

Modul Praktikum Jaringan Komunikasi 08

Protokol Komunikasi Nirkabel Berbasis Mikrokontroler

1. Tujuan

Tujuan dari praktikum “PJK-08 : Protokol Komunikasi Nirkabel Berbasis Mikrokontroler” ini adalah :

- a. Mengetahui dasar komunikasi nirkabel berbasis mikrokontroler
- b. Mengetahui protokol komunikasi *Bluetooth Low Energy* (BLE)
- c. Mengetahui protokol komunikasi *Long Range* (LoRa)
- d. Mengetahui protokol komunikasi berbasis frekuensi radio menggunakan RF433

2. Kompetensi Dasar

Dalam menjalani praktikum “PJK-08 : Protokol Komunikasi Nirkabel Berbasis Mikrokontroler” ini, dibutuhkan kompetensi dasar sebagai syarat utama untuk menuntaskan praktikum. Kompetensi dasar yang dibutuhkan adalah :

- a. Dasar Informatika
- b. Praktikum Dasar Informatika
- c. Elektromagnetika
- d. Pengolahan Data
- e. Jaringan Komunikasi
- f. Elektronika Digital

3. Dasar Teori

3.1 Komunikasi Nirkabel

Komunikasi nirkabel menekankan pada protokol komunikasi antar mikrokontroler tanpa menggunakan kabel. Komunikasi nirkabel dapat mentransmisikan suara dan data tanpa kabel. Sebagai pengganti koneksi fisik, data bergerak melalui sinyal elektromagnetik yang disiarkan dari fasilitas pengiriman ke perangkat perantara dan pengguna akhir. Pemancar nirkabel pertama dikenal pada awal abad ke-20 menggunakan radio telegrafi, yaitu komunikasi radio menggunakan kode Morse atau sinyal berkode lainnya yang kemudian dengan sistem modulasi memungkinkan pengiriman suara dan musik secara nirkabel. Pemancar nirkabel menggunakan gelombang elektromagnetik untuk membawa suara, data, video, atau sinyal melalui jalur komunikasi.

3.2 ESP32

Mikrokontroler ESP32 merupakan mikrokontroler SoC (*System on Chip*) terpadu dengan dilengkapi WiFi 802.11 b/g/n, *Bluetooth* versi 4.2, dan berbagai peripheral. ESP32 adalah *chip* yang cukup lengkap, terdapat prosesor, penyimpanan, dan akses pada GPIO (*General Purpose Input Output*). ESP32 bisa digunakan untuk rangkaian pengganti pada Arduino, ESP32 memiliki kemampuan untuk mendukung terkoneksi ke WIFI secara langsung.



Gambar 3.1. ESP32

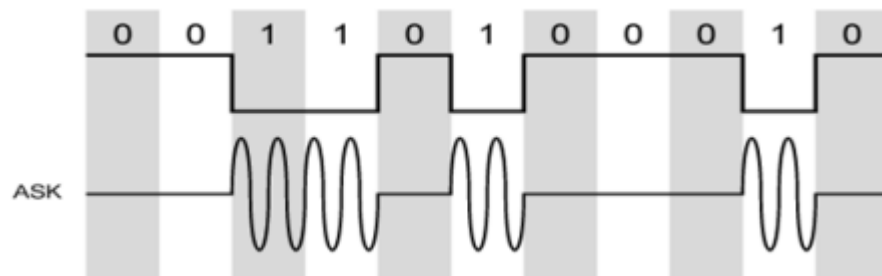
Adapun spesifikasi dari ESP32 adalah sebagai berikut: Board ini memiliki dua versi, yaitu 30 GPIO dan 36 GPIO. Keduanya memiliki fungsi yang sama tetapi versi yang 30 GPIO dipilih karena memiliki dua pin GND. Semua pin diberi label di bagian atas *board* sehingga mudah untuk dikenali. Board ini memiliki *interface USB to UART* yang mudah diprogram dengan program pengembangan aplikasi seperti Arduino IDE [3]. Sumber daya board bisa diberikan melalui konektor *micro USB*.

3.3 Teknik Modulasi dan Demodulasi (ASK, FSK, PSK)

3.3.1 ASK (*Amplitudo Shift Keying*)

ASK merupakan jenis modulasi digital yang paling sederhana, dimana sinyal *carrier* dimodulasi berdasarkan amplitudo sinyal digital. Umumnya, kita membutuhkan dua buah sinyal $s_1(t)$ dan $s_2(t)$ untuk transmisi biner. Jika *transmitter* ingin mentransmisikan bit 1, $s_1(t)$ digunakan untuk interval pensinyalan $(0, T_b)$. Sedangkan untuk mentransmisikan bit 0, $s_2(t)$ digunakan pada interval $(0, T_b)$ [4]. Untuk ASK sinyal transmisi dapat dituliskan sbb: Sinyal direpresentasikan dalam dua kondisi perubahan amplitudo gelombang pembawa Sinyal “1”

direpresentasikan dengan status “ON” (ada gelombang pembawa), Sinyal “0” direpresentasikan dengan status “OFF” (tidak ada gelombang pembawa).

