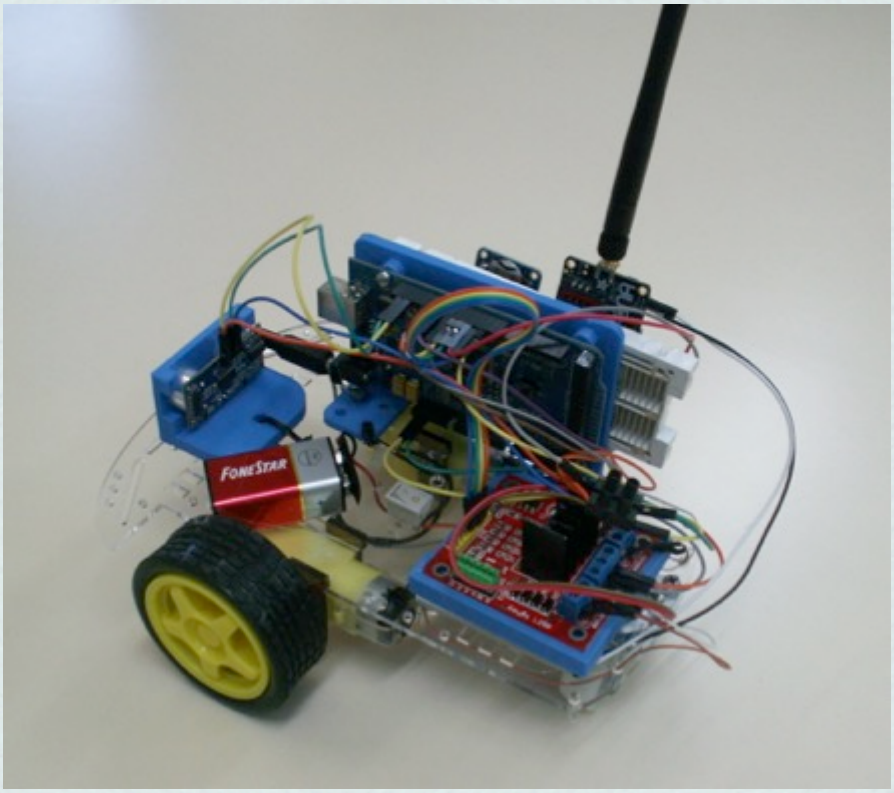
VÉHICULO MOVIL

GPS - GSM



Autores:
Agustín Lechuga Carretero
Luis Daniel Figuereo Morales

Tutores:
Joaquín Moreno Marchal
Agustín Carmona Lorente

ÍNDICE

Introducción

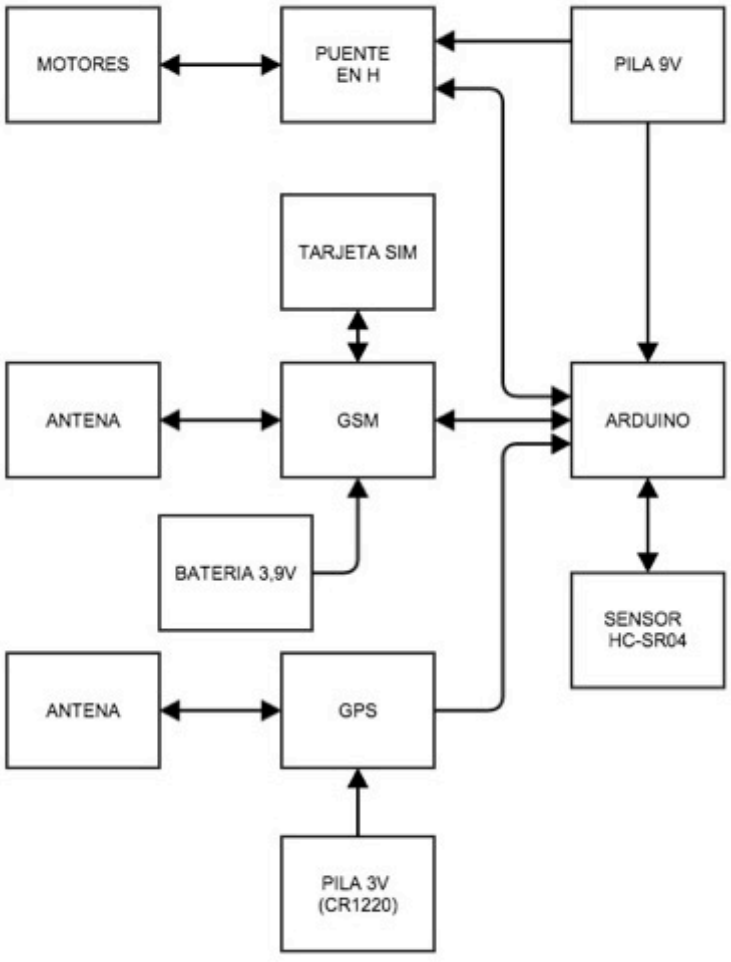
La realización de este proyecto viene dada la idea del programa C3IR del grado en ingeniería radioelectrónica, el cual se vasa en la creatividad, colaboración y comunicación.

