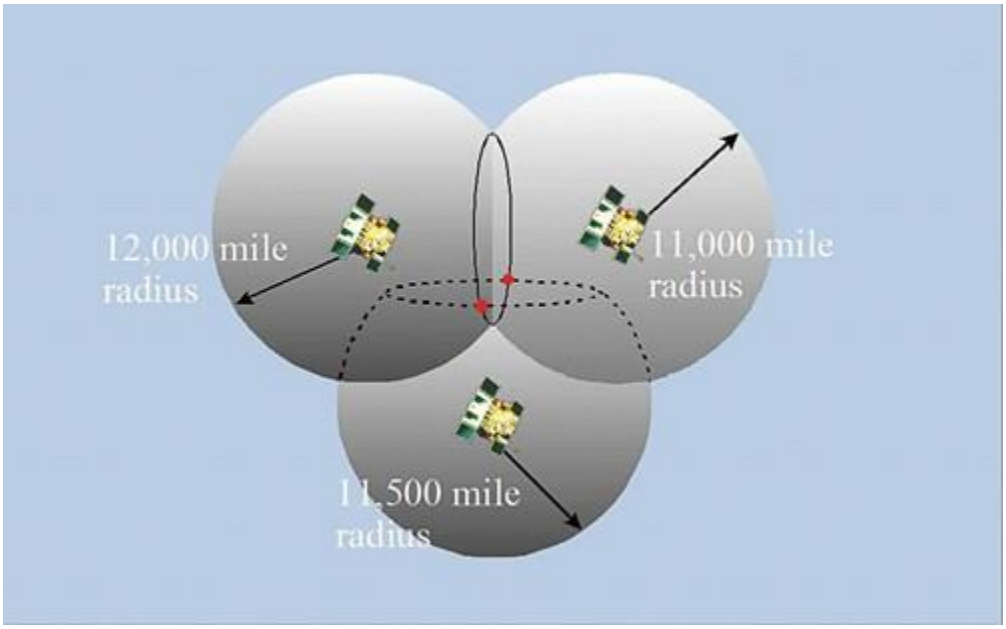
Kết nối bo mạch định vị GPS

1. Nguyên lý xác định vị trí bằng GPS

Hệ thống định vị GPS (Global Positioning System) là một hệ thống định vị vận hành dựa vào 27 vệ tinh (trong đó có 3 vệ tinh dự phòng) chuyển động trên các quỹ đạo quanh trái đất, do Mỹ phát triển ban đầu cho mục đích quân sự, nay đã mở rộng ra cho cả dân sự. Các vệ tinh được bố trí sao cho tại bất kỳ thời điểm nào và ở nơi nào trên mặt đất, cũng có thể thấy ít nhất 4 vệ tinh.



Nguyên lý: 3 mặt cầu giao nhau tại 2 điểm trong đó 1 điểm là vị trí của máy thu GPS trên mặt đất. Điểm giao cắt thứ hai là một nơi nào đó lơ lửng trong không gian, cách xa trái đất hàng ngàn km nên có thể bỏ qua.

Vệ tinh phát ra các tín hiệu gồm vị trí của chúng và thời điểm phát tín hiệu. Nhiệm vụ của máy thu GPS là xác định vị trí của 4 vệ tinh, tính toán khoảng cách tới các vệ tinh để từ đó tự xác định vị trí của chính nó theo công thức:

Quãng đường = Vận tốc x Thời gian.

Trên lý thuyết thì chỉ cần có 3 vệ tinh là đủ nhưng để cải thiện tính chính xác của việc xác định thời gian cần sử dụng thêm một vệ tinh thứ 4, vì chỉ cần sai số 1

phần triệu giây giữa đồng hồ vệ tinh và máy thu cũng có thể dẫn đến định vị sai lệch hàng trăm mét.

Như vậy để xác định vị trí của mình trên mặt đất, máy thu GPS phải tính để biết khoảng cách tới 4 vệ tinh và vị trí chính xác của các vệ tinh trên quỹ đạo.

Cách tính toán GPS

- **Vận tốc (~ vận tốc ánh sánh):** Vệ tinh sử dụng sóng radio tần số cao để truyền tín hiệu, tốc độ của sóng này tương đương tốc độ của ánh sáng khoảng 300.000km/giây trong chân không (Giả định là tín hiệu được truyền thẳng, bỏ qua lực cản không khí và sự phản xạ tín hiệu khi gặp các vật chắn lớn như các dãy núi, tòa nhà cao tầng..)
- **Thời gian:** Vào một thời điểm nào đó trong ngày, một vệ tinh bắt đầu truyền một chuỗi dài tín hiệu số, được gọi là mã giả ngẫu nhiên. Cùng lúc, máy thu cũng bắt đầu tạo ra chuỗi mã giống hệt, sau đó một chút mới nhận được chuỗi tín hiệu của vệ tinh. So sánh 2 chuỗi mã này ta xác đinh được khoảng thời gian (độ trễ) truyền tín hiệu từ vệ tinh tới máy thu.