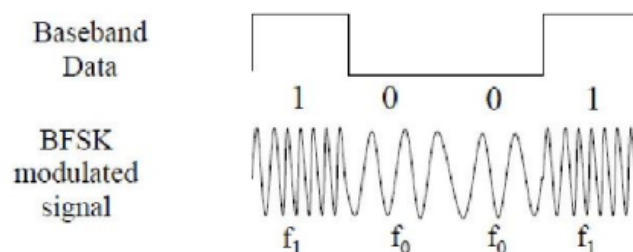


Gambar 3.2. Modulasi sinyal ASK [5]

Amplitudo shift keying (ASK) dalam konteks komunikasi digital adalah proses modulasi, yang menanamkan untuk dua atau lebih tingkat amplitudo diskrit sinusoid. Hal ini juga terkait dengan jumlah tingkat diadopsi oleh pesan digital. Untuk urutan pesan biner ada dua tingkat, salah satunya biasanya nol. Jadi gelombang termodulasi terdiri dari semburan sinusoidal sebuah.

3.3.2 FSK (*Frequency Shift Keying*)

Dalam modulasi FM, frekuensi *carrier* diubah-ubah harganya mengikuti harga sinyal pemodulasinya (analog) dengan amplitudo pembawa yang tetap. Jika sinyal yang memodulasi tersebut hanya mempunyai dua harga tegangan 0 dan 1 (biner/ digital), maka proses modulasi tersebut dapat diartikan sebagai proses penguncian frekuensi sinyal. Hasil gelombang FM yang dimodulasi oleh data biner ini kita sebut dengan *Frekuensi Shift Keying* (FSK).

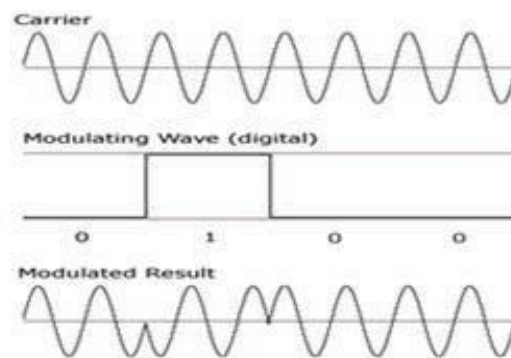


Gambar 3.3. Sinyal sebelum dan sesudah dimodulasi dengan FSK [5]

Dalam sistem FSK (*Frequency Shift Keying*), maka simbol 1 dan 0 ditransmisikan secara berbeda antara satu sama lain dalam satu atau dua buah sinyal sinusoidal yang berbeda besar frekuensinya.

3.3.3 PSK (*Phase Shift Keying*)

Phase Shift Keying (PSK) atau pengiriman sinyal digital melalui pergeseran fasa. Metode ini merupakan suatu bentuk modulasi fasa yang memungkinkan fungsi pemodulasi fasa gelombang termodulasi di antara nilai nilai diskrit yang telah ditetapkan sebelumnya.



Gambar 3.4. Gelombang modulasi PSK [5]

Dalam proses modulasi ini fasa dari frekuensi gelombang pembawa berubah-ubah sesuai dengan perubahan status sinyal informasi digital. Sudut fasa harus mempunyai acuan kepada pemancar dan penerima guna memudahkan untuk memperoleh stabilitas. Dalam keadaan seperti ini, fasa yang ada dapat dideteksi bila fasa sebelumnya telah diketahui. Hasil dari perbandingan ini dipakai sebagai patokan.

3.4 RF433

Modul *Radio Frequency* (RF) merupakan komponen yang dapat mendeteksi sinyal gelombang elektromagnetik yang digunakan oleh sistem komunikasi untuk mengirimkan informasi melalui udara dari satu titik ke titik lainnya yang merambat di antara antena pengirim dan antena penerima. Modul RF dapat ditunjukkan pada gambar berikut.



Gambar 3.5. RF Module

Modul RF memiliki dua perangkat elektronik untuk mengirim sinyal gelombang elektromagnetik pada perangkat pemancar dan untuk menerima sinyal gelombang elektromagnetik pada perangkat penerima. Pemancar RF menerima data *serial* dan mentransmisikan secara nirkabel melalui antenna dengan jarak transmisi 1 km. Data yang dikirim akan diterima oleh Penerima RF yang beroperasi pada frekuensi yang sama dengan Pemancar RF yaitu 433 MHz. Frekuensi yang digunakan dalam proses pengiriman dan penerimaan harus sama, sehingga tidak terjadi kegagalan dalam proses komunikasi. Dengan besar frekuensi tersebut menjadikan sistem komunikasi dengan menggunakan modul ini memiliki keunggulan penetrasi yang kuat dan sinyal yang stabil. Beberapa keuntungan transmisi komunikasi menggunakan modul RF433 adalah sebagai berikut:

1. Penetrasi sinyal yang kuat

Sinyal RF433 dapat menembus dinding dan rintangan, sehingga jangkauannya lebih luas. Sinyal RF433 cenderung dapat digunakan sekitar 100 meter melintasi bangunan yang berarti ia memiliki kemampuan transmisi sinyal yang cukup kuat dan transmisinya juga relatif stabil dan dapat memiliki jangkauan transmisi yang lebih luas.