La finalidad de este proyecto es la creación de un prototipo del control de un coche mediante GSM y GPS que este nos mande los datos de su posición en cada momento a un dispositivo móvil y poder controlarlo mediante SMS.

Descripción del proyecto

Para este cometido utilizaremos un coche montado en el laboratorio con piezas reutilizadas de distintos aparatos, un Arduino y piezas diseñadas mediante impresora 3D, también otros componentes para Arduino como el GPS Adafruit y al GSM Adafruit.

Esquema:



Funcionamiento

Este prototipo consiste en el manejo de un vehículo RC mediante un móvil escribiendo unos mensajes con un código específico para que este haga las funciones que tiene programadas.

Todos estos mensajes deber ir dirigidos al número de la tarjeta SIM que tenemos colocada en el módulo GSM.

Le hemos implementado 3 funciones diferentes, cada una más interesante que la anterior:

1. "Función de control de dirección"

En esta función controlamos el coche y lo dirigimos a las direcciones que nosotros deseemos. Para esto tenemos que escribir unos sencillos comandos en los mensajes.

-Para que se dirija a delante escribimos **#u**

-Para que se dirija a la izquierda escribimos **#l**

-Para que se dirija a la derecha escribimos **#r**

-Para que se dirija atrás escribimos **#d**

-Para que se pare escribimos **#s**

2. "Función búsqueda de Obstáculos"

Para su segunda función el vehículo se moverá continuamente hacia delante hasta que encuentre un obstáculo a menos de 15cm. Cuando lo encuentre, el vehículo se parara, retrocederá y girara a la derecha hasta que encuentre un nuevo obstáculo.

