

6) Como implementaría un sensor inteligente de posicionamiento global.

¿Que es un gps?

GPS Global Positioning System. Es un sistema que permite determinar en toda la Tierra la posición de cualquier objeto

¿Como funciona?

El GPS funciona mediante una red de como mínimo 24 satélites en órbita sobre el planeta Tierra, a aproximadamente 20.000 km de altura, con órbitas distribuidas para que en todo momento haya al menos 4 satélites visibles en cualquier punto de la tierra.

Determinación de la posición

Para **determinar su posición**, se necesita **4 o más satélites** y utiliza la **trilateración**. La trilateración es un método matemático para determinar las posiciones relativas de objetos usando la geometría de triángulos de forma análoga a la triangulación.

Modulo

El modulo posee u-blox neo 6M, una memoria EEPROM en la cual vienen datos con las configuraciones necesarias para utilizar el modulo u-blox neo 6M. Además de una antena cerámica.

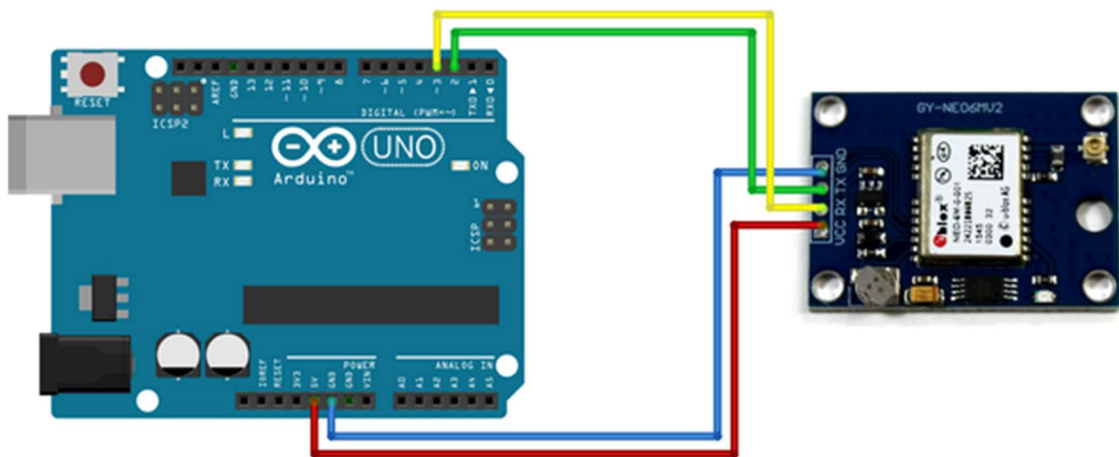
Características principales

- Precisión: Posición: 2.5m. velocidad 0,1m/s. Orientación 0.5°. Lo suficientemente bueno para obtener una posición global aceptable.
- Alimentación: 3.3v – 5V
- Comunicación: UART

Material

- Arduino Uno
- Cables dupond M-H
- Módulo GPS Neo GPS6MV2

Diagrama de conexión



Desarrollo

El modulo utiliza el protocolo UART por lo que para conectar el modulo con el micro solo hace falta simular un puerto serial con la librería **SoftwareSerial.h** que viene instalada en arduino por defecto.



NOTA: Para mejor recepción y conexión con los satélites es recomendable dejar el sensor en un lugar abierto o cerca de una ventana para mejorar la recepción de la señal satelital. Cuando tenga la suficiente señal un led encenderá.

/*

Ejemplo para obtener lectura en crudo u-blox NEO-6M
protocolo NMEA

*/

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

```
const byte RX = 2;
```

```
const byte TX = 3;
```

```
SoftwareSerial gps(RX, TX);
```

```
void setup()
```

```
{
```

```
  Serial.begin(9600);
```

```
  gps.begin(9600);
```

```
}
```

```
void loop()
```

```
{
```

```
  if (gps.available())
```

```
  {
```

```
    char data;
```

```
    data = gps.read();
```

```
    Serial.print(data);
```

```
  }
```

Cuando abrimos el monitor serial podemos observar todos los datos que el gps esta mandando cada segundo. Los datos se encuentran interpretados en un protocolo **NMEA (National Marine Electronics Asociation)**.

Interpretación de datos

NMEA 2.0					
Name	Garmin	Magellan	Lowrance	SiRF	Notes:
GPAPB	N	Y	Y	N	Auto Pilot B
GPBOD	Y	N	N	N	bearing, origin to destination - earlier G-12's do not transmit this
GPGGA	Y	Y	Y	Y	fix data
GPGLL	Y	Y	Y	Y	Lat/Lon data - earlier G-12's do not transmit this
GPGSA	Y	Y	Y	Y	overall satellite reception data, missing on some Garmin models
GPGSV	Y	Y	Y	Y	detailed satellite data, missing on some Garmin models
GPRMB	Y	Y	Y	N	minimum recommended data when following a route
GPRMC	Y	Y	Y	Y	minimum recommended data
GP RTE	Y	U	U	N	route data, only when there is an active route. (this is sometimes bidirectional)
GPWPL	Y	Y	U	N	waypoint data, only when there is an active route (this is sometimes bidirectional)

Podemos ver que GPGLL nos entrega Latitud y longitud pero el mas utilizado es el GPRMC el cual nos entrega información bastante útil.