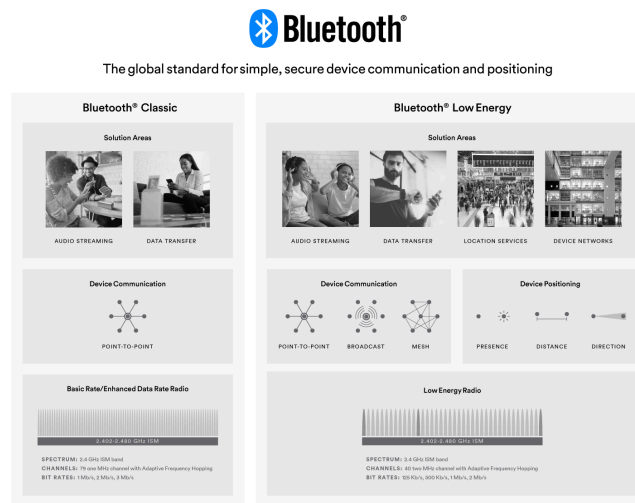
2. Penggunaan yang relatif mudah

Protokol komunikasi RF433 relatif sederhana dan mudah diimplementasikan, pengguna dapat mewujudkan komunikasi dan kontrol antar perangkat melalui pengoperasian yang sederhana.

3.5 Bluetooth Low Energy (BLE)

Bluetooth Low Energy (BLE) adalah teknologi komunikasi nirkabel yang dirancang untuk aplikasi yang memerlukan konsumsi daya rendah. BLE dirancang untuk beroperasi dengan perangkat yang memiliki daya terbatas, seperti sensor, *wearable devices*, dan IoT

(*Internet of Things*). Protokol ini memungkinkan transfer data dengan latensi rendah dan efisiensi energi yang tinggi.



Gambar 3.6. Perbedaan Bluetooth dan BLE [1]

BLE menggunakan teknik modulasi frekuensi yang disebut *Frequency Hopping Spread Spectrum* (FHSS) untuk mengurangi interferensi sinyal dan meningkatkan kualitas komunikasi. BLE beroperasi pada frekuensi ISM (*Industrial, Scientific, and Medical*) 2.4 GHz, sama seperti teknologi Bluetooth klasik, namun dengan pendekatan yang lebih efisien dalam hal konsumsi energi [2]. BLE terdiri dari dua mode operasi utama:

- Advertising Mode:** Perangkat BLE mengirimkan *broadcast* paket data untuk memberitahukan kehadirannya kepada perangkat lain.
- Connection Mode:** Setelah perangkat lain menerima sinyal dari paket *advertising*, mereka dapat membentuk koneksi untuk bertukar data lebih lanjut.

Selain itu, protokol BLE dibagi menjadi beberapa lapisan utama. Lapisan protokol BLE antara lain adalah :

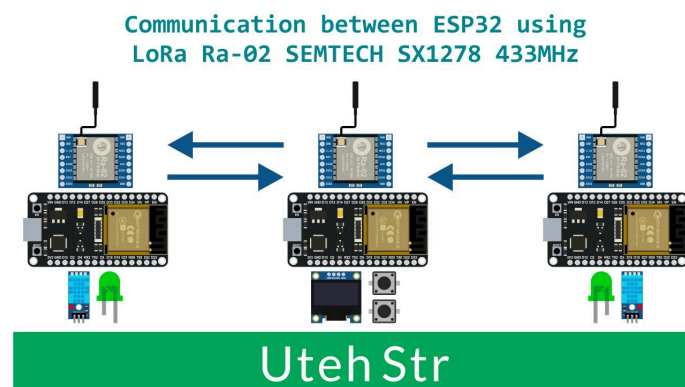
- Physical Layer:** Ini adalah lapisan yang bertanggung jawab atas transmisi sinyal radio pada frekuensi 2.4 GHz.
- Link Layer:** Bertanggung jawab untuk *data framing*, manajemen koneksi, dan kontrol transmisi.
- Host Controller Interface (HCI):** Menghubungkan antara lapisan perangkat keras (hardware) dan perangkat lunak (software).

- d. **Logical Link Control and Adaptation Protocol (L2CAP)**: Mengelola fragmentasi dan penggabungan data serta mengatur multiplexing data.
- e. **Attribute Protocol (ATT)** dan **Generic Attribute Profile (GATT)**: Digunakan untuk mengelola bagaimana data diatur dan ditransfer di antara perangkat BLE. Data disimpan dalam bentuk *attributes*, dan GATT menyediakan layanan yang memungkinkan perangkat untuk mengakses dan memodifikasi *attributes*.

