

Antarmuka ublox NEO-6M GPS Module dengan NodeMCU ESP8266

Taryana Suryana
081221480577

Teknik Informatika

Universitas Komputer Indonesia
Jln.Dipatiukur 112-114 Bandung

taryanarx@email.unikom.ac.id – taryanarx@gmail.com

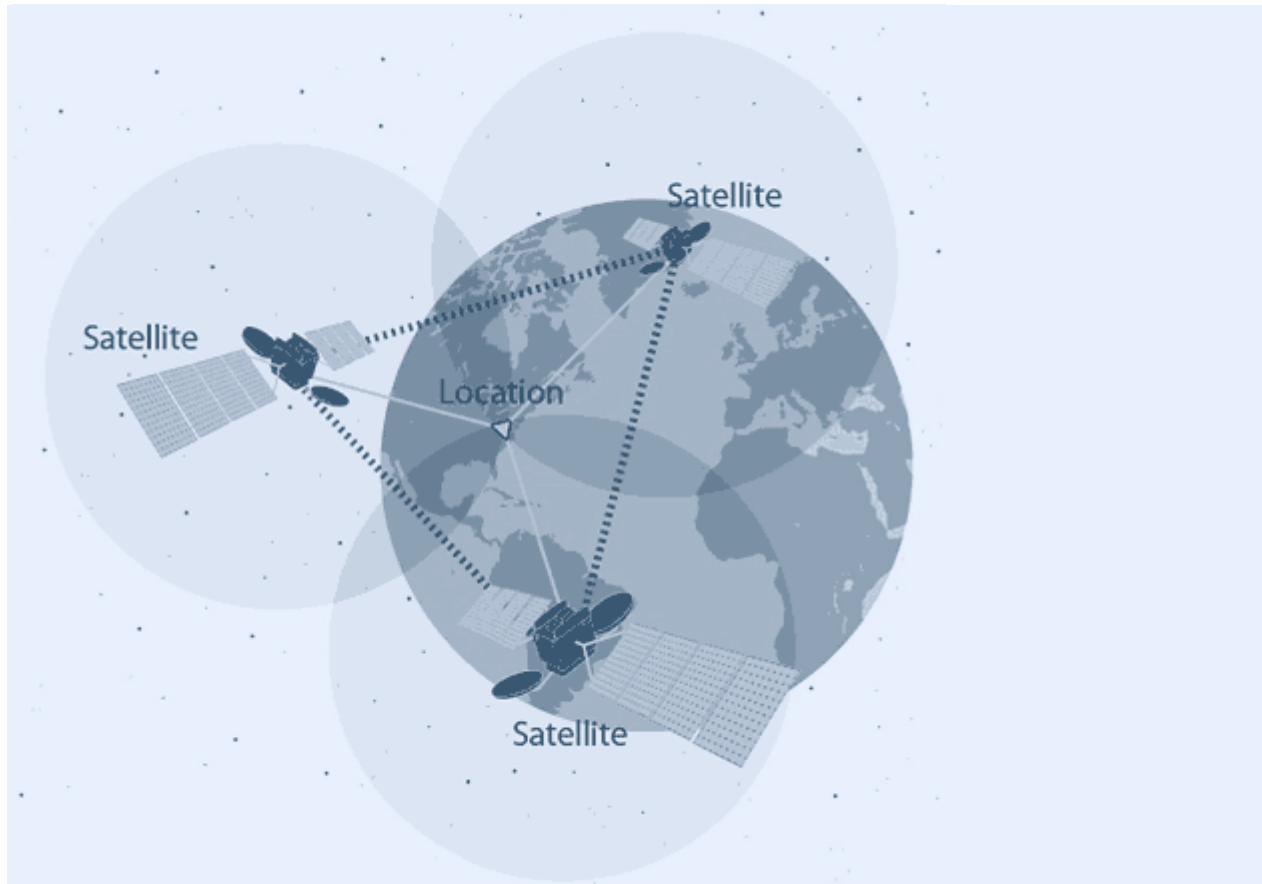
[Jurnal Komputa Unikom 2021](#)

Jika anda ingin menambahkan kemampuan mendeteksi lokasi pada aplikasi yang dibangun maka anda dapat melakukannya dengan menambahkan Modul GPS NEO-6M, modul ini yang dapat melacak hingga 22 satelit dan mengidentifikasi lokasi di mana pun di dunia. Modul ini memiliki daya rendah (cocok untuk perangkat bertenaga baterai), murah, mudah untuk berinteraksi dan sangat populer di kalangan penggemar.

Bagaimana cara kerja GPS?

Penerima GPS sebenarnya bekerja dengan mencari tahu seberapa jauh mereka dari sejumlah satelit. Mereka diprogram untuk mengetahui di mana satelit GPS berada pada waktu tertentu.

Satelit mengirimkan informasi tentang posisi mereka dan waktu saat ini dalam bentuk sinyal radio menuju Bumi. Sinyal-sinyal ini mengidentifikasi satelit dan memberitahu penerima di mana mereka berada.



Penerima kemudian menghitung seberapa jauh setiap satelit dengan mencari tahu berapa lama waktu yang dibutuhkan untuk sinyal tiba. Setelah memiliki informasi tentang seberapa jauh setidaknya tiga satelit dan di mana mereka berada di luar angkasa, ia dapat menentukan lokasi Anda di Bumi. Proses ini dikenal sebagai Trilaterasi.