Để đo chính xác đồng hồ sử dụng trên vệ tinh là đồng hồ nguyên tử có độ chính xác cao. Còn đồng hồ ở máy thu thường là đồng hồ quartz, và chúng được hiệu chỉnh liên tục dựa vào tín hiệu nhận được từ các vệ tinh để đồng bộ thời gian chính xác theo đồng hồ nguyên tử trên vệ tinh.

- **Khoảng cách tới 4 vệ tinh:** Có được thời gian và vận tốc ta có thể xác định được khoảng cách của 4 vệ tinh đến máy thu GPS.
- **Xác định vị trí vệ tinh:** Mỗi máy thu đều cập nhật và lưu trữ định kỳ một bảng tra cứu (gọi là almanac data) vị trí gần đúng của từng vệ tinh chuyển động trên quỹ đạo vào bất kỳ thời điểm nào.

Tính được khoảng cách tới 4 vệ tinh và vị trí chính xác của các vệ tinh trên quỹ đạo máy thu GPS có thể cho chúng ta biết được kinh độ, vĩ độ, và cao độ của vị trí hiện tại. Sau đó sử dụng những dữ liệu này vào các mục đích như hiển thị lên bảng đồ google, tính toán vận tốc, xác định hướng đi,…

2. Hệ thống định vị GPS

Một hệ thống định vị GPS bao gồm:

- **Phần vũ trụ:**

Gồm 24 vệ tinh (3 vệ tinh dự phòng) quay xung quanh trái đất 2 lần trong ngày quỷ đạo rất chính xác. Độ cao của vệ tinh so với mặt đất là 20.183 km. chu kỳ quy xung quanh trái đất là 11 giờ 57`58″. Phần vũ trụ sẽ đảm bảo cho bất kỳ vị trí nào trên trái dất đều có thể quan sát đươc 4 vệ tinh ở góc độ 15 độ (Nếu góc ở ngưỡng 10 độ thì có thể quan sát được 10 vệ tinh và ở góc ngưỡng 5 độ có thể quan sát được 12 vệ tinh).

- Nhiệm vụ:
  - Ghi nhận và lưu trữ các thông tin truyền đi từ phần điều khiển
  - Sử dụng dữ liệu có chọn lọc trên vệ tinh.
  - Duy trì chính xác độ cao của thời gian bằng các đồng hồ nguyên tử.
  - Chuyển tiếp thông tin đến người dùng.
  - Thay đổi quỹ đạo bay của vệ tinh theo sự điều khiển của mặt đất.

- **Phần điều khiển:**

Gồm một trạm điều khiển chính, 5 trạm thu số liệu, 3 trạm truyền số liệu.

Trạm diều khiển chính đặt tại Colorade Spring (Mỹ) có nhiệm vụ thu thập các dữ liệu theo dõi vệ tinh từ các trạm thu số liệu để xử lý. Công nghệ xử lý gồm : Tính lịch thiên văn, tính và hiệu chỉnh đồng hồ, hiệu chỉnh quỹ đạo điều khiển, thay thế các vệ tinh ngừng hoạt động bằng các vệ tinh dự phòng.

5 trạm thu dữ liệu được đặt tại Hawai, Collrade Sping , Ascesion (Ban Đại Tây Dương), Diago Garia (Ấn Độ Dương), Kwayalien (Nam Thái Bình Dương). Có nhiệm vụ theo dõi các tín hiệu vệ tinh để kiểm soát và dự đoán quỹ đạo của chúng. Mỗi trạm được trang bị những máy thu P-code để thu các tín hiệu của vệ tinh, sau đó truyền về trạm điều khiển chính.

