

Komunikasi serial asinkronus dengan modul GPS (Ublox Neo 6M)

Komunikasi Serial

Mari teman-teman kita bersama-sama belajar apa itu komunikasi serial, disini terdapat dua tipe komunikasi serial yaitu asinkron dan sinkron. Pada beberapa percobaan yang lalu kita telah menggunakan komunikasi serial sinkronus seperti I2C dan SPI, untuk link percobaanya seperti dalam list tabel berikut.

I2C	<ul style="list-style-type: none">• Percobaan eksternal sensor• Percobaan display
SPI	<ul style="list-style-type: none">• Percobaan sensor Internal (menggunakan round TFT)

Beberapa pengertian tentang komunikasi serial yang kami rangkum dari laman <https://randomnerdtutorial.com>.

Asinkron Komunikasi asinkron adalah jenis komunikasi serial yang paling umum. Ini digunakan dalam berbagai aplikasi, termasuk koneksi komputer ke perangkat periferal, seperti printer dan modem.Komunikasi asinkron ditandai dengan karakteristik berikut:

- Data ditransmisikan satu byte pada satu waktu.
- Setiap byte data diapit oleh bit start dan stop.
- Tidak ada jam sinyal yang digunakan untuk menyinkronkan pengirim dan penerima.
- Kecepatan data ditentukan oleh baud rate.

Sinkron Komunikasi sinkron adalah jenis komunikasi serial yang digunakan untuk mentransmisikan data secara real time. Ini digunakan dalam berbagai aplikasi, termasuk komunikasi video dan audio. Komunikasi sinkron ditandai dengan karakteristik berikut:

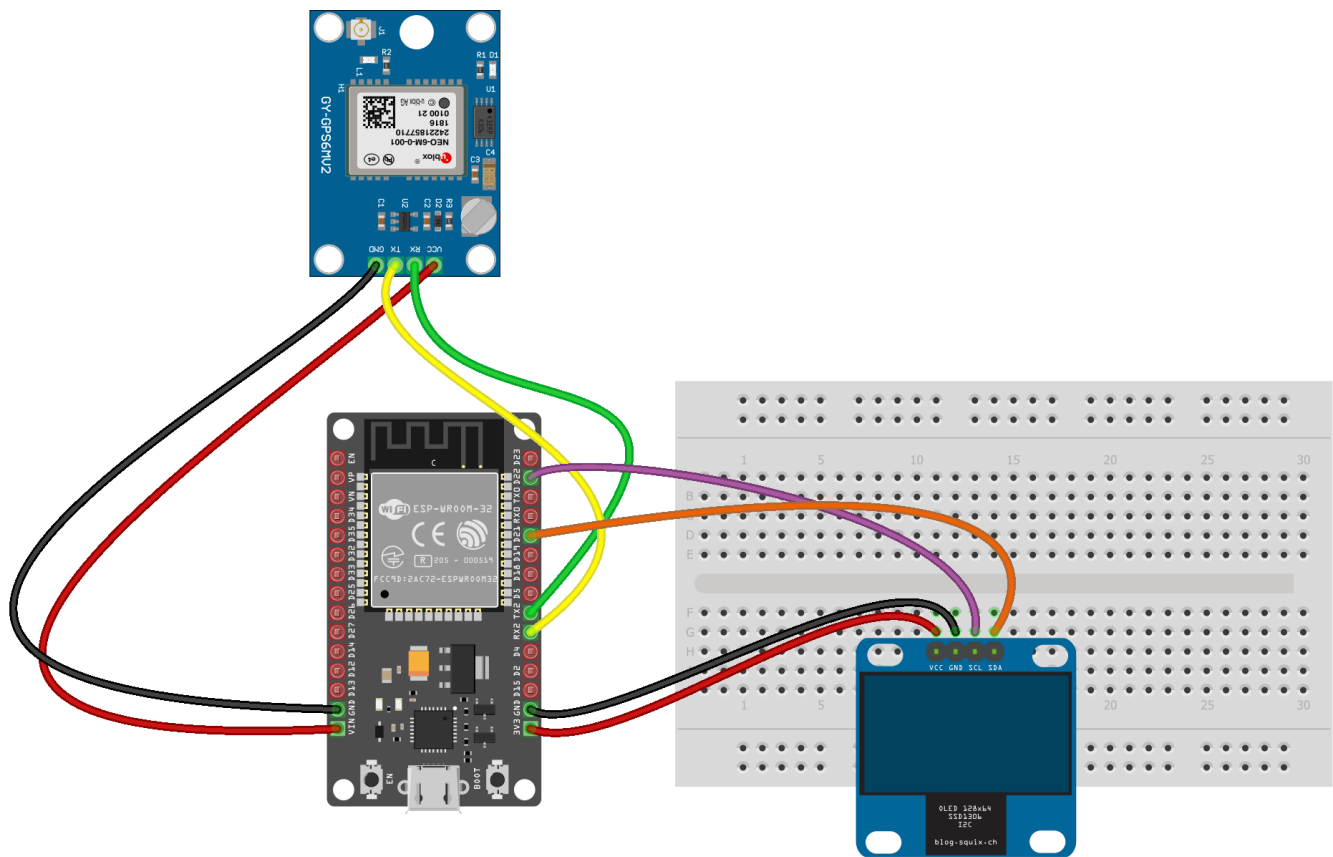
- Data ditransmisikan dalam blok, atau frame.
- Setiap frame data diapit oleh bit start dan stop.
- Jam sinyal digunakan untuk menyinkronkan pengirim dan penerima.
- Kecepatan data ditentukan oleh clock rate.