Tinjauan Perangkat Keras Modul GPS NEO-6M

Chip GPS NEO-6M

Inti dari modul ini adalah chip GPS NEO-6M dari u-blox. Chip ini berukuran kurang dari ukuran prangko tetapi mengemas sejumlah fitur yang mengejutkan ke dalam bingkai kecilnya.



Gambar 1. Neo-6m GPS Chip

Modul ini dapat melacak hingga 22 satelit pada 50 saluran dan mencapai tingkat sensitivitas tertinggi di industri yaitu pelacakan -161 dB, sementara hanya menggunakan arus suplai 45mA.

Tidak seperti modul GPS lainnya, ia dapat melakukan pembaruan lokasi hingga 5 detik dengan akurasi posisi Horizontal 2.5m. Mesin pemosisian u-blox 6 juga menawarkan Time-To-First-Fix (TTFF) di bawah 1 detik.

Salah satu fitur terbaik yang disediakan chip ini adalah *Power Save Mode (PSM)*. Hal ini memungkinkan pengurangan konsumsi daya sistem dengan secara selektif mengalihkan bagian penerima ON dan OFF. Ini secara dramatis mengurangi konsumsi daya modul menjadi hanya 11mA sehingga cocok untuk aplikasi sensitif daya seperti jam tangan GPS.

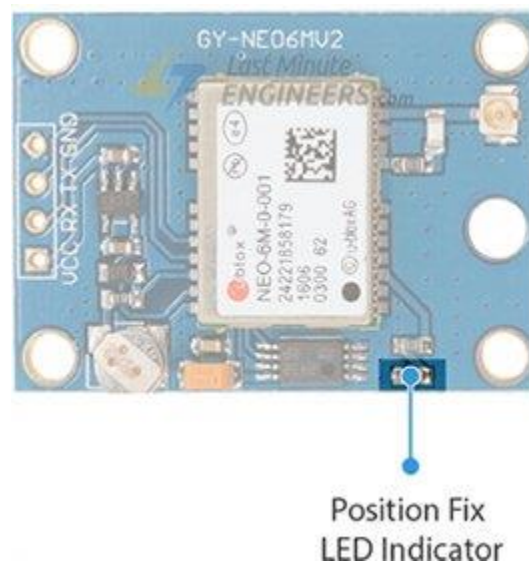
Pin data yang diperlukan dari chip GPS NEO-6M dipecah menjadi header pitch 0,1". Ini termasuk pin yang diperlukan untuk komunikasi dengan mikrokontroler melalui UART. Modul ini mendukung baud rate dari 4800bps hingga 230400bps dengan baud default 9600.

Berikut spesifikasi lengkapnya:

Jenis Penerima	50 saluran, GPS L1 (1575.42Mhz)
Akurasi Posisi Horizontal	2.5m
Tingkat Pembaruan Navigasi	1Hz (maksimum 5Hz)
Waktu Tangkap	Awal yang keren: 27s Hot start: 1s
Sensitivitas Navigasi	-161dBm
Protokol komunikasi	NMEA, Biner UBX, RTCM
Tingkat Baud Seri	4800-230400 (default 9600)

Suhu Operasional	-40 °C ~ 85 °C
Tegangan Operasi	2.7V ~ 3.6V
Operasi Saat Ini	45mA
Impedansi TXD/RXD	510Ω

Indikator LED Perbaiki Posisi



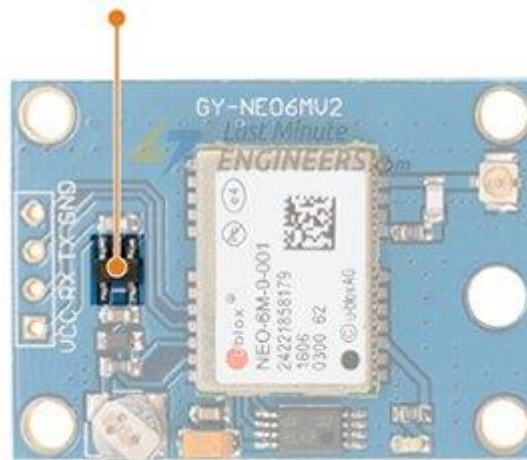
Gambar 2. Posisi Led Indikator

Terdapat LED pada Modul GPS NEO-6M yang menunjukkan status Position Fix. Ini akan berkedip pada berbagai tingkat tergantung pada statusnya:

1. Tidak Berkedip – Ini mencari satelit.
2. Berkedip setiap 1 detik – Perbaiki Posisi ditemukan (Modul dapat melihat cukup banyak satelit).