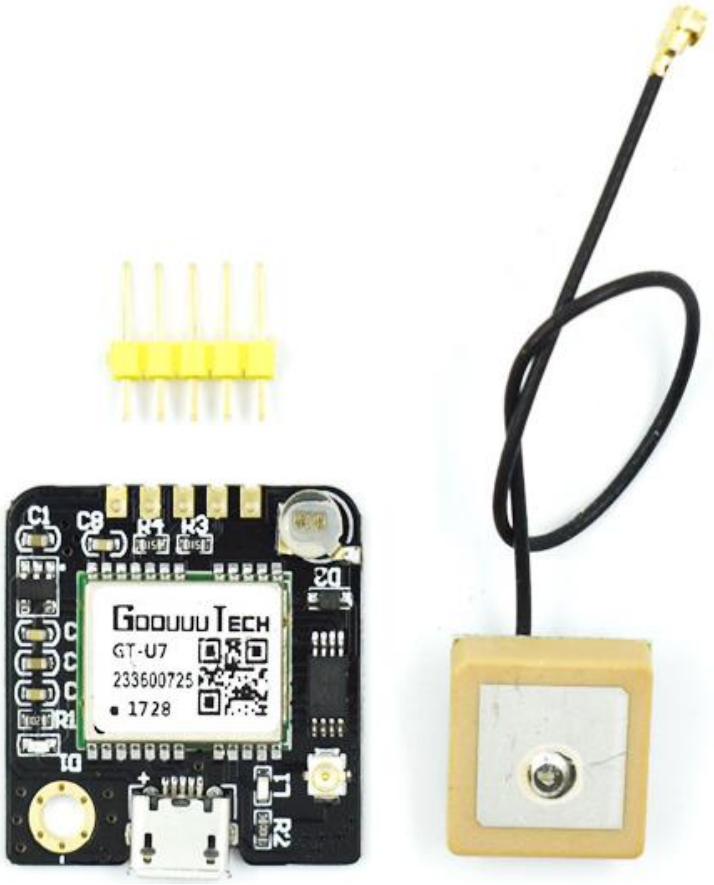
3 trạm truyền số liệu đặt tại Ascensionm Diago Garia , Kwayalein có khả năng chuyển số liệu lên vệ tinh gồm lịch thiên văn mới, hiệu chỉnh đồng hồ, các thông điệp cần phát các lệnh điều khiển từ xa.

- **Phần sử dụng:**

Là thiết bị thu nhận và sử dụng tín hiệu GPS có mục đích. Thiết bị này bao gồm phần cứng để thu nhận sóng, phần mềm để giải mã sóng, tính toán, và phần giao diện như ứng dụng google map, phần mềm la bàn, thiết bị định vị,vv…

3. Mô-đun định vị GPS

Trên thị trường có rất nhiều loại mô-đun GPS khác nhau, nhưng thiết bị cần độ chính xác cao như dùng cho quan trắc, vẽ bản đồ thì giá thành rất đắt. Tuy nhiên bên cạnh đó có những mô-đun phục vụ học tập, phục vụ đời sống thì độ chính xác vừa phải và giá thành thấp hơn nhiều.



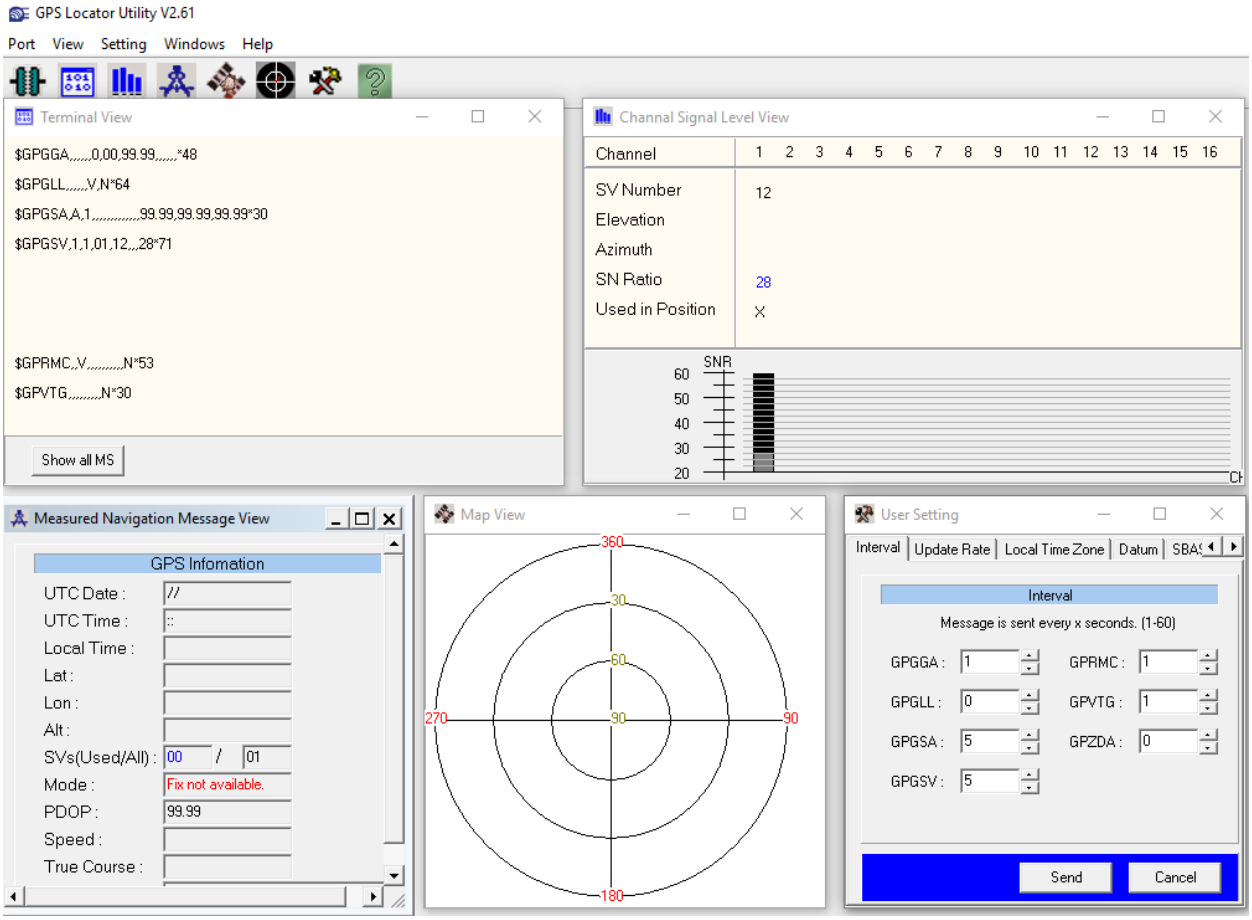
Mô-đun định vị GPS sử dụng trong bài học là mô-đun NEO-6M với những tính năng sau:

