

## ¿QUÉ ES UN GPS NEO-6?

Los dispositivos NEO-6 son una familia de receptores fabricados por U-Blox, que pueden ser conectados con facilidad a un autómata o procesador como Arduino.

Los GPS NEO-6 disponen de interface de comunicación UART, SPI, DDC (I2C) y USB. Soportan los protocolos NMEA, UBX binary y RTCM.

La familia de receptores GPS NEO-6 están diseñados para tener un pequeño tamaño, pequeño coste, y pequeño consumo. La intensidad de corriente necesaria es de unos 37mA en modo de medición continuo.

La tensión de alimentación es de 2.7 a 3.6V para los modelos NEO-6Q/6M, y 1.75-2.0V para los modelos NEO-6G.

Frecuentemente se encuentran integrados en módulos que incorporan la electrónica necesaria para conectarla de forma sencilla a un Arduino. En la mayoría de los módulos, esto incluye un regulador de voltaje que permite alimentar directamente a 5V.

El GPS NEO-6 tiene un tiempo de encendido cold y warm de unos 30s, y en hot 1 segundo. La frecuencia máxima de medición es de 5Hz.

La precisión que en posición es de 2.5m, en velocidad 0,1m/s y en orientación 0.5º, valores más que aceptables para un sistema de posicionamiento GPS.

Los GPS son muy empleados en proyectos de electrónica y Arduino, especialmente en el caso de robots y vehículos como cuadricópteros.

## ESQUEMA DE MONTAJE

La conexión es sencilla, ya que vamos la UART disponible en el receptor de GPS. En primer lugar alimentamos el módulo conectando Vcc y Gnd del módulo, a 5V y Gnd de Arduino.

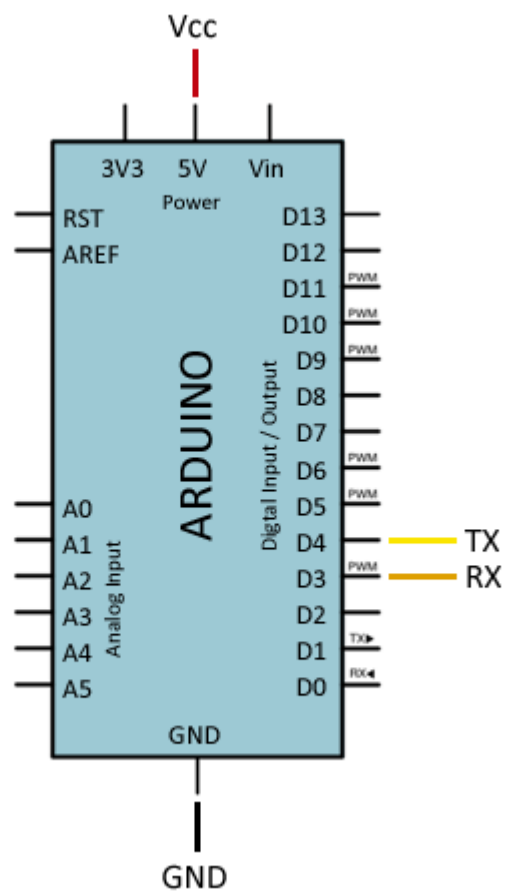
Para que la conexión no interfiera con el puerto serie normal vamos a usar la librería SoftSerial para establecer una comunicación empleando cualquiera de los pines.

En el ejemplo usaremos el Pin 3 para RX y el Pin 4 para TX. Recordar que en la comunicación serial los pines se conectan con sus opuestos, es decir, RX - TX y viceversa, por lo que el esquema es el siguiente.

GND — GND  
 (D4) RX — TX  
 (D3) TX — RX  
 5V — Vcc



Mientras que la conexión con el módulo, vista desde Arduino, quedaría de la siguiente forma.



