

Tarea 5, Grupo 2

Ejercicio 1-B

Implementación de un Sensor Inteligente de Posicionamiento Global

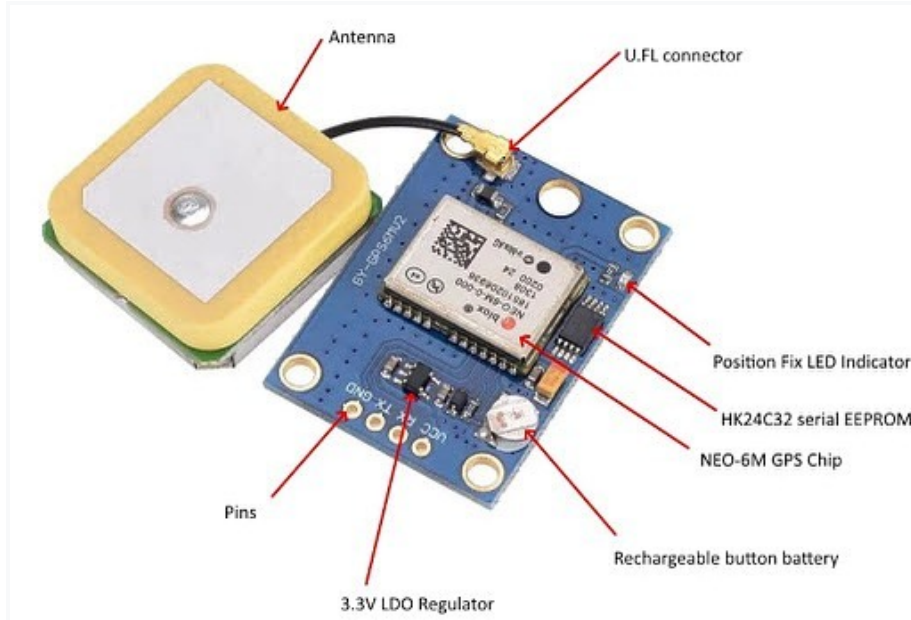
Este informe técnico describe la implementación de un sensor inteligente de posicionamiento global (GPS) utilizando el módulo GY-GPS6MV2 integrado a una placa de desarrollo Arduino. El proyecto se enfoca en la obtención de datos de posición precisa, incluyendo latitud, longitud, altitud, velocidad y rumbo, además de explorar las funcionalidades y prestaciones que se pueden lograr con este sensor, incluyendo aplicaciones inteligentes.

El uso de sistemas de posicionamiento global se ha vuelto esencial en una amplia variedad de aplicaciones, desde la navegación hasta la logística y la monitorización de activos. Además del sistema GPS estadounidense, hay otros sistemas de posicionamiento global desarrollados por diferentes países, como el sistema GLONASS de Rusia, el sistema Galileo de la Unión Europea y el sistema BeiDou de China. Estos sistemas son compatibles entre sí y proporcionan redundancia y mayor precisión cuando se utilizan en conjunto.

Este proyecto se centra en la implementación de un prototipo de sensor GPS utilizando el módulo GY-GPS6MV2 junto con un Arduino Uno para demostrar las capacidades de un dispositivo de seguimiento y localización. Para que el módulo GPS funcione a la perfección se recomienda hacer las pruebas en un ambiente abierto o cercano a la ventana para una correcta recepción de la señal.

Modulo GY-GPS6MV2

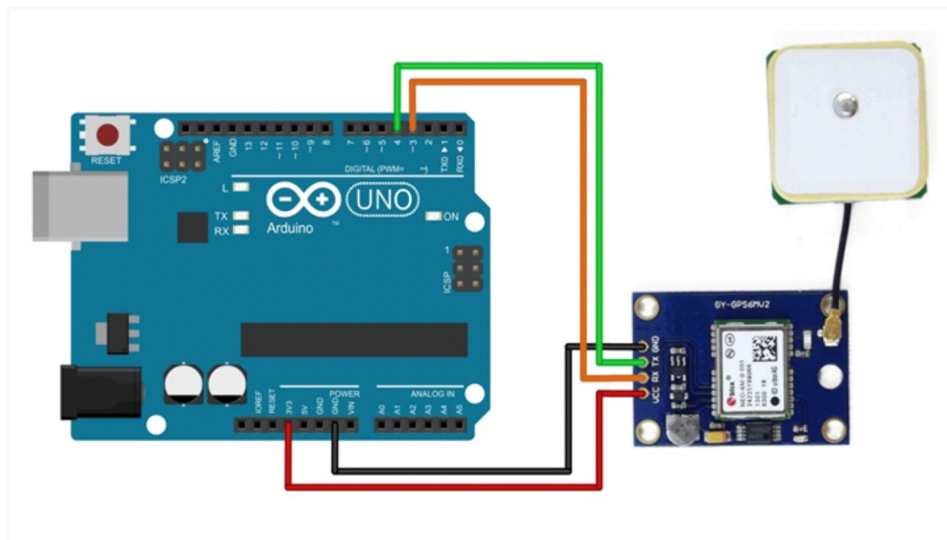
El módulo GPS en su modelo GY-GPS6MV2 viene con un módulo de serie U-Blox NEO 6M equipado en el PCB, una EEPROM con configuración de fábrica, una pila de botón para mantener los datos de configuración en la memoria EEPROM, un indicador LED y una antena cerámica. También posee los pines o conectores Vcc, Rx, Tx y Gnd por el que se puede conectar a algún microcontrolador mediante una interfaz serial.