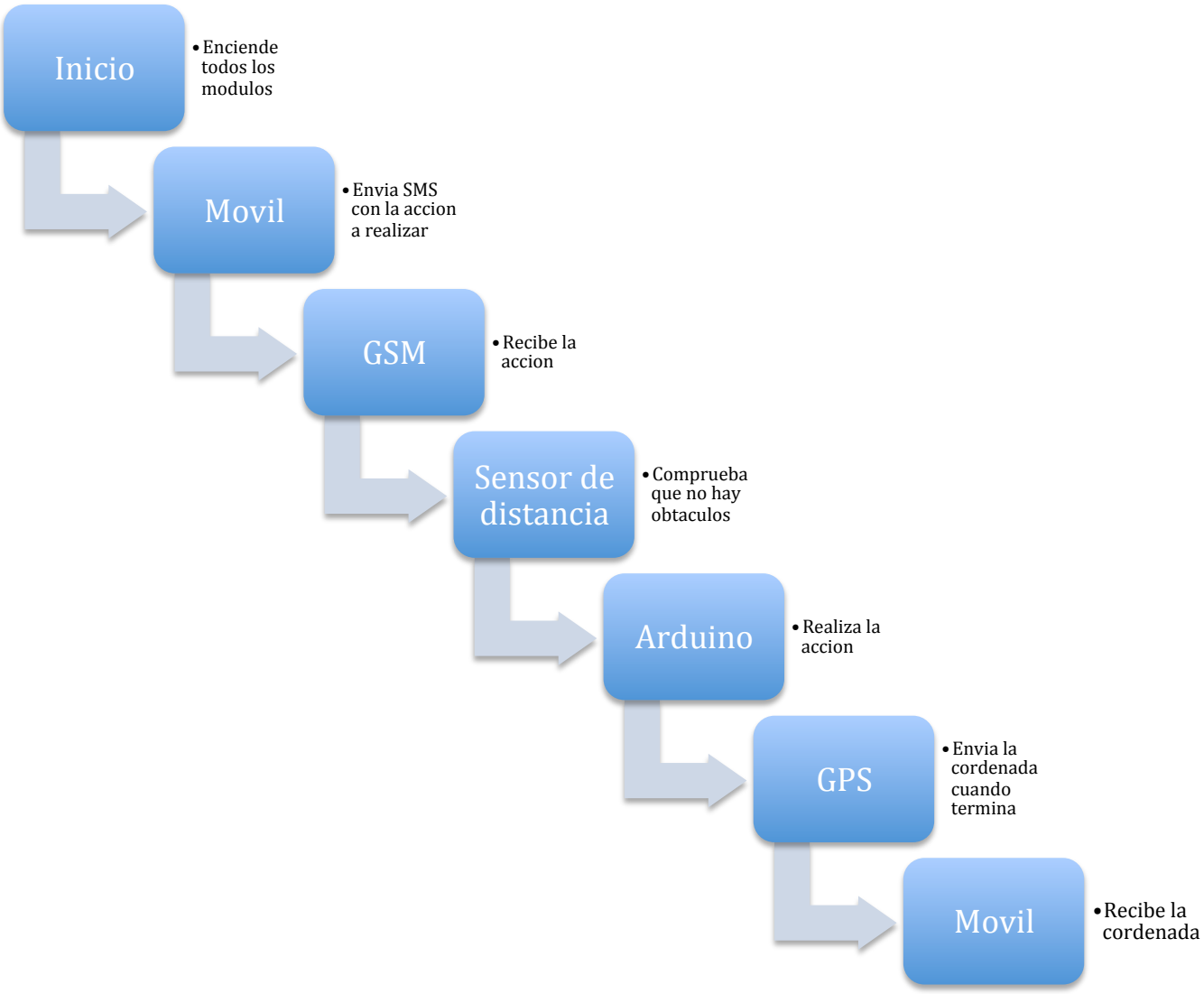
-Para que realice esta función tenemos que escribir **#a**

3. "Función obtención de Datos"

Con esta función al enviar un mensaje al número de la tarjeta SIM este nos responderá los datos de la posición donde se encuentra nuestro vehículo.

Los datos que recibiremos son los siguientes: Hora, Fecha, Calidad de la señal, Longitud, Latitud, Velocidad en Nudos, Angulo y altitud de los satélites y Nº e satélites conectados al GPS.

- Para que realice esta función tenemos que escribir **#p**

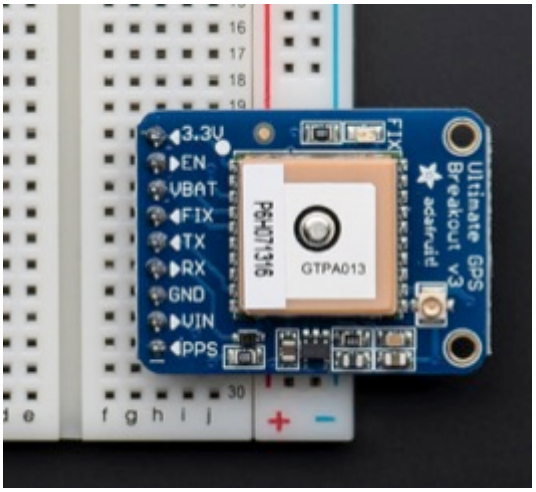

Inicio
• Enciende todos los modulos
Movil
• Envia SMS con la accion a realizar
GSM
• Recibe la accion
Sensor de distancia
• Comprueba que no hay obtaculos
Arduino
• Realiza la accion
GPS
• Envia la cordenada cuando termina
Movil
• Recibe la cordenada