## EJEMPLOS DE CÓDIGO

### OBTENER LECTURA NMEA

El código necesario es igualmente muy sencillo, al realizarse a través de puerto serie. Empleamos la librería `SoftSerial` para crear un puerto en cualquier par de pines de Arduino.

```
#include <SoftwareSerial.h>

const int RX = 4;

const int TX = 3;

SoftwareSerial gps(RX, TX);

void setup()

{

    Serial.begin(115200);

    gps.begin(9600);

}

void loop()

{

    if (gps.available())
```

```

{

    char data;

    data = gps.read();

    Serial.print(data);

}

}

```

Los datos obtenidos del NEO-6 están en formato \$GPRMC, uno de las secuencias disponibles en el protocolo NMEA (National Marine Electronics Association).

Esta secuencia tiene la forma

\$GPRMC,hhmmss.ss,A,IIII.II,a,yyyyy.yy,a,x.x,x.x,ddmmyy,x.x,a\*hh

Siendo,

hhmmss.ss	Hora UTC
A	Estado receptor (A = OK, V = warning)
IIII.II,a	Latitud (a = N o S)
yyyy.yy,a	Longitud (a = E o W)

x.x	Velocidad en nudos in knots
x.x	Curso en grados
ddmmyy	Fecha UT
x.x,a	Variación magnética en grados (a = E o W)
*hh	Checksum

Por ejemplo, la secuencia

```
$GPRMC,225446,A,4916.45,N,12311.12,W,000.5,054.7,191194,020.3,E*68
```

Tiene el siguiente significado

225446	Hora 22:54:46 UTC
A	Estado receptor A = OK
4916.45,N	Latitud 49º 16.45 min Norte

12311.12,W	Longitud 123º11.12 min Oeste
000.5	Velocidad 0.5 nudos
054.7	Curso 54.7º
191194	Fecha 19 Noviembre 1994
020.3,E	Variación magnética 20.3º East
*68	Checksum

## OBTENER COORDENADAS CON TINYGPS

Para interpretar la secuencia \$GPRMC de forma sencilla disponemos de la [librería TinyGPS](#), que nos permite parsear los datos obtenidos del módulo GPS.

El siguiente ejemplo emplea la librería TinyGPS para obtener las coordenadas (latitud y longitud) y mostrarlas en la pantalla.

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

```
#include <TinyGPS.h>

TinyGPS gps;

SoftwareSerial softSerial(4, 3);

void setup()

{

    Serial.begin(115200);

    softSerial.begin(9600);

}

void loop()

{

    bool newData = false;

    unsigned long chars;

    unsigned short sentences, failed;

    // Intentar recibir secuencia durante un segundo

    for (unsigned long start = millis(); millis() - start < 1000;)

    {
```

```

while (softSerial.available())

{

    char c = softSerial.read();

    if (gps.encode(c)) // Nueva secuencia recibida

        newData = true;

}

}

if (newData)

{

    float flat, flon;

    unsigned long age;

    gps.f_get_position(&flat, &flon, &age);

    Serial.print("LAT=");

    Serial.print(flat == TinyGPS::GPS_INVALID_F_ANGLE ? 0.0 : flat, 6);

    Serial.print(" LON=");

    Serial.print(flton == TinyGPS::GPS_INVALID_F_ANGLE ? 0.0 : flon, 6);

    Serial.print(" SAT=");

    Serial.print(gps.satellites() == TinyGPS::GPS_INVALID_SATELLITES ?
0 : gps.satellites());

```



```
        Serial.print(" PREC=");

        Serial.print(gps.hdop() == TinyGPS::GPS_INVALID_HDOP ? 0 :
gps.hdop());

    }

    gps.stats(&chars, &sentences, &failed);

    Serial.print(" CHARS=");

    Serial.print(chars);

    Serial.print(" SENTENCES=");

    Serial.print(sentences);

    Serial.print(" CSUM ERR=");

    Serial.println(failed);

}
```