Conexiones:

Para establecer una comunicación efectiva entre el Arduino Uno y el módulo GPS GY-GPS6MV2, se deben realizar las siguientes conexiones:

- Conecte el pin 3.3V del Arduino Uno al pin Vcc del módulo GPS para proporcionar la alimentación de 3.3V requerida por el GPS.
- Conecte el pin GND del Arduino Uno al pin GND del módulo GPS para asegurar una referencia común de tierra.
- Conecte el pin digital 4 del Arduino Uno al pin Tx (transmisor) del módulo GPS para recibir datos desde el GPS.
- Conecte el pin digital 3 del Arduino Uno al pin Rx (receptor) del módulo GPS para transmitir comandos y configuraciones al GPS.
- El módulo GPS se puede alimentar con una tensión de 5V ya que posee un regulador integrado dentro de sí.



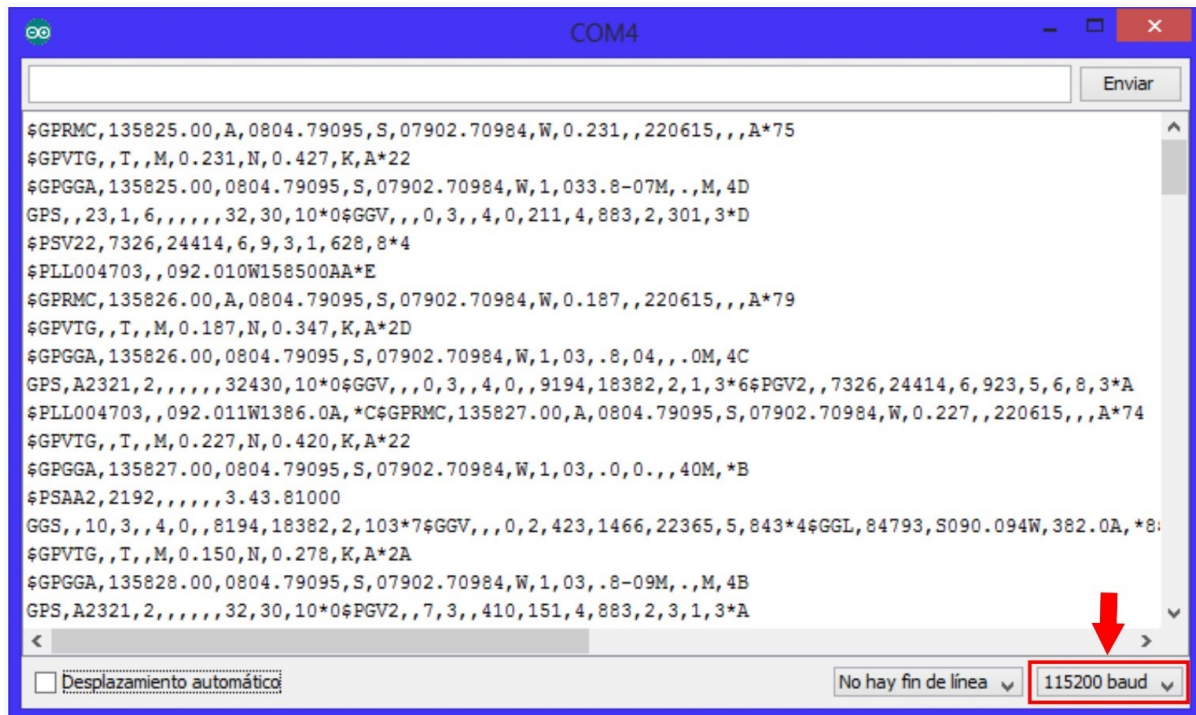
Programacion y configuracion

Se carga el siguiente código a la placa Arduino, en el cual se puede apreciar que se emplea la librería SoftwareSerial para emular un puerto serie.

