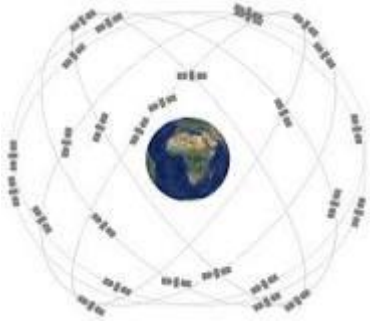
SENSOR INTELIGENTE DE POSICIONAMIENTO GLOBAL

Qué es un sensor de posicionamiento global?



El Sistema de Posicionamiento Global (GPS) es un sistema de radionavegación de los Estados Unidos de América, basado en el espacio, que proporciona servicios fiables de posicionamiento, navegación, y cronometría gratuita e ininterrumpidamente a usuarios civiles en todo el mundo.

¿Qué es el GPS y para qué se utiliza?

Hoy en día, los dispositivos GPS se conectan en segundos y están incluidos en múltiples aparatos, desde 'smartphones' hasta cámaras de fotos, pasando por relojes deportivos o pulseras cuantificadoras. Incluso los drones tienen hoy en día dispositivo GPS incorporado.

. GPS El GPS proporciona a los usuarios información sobre posicionamiento y navegación. Este sistema está constituido por tres segmentos: el espacial, el de control y el del usuario. Fue el primer sistema de radionavegación basado en satélites que no se veía afectado por condiciones atmosféricas o climatológicas. Los sistemas de GPS son capaces de geolocalizar personas u objetos gracias a un principio matemático llamado triangulación. La triangulación tiene un principio muy sencillo: un receptor se comunica con 3 satélites y se mide la distancia a la que está de cada uno. Calculando dicha distancia, se puede extraer la ubicación exacta del receptor. El GPS dio origen a los sistemas de triangulación geodésicos, que eran bidimensionales y provenían de latitud y longitud.

Hoy en día, la comunidad científica ha instalado estaciones distribuidas a nivel global para comunicarse con los satélites y determinar así la posición correcta.

En este proyecto, se usa el modelo GPS Click, el cual es una placa accesoria en mikroBus. La placa se puede interconectar con un microcontrolador a través de UART o I2C conexión, o los datos se pueden adquirir usando PC aplicación a través de conexión USB. Cuenta con conector y antenas pasivas. Puede operar con una fuente de alimentación de 3.3V.

El clic de GPS puede rastrear simultáneamente hasta 16 satélites mientras busca otros nuevos. El TTFF (tiempo hasta la primera corrección) del módulo LEA-6S es de menos de un segundo; esta es la medida de tiempo necesaria para que un receptor GPS obtenga señales de satélite y datos de navegación y, en base a esta información, calcule una posición. El data sheet del GPS se encuentra en la siguiente referencia (Mikroe 2016).

Módulo GPS e Interfaz con Arduino UNO

¿Qué es el GPS?

El Sistema de Posicionamiento Global (GPS) es un sistema de navegación por satélite compuesto por al menos 24 satélites. El GPS funciona en cualquier condición climática, en cualquier parte del mundo, las 24 horas del día, sin cargos de suscripción ni de configuración.

Cómo funciona el GPS

Los satélites GPS rodean la Tierra dos veces al día en una órbita precisa. Cada satélite transmite una señal única y parámetros orbitales que permiten a los dispositivos GPS decodificar y calcular la ubicación precisa del satélite. Los receptores GPS utilizan esta información y trilateración para calcular la ubicación exacta de un usuario. Esencialmente, el receptor GPS mide la distancia a cada satélite por la cantidad de tiempo que toma recibir una señal transmitida. Con mediciones de distancia de unos pocos satélites más, el receptor puede determinar la posición de un usuario y mostrarla.

Para calcular su posición 2-D (latitud y longitud) y el movimiento de la pista, un receptor GPS debe estar bloqueado a la señal de al menos 3 satélites. Con 4 o más satélites a la vista, el receptor puede determinar su posición tridimensional (latitud, longitud y altitud). Generalmente, un receptor GPS rastreará 8 o más satélites, pero eso depende de la hora del día y del lugar en el que se encuentre en la tierra.