Keunggulan utama BLE terletak pada efisiensi energi yang digunakan. BLE mengonsumsi daya yang jauh lebih rendah karena dirancang untuk bekerja dalam mode siaga yang panjang dan hanya mengaktifkan komunikasi saat diperlukan. Ini membuat BLE sangat tepat untuk aplikasi yang membutuhkan pengoperasian baterai dalam waktu lama, seperti perangkat *wearable* atau sensor.

3.6 Long Range (LoRa)

Protokol komunikasi LoRa atau Long Range merupakan teknologi komunikasi nirkabel jarak jauh yang memanfaatkan teknik modulasi spektrum *Chirp Spread Spectrum* (CSS) serta menggunakan daya rendah. LoRa beroperasi pada frekuensi di atas 433/868 MHz di wilayah Eropa, sementara di Amerika Serikat menggunakan frekuensi 915 MHz. Teknologi ini merupakan sistem telekomunikasi jarak jauh dengan konsumsi daya dan bitrate yang rendah, yang dirancang sebagai solusi infrastruktur untuk *Internet of Things* (IoT).



Gambar 3.7. Komunikasi antar ESP32 menggunakan LoRa [6]

LoRa memiliki arsitektur yang dibagi menjadi 2 bagian yaitu LoRa dan LoRaWAN. Pada bagian LoRa memiliki dua lapisan yaitu, *Radio Frequency (RF) Layer* dan *Physical (PHY) Layer*.

A. *Radio Frequency (RF) Layer*

RF Layer berjalan di ISM band yang mengikuti standar regional negara masing-masing. Pada regional EU memiliki ISM band sebesar 863-870 Mhz, untuk regional US memiliki ISM band sebesar 902-928 Mhz, untuk regional AU memiliki ISM band sebesar 915-928 Mhz, untuk regional CA memiliki ISM band sebesar 779-787 Mhz, sedangkan untuk regional CN memiliki ISM band sebesar 779-787 dan 470-510.

B. *Physical (PHY) Layer*

Lapisan *physical* (PHY) merupakan lapisan yang terdapat radio dan bagian modulasi yang digunakan oleh LoRa, modulasi yang digunakan adalah modulasi spektrum dari *Chirp Spread Spectrum* (CSS) yang memiliki kemampuan penggunaan daya rendah.

Pada bagian lapisan *physical* LoRa terdapat beberapa parameter yang perlu diperhatikan dalam modulasi LoRa antara lain, *Bandwidth* (BW), *Spreading factor* (SF) dan *Code Rate*.

A. *Bandwidth* (BW)

Bandwidth merupakan parameter yang berperan untuk menentukan *chip rate* yang digunakan untuk mengirimkan data. Pada modulasi LoRa, *bandwidth* diatur sebesar 125-,250,500-kHz.

B. *Spreading Factor* (SF)

Spreading factor merupakan jumlah *chip* yang ada untuk merepresentasikan simbol. Semakin tinggi SF maka chip yang digunakan untuk merepresentasikan simbol lebih banyak. Nilai *Spreading factor* dapat diatur dengan rentang 7 hingga 12.

C. *Code Rate*

Code Rate digunakan untuk mengatasi *packet error* karena adanya interferensi, besaran nilai coding rate perlu disesuaikan dengan kondisi jalur pada proses transmisi data. Jika terdapat banyak interferensi/gangguan pada sebuah jalur maka nilai *coding rate* perlu ditingkatkan, namun meningkatnya nilai *coding rate* akan menambah durasi dari sebuah transmisi. Nilai yang dapat diatur sebesar 5 bit, 6 bit, 7 bit, atau 8 bit.

Selain itu LoRa juga memiliki parameter untuk mengukur kuat sinyal. *Received Signal Strength Indication* (RSSI) merupakan kekuatan sinyal yang diukur dalam bentuk dbm

dan menggunakan nilai negatif, nilai RSSI dapat digunakan sebagai parameter seberapa baik *receiver* dapat mendengar sinyal dari *sender*. Jika nilai RSSI semakin dekat dengan 0 maka kualitas sinyal antara *sender* dan *receiver* menjadi lebih baik. Nilai RSSI minimum yang ada pada LoRa bernilai -120 dbm, sedangkan untuk sinyal terkuat berada pada nilai -30 dbm.

4. Alat dan Bahan

Alat dan bahan yang digunakan untuk menjalankan praktikum “PJK-08 : Protokol Komunikasi Nirkabel Berbasis Mikrokontroler” antara lain adalah:

No	Alat	Jumlah
1	Laptop dengan <i>software</i> Arduino IDE	2
2	Mikrokontroler Arduino UNO R3	2
3	Mikrokontroler ESP32-Wroom 32D	2
4	Modul LoRa E220-900T22D	2
5	Modul RF433 FS100A Transmitter	1
6	Modul RF433 080408 Receiver	1
7	USB to TTL USB2TT_004	2
8	Meteran proyek	1
9	Project Board	1
10	Kabel jumper yang layak untuk digunakan	secukupnya

5. Langkah Percobaan

A. Menyiapkan ESP32 dengan Arduino IDE

Untuk menyiapkan ESP32 agar terhubung dengan *software* Arduino IDE, dapat mengikuti langkah berikut :

- 1) Pastikan laptop yang akan digunakan untuk praktikum sudah terinstall *software* **Arduino IDE**.