Jenis komunikasi serial lainnya

Selain komunikasi asinkron dan sinkron, ada beberapa jenis komunikasi serial lainnya, antara lain:

- **Komunikasi half-duplex** adalah jenis komunikasi serial di mana pengirim dan penerima dapat berbagi saluran komunikasi. Namun, hanya satu pihak yang dapat mentransmisikan data pada satu waktu.
- **Komunikasi full-duplex** adalah jenis komunikasi serial di mana pengirim dan penerima dapat mentransmisikan data secara bersamaan.
- **Komunikasi simplex** adalah jenis komunikasi serial di mana hanya satu pihak yang dapat mentransmisikan data. Pihak lain hanya dapat menerima data.

Percobaan kita yang lalu, telah berhasil menggunakan serial komunikasi I2C dan SPI, dimana keduanya adalah tipe komunikasi serial sinkronus yang semuanya disinkronisasi oleh sinyal waktu (Clock atau Serial Clock). Pada kesempatan kali ini, kami tertantang untuk mencoba membuat contoh rangkaian untuk berkomunikasi secara asinkronus dengan protokol UART. Kebetulan kami memiliki bekas modul GPS yang dulu digunakan untuk rangkaian penjejak pergerakan kapal, dan sepertinya masih berfungsi. berikut adalah sekema rangkaian yang kita gunakan.