- Điện áp nguồn cấp: từ 3.6V-5V (hoặc lấy trực tiếp từ nguồn USB).
- Giao tiếp UART tốc độ 9600 (có thể điều chỉnh được).
- Antenna chuẩn IPEX giúp kết nối với vệ tinh nhanh hơn.
- Có Pin sạc được tích hợp trên bo mạch.
- Tích hợp bộ nhớ EEPROM cho phép lưu trữ các thông số thiết lập từ người dùng.

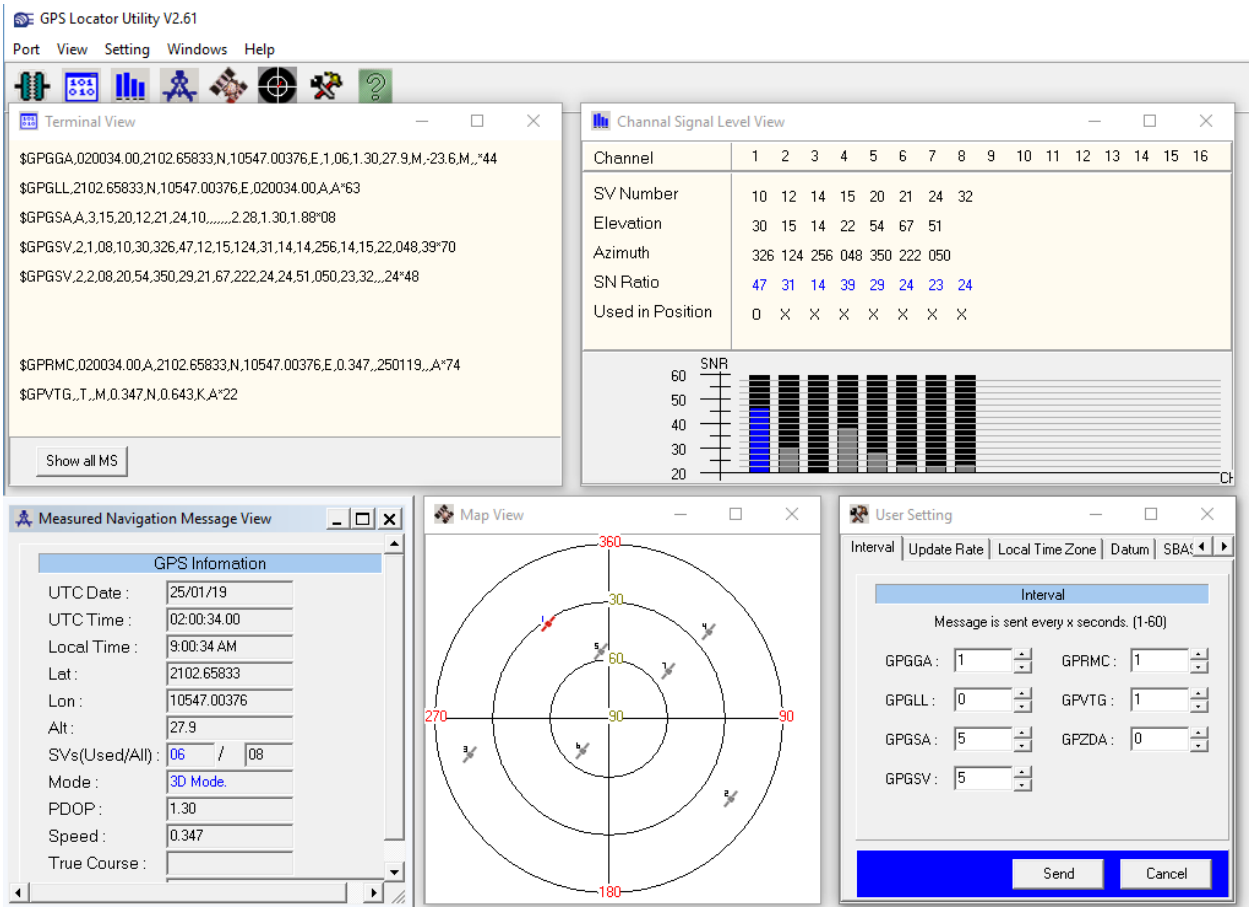
Sử dụng phần mềm GPS Locator Utility để quan sát dữ liệu gửi lên từ mô-đun GPS NEO-6M tại địa chỉ: http://track.sanav.com/software.htm