Descripción de los componentes

Arduino: es una plataforma de hardware libre, basada en una placa con un microcontrolador y un entorno de desarrollo, diseñada para facilitar el uso de la electrónica en proyectos multidisciplinares. El hardware consiste en una placa con un microcontrolador Atmel AVR y puertos de entrada/salida.

Modulo GSM: Esta basado en la última versión del conocido SIM800 que permite realizar llamadas de voz, enviar y recibir SMS y datos.
Requiere de un microcontrolador para gestionarlo. En este caso utilizaremos un Arduino pero cualquier microcontrolador de 3-5V con una salid UART podrá controlarlo perfectamente.

Modulo GPS: Tiene una alta sensibilidad (-165dBm), funciona con 5V. Puede alcanzar una velocidad de refresco de hasta 10Hz (programable) y soporta hasta 66 canales para capturar hasta 22 satélites simultáneamente.
Está construido a partir del chip MTK3339, un GPS de alta calidad y grandes prestaciones con antena integrada y que consume tan solo 20mA.
Tiene una flash interna que permite data-loggin, pudiendo guardar unas 16 horas de información.



Puentes en H L298: Éste controlador te permitirá controlar hasta 2 potentes motores de corriente continua de forma muy sencilla y eficaz. El controlador permite controlar el sentido de giro y velocidad mediante señales TTL que podrás sacar de tu microcontrolador. Cada puente está separado y dispone de indicadores LED de funcionamiento.

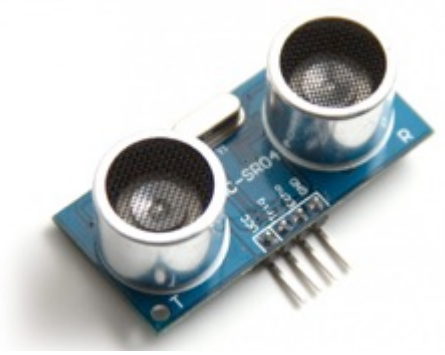
Motores: dos motores de 65mm con reductora con ruedas y soportes.

Antenas:
Para el GSM utilizamos una Antena GSM con conector SMA, esta pequeña antena de alta calidad es quadribanda y dispone de un conector SMA. soporta frecuencias de entre 824Mhz a 1990Mhz.

Para el GPS utilizamos una antena GPS 3V con base magnética para módulos GPS de 3V.

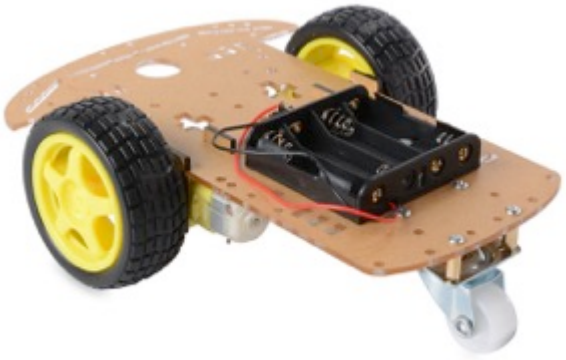
Batería: Para la utilización de modulo GSM necesitamos una batería recargable de 3,9V

Sensor distancia HC-SR04: Sensor de ultrasonidos HC-SR04 tiene una buena precisión, alcance, y la calidad.