Pengatur LDO 3.3V

3.3V LDO Regulator

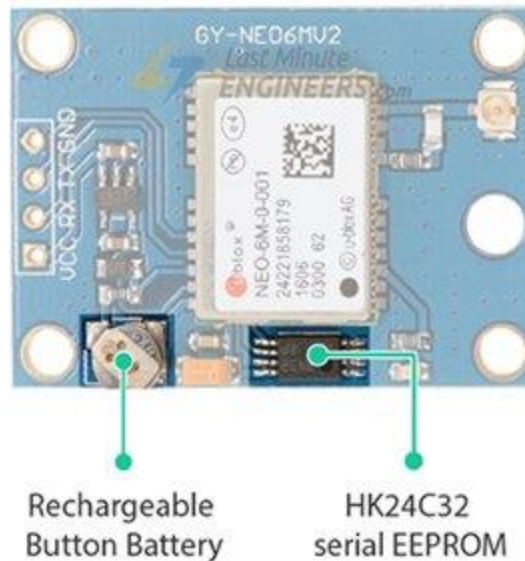


Gambar 3. Regulator Pengatur Tegangan

Tegangan operasi chip NEO-6M adalah dari 2,7 hingga 3,6V. Tetapi kabar baiknya adalah, modul ini dilengkapi dengan regulator 3V3 dropout ultra-rendah MIC5205 dari [MICREL](#) .

Pin logika juga toleran 5 volt, sehingga kita dapat dengan mudah menghubungkannya ke Arduino atau mikrokontroler logika 5V tanpa menggunakan konverter level logika.

Baterai & EEPROM



Rechargeable
Button Battery

HK24C32
serial EEPROM

Gambar 4. Battery dan EEPROM

Modul ini dilengkapi dengan EEPROM serial dua kawat HK24C32 . Ini berukuran 4KB dan terhubung ke chip NEO-6M melalui I2C.

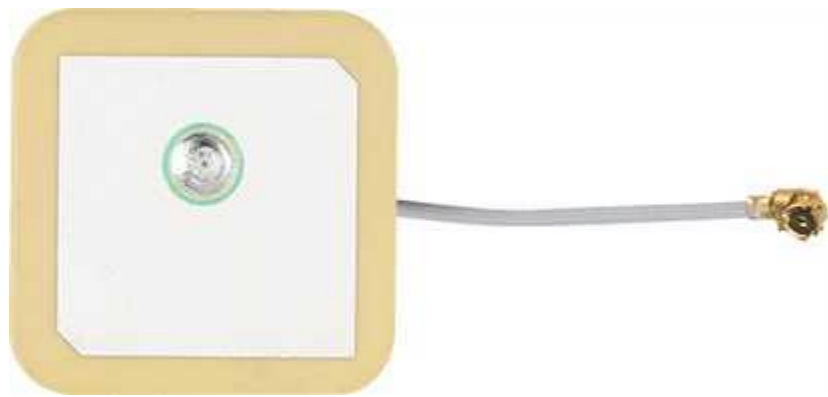
Modul ini juga berisi baterai tombol isi ulang yang bertindak sebagai super-kapasitor. EEPROM bersama dengan baterai membantu mempertahankan RAM yang didukung

baterai (BBR). BBR berisi data jam, data posisi terbaru (data orbit GNSS) dan konfigurasi modul. Tapi itu tidak dimaksudkan untuk penyimpanan data permanen. Karena baterai mempertahankan jam dan posisi terakhir, waktu untuk perbaikan pertama (TTFF) berkurang secara signifikan menjadi 1 detik. Ini memungkinkan penguncian posisi yang jauh lebih cepat. Tanpa baterai, GPS selalu start dingin sehingga penguncian GPS awal membutuhkan waktu lebih lama.

Baterai terisi secara otomatis saat daya diterapkan dan menyimpan data hingga dua minggu tanpa daya.

Antena

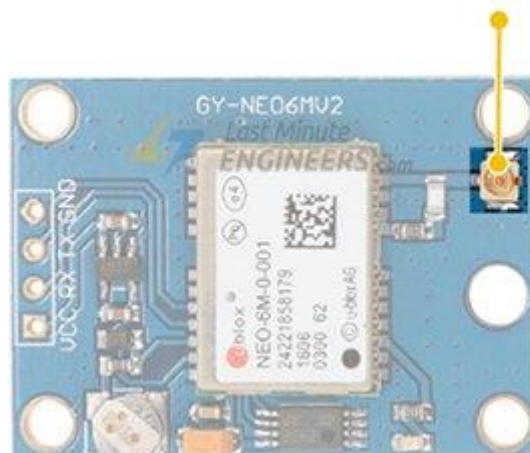
Antena diperlukan untuk menggunakan modul untuk segala jenis komunikasi. Jadi, modul ini dilengkapi dengan antena patch yang memiliki sensitivitas -161 dBm.



Gambar 5. Bentuk Antena

Anda dapat memasang antena ini ke konektor U.FL kecil yang terletak di modul.