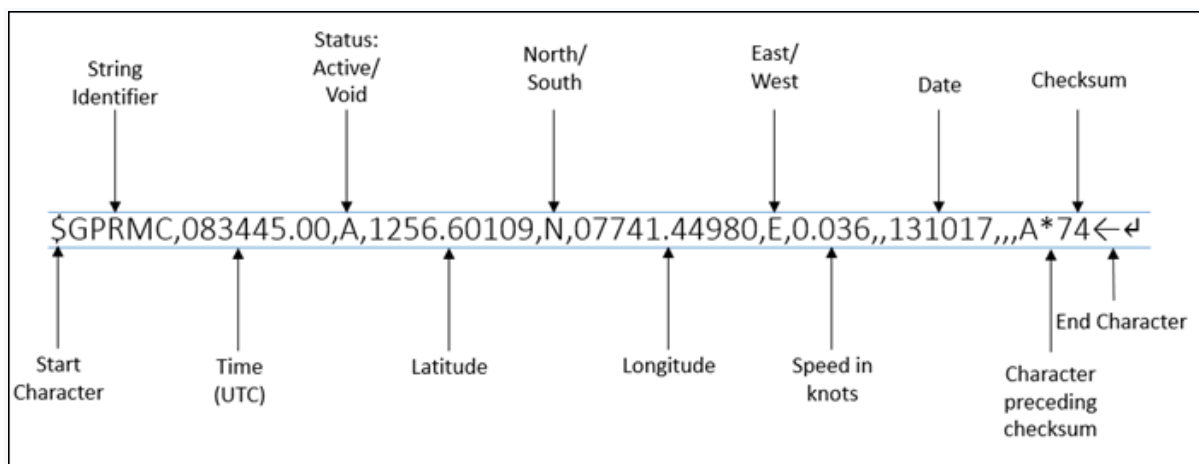
```
#include <SoftwareSerial.h>
SoftwareSerial gps(4,3);
char dato=' ';
void setup()
{
  Serial.begin(115200);
  gps.begin(9600);
}
void loop()
{
  if(gps.available())
  {
    dato=gps.read();
    Serial.print(dato);
  }
}
```

Como podemos ver, lo que hace nuestro programa es leer constantemente el módulo GPS a una velocidad de 9600 baudios que es la velocidad por la que viene configurado por defecto el módulo GPS y enviar dichos datos a la PC a través del puerto serie físico para

poder visualizarlos en el Monitor Serial. Al abrir nuestro Monitor Serial a una velocidad de 115200 baudios. Podremos ver los datos que recibimos.



Los datos que recibimos en nuestro módulo GPS siguen el protocolo NMEA (siglas de National Marine Electronics Association), las cuales son sentencias estándares para la recepción de datos GPS. Una de ellas y la más usada son las sentencias \$GPRMC . Cuya trama tiene la siguiente estructura:



De la trama de datos que envía el módulo GPS es posible obtener varias variables. Se puede utilizar la biblioteca TinyGPS para Arduino. Esta biblioteca simplifica la obtención de datos de GPS y proporciona funciones específicas para extraer información crucial, como latitud, longitud, altitud, velocidad y rumbo. Las funciones más utilizadas incluyen:

TinyGPS gps;: Inicializa un objeto TinyGPS.
gps.encode(Serial.read()): Actualiza y procesa los datos del GPS.
gps.location.lat(): Obtiene la latitud actual en grados decimales.
gps.location.lng(): Obtiene la longitud actual en grados decimales.
gps.altitude.meters(): Obtiene la altitud actual en metros sobre el nivel del mar.
gps.speed.mps(): Obtiene la velocidad actual en metros por segundo.
gps.course.deg(): Obtiene el rumbo actual en grados desde el norte verdadero.

Funcionalidades y Prestaciones:

Este prototipo de sensor inteligente de posicionamiento global implementado con el módulo GY-GPS6MV2 y Arduino puede ofrecer una amplia gama de funcionalidades y prestaciones, que incluyen:

- Seguimiento en Tiempo Real: Monitoreo y seguimiento en tiempo real de la ubicación del dispositivo en un mapa.
- Registro de Rutas: Capacidad para registrar rutas y generar historiales de movimientos.
- Geolocalización: Obtención de coordenadas precisas para geolocalización de objetos o activos.
- Alarma de Proximidad: Configuración de alarmas cuando el dispositivo se acerca o se aleja de ubicaciones predefinidas.
- Geo-fencing: Establecimiento de áreas geográficas virtuales con notificaciones cuando el dispositivo entra o sale de ellas.
- Navegación: Posibilidad de crear aplicaciones de navegación paso a paso.
- Monitoreo de Velocidad y Rumbo: Registro de velocidad y rumbo en tiempo real, útil para aplicaciones de control de vehículos.
- Integración de Mapas: Integración con servicios de mapas para visualización y geocodificación.
- Visualización de Datos: Presentación de datos en una pantalla LCD, aplicación móvil o plataforma web.
- Registro de Datos Históricos: Almacenamiento de datos de posición para análisis y seguimiento a largo plazo.