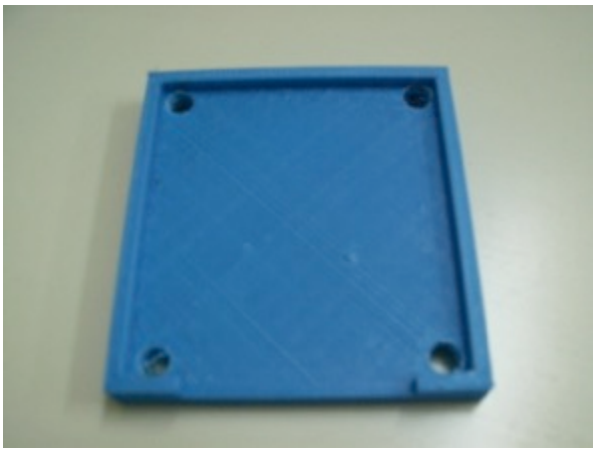


Pilas: Para alimentar al Arduino y a los motores utilizamos una pila de 9V y para el GPS utilizamos una pila CR1220 de 3V.

Estructura: La estructura es de metacrilato cortada con laser y con algunas perforaciones para poder sujetar los componentes del proyecto. También se le a añadido alguna piezas realizadas con impresora 3D.

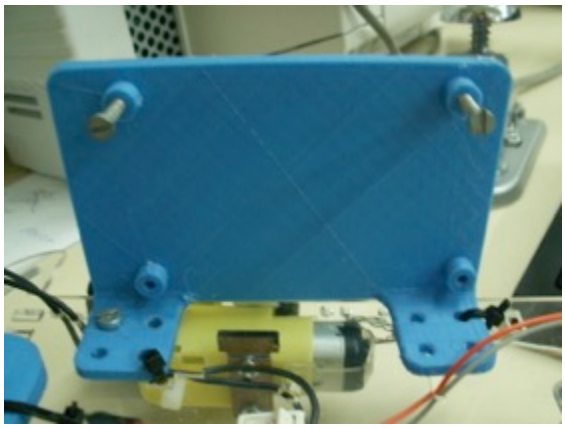


Soporte para Puente en H: Es un soporte creado mediante una impresora 3D para poder colocar el Puente en H.



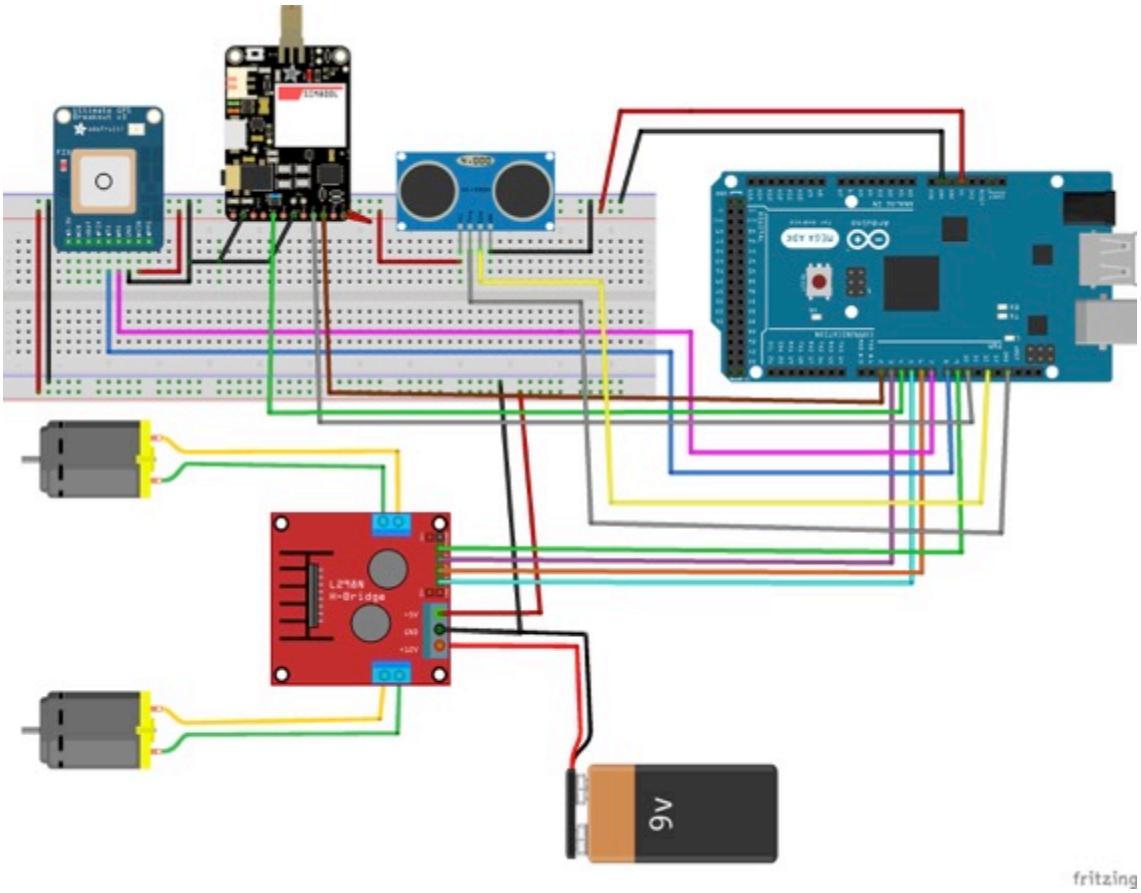
Soporte para Sensor de Distancia: Es un soporte creado mediante una impresora 3D para poder colocar el Sensor de Distancia.