fritzing

Berikut adalah kode *sketch* yang akan kita gunakan untuk menjalankan percobaan kita. Idenya adalah menampilkan informasi dari modul GPS ke display OLED.

```
/* Sketch Kode untuk ESP32 - GPS */
```

```
/* by Sandi Wibowo*/
```

```
#include <TinyGPSPlus.h>
```

```
#include <SPI.h>
```

```
#include <Wire.h>
```

```
#include <Adafruit_GFX.h>
```

```
#include <Adafruit_SSD1306.h>
```

```
TinyGPSPlus gps;
```

```
#define TX_PIN 17
```

```
#define RX_PIN 16
```

```
#define SCREEN_WIDTH 128
```

```
#define SCREEN_HEIGHT 64
```

```
#define OLED_RESET -1
```

```
#define SCREEN_ADDRESS 0x3C
```

```
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);
```

```
void setup() {
```

```
    Serial.begin(115200);
```

```
    delay(1000);
```

```
    Serial2.begin(9600, SERIAL_8N1, RX_PIN, TX_PIN);
```

```
    if(!display.begin(SSD1306_SWITCHCAPVCC, SCREEN_ADDRESS)) {
```

```
        Serial.println(F("SSD1306 allocation failed"));
```

```
        for(;;);
```

```
    }
```

```
    display.display();
```

```
    delay(5000);
```

```
    delay(2000);
```

```
    display.clearDisplay();
```

```
    display.setTextColor(WHITE);
```

```
    display.setTextSize(1);
```

```
    display.setCursor(25, 31);
```

```
    display.print("SMART-X ITB");
```

```
    display.display();
```

```
    delay(5000);
```

```
    display.clearDisplay();
```

```
    delay(500);
```

```
}
```

```
void updateSerial(){
```

```
    delay(500);
```

```
    while (Serial.available()) {
```

```
        Serial2.write(Serial.read()); //Forward what Serial received to Software Serial Port
```

```
    }
```

```
    while (Serial2.available()) {
```

```
        Serial.write(Serial2.read()); //Forward what Software Serial received to Serial Port
```

```
    }
```

```
}
```

```
void displayInfo()
```

```
{
```

```
    Serial.print(F("Location: "));
```

```
if (gps.location.isValid()){
    Serial.print(gps.location.lat(), 6);
    Serial.print(F(", "));
    Serial.print(gps.location.lng(), 6);
    Serial.println();
}
else
{
    Serial.print(F("INVALID"));
}
}
```

```
void tampilkanOled() {
    display.clearDisplay();
    display.setCursor(0, 0);
    display.print("SMART-X | GPS Rec");
    if (gps.location.isValid()){
        display.setCursor(0,11);
        display.print("Lat : ");
        display.print(gps.location.lat(),7);
        display.setCursor(0,21);
        display.print("Lon : ");
        display.print(gps.location.lng(),6);
        display.setCursor(0,31);
        display.print("Sat : ");
        display.print(gps.satellites.value());
        display.setCursor(55,31);
        display.print("Alt : ");
        display.print(gps.altitude.meters());
        display.setCursor(0,41);
        display.print("HDOP : ");
        display.print(gps.hdop.hdop());
        display.setCursor(0,51);
        display.print("V : ");
        display.print(gps.speed.kmph());
        display.setCursor(55,51);
        display.print("Dir : ");
        display.print(gps.course.deg());
    } else {
        display.setCursor(0,11);
```

```

    display.print("Invalid");
}
display.display();
}

void loop() {
    while (Serial2.available() > 0)
        if (gps.encode(Serial2.read()))
            tampilkanOled();
    if (millis() > 5000 && gps.charsProcessed() < 10)
    {
        Serial.println(F("ESP 32, Tidak terhubung Sensor"));
        while (true);
    }
}

```

Hasil running *sketch* diatas, secara lengkap ditampilkan dalam video berikut.

<https://www.youtube.com/embed/kDVRbE0qMQs>

Debug komunikasi serial modul Ublox Neo 6M

- **Kesalahan tegangan Vcc modul Ublox Neo 6M** semula saya kira adalah 3.3V berdasarkan *datasheet* yang saya baca ([Datasheet](#)). Dan ternyata yang saya baca adalah tipe Neo 6G, sementara untuk varian 6M supply dapat diberikan dari 2.7 sampai dengan 36 Volt.
- **Kesalahan pemasangan kabel komunikasi**, yang semula saya lakukan adalah pin **Rx** GPS saya sambungkan ke pin **Rx** ESP 32 dan sebaliknya, ternyata saya baru ingat ini asinkronus dan *full-duplex*, jadi Received (**Rx**) GPS adalah hasil dari Tranceive (**Tx**) ESP32, dan sebaliknya Tranceive (**Tx**) GPS akan diterima sebagai Receive (**Rx**) di ESP32.
- **Masalah indoor reciever**, modul GPS tidak akan memberikan indikator led-blink jika dianggap kalkulasi posisi nya belum fix, sehingga status `gps.location.isValid()` bernilai 0 ketika satelit belum cukup, proses validasi ini memakan waktu cukup lama, sampai sempat saya memvonis modul GPS ini Rusak.