Direkomendasikan menggunakan Software Arduino IDE v1.8.18.

- 2) Pada *software* Arduino IDE, pilih **File > Preferences**. Salin dan tempel link berikut :

Unset

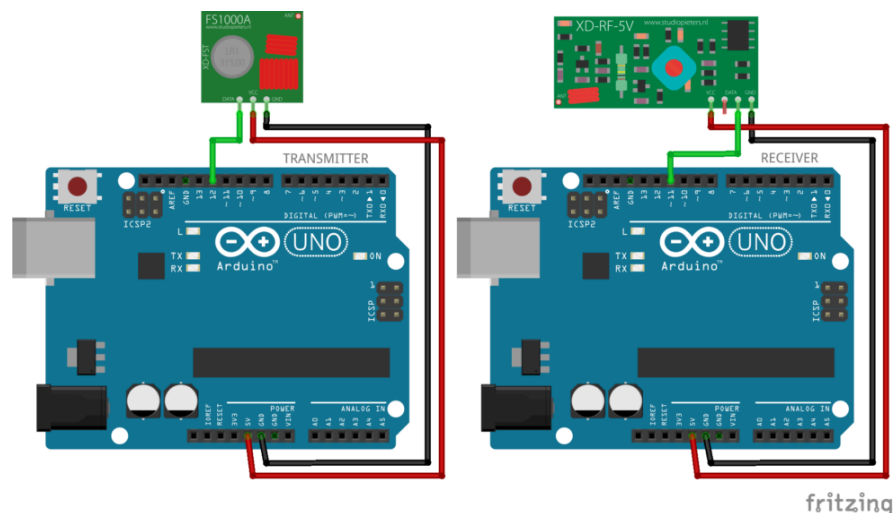
https://dl.espressif.com/dl/package_esp32_index.json

Pada bagian “Additional Board Manager” lalu klik OK.

- 3) Pergi ke **Tools > Board > Board Manager** dan install “**ESP32 by Espressif Systems**” board.
- 4) Setelah instalasi sudah selesai, pilih Board ESP32 pada menu **Tools > Board > ESP32 > uPesy WROOM Devkit**.
- 5) Pilih Port COM yang terhubung dengan mikrokontroler ESP32 pada laptop.

B. Percobaan 1 : Protokol Komunikasi berbasis Frekuensi Radio dengan RF433

- 1) Persiapkan semua alat dan bahan yang akan digunakan untuk praktikum.
- 2) Siapkan dua buah Arduino UNO, satu buah modul RF433 *transmitter*, dan satu buah modul RF433 *receiver*.
- 3) Rangkai modul RF433 *transmitter* dan RF433 *receiver* dengan masing-masing Arduino UNO seperti pada gambar berikut :



Gambar 5.1. Rangkaian modul RF433 dengan Arduino UNO

- 4) Siapkan dua buah laptop dengan *software* Arduino IDE yang sudah terinstall board “Arduino Uno”. Laptop pertama akan dihubungkan dengan modul RF433 *transmitter* dan laptop kedua akan dihubungkan dengan modul RF433 *receiver*.
- 5) Untuk mengatur agar masing-masing RF433 dapat mengirim dan menerima data satu dengan yang lainnya, dapat menggunakan kode program yang terdapat pada **File > Examples > SoftwareSerial > “SoftwareSerialExample”**.
- 6) Cocokkan pin RX dan TX pada masing-masing Arduino UNO dengan kode program dan dengan masing-masing RF433 sesuai dengan perannya.
- 7) Data yang akan dikirimkan oleh RF433 *transmitter* dapat diatur pada kode program yang terdapat di dalam *void setup* dan juga dapat diatur pada serial monitor yang terhubung pada RF433 *transmitter*.
- 8) Atur *baudrate* pengiriman data pada RF433 *transmitter* dan RF433 *receiver* pada kode program berikut :

```
// set the data rate for the SoftwareSerial port
mySerial.begin(4800);
mySerial.println("Hello, world?");
}
```

- 9) Setelah kode program untuk masing-masing RF433 siap, *build* kode program tersebut ke masing-masing Arduino UNO.
- 10) Lakukan pengamatan terhadap data yang diterima oleh RF433 *receiver* pada *serial monitor* yang terhubung dengan RF433 *receiver*.
- 11) Variasikan jarak antar modul RF433 dan *baudrate* dari kedua RF433. Catat hasil pengamatan di Laporan Sementara.

C. Percobaan 2 : Protokol Komunikasi berbasis Bluetooth Low Energy (BLE)

- 1) Persiapkan semua alat yang akan digunakan untuk praktikum.
- 2) Siapkan dua buah ESP32-Wroom 32D. Satu ESP32 akan bertindak sebagai BLE *Client* dan satu ESP32 akan bertindak sebagai BLE *Server*.
- 3) Siapkan dua buah laptop dengan *software* Arduino IDE yang sudah terinstall board “uPesy ESP32 WROOM Devkit”. Laptop pertama akan dihubungkan dengan ESP32 BLE *Client* dan laptop kedua akan dihubungkan dengan ESP32 BLE *Server*.
- 4) Pada laptop yang terhubung dengan ESP32 BLE *Client*, buka kode program pada **File > Examples > BLE > Client**.