Soporte para Arduino: Es un soporte creado mediante una impresora 3D para poder colocar el Arduino.



Esquema de conexiones

En el siguiente esquema podemos observar mediante una imagen creada por el programa Fritzing el sistema de conexionado de los diferentes pines del circuito.



Programa

En esta parte expondremos el programa completo creado para el funcionamiento de este prototipo, la explicación de cada parte se encuentra en los comentarios escritos en la parte derecha de cada sentencia.

- En esta parte escribimos las llamadas a librería de los módulos y declaramos los pines que vamos a utilizar.

```
#include <SoftwareSerial.h> // son las llamada a libreria para que funcionen el GPS y el GSM
#include <Adafruit_GPS.h>
char estado;
Adafruit_GPS GPS(&mySerial);
#define GPSECHO  true
boolean usingInterrupt = false;
void useInterrupt(boolean);
SoftwareSerial SIM800L(2, 10); //pines del GSM
SoftwareSerial mySerial(8, 7); //pines del GPS

int MotorIzqA = 5;        //Define  el pin 5 como variable para el motor izquierdo
int MotorIzqB = 6;        //Define  el pin 6 como variable para el motor izquierdo
int MotorDerA = 3;        //Define  el pin 9 como variable para el motor derecho
int MotorDerB = 9;       //Define  el pin 10 como variable para el motor derecho
int vel = 255;
int vell = 120;           // Velocidad de los motores (0-255)
int estadol = 'o';         // Empieza parado

int Recibe = 12;           // Define el pin dos como el que emite el eco del sensor
int Emite = 13;            // Define el pin dos como el que recibe el eco del sensor
int duracion, distancia;  // Para Calcular distacia
```

