimana cara memastikan bahwa koneksi antara Arduino dan jaringan berhasil setelah program diunggah dan dijalankan?
4. Jelaskan mengapa komunikasi menggunakan kabel Ethernet dianggap lebih stabil dibandingkan komunikasi nirkabel, khususnya pada sistem berbasis mikrokontroler!