Cài đặt phần mềm với license miễn phí: **demover**

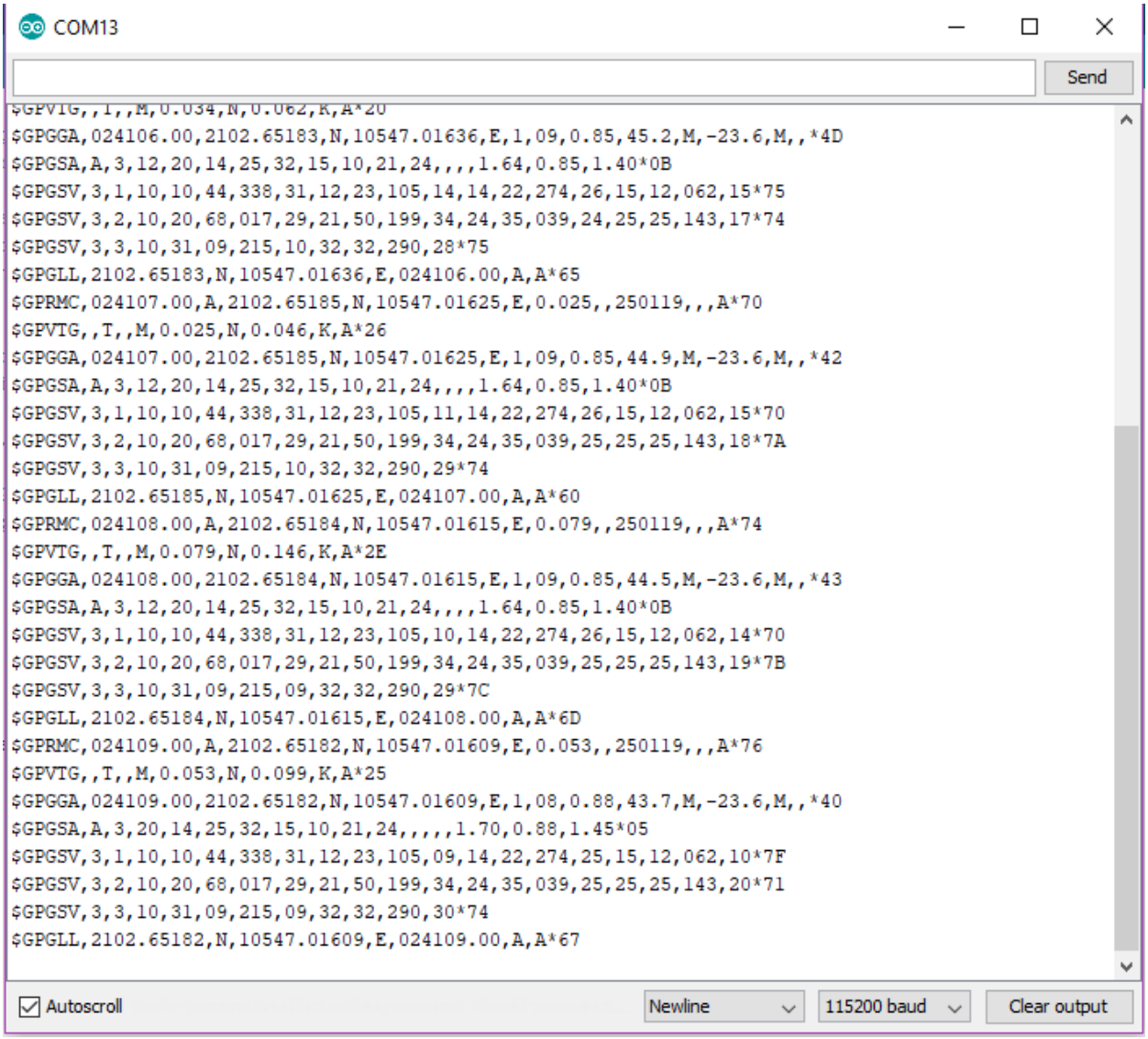
Cài đặt xong ta sử dụng phần mềm để lấy thông tin GPS như hình dưới đây:



Khi có đầy đủ lượng vệ tinh cần thiết thì vị trí của mô-đun GPS sẽ có thêm tọa độ vĩ độ (Latitude) và kinh độ (Longitude).