U.FL connector



Gambar 6. Konektor U.FL untuk sambungan Antena

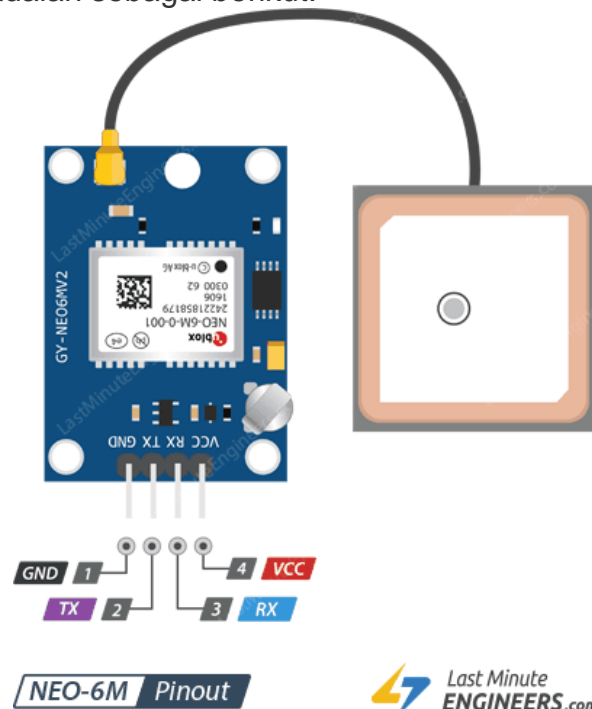
Antena patch sangat bagus untuk sebagian besar proyek. Tetapi jika Anda ingin mendapatkan lebih banyak sensitivitas atau meletakkan modul Anda di dalam wadah logam, Anda juga dapat memasang antena GPS aktif 3V melalui konektor U.FL.

TIP

Konektor U.FL berukuran kecil, halus, dan tidak memiliki nilai tegangan. Untuk mencegah kerusakan koneksi U.FL, kami sarankan untuk memasang kabel U.FL melalui lubang pemasangan, lalu pasang konektor U.FL.

Pinout Modul GPS NEO-6M

Modul GPS NEO-6M memiliki total 4 pin yang menghubungkannya ke dunia luar. Sambungannya adalah sebagai berikut:



Gambar 7. PinOut NEO-6M

GND adalah Ground Pin dan perlu dihubungkan ke pin GND di Arduino.

TxD (Pemancar) pin digunakan untuk komunikasi serial.

RxD (Penerima) pin digunakan untuk komunikasi serial.

VCC memasok daya untuk modul. Anda bisa langsung menghubungkannya ke pin 5V pada Arduino.

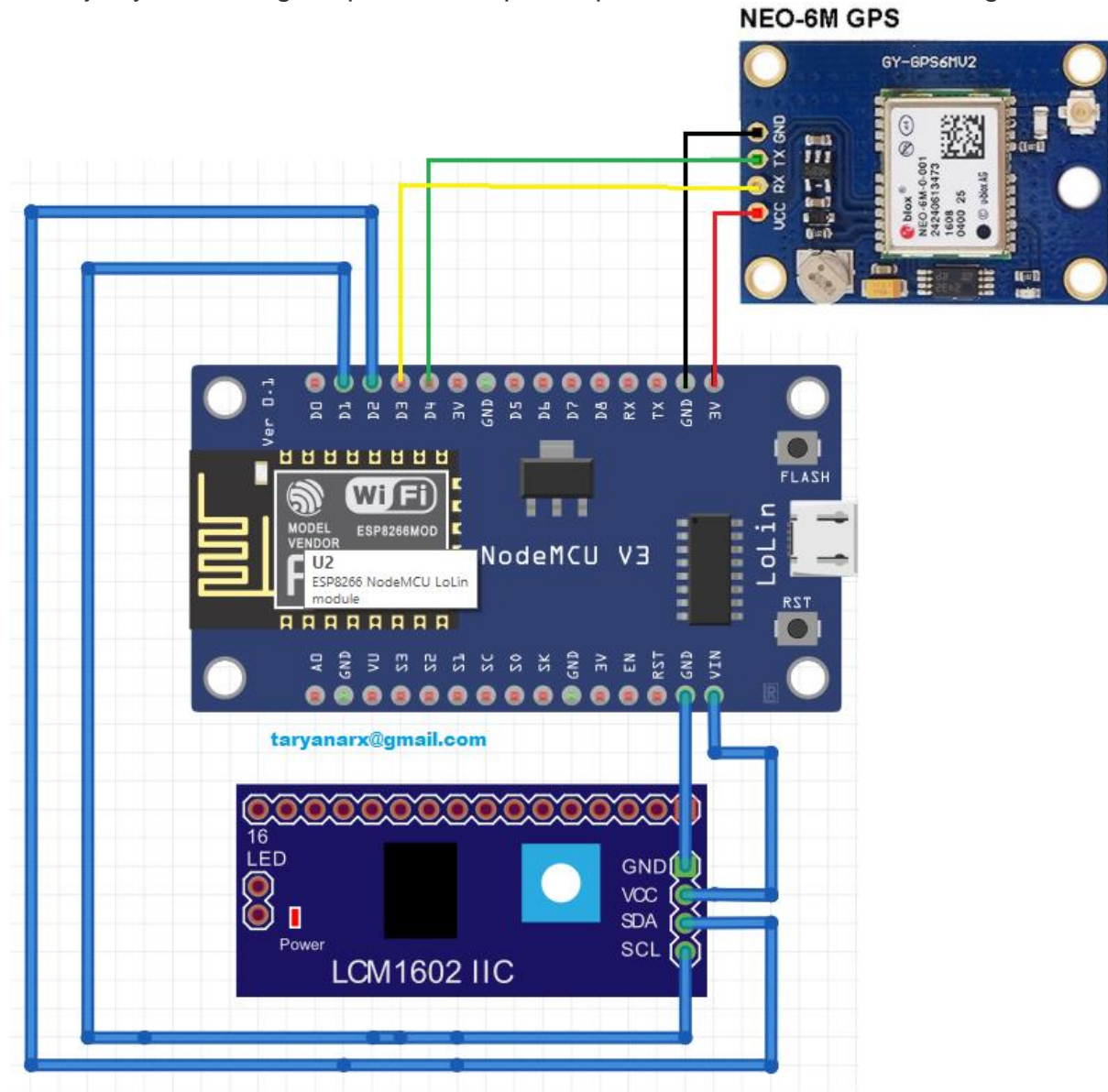
Pengkabelan modul GPS NEO-6M dengan NodeMCU ESP8266

Sekarang setelah kita mengetahui segalanya tentang modul, kita dapat mulai menghubungkannya ke NodeMCU kita!

