<https://www.luisllamas.es/localizacion-gps-con-arduino-y-los-modulos-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. |

**USO:** UN DRON, SE LE DA LAS COORDENADAS, Y CONSTANTEMENTE VERIFICA SI SE VA ACERCANDO. AL INICIAR VERIFICA SU POSICION Y SU DESTINO, PODRIA TENER UN SENSOR BRUJULA, O SE CALIBRA AL INICIO LA DIRECCION. LUEGO DE DEJAR EL PAQUETE VUELVE A LA COORDENADA INICIAL.

De pequeño tamaño, costo ($4) y consume. La corriente es de unos 37mA en modo de medición continuo. Alimentación:

* 2.7 a 3.6V para los modelos NEO-6Q/6M,
* 1.75-2.0V para los modelos NEO-6G.

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.

#include <SoftwareSerial.h>

// PIN 4 a RX

// PIN 3 a TX

SoftwareSerial gps**(4**, 3**)**;

void setup**()**

**{**

Serial.begin**(**9600**)**;

gps.begin**(**9600**)**;

}

void loop**()**

**{**

if **(**gps.available**())**

**{**

char data = gps.read**()**;

Serial.print**(**data**)**;

delay(60 \* 1000);

**}**

**}**

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

$GPRMC,hhmmss.ss,A,llll.ll,a,yyyyy.yy,a,x.x,x.x,ddmmyy,x.x,a\*hh

Los datos están en formato $GPRMC, una de las secuencias disponibles en el protocolo NMEA (National Marine Electronics Asociation).

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

|  |  |  |
| --- | --- | --- |
| arr[0] | 225446 | Hora 22:54:46 **UTC** (restar 4 horas para S.S.) |
| arr[1] | A | Estado receptor **A = OK** |
| arr[2] | 4916.45 | Latitud 49º 16.45 min |
| arr[3] | N | Norte |
| arr[4] | 12311.12 | Longitud 123º11.12 min |
| arr[5] | W | West Oeste |
| arr[6] | 000.5 | Velocidad 0.5 nudos |
| arr[7] | 054.7 | Curso 54.7º |
| arr[8] | 191194 | Fecha 19 Noviembre 1994 |
| arr[9] | 020.3 | Variación magnética 20.3º |
| arr[10] | E | East, de variacion magnetica |
| arr[11] | \*68 | Checksum |

|  |  |
| --- | --- |
| **COLEGIO** | **MI CASA** |
|  |  |

Latitud = (13.7069 – 13.6707) \* 3.14159 / 180 \* 10,000 = **6.3 kms** (+) hacia el Sur

Longitud = (89.24504 – 80.27682) \* 3.14159 / 180 \* 10,000 = **-5.5 kms** (-) hacia el Oeste

Como encontramos la distancia a recorrer en linea recta? **8.4 kms**.

**Lenguage C, Python, Javascript**

|  |  |
| --- | --- |
| int iArr[3];  iArr[0] = 1;  iArr[1]= 5;  iArr[2] = 20;  for (int i=0; i< 3; i++)  Serial.println(iArr[i]);  1  5  20 | string sArr[10];  // limpiar variable  for (int i=0; i<10; i++)  sArr[i]="";  sArr[0] , sArr[1], … sArr[9] |

char data = gps.read();

String arr[12] = **GetArray**(data.toString());

for (int i=0; i<13; i++)

Serial.println(arr[i]);

string[] **GetArray**(String msg)

{

String arr[12];

for (int i=0; i<11; i++) arr[i]="";

int from = 0;

int pos = 0;

int long = msg.length();

for (int i=0; i < long; i++)

{

if (msg.charAt(i) == ',')

{

arr[pos++] = msg.substring(from, i);

from=(i+1);

}

}

if (from < long);

arr[pos] = msg.substring(from, long);

return arr;

}

Brujula, **HMC5883L**



#include <SoftwareSerial.h>

// PIN 4 a RX

// PIN 3 a TX

SoftwareSerial gps**(4**, 3**)**;

void setup**()**

**{**

Serial.begin**(**9600**)**;

gps.begin**(**9600**)**;

}

void loop() {

String data = gps.read();

String arr[12] = GetArray(data);

delay(5 \* 1000);

// arr[2] arr[4]

}

string[] **GetArray**(String msg)

{

String arr[12];

for (int i=0; i<11; i++) arr[i]="";

int from = 0;

int pos = 0;

int long = msg.length();

for (int i=0; i < long; i++)

{

if (msg.charAt(i) == ',')

{

arr[pos++] = msg.substring(from, i);

from=(i+1);

}

}

if (from < long);

arr[pos] = msg.substring(from, long);

return arr;

}