- 5) Atur *service* UUID dan *characteristic* UUID dengan mengubah bagian kode program berikut :

```
// The remote service we wish to connect to.  
static BLEUUID serviceUUID("4fafc201-1fb5-459e-8fcc-c5c9c331914b");  
// The characteristic of the remote service we are interested in.  
static BLEUUID charUUID("beb5483e-36e1-4688-b7f5-ea07361b26a8");
```

Service UUID dan *characteristic* UUID pada BLE dapat di-generate menggunakan “**UUID Generator**” di internet.

- 6) Lakukan hal yang sama pada laptop yang terhubung dengan ESP32 BLE *Server*.
7) Atur data yang akan dikirimkan dengan mengubah bagian kode program berikut :

```
pCharacteristic->setValue("Hello World says Neil");  
pService->start();
```

Pengaturan data yang akan dikirimkan juga dapat diatur pada serial monitor yang terhubung dengan ESP32 BLE *Server*.

- 8) Amati data yang diterima oleh *Client* dengan melihat serial monitor yang terhubung dengan ESP32 BLE *Client*.
9) Variasikan *service* UUID dan *characteristic* UUID dari kedua BLE serta variasikan jarak dari kedua ESP32.
10) Catat hasil pengamatan di Laporan Sementara.

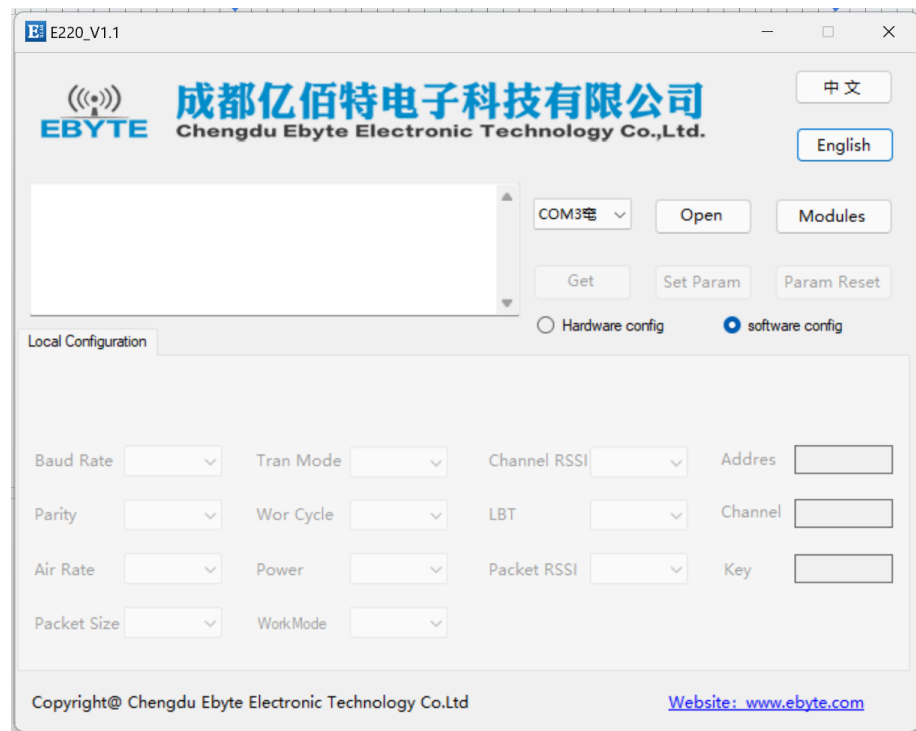
D. Percobaan 3 : Protokol Komunikasi berbasis Long Range (LoRa)

a. Mengkonfigurasi LoRa

- 1) Siapkan dua buah modul LoRa E220-900T22D dan dua buah USBTTL USB2TT_004.
- 2) Hubungkan modul LoRa E220-900T22D dengan USB to TTL USB2TT-004 sesuai dengan konfigurasi pin out adalah sebagai berikut :

USB to TTL	LoRa E220-900T22D
5V	Vin
GND	GND
R	TX
T	RX

- 3) Sambungkan USB to TTL yang sudah dihubungkan dengan modul LoRa ke laptop.
- 4) Download dan install *software* “**RF_Setting(E220-RD) V1.0**” melalui tautan berikut :
<https://drive.google.com/file/d/11WA5oF2h37Qmhn7yxZ0HnLFOo1sF4kpy/view>
- 5) Buka *software* “RF_Setting(E220-RD) V1.0” hingga muncul jendela program seperti pada **Gambar 5.2** berikut :