Nếu theo dõi trên cổng COM ta sẽ thấy luồng dữ liệu gửi lên như hình sau:



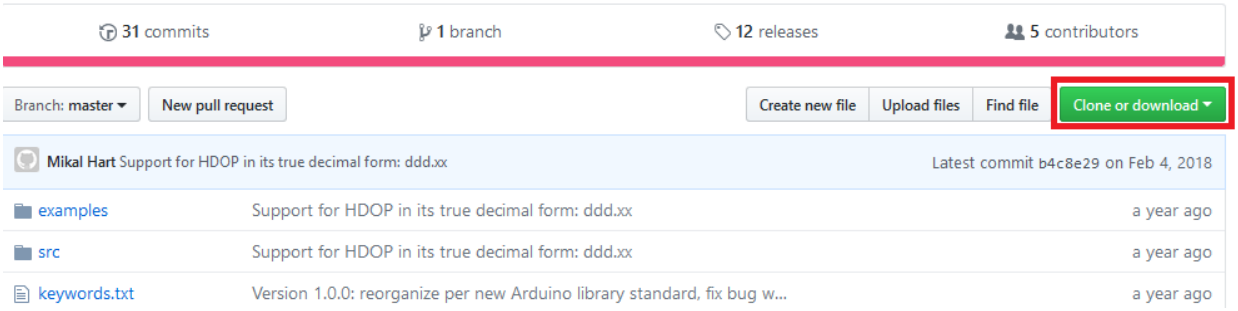
Để phân tích ra các thông số có ý nghĩa ta phải giải mã bản tin này.

4. Kết nối ESP8266 với mô-đun GPS

Để cài đặt thư viện cho mô-đun GPS NEO-6M ta cài đặt thư viện **TinyGPS++** cho **ESP8266** sử dụng thư viện tại địa chỉ sau (nhớ Fork về tài khoản Github của mình để thuận tiện khi sử dụng sau này):

https://github.com/mikalhart/TinyGPSPlus

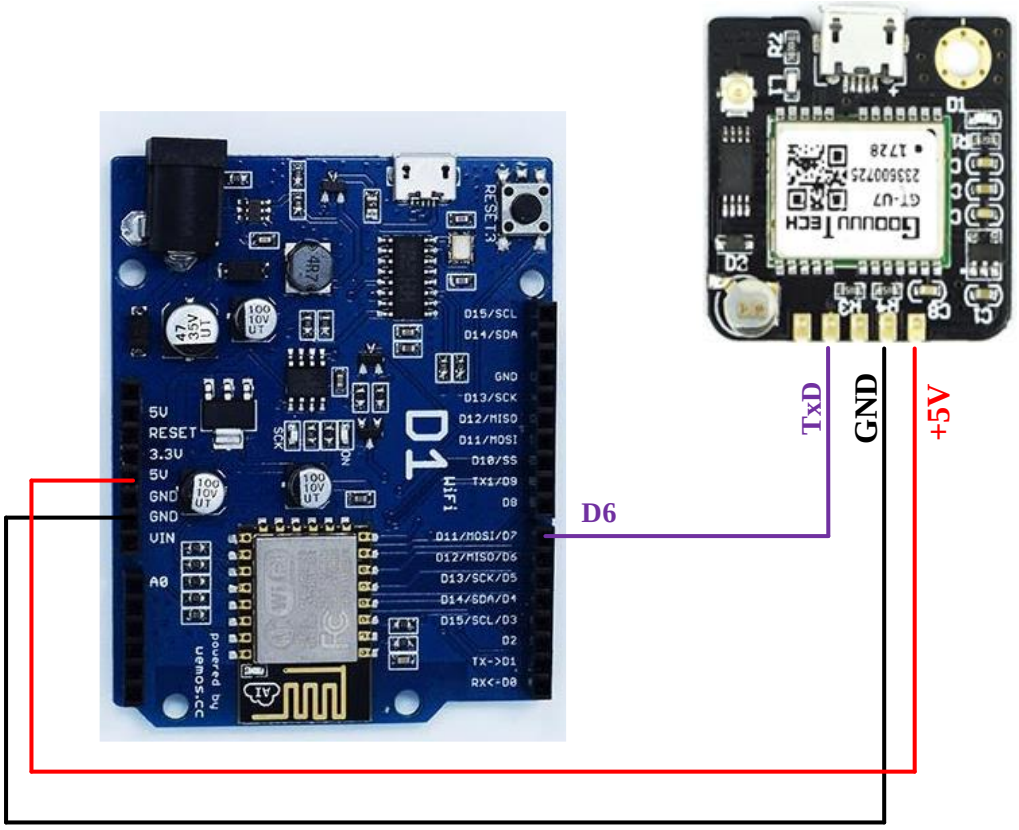
Bấm vào đường link trên, sau đó ta chọn vào mục "***Clone or download***" và chọn "***Download ZIP***" để tải về.