Una vez que se ha determinado su posición, la unidad GPS puede calcular otra información, como por ejemplo:

- Velocidad
- Rango de movimiento
- Camino
- Distancia del viaje
- Distancia hasta el destino

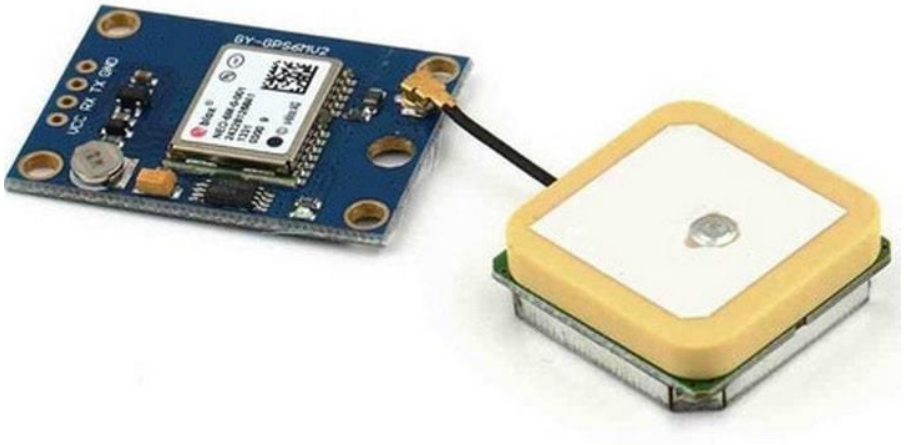
¿Cuál es la señal?

Satélites GPS transmiten al menos 2 señales de radio de baja potencia. Las señales viajan por línea de visión, lo que significa que pasarán a través de nubes, vidrio y plástico, pero no a través de la mayoría de los objetos sólidos, como edificios y montañas. Sin embargo, los receptores modernos son más sensibles y por lo general pueden rastrear a través de las casas.

Una señal GPS contiene 3 tipos diferentes de información:

- El código pseudoaleatorio es un código de identificación que identifica qué satélite está transmitiendo información. Puede ver de qué satélites está recibiendo señales en la página de satélites de su dispositivo.
- Los datos de efemérides son necesarios para determinar la posición de un satélite y proporcionan información importante sobre la salud del satélite, la fecha y la hora actuales.
- Los datos del almanaque le indican al receptor GPS dónde debe estar cada satélite GPS en cualquier momento del día y muestran la información orbital de ese satélite y de todos los demás satélites del sistema.

Módulo GPS NEO-6M y Arduino



- El Sistema de Posicionamiento Global (GPS) utiliza las señales enviadas por los satélites en las estaciones espaciales y terrestres de la Tierra para determinar con precisión su posición en la Tierra.
- El módulo receptor GPS NEO-6M utiliza la comunicación USART para comunicarse con el microcontrolador o terminal de PC.
- Recibe información como la latitud, longitud, altitud, hora UTC, etc. de los satélites en forma de cadena NMEA. Esta cadena necesita ser analizada para extraer la información que queremos usar.

El módulo GPS NEO-6M tiene las siguientes características

- Tiene una antena externa y una EEPROM incorporada.
- Interfaz: RS232 TTL
- Fuente de alimentación: 3V a 5V
- Velocidad de transmisión predeterminada: 9600 bps
- Funciona con frases NMEA estándar

El módulo GPS NEO-6M también es compatible con otras tarjetas de microcontrolador. Por ejemplo con la Raspberry Pi.

Ejemplo

Vamos a mostrar los datos, latitud, longitud, altitud y tiempo, recibidos por el módulo receptor GPS en el monitor serie de nuestro Arduino Uno.
Aquí usaremos la biblioteca TinyGPSPlus de Mikal Hart de GitHub. Puedes descargar esta biblioteca desde aquí.
Extrae la biblioteca y añádela a la ruta de la carpeta de bibliotecas de Arduino IDE.
Para más información sobre cómo añadir una biblioteca personalizada al IDE de Arduino y usar ejemplos de él, consulta Añadir biblioteca al IDE de Arduino en la sección tutoriales.
Aquí, hemos creado un scketch simplificado utilizando las funciones y el archivo de cabecera proporcionado por el Autor de esta biblioteca.