- En el Setup escribimos las velocidades de transmisión del GSM y del GPS además de la declaración de los pines de entrada y salida y la activación del GSM.

```
void setup()
{
  Serial.begin(19200);  // velocidad de tansmision del GSM
  GPS.begin(9600);      // velocidad de transmision del GPS
  GPS.sendCommand(PMTK_SET_NMEA_OUTPUT_RMCGGA); // estipula la cantidad de datos requeridos por el GPS
  GPS.sendCommand(PMTK_SET_NMEA_UPDATE_1HZ);   // velocidad a la que renueva los datos
  GPS.sendCommand(PGCMD_ANTENNA);              // actualizaciones de la antena GPS
  useInterrupt(true);

  pinMode(MotorDerA, OUTPUT);   // Define los puertos como salida
  pinMode(MotorDerB, OUTPUT);   // Define los puertos como salida
  pinMode(MotorIzqA, OUTPUT);   // Define los puertos como salida
  pinMode(MotorIzqB, OUTPUT);   // Define los puertos como salida

  pinMode(Recibe,INPUT);   // Define el pin 2 como entrada (Recibe)
  pinMode(Emite,OUTPUT);   // Define el pin 3 como salida  (Emite)
  pinMode(13,OUTPUT);      // Encendemos un pequeño LED cuando encontramos un obstaculo

  // Activa el GSM
  SIM800L.begin(19200);
  delay(20000);  // da tiempo para conectarse a la red
  SIM800L.print("AT+CMGF=1\r");  // Pone el SMS en modo texto
  delay(100);
  SIM800L.print("AT+CNMI=2,2,0,0,0\r");
  delay(100);
  Serial.println("Preparado"); // ponemos PREPARADO para introducir los comandos
}
```

- Aquí exponemos los comandos para la librería del GPS

```
SIGNAL(TIMER0_COMPA_vect) { // Son los comandos tomados por la libreria del GPS para optener los datos
  char c = GPS.read();

#ifdef UDR0
  if (GPSECHO)
    if (c) UDR0 = c;

#endif
}

void useInterrupt(boolean v) {
  if (v) {

    OCR0A = 0xAF;
    TIMSK0 |= _BV(OCIE0A);
    usingInterrupt = true;
  } else {

    TIMSK0 &= ~_BV(OCIE0A);
    usingInterrupt = false;
  }
}
uint32_t timer = millis();
```