Mulailah dengan menghubungkan antena patch ke konektor U.FL. Ingatlah untuk memasang kabel U.FL melalui salah satu lubang pemasangan untuk koneksi yang kuat. Modul ketika membelinya biasanya dilengkapi dengan pin header yang belum disolder. Jadi, Anda harus menyoldernya.

Sekarang, sambungkan pin Rx dan Tx pada modul ke pin digital D3 dan D4 masing-masing pada NodeMCU; karena kita akan menggunakan serial perangkat lunak untuk berbicara dengan modul.

Selanjutnya sambungkan pin VCC ke pin 3V pada NodeMCU dan GND ke ground.



Gambar 8. Menhubungkan Modul GPS NEO-6M ke NodeMCU

Kode Arduino – Membaca Data GPS

Hal terbaik tentang penerima GPS apa pun adalah mereka mulai mengeluarkan data segera setelah Anda menghidupkannya.

Cara terbaik untuk menguji data ini adalah dengan menggunakan Arduino sebagai USB to TTL Converter.

Program berikut melakukan hal itu.

```
#include <TinyGPS++.h>
#include <SoftwareSerial.h>
int RXPin = D3;
int TXPin = D4;

//Default baud of NEO-6M is 9600
int GPSPbaud = 9600;

// Create a software serial port called "gpsSerial"
SoftwareSerial gpsSerial(RXPin, TXPin);

void setup()
{
  // Start the Arduino hardware serial port at 9600 baud
  Serial.begin(9600);

  // Start the software serial port at the GPS's default baud
  gpsSerial.begin(GPSPbaud);
}

void loop()
{
  // Displays information when new sentence is available.
  while (gpsSerial.available() > 0)
    Serial.write(gpsSerial.read());
}
```

Unggah program dan buka monitor serial dari Arduino IDE. Ingatlah untuk memilih 9600 baud. Anda akan melihat teks seperti berikut:

```
$GPGGA,092457.000,1888.8888,N,17818.8888,E,6,06,4.0,453.2,M,0.0,M,,*6D
$GPGLL,1888.8888,N,17818.8888,E,092457.000,A,E*59
$GPGSA,A,3,14,18,32,11,08,27,,,,,,,,,5.3,4.0,3.5*36
$GPGSV,3,1,12,08,16,273,28,10,46,042,,11,07,315,37,14,56,315,41*7E
$GPGSV,3,2,12,18,21,320,41,20,36,081,,21,29,152,,25,10,098,*72
$GPGSV,3,3,12,26,06,185,,27,23,241,26,31,54,182,,32,50,353,34*7D
$GPRMC,092457.000,A,1888.8888,N,17818.8888,E,2.33,332.91,311018,,,E*6C
$GPVTG,332.91,T,,M,2.33,N,4.32,K,E*34
$GPZDA,092457.000,31,10,2018,00,00*53
$GPTXT,01,01,01,ANTENNA OK*35
```

Data yang Anda peroleh melalui antarmuka serial sebenarnya adalah kalimat NMEA . NMEA adalah singkatan dari [National Marine Electronics Association](https://www.nmea.org/) . Ini adalah format pesan standar untuk Hampir semua penerima GPS.

Standar NMEA diformat dalam baris data yang disebut kalimat. Setiap kalimat dipisahkan koma untuk memudahkan penguraian oleh komputer dan mikrokontroler.

Kalimat NMEA ini dikirim pada interval yang disebut tingkat pembaruan.

Modul GPS NEO-6M memperbarui informasi ini sekali per detik (frekuensi 1Hz) secara default. Tetapi Anda dapat mengonfigurasinya hingga 5 pembaruan per detik (frekuensi 5Hz).

Mengurai Kalimat NMEA

Ada banyak kalimat dalam standar NMEA, yang paling umum adalah:

1. \$GPRMC (Global Positioning Recommended Minimum Coordinates) menyediakan waktu, tanggal, garis lintang, garis bujur, ketinggian, dan perkiraan kecepatan.
2. Kalimat \$GPGGA menyediakan data perbaikan penting yang menyediakan lokasi 3D dan data akurasi.

Mari kita ambil contoh kalimat \$GPRMC NMEA dari penerima GPS.