Ejemplo

```

$GPGGA,185841.00,2045.31591,N,10322.04436,W,2,10,0.92,1583.6,M,-16.8,M,,0000*56
$GPGSA,A,3,22,46,31,01,26,16,03,11,14,23,,,1.66,0.92,1.38*07
$GPGSV,4,1,15,01,28,241,26,03,46,320,37,04,23,312,23,11,08,224,34*7B
$GPGSV,4,2,15,14,31,065,28,16,53,147,32,18,22,215,28,22,69,300,40*7A
$GPGSV,4,3,15,23,19,309,34,26,60,090,33,27,04,162,18,31,31,032,35*7F
$GPGSV,4,4,15,32,14,089,,46,52,233,44,51,65,191,32*40
$GPGLL,2045.31591,N,10322.04436,W,185841.00,A,D*74
$GPRMC,185842.00,A,2045.31591,N,10322.04437,W,0.025,,100120,,,D*6A
$GPVTG,,T,,M,0.025,N,0.047,K,D*22
$GPGGA,185842.00,2045.31591,N,10322.04437,W,2,10,0.92,1583.6,M,-16.8,M,,0000*54
$GPGSA,A,3,22,46,31,01,26,16,03,11,14,23,,,1.66,0.92,1.38*07
$GPGSV,4,1,15,01,28,241,26,03,46,320,37,04,23,312,23,11,08,224,34*7B
$GPGSV,4,2,15,14,31,065,28,16,53,147,32,18,22,215,28,22,69,300,40*7A
$GPGSV,4,3,15,23,19,309,34,26,60,090,33,27,04,162,18,31,31,032,35*7F
$GPGSV,4,4,15,32,14,089,,46,52,233,44,51,65,191,33*41
$GPGLL,2045.31591,N,10322.04437,W,185842.00,A,D*76
  
```

\$GPRMC,185842.00,A,2045.31591,N,10322.04437,W,0.025,,100120,,,D*6A

Desglose del mensaje

- 185842.00 - Indica la Hora GMT 18:58:42
- A - significa que la información es correcta, de otra forma seria V
- 2045.31591 - significa la longitud 20°45.31591
- N - Norte
- 10322.04437 - significa la latitud 103°22.04347
- W - Oeste
- 0.025 - Velocidad en nudos
- 100120 -Fecha 10/enero/20

Utilizando la librería **TinyGPS**.

```

/*
 * Ejemplo obtención de posición con gps
 * Taloselectronics.com
 * Rafael Lozano Rolon
 */
#include <SoftwareSerial.h> //incluimos SoftwareSerial
#include <TinyGPS.h> //incluimos TinyGPS
  
```

```

const byte RX = 2;
const byte TX = 3;
  
```

```

SoftwareSerial serialgps(RX, TX);
TinyGPS gps; //Declaramos el objeto gps
  
```

```

//Declaramos la variables para la obtención de datos
int year;
byte month, day, hour, minute, second, hundredths;
unsigned long chars;
unsigned short sentences, failed_checksum;
  
```



```

void setup()
{
    Serial.begin(9600); // Iniciamos el puerto serie
    serialgps.begin(9600); // Iniciamos el puerto serie del gps
}

void loop()
{
    while (serialgps.available())
    {
        int c = serialgps.read();

        if (gps.encode(c))
        {
            float latitude, longitude;
            gps.f_get_position(&latitude, &longitude);
            Serial.print("Latitud/Longitud: ");
            Serial.print(latitude, 5);
            Serial.print(", ");
            Serial.println(longitude, 5);

            gps.crack_datetime(&year, &month, &day, &hour, &minute, &second, &hundredths);
            Serial.print("Fecha: "); Serial.print(day, DEC); Serial.print("/");
            Serial.print(month, DEC); Serial.print("/"); Serial.print(year);
            Serial.print(" Hora: "); Serial.print(hour, DEC); Serial.print(":");
            Serial.print(minute, DEC); Serial.print(":"); Serial.print(second, DEC);
            Serial.print("."); Serial.println(hundredths, DEC);
            Serial.print("Altitud (metros): ");
            Serial.println(gps.f_altitude());
            Serial.print("Rumbo (grados): "); Serial.println(gps.f_course());
            Serial.print("Velocidad(kmph): ");
            Serial.println(gps.f_speed_kmph());
            Serial.print("Satelites: "); Serial.println(gps.satellites());
            Serial.println();
            gps.stats(&chars, &sentences, &failed_checksum);
        }
    }
}

```

Monitor Serial