Gambar 5.2 Tampilan *software* RF_Setting(E220-RD) V1.0

- 6) Atur port laptop yang tersambung dengan modul LoRa pada bagian ini
- 7) Setelah port sudah terpilih, kemudian pilih “Hardware config” lalu tekan “Open”.
- 8) Untuk menghubungkan *software* dengan port yang dipilih, tekan “Get” hingga muncul *pop-up window* “Read Param Success !!”
- 9) Ubah mode “Channel RSSI” dan “Packet RSSI” menjadi “Enabled”.
- 10) Pengaturan *address* dan *channel* dapat dilakukan pada bagian “Address” dan “Channel”. Parameter *channel* dan *address* LoRa adalah sebagai berikut :

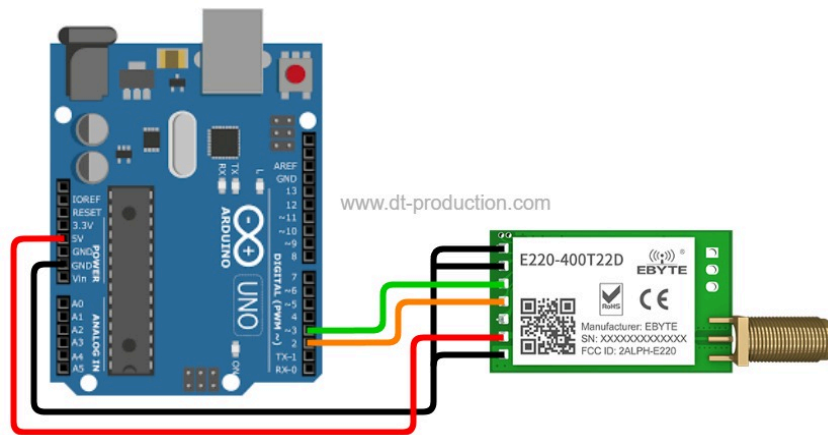
- Address : 0 - 65535
- Channel 0 - 83

11) Setelah LoRa sudah terkonfigurasi, tekan tombol “Set Param” untuk menyimpan pengaturan pada modul LoRa.

12) Tutup *software* “RF_Setting(E220-RD) V1.0”.

b. LoRa Transmitter dan Receiver

- 1) Siapkan dua buah mikrokontroler Arduino UNO.
- 2) Sambungkan kedua modul LoRa dengan masing-masing mikrokontroler Arduino UNO seperti rangkaian pada **Gambar 5.3**.



Gambar 5.3 Rangkaian modul LoRa dengan Arduino UNO

- 3) Siapkan dua buah laptop dengan *software* Arduino IDE dan sambungkan kedua mikrokontroler Arduino UNO ke masing-masing laptop.
- 4) Buka *software* Arduino IDE dan buat kode program untuk membangkitkan LoRa Transmitter dan Receiver seperti pada gambar berikut :

```

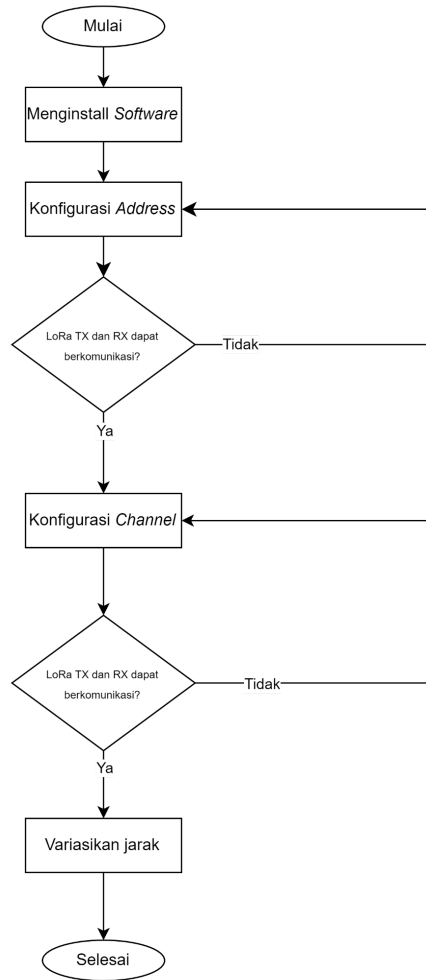
1 #include <LoRa_E220.h>
2 #include <SoftwareSerial.h>
3
4 SoftwareSerial Lora(2, 3); // RX, TX
5 LoRa_E220 e220ttl(&Lora);
6
7 void setup() {
8     Serial.begin(9600);
9     Lora.begin(9600);
10    delay(500);
11    e220ttl.begin();
12    Serial.println("Start receiving!");
13 }
14
15 void loop() {
16     if (Serial.available()) {
17         Serial.print("Data Dikirim : ");
18         String DataDikirim = Serial.readStringUntil('\n');
19         Serial.println(DataDikirim);
20         Lora.println(DataDikirim.c_str());
21     }
22
23     if (e220ttl.available() > 1) {
24         Serial.print("Data Diterima : ");
25         ResponseContainer pesan = e220ttl.receiveMessage();
26
27         if (pesan.status.code == 1) {
28             Serial.println(pesan.data);
29         } else {
30             Serial.println("Error: " + String(pesan.status.getResponseDescription()));
31         }
32     }
33     delay(100);
34 }