Si ves la mayoría de los datos en formato *****, toma el GPS conectado a Arduino en un espacio abierto (balcón por ejemplo). El GPS puede requerir algún tiempo para fijar los satélites. Dale unos 20-30 segundos para que pueda empezar a darte los datos correctos. Por lo general, no se tarda más de 5 segundos en fijar los satélites si estás en un espacio abierto, pero ocasionalmente puede tomar más tiempo (por ejemplo, si el receptor GPS no puede ver 3 o más satélites).

Diagrama de interfaz

Código para mostrar los parámetros GPS en el serial monitor

```
#include <TinyGPS++.h>
#include
/* Create object named bt of the class SoftwareSerial */
SoftwareSerial GPS_SoftSerial(4, 3);/* (Rx, Tx) */
/* Create an object named gps of the class TinyGPSPlus */
TinyGPSPlus gps;
volatile float minutes, seconds;
volatile int degree, secs, mins;
void setup() {
Serial.begin(9600); /* Define baud rate for serial
communication */
```

```
GPS_SoftSerial.begin(9600); /* Define baud rate for
software serial communication */
}
void loop() {
smartDelay(1000); /* Generate precise delay of 1ms */
unsigned long start;
double lat_val, lng_val, alt_m_val;
uint8_t hr_val, min_val, sec_val;
bool loc_valid, alt_valid, time_valid;
lat_val = gps.location.lat(); /* Get latitude data */
loc_valid = gps.location.isValid(); /* Check if valid
location data is available */
lng_val = gps.location.lng(); /* Get longtitude data */
alt_m_val = gps.altitude.meters(); /* Get altitude data in
meters */
alt_valid = gps.altitude.isValid(); /* Check if valid
altitude data is available */
hr_val = gps.time.hour(); /* Get hour */
min_val = gps.time.minute(); /* Get minutes */
sec_val = gps.time.second(); /* Get seconds */
time_valid = gps.time.isValid(); /* Check if valid time
data is available */
if (!loc_valid)
{
Serial.print("Latitude : ");
Serial.println("*****");
Serial.print("Longitude : ");
Serial.println("*****");
}
else
{
DegMinSec(lat_val);
Serial.print("Latitude in Decimal Degrees : ");
Serial.println(lat_val, 6);
Serial.print("Latitude in Degrees Minutes Seconds : ");
Serial.print(degree);
```

```
Serial.print("\t");
Serial.print(mins);
Serial.print("\t");
Serial.println(secs);
DegMinSec(lng_val); /* Convert the decimal degree value
into degrees minutes seconds form */
Serial.print("Longitude in Decimal Degrees : ");
Serial.println(lng_val, 6);
Serial.print("Longitude in Degrees Minutes Seconds : ");
Serial.print(degree);
Serial.print("\t");
Serial.print(mins);
Serial.print("\t");
Serial.println(secs);
}
if (!alt_valid)
{
Serial.print("Altitude : ");
Serial.println("*****");
}
else
{
Serial.print("Altitude : ");
Serial.println(alt_m_val, 6);
}
if (!time_valid)
{
Serial.print("Time : ");
Serial.println("*****");
}
else
{
char time_string[32];
sprintf(time_string, "Time : %02d/%02d/%02d \n", hr_val,
min_val, sec_val);
Serial.print(time_string);
```

```
}
}
static void smartDelay(unsigned long ms)
{
unsigned long start = millis();
do
{
while (GPS_SoftSerial.available()) /* Encode data read from
GPS while data is available on serial port */
gps.encode(GPS_SoftSerial.read());
/* Encode basically is used to parse the string received by
the GPS and to store it in a buffer so that information can
be extracted from it */
} while (millis() - start < ms);
}
void DegMinSec( double tot_val) /* Convert data in decimal
degrees into degrees minutes seconds form */
{
degree = (int)tot_val;
minutes = tot_val - degree;
seconds = 60 * minutes;
minutes = (int)seconds;
mins = (int)minutes;
seconds = seconds - minutes;
seconds = 60 * seconds;
secs = (int)seconds;
}
```