\$GPRMC, 123519, A, 4807.038, N, 01131.000, E, 022.4, 084.4, 230394, 003.1, W*6A

\$	Setiap kalimat NMEA dimulai dengan karakter \$.
GPRMC	Posisi Global Direkomendasikan Koordinat Minir
123519	Waktu saat ini di UTC – 12:35:19
SEBUAH	Status A=aktif atau V=Void.
4807.038,N	Lintang 48 derajat 07.038' N
01131.000,E	Bujur 11 derajat 31.000' BT
022.4	Kecepatan di atas tanah dalam knot
084.4	Lacak sudut dalam derajat Benar
220318	Tanggal Saat Ini – 22 Maret 2018
003.1,W	Variasi Magnetik
*6A	Data checksum, selalu diawali dengan *

Untuk informasi lebih lanjut tentang kalimat NMEA dan data apa yang dikandungnya, lihat gpsinformation.org

Kode Arduino – Library TinyGPS

Seringkali untuk proyek kita, kita perlu mengurai kalimat NMEA menjadi informasi yang berguna. Untuk menyederhanakan pekerjaan kami, kami memiliki perpustakaan yang disebut perpustakaan TinyGPS++ .

Pustaka ini melakukan banyak pekerjaan berat yang diperlukan untuk menerima data dari modul GPS, seperti membaca dan mengekstrak data yang berguna di latar belakang. Jadi, kita tidak perlu khawatir tentang pekerjaan parsing yang menjijikkan. Terima kasih kepada Mikal Hart atas kontribusinya yang luar biasa. Situs webnya Arduiniana.org memiliki ikhtisar lengkap tentang semua kemampuan perpustakaan TinyGPS++.

Unduh perpustakaan terlebih dahulu, dengan mengunjungi [repo GitHub](#) atau, cukup klik tombol ini untuk mengunduh zip: [TinyGPSPlus.zip](#)
Untuk menginstalnya, buka Arduino IDE, buka Sketch > Include Library > Add .ZIP Library, lalu pilih file TinyGPSPlus ZIP yang baru saja Anda download. Jika Anda memerlukan detail lebih lanjut tentang menginstal perpustakaan, kunjungi tutorial [Memasang Perpustakaan Arduino ini](#) .

Setelah library terinstal, Anda dapat menyalin sketsa di bawah ini ke Arduino IDE. Sketsa pengujian berikut akan mencetak informasi lokasi (Lintang, Bujur & Ketinggian) dan UTC (Tanggal & Waktu) pada monitor serial. Coba sketsanya; dan kemudian kami akan menjelaskannya secara rinci.

```
#include <TinyGPS++.h>
#include <SoftwareSerial.h>

// Choose two Arduino pins to use for software serial
int RXPin = D3;
int TXPin = D4;

int GPSBaud = 9600;

// Create a TinyGPS++ object
TinyGPSPlus gps;

// Create a software serial port called "gpsSerial"
SoftwareSerial gpsSerial(TXPin, RXPin);

void setup()
{
  // Start the Arduino hardware serial port at 9600 baud
  Serial.begin(9600);

  // Start the software serial port at the GPS's default baud
  gpsSerial.begin(GPSBaud);
}

void loop()
{
  // This sketch displays information every time a new sentence is correctly encoded.
  while (gpsSerial.available() > 0)
    if (gps.encode(gpsSerial.read()))
      displayInfo();

  // If 5000 milliseconds pass and there are no characters coming in
  // over the software serial port, show a "No GPS detected" error
  if (millis() > 5000 && gps.charsProcessed() < 10)
  {
    Serial.println("No GPS detected");
    while(true);
  }
}

void displayInfo()
```

```

{
  if (gps.location.isValid())
  {
    Serial.print("Latitude: ");
    Serial.println(gps.location.lat(), 6);
    Serial.print("Longitude: ");
    Serial.println(gps.location.lng(), 6);
    Serial.print("Altitude: ");
    Serial.println(gps.altitude.meters());
  }
  else
  {
    Serial.println("Location: Not Available");
  }

  Serial.print("Date: ");
  if (gps.date.isValid())
  {
    Serial.print(gps.date.month());
    Serial.print("/");
    Serial.print(gps.date.day());
    Serial.print("/");
    Serial.println(gps.date.year());
  }
  else
  {
    Serial.println("Not Available");
  }

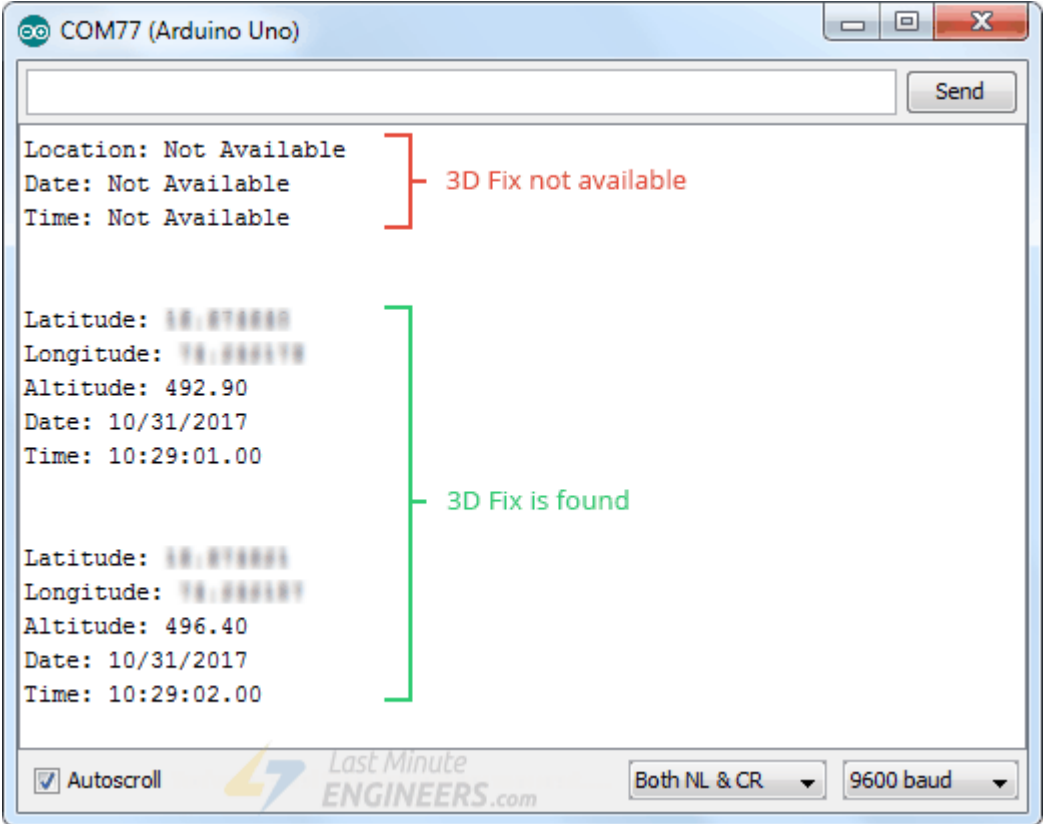
  Serial.print("Time: ");
  if (gps.time.isValid())
  {
    if (gps.time.hour() < 10) Serial.print(F("0"));
    Serial.print(gps.time.hour());
    Serial.print(":");
    if (gps.time.minute() < 10) Serial.print(F("0"));
    Serial.print(gps.time.minute());
    Serial.print(":");
    if (gps.time.second() < 10) Serial.print(F("0"));
    Serial.print(gps.time.second());
    Serial.print(".");
    if (gps.time.centisecond() < 10) Serial.print(F("0"));
    Serial.println(gps.time.centisecond());
  }
  else
  {