- En el loop escribimos las 3 funciones diferentes que va a realizar nuestro prototipo: el movimiento en las diferentes direcciones, el modo "Búsqueda de Obstáculos" y obtención de datos de posición.

```
void loop()
{

  if(SIM800L.available() >0) // enviamos un mensaje de texto con #,
  estado1 = Serial.read();   //+ la letra de la funcion que queremos utilizar
  {
   estado=SIM800L.read();
    if (estado=='#')
    {
      delay(10);

 estado=SIM800L.read();
 if(estado=='u'){                  //  direccion hacia delante
    analogWrite(MotorDerB, 0);
    analogWrite(MotorIzqB, 0);
    analogWrite(MotorDerA, vel);
    analogWrite(MotorIzqA, vel);
  }
  estado=SIM800L.read();
  if(estado=='l'){                  //  direccion hacia la izquierda
    analogWrite(MotorDerB, 0);
    analogWrite(MotorIzqB, 0);
    analogWrite(MotorDerA, vell);
    analogWrite(MotorIzqA, 0);
  }
```

```
  }
  estado=SIM800L.read();
  if(estado=='s'){                  //  paro
    analogWrite(MotorDerB, 0);
    analogWrite(MotorIzqB, 0);
    analogWrite(MotorDerA, 0);
    analogWrite(MotorIzqA, 0);
  }
  estado=SIM800L.read();
  if(estado=='r'){                  // direccion hacia la derecha
    analogWrite(MotorDerB, 0);
    analogWrite(MotorIzqB, 0);
    analogWrite(MotorDerA, 0);
    analogWrite(MotorIzqA, vell);
  }
  estado=SIM800L.read();
  if(estado=='d'){                  //  direccion hacia atras
    analogWrite(MotorDerA, 0);
    analogWrite(MotorIzqA, 0);
    analogWrite(MotorDerB, vel);
    analogWrite(MotorIzqB, vel);
  }

 estado=SIM800L.read();                                  //Recibimos un mensaje de texto con la posicion,
  if(estado=='p'){                                       //tiempo, fecha,calidad, nº de satelites....
    SIM800L.print("AT+CMGF=1\r");                       // Comando para enviar el mensaje
    delay(100);
    SIM800L.println("AT + CMGS = \"+34633358029\"");     // Movil al que se le envia el mensaje
    delay(100);
    SIM800L.print(GPS.hour, DEC); SIM800L.print(':');
    SIM800L.print(GPS.minute, DEC); SIM800L.print(':');
    SIM800L.print(GPS.seconds, DEC); SIM800L.print('.');
    SIM800L.println(GPS.milliseconds);
    SIM800L.print("Date: ");
    SIM800L.print(GPS.day, DEC); SIM800L.print('/');
    SIM800L.print(GPS.month, DEC); SIM800L.print("/20");
    SIM800L.println(GPS.year, DEC);
    SIM800L.print("Fix: "); SIM800L.print((int)GPS.fix);
    SIM800L.print(" quality: "); SIM800L.println((int)GPS.fixquality);
    if (GPS.fix) {
      SIM800L.print("Location: ");
      SIM800L.print(GPS.latitude, 4); SIM800L.print(GPS.lat);
      SIM800L.print(", ");
      SIM800L.print(GPS.longitude, 4); SIM800L.println(GPS.lon);
      SIM800L.print("Location (in degrees, works with Google Maps): ");
      SIM800L.print(GPS.latitudeDegrees, 4);
      SIM800L.print(", ");
      SIM800L.println(GPS.longitudeDegrees, 4);

      SIM800L.print("Speed (knots): "); SIM800L.println(GPS.speed);
      SIM800L.print("Angle: "); SIM800L.println(GPS.angle);
      SIM800L.print("Altitude: "); SIM800L.println(GPS.altitude);
      SIM800L.print("Satellites: "); SIM800L.println((int)GPS.satellites);
  delay(100);
  SIM800L.println((char)26);                        // Comando para escribirlo en codigo ASCII
  delay(100);
  SIM800L.println();
  delay(5000);                                      // Da al modulo tiempor para enviar el SMS

}

  }
```

```
}
estado=SIM800L.read();
if (estado =='a'){              // se mueve buscando un obstaculo

   digitalWrite(Emite, HIGH);   // Genera el pulso del sensor a 10us
   delay(0.01);
   digitalWrite(Emite, LOW);    // Apaga el Emisor del pulso

   duracion = pulseIn(Recibe, HIGH);              // Lee el tiempo del Eco
   distancia = (duracion/2) / 29;                 // Calcula la distancia en centimetros
   delay(10);                                     // Espera 1 ns

   if (distancia <= 15 && distancia >=2){         // si la distancia es menor de 15cm
      digitalWrite(13,HIGH);                      // Enciende LED

      analogWrite(MotorDerB, 0);                  // Parar los motores por 300ms
      analogWrite(MotorIzqB, 0);
      analogWrite(MotorDerA, 0);
      analogWrite(MotorIzqA, 0);
      delay (300);

      analogWrite(MotorDerB, vel);          // Pone marcha atras durante 500ms
      analogWrite(MotorIzqB, vel);
      delay(500);


      analogWrite(MotorDerB, 0);            // Girar a derecha durante 500ms
      analogWrite(MotorIzqB, 0);
      analogWrite(MotorDerA, 0);
      analogWrite(MotorIzqA, vel);
      delay(500);

      digitalWrite(13,LOW);
   }
   else{                                    // Si no hay obstaculos se desplaza al frente
       analogWrite(MotorDerB, 0);
       analogWrite(MotorIzqB, 0);
       analogWrite(MotorDerA, vel);
       analogWrite(MotorIzqA, vel);
   }
}
estado=SIM800L.read();
if(estado=='o'){                            // Boton OFF, detiene los motores no hace nada
   analogWrite(MotorDerB, 0);
   analogWrite(MotorIzqB, 0);
   analogWrite(MotorDerA, 0);
   analogWrite(MotorIzqA, 0);
            delay(10);
          }
        }
        SIM800L.println("AT+CMGD=1,4"); // borra todos los mensajes
}
}
```

Bibliografía

www.adafruit.com
www.bricogeek.com
www.arduino.cc