Tiếp đó ta mở phần mềm Arduino IDE và bấm vào "***Sketch → Include Library → Add .ZIP Library…***", sau đó chọn vào đường dẫn chứa file .ZIP của thư viện **TinyGPS++** vừa tải về, chọn "***Open***" để cài đặt thư viện .ZIP đó.

Sau khi cài đặt thư viện xong, ta có thể sử dụng một Example của thư viện đó để thử nghiệm mô-đun định vị GPS NEO-6M với thư viện **TinyGPS++**. Bấm vào "***File → Examples → TinyGPS++ → FullExample***".

Tiến hành lắp mạch theo sơ đồ sau:



Chương trình mẫu sau khi được lấy ra ta phải sửa một chút ít cho phù hợp, ở đây ta sửa theo nội dung các chữ màu đỏ.

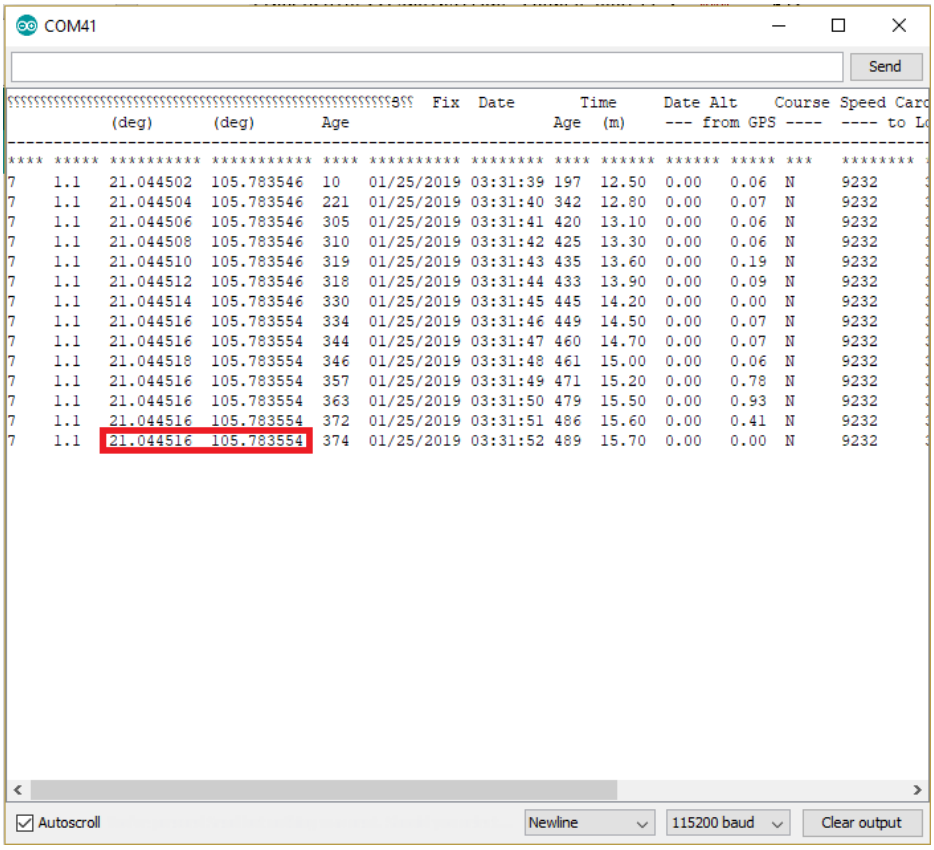
```
#include <TinyGPS++.h>
#include <SoftwareSerial.h>
/*
This sample code demonstrates the normal use of a TinyGPS++
(TinyGPSPlus) object.
It requires the use of SoftwareSerial, and assumes that you have a
4800-baud serial GPS device hooked up on pins 4(rx) and 3(tx).
*/
static const int RXPin = D6, TXPin = D7;
static const uint32_t GPSBaud = 9600;

// The TinyGPS++ object
TinyGPSPlus gps;

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

void setup()
{
     Serial.begin(115200);
     ss.begin(GPSBaud);

     Serial.println(F("FullExample.ino"));
     Serial.println(F("An extensive example of many interesting
TinyGPS++ features"));
     Serial.print(F("Testing TinyGPS++ library v. "));
Serial.println(TinyGPSPlus::libraryVersion());
     Serial.println(F("by Mikal Hart"));
     Serial.println();
     Serial.println(F("Sats HDOP  Latitude   Longitude   Fix  Date
Time     Date Alt    Course Speed Card  Distance Course Card  Chars
Sentences Checksum"));
     Serial.println(F("           (deg)      (deg)       Age
Age  (m)    --- from GPS ----  ---- to London  ----  RX    RX
Fail"));
     Serial.println(F("----------------------------------------------
-------------------------------------------------------------------
------------------"));
}

void loop()
{
```

```
      static const double LONDON_LAT = 51.508131, LONDON_LON = -
0.128002;

      printInt(gps.satellites.value(), gps.satellites.isValid(), 5);
      printFloat(gps.hdop.hdop(), gps.hdop.isValid(), 6, 1);
      printFloat(gps.location.lat(), gps.location.isValid(), 11, 6);
      printFloat(gps.location.lng(), gps.location.isValid(), 12, 6);
      printInt(gps.location.age(), gps.location.isValid(), 5);
      printDateTime(gps.date, gps.time);
      printFloat(gps.altitude.meters(), gps.altitude.isValid(), 7, 2);
      printFloat(gps.course.deg(), gps.course.isValid(), 7, 2);
      printFloat(gps.speed.kmph(), gps.speed.isValid(), 6, 2);
      printStr(gps.course.isValid() ?
TinyGPSPlus::cardinal(gps.course.deg()) : "*** ", 6);

      unsigned long distanceKmToLondon =
            (unsigned long)TinyGPSPlus::distanceBetween(
                  gps.location.lat(),
                  gps.location.lng(),
                  LONDON_LAT,
                  LONDON_LON) / 1000;
      printInt(distanceKmToLondon, gps.location.isValid(), 9);

      double courseToLondon =