```

```
Serial.println("Not Available");
}

Serial.println();
Serial.println();
delay(1000);
}
```

Ini adalah contoh keluaran yang dapat dilihat di serial monior



Penjelasan Kode:

Sketsa dimulai dengan menyertakan perpustakaan TinyGPS++ dan perpustakaan serial perangkat lunak. Kemudian, kami mendefinisikan pin arduino yang terhubung dengan modul GPS NEO-6M dan variabel yang menyimpan baud rate GPS default.

Membuat TinyGPSPPlusobjek akan membantu mengakses fungsi khusus yang terkait dengan Library. Selanjutnya, kita membuat port serial perangkat lunak yang disebut gpsSerial agar kita dapat berkomunikasi dengan modul.

```
#include <TinyGPS++.h>
#include <SoftwareSerial.h>

int RXPin = D3;
int TXPin = D4;
int GPSBaud = 9600;

TinyGPSPPlus gps;
SoftwareSerial gpsSerial(TXPin, RXPin);
```

Dalam fungsi setup tersebut, kita perlu memulai komunikasi serial dengan PC serta modul GPS.

```
void setup()
{
  Serial.begin(9600);
  gpsSerial.begin(GPSBaud);
}
```

Dalam fungsi loop(), kami memanggil fungsi displayInfo(), fungsi kustom yang mencetak informasi lokasi (Lintang, Bujur & Ketinggian) dan UTC (Tanggal & Waktu) pada monitor serial, setiap kali kalimat NMEA baru dikodekan dengan benar.

Jika 5000 milidetik berlalu dan tidak ada karakter yang masuk melalui port serial perangkat lunak, kami menunjukkan kesalahan Tidak Ada GPS yang terdeteksi.

```
void loop()
{
  while (gpsSerial.available() > 0)
    if (gps.encode(gpsSerial.read()))
      displayInfo();

  if (millis() > 5000 && gps.charsProcessed() < 10)
  {
    Serial.println(F("No GPS detected"));
    while(true);
  }
}
```


Fungsi Berguna Lainnya Di Library TinyGPS++

Ada beberapa fungsi berguna yang dapat Anda gunakan dengan objek TinyGPS++. Beberapa di antaranya tercantum di bawah ini:

1. `gps.speed.value()` fungsi mengembalikan kecepatan gerak saat ini dalam 100 knot.
2. `gps.course.value()` fungsi mengembalikan kursus dasar saat ini dalam seperseratus derajat.
3. `gps.satellites.value()` fungsi mengembalikan jumlah satelit yang terlihat dan berpartisipasi.
4. `gps.hdop.value()` fungsi mengembalikan pengurangan presisi horizontal.
5. Jika Anda ingin mengetahui berapa umur data suatu objek, panggil `age()` metodenya, yang mengembalikan jumlah milidetik sejak pembaruan terakhirnya. Jika ini mengembalikan nilai yang lebih besar dari 1500 atau lebih, ini mungkin merupakan tanda masalah seperti perbaikan yang hilang.
6. Jika Anda ingin mengekstrak data dari kalimat NMEA lainnya. Anda dapat menggunakan fungsi ekstraksi kustom perpustakaan dengan memberi tahu TinyGPS++ nama kalimat dan nomor bidang yang Anda minati, seperti ini: `TinyGPSCustom magneticVariation(gps, "GPRMC", 10)` Dan Anda dapat menanyakannya seperti yang lain: `magneticVariation.value()`
7. **Perangkat lunak pusat-U**
8. U-center dari [u-blox](#) adalah alat yang ampuh untuk evaluasi, analisis kinerja dan konfigurasi penerima GPS u-blox termasuk NEO-6M. Ini adalah alat gratis tetapi hanya dapat digunakan pada platform Windows.
9. Ini dapat menampilkan visualisasi data terstruktur dan grafis realtime dari penerima GPS apa pun seperti:
 10. Tampilan ringkasan satelit
 11. Tampilan ringkasan navigasi
 12. Kompas, speedometer, jam, altimeter
 13. Tampilan grafik dari dua parameter pilihan
 14. Fungsi perekaman dan pemutaran data

Perintah Dasar

```
Serial.println(gps.location.lat(), 6); // Latitude in degrees (double)
Serial.println(gps.location.lng(), 6); // Longitude in degrees (double)
Serial.print(gps.location.rawLat().negative ? "-" : "+");
Serial.println(gps.location.rawLat().deg); // Raw latitude in whole degrees
Serial.println(gps.location.rawLat().billionths); // ... and billionths
(u16/u32)
Serial.print(gps.location.rawLng().negative ? "-" : "+");
Serial.println(gps.location.rawLng().deg); // Raw longitude in whole degrees
Serial.println(gps.location.rawLng().billionths); // ... and billionths
(u16/u32)
Serial.println(gps.date.value()); // Raw date in DDMMYY format (u32)
Serial.println(gps.date.year()); // Year (2000+) (u16)
Serial.println(gps.date.month()); // Month (1-12) (u8)
Serial.println(gps.date.day()); // Day (1-31) (u8)
Serial.println(gps.time.value()); // Raw time in HHMMSSCC format (u32)
```

```

Serial.println(gps.time.hour()); // Hour (0-23) (u8)
Serial.println(gps.time.minute()); // Minute (0-59) (u8)
Serial.println(gps.time.second()); // Second (0-59) (u8)
Serial.println(gps.time.centisecond()); // 100ths of a second (0-99) (u8)
Serial.println(gps.speed.value()); // Raw speed in 100ths of a knot (i32)
Serial.println(gps.speed.knots()); // Speed in knots (double)
Serial.println(gps.speed.mph()); // Speed in miles per hour (double)
Serial.println(gps.speed.mps()); // Speed in meters per second (double)
Serial.println(gps.speed.kmph()); // Speed in kilometers per hour (double)
Serial.println(gps.course.value()); // Raw course in 100ths of a degree (i32)
Serial.println(gps.course.deg()); // Course in degrees (double)
Serial.println(gps.altitude.value()); // Raw altitude in centimeters (i32)
Serial.println(gps.altitude.meters()); // Altitude in meters (double)
Serial.println(gps.altitude.miles()); // Altitude in miles (double)
Serial.println(gps.altitude.kilometers()); // Altitude in kilometers (double)
Serial.println(gps.altitude.feet()); // Altitude in feet (double)
Serial.println(gps.satellites.value()); // Number of satellites in use (u32)
Serial.println(gps.hdop.value()); // Horizontal Dim. of Precision (100ths-
i32)

```

Contoh Sketch Program

```

//*****
//NamaFile:gps
//Menampilkan Informasi Lat/Lon dengan NEO-6M GPS Modul
//taryanarx@gmail.com
//*****
#include <Wire.h> // Library komunikasi I2C
#include <LiquidCrystal_I2C.h> // Library modul I2C LCD

#include <TinyGPS++.h>
#include <SoftwareSerial.h>

// default address 0x27
// tipe LCD 16x2 (16,2)
LiquidCrystal_I2C lcd = LiquidCrystal_I2C(0x27, 16, 2);

static const int RXPin = D3, TXPin = D4;
static const uint32_t GPSPbaud = 9600;

// The TinyGPS++ object
TinyGPSPlus gps;

// The serial connection to the GPS device
SoftwareSerial ss(TXPin, RXPin);

void setup(){
  lcd.init();
  lcd.backlight();

```

```
lcd.clear();
//      1234567890123456
lcd.print(" Radio Amatur");
lcd.setCursor(0, 1);
lcd.print("YC1JEA");
Serial.begin(9600);
ss.begin(GPSBaud);
}

void loop(){
// This sketch displays information every time a new sentence is correctly encoded.

while (ss.available() > 0){
  gps.encode(ss.read());
  if (gps.location.isUpdated()){
    lcd.clear();
    //      1234567890123456
    lcd.print("Lat:");
    lcd.print(gps.location.lat(), 6);

    lcd.setCursor(0, 1);
    lcd.print("Lon:");
    lcd.print(gps.location.lng(), 6);
  }
}
}
```

Daftar Pustaka

<https://lastminuteengineers.com/neo6m-gps-arduino-tutorial/>
<http://arduinoiana.org/libraries/tinygpsplus/>