```

Gambar 5.4 Kode sumber LoRa Transmitter dan LoRa Receiver

- 5) Setelah kode program untuk kedua modul LoRa sudah siap, *build* kode program tersebut ke masing-masing Arduino UNO.
- 6) Atur data yang akan dikirim oleh LoRa Transmitter dengan memasukkan karakter pada *serial monitor* yang terhubung pada LoRa Transmitter.
- 7) Lakukan variasi terhadap *Address* dan *Channel* dengan mengikuti langkah-langkah yang sudah dijelaskan sebelumnya. Variasi dilakukan juga untuk jarak antar LoRa Transmitter dan Receiver pada *Address* dan *Channel* yang berhasil mentransmisikan data.
- 8) Amati data yang diterima oleh LoRa Receiver dengan melihat *serial monitor* yang terhubung dengan LoRa Receiver.
- 9) Catat hasil pengamatan di Laporan Sementara.

Untuk lebih jelasnya, rangkaian “Percobaan 3 : Protokol Komunikasi Long Range (LoRa) ini dapat dilihat pada diagram alir berikut :



Gambar 5.5 Diagram alir percobaan 3

6. Laporan Sementara

A. Percobaan 1 : Protokol Komunikasi berbasis Frekuensi Radio dengan RF433

a) Variasi Jarak

No	Jarak	Data yang dikirim	Data yang diterima	Kondisi data
1	1 m			
2	5 m			
3	10 m			
4	20 m			
5	50 m			

b) Variasi Baudrate

No	Baudrate	Data yang dikirim	Data yang diterima	Kondisi Data
1	1200			
2	2400			
3	9600			
4	19200			

B. Percobaan 2 : Protokol Komunikasi berbasis Bluetooth Low Energy (BLE)

a) Variasi Address

No	BLE Client			BLE Receiver		
	Service UUID	Characteristic UUID	Data yang dikirim	Service UUID	Characteristic UUID	Data yang diterima

b) Variasi Jarak

No	Jarak	Data yang dikirim	Data yang diterima
1	1 m		
2	5 m		
3	10 m		
4	20 m		

C. Percobaan 3 : Protokol Komunikasi berbasis Long Range (LoRa)

a) Variasi *Channel*

No	LoRa Transmitter		LoRa Receiver	
	Channel	Data yang dikirim*	Channel	Data yang diterima
1				
2				

*Ganti dengan nama asisten praktikum masing-masing

Konfigurasi variasi *Channel* antar LoRa Transmitter dengan LoRa Receiver adalah sebagai berikut :

- Channel 1 // Channel 1
- Channel 1 // Channel 2

b) Variasi *Address*

No	LoRa Transmitter		LoRa Receiver	
	<i>Address</i>	Data yang dikirim*	<i>Address</i>	Data yang diterima
1				
2				

* Ganti dengan nama asisten praktikum masing-masing

Konfigurasi variasi *Address* antar LoRa Transmitter dengan LoRa Receiver adalah sebagai berikut :

- Address 1 // Address 1
- Address 1 // Address 2

c) Variasi Jarak

No	Jarak	Data yang dikirim	Data yang diterima
1	1 m		
2	10 m		
3	20 m		
4	100 m		
5	500 m*		

*Tentatif tergantung situasi dan kondisi

Daftar Pustaka

- [1] Bluetooth, “Bluetooth Technology Overview,” *Bluetooth® Technology Website*, 2024. <https://www.bluetooth.com/learn-about-bluetooth/tech-overview/>
- [2] C. Gomez, J. Oller, and J. Paradells, “Overview and Evaluation of Bluetooth Low Energy: An Emerging Low-Power Wireless Technology,” *Sensors*, vol. 12, no. 9, pp. 11734–11753, Aug. 2012, doi: <https://doi.org/10.3390/s120911734>.
- [3] Espressif, “ESP32 Overview | Espressif Systems,” *www.espressif.com*, 2023. <https://www.espressif.com/en/products/socs/esp32>
- [4] T. Slattery and J. Burke, “What Is Modulation? Definition from SearchNetworking,” *SearchNetworking*, Oct. 2021. <https://www.techtarget.com/searchnetworking/definition/modulation>
- [5] “ASK Vs FSK Vs PSK-Difference between ASK,FSK,PSK Modulation,” *www.rfwireless-world.com*. <https://www.rfwireless-world.com/Terminology/ASK-vs-FSK-vs-PSK.html>
- [6] E. D. Widiyanto, A. A. Faizal, D. Eridani, R. D. O. Augustinus, and M. S. Pakpahan, “Simple LoRa Protocol: Protokol Komunikasi LoRa Untuk Sistem Pemantauan Multisensor,” *TELKA - Telekomunikasi, Elektronika, Komputasi dan Kontrol*, vol. 5, no. 2, pp. 83–92, Nov. 2019, doi: <https://doi.org/10.15575/telka.v5n2.83-92>.