            TinyGPSPlus::courseTo(
                  gps.location.lat(),
                  gps.location.lng(),
                  LONDON_LAT,
                  LONDON_LON);

      printFloat(courseToLondon, gps.location.isValid(), 7, 2);

      const char *cardinalToLondon =
TinyGPSPlus::cardinal(courseToLondon);

      printStr(gps.location.isValid() ? cardinalToLondon : "*** ", 6);

      printInt(gps.charsProcessed(), true, 6);
      printInt(gps.sentencesWithFix(), true, 10);
      printInt(gps.failedChecksum(), true, 9);
      Serial.println();

      smartDelay(5000);

      if (millis() > 5000 && gps.charsProcessed() < 10)
            Serial.println(F("No GPS data received: check wiring"));
```

```
}

// This custom version of delay() ensures that the gps object
// is being "fed".
static void smartDelay(unsigned long ms)
{
	unsigned long start = millis();
	do
	{
		while (ss.available())
			gps.encode(ss.read());
	} while (millis() - start < ms);
}

static void printFloat(float val, bool valid, int len, int prec)
{
	if (!valid)
	{
		while (len-- > 1)
			Serial.print('*');
		Serial.print(' ');
	}
	else
	{
		Serial.print(val, prec);
		int vi = abs((int)val);
		int flen = prec + (val < 0.0 ? 2 : 1); // . and -
		flen += vi >= 1000 ? 4 : vi >= 100 ? 3 : vi >= 10 ? 2 : 1;
		for (int i = flen; i<len; ++i)
			Serial.print(' ');
	}
	smartDelay(0);
}

static void printInt(unsigned long val, bool valid, int len)
{
	char sz[32] = "******************";
	if (valid)
		sprintf(sz, "%ld", val);
	sz[len] = 0;
	for (int i = strlen(sz); i<len; ++i)
		sz[i] = ' ';
	if (len > 0)
		sz[len - 1] = ' ';
	Serial.print(sz);
	smartDelay(0);
```

```
}

static void printDateTime(TinyGPSDate &d, TinyGPSTime &t)
{
      if (!d.isValid())
      {
            Serial.print(F("********** "));
      }
      else
      {
            char sz[32];
            sprintf(sz, "%02d/%02d/%02d ", d.month(), d.day(),
d.year());
            Serial.print(sz);
      }

      if (!t.isValid())
      {
            Serial.print(F("******** "));
      }
      else
      {
            char sz[32];
            sprintf(sz, "%02d:%02d:%02d ", t.hour(), t.minute(),
t.second());
            Serial.print(sz);
      }

      printInt(d.age(), d.isValid(), 5);
      smartDelay(0);
}

static void printStr(const char *str, int len)
{
      int slen = strlen(str);
      for (int i = 0; i<len; ++i)
            Serial.print(i<slen ? str[i] : ' ');
      smartDelay(0);
}
```

Nạp code và theo dõi kết quả trên Serial Monitor. Sử dụng tọa độ này ta dùng trên Google Maps để hiển thị vị trí, đối chiếu với vị trí thực nhận xét về sai số của phép đo.



Vào trang web sau: https://www.google.com/maps/

Nhập tọa độ: `21.044516  105.783554` như vừa lấy được ở trên, ta thu được kết quả sau:

Bài tập:

Ghép nối với màn hình OLED, hiển thị các thông tin GPS như sau:

- Vĩ độ.
- Kinh độ.
- Độ cao.
- Tốc độ theo km/h.
- Khoảng cách tại điểm đo với vị trí trung tâm (tọa độ lấy tại điểm nào đó do học viên định nghĩa).
- Ngày tháng năm, thời gian.