

ESCUELA DE EDUCACION SECUNDARIA TECNICA N° 5

“2 DE ABRIL” – TEMPERLEY – BUENOS AIRES



## Car Guider

MATERIA : PROYECTO Y DISEÑO ELECTRONICO

FECHA : 4/11/2022

AUTORES:

Grupo: Robobrain

Alumnos: Benítez Gabriel  
Benítez Bruno  
Campusano Nicolás

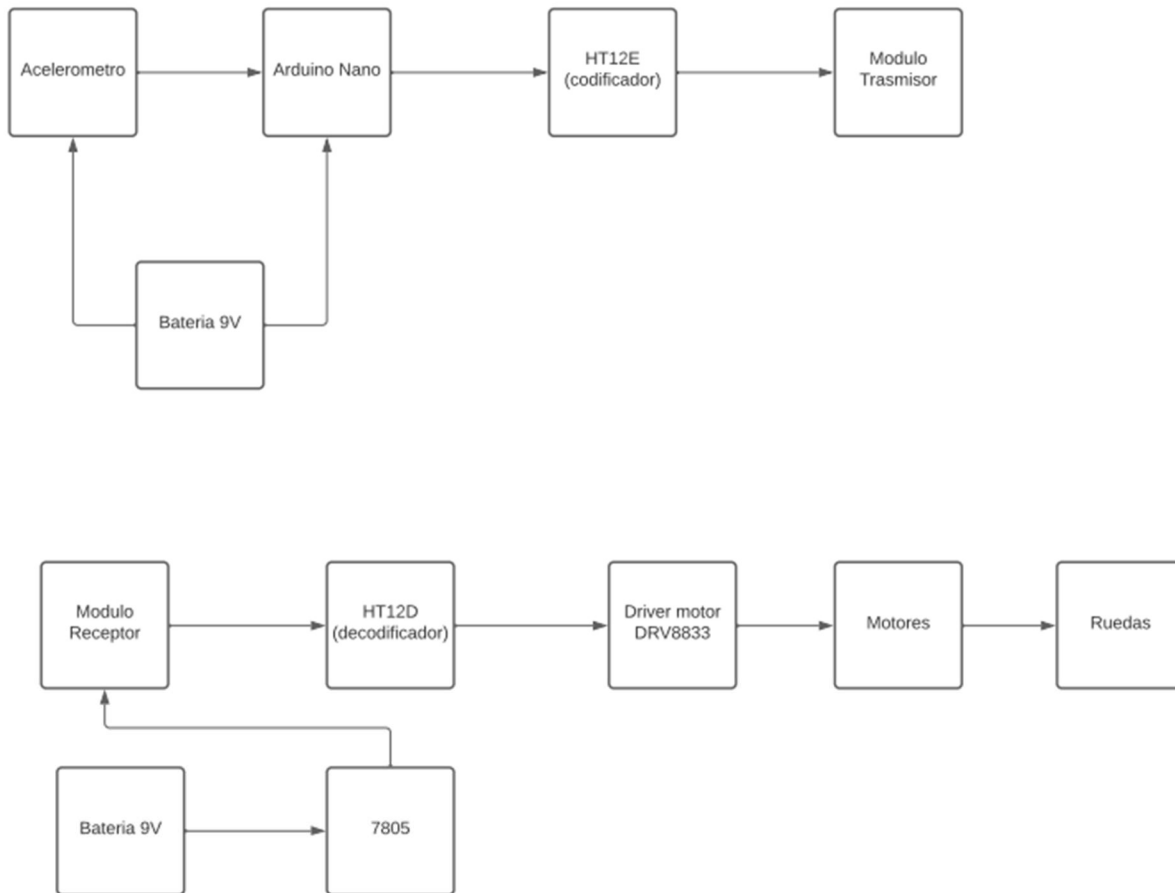
NOTA:

PROFESOR: ING. MARTIN LEGUIZAMON

## CONTENIDO

Descripción general.....	3
Etapa acelerómetro.....	4
Circuito, Schematic.....	4
PCB,resumen.....	5
Etapa Auto.....	6
Circuito, Schematic.....	6
PCB, resumen.....	7
Código Arduino.....	8
Explicación código.....	9
Costos.....	10
Componentes.....	10
Recursos disponibles.....	11
Bibliografía.....	11

## Descripción general:



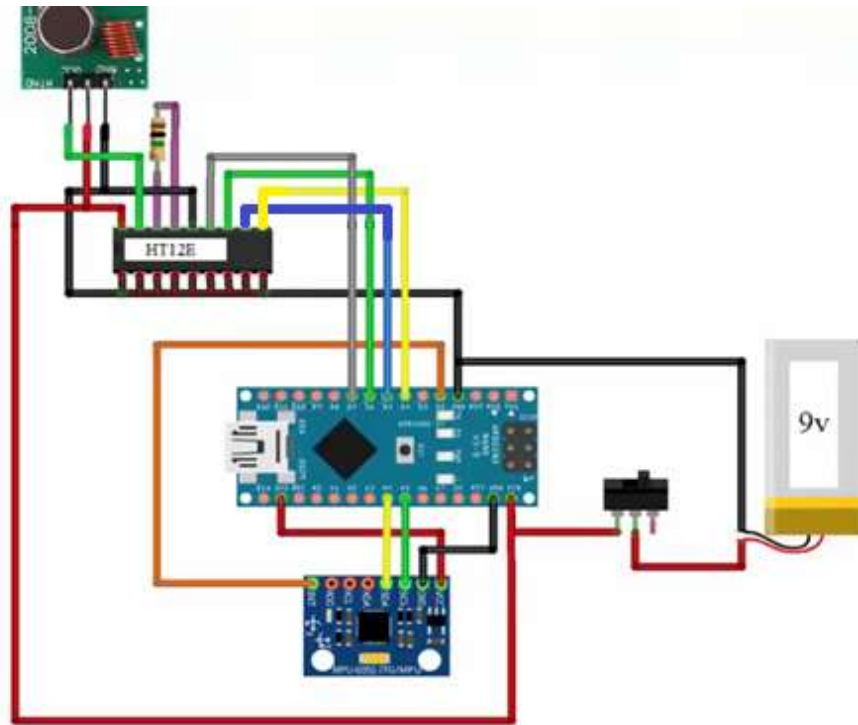
El proyecto trata sobre el desarrollo de un auto controlado remotamente a través de gestores de mano donde los movimientos de la mano del usuario que tendrá puesto un guante le darán el movimiento. Este guante tendrá un acelerómetro conectado a un Arduino nano el cual mediante un codificador HT12E se conecta al transmisor de la señal módulo RF emisor, cuyos datos se transmitirán hacia el auto que contiene el receptor RF donde este está conectado al decodificador HT12E que enviará una señal decodificada al Driver motor DRV8833 y este le dará el movimiento a las ruedas. Además el guante dispondrá de una batería de 9v que alimentará el circuito. Para la alimentación del auto utilizaremos una batería de 9v y el Driver8833 tendrá su alimentación aparte con un porta pilas para 2 pilas 18650. Esto debido a que si el driver comparte la misma alimentación con el integrado le entrega ruido al mismo haciendo que se cuelgue y dejando de responder. Además para que esto no suceda se deben soldar 2 capacitores de 100u en los bornes de los motores para que absorba el ruido.

Como se expresó anteriormente este proyecto cuenta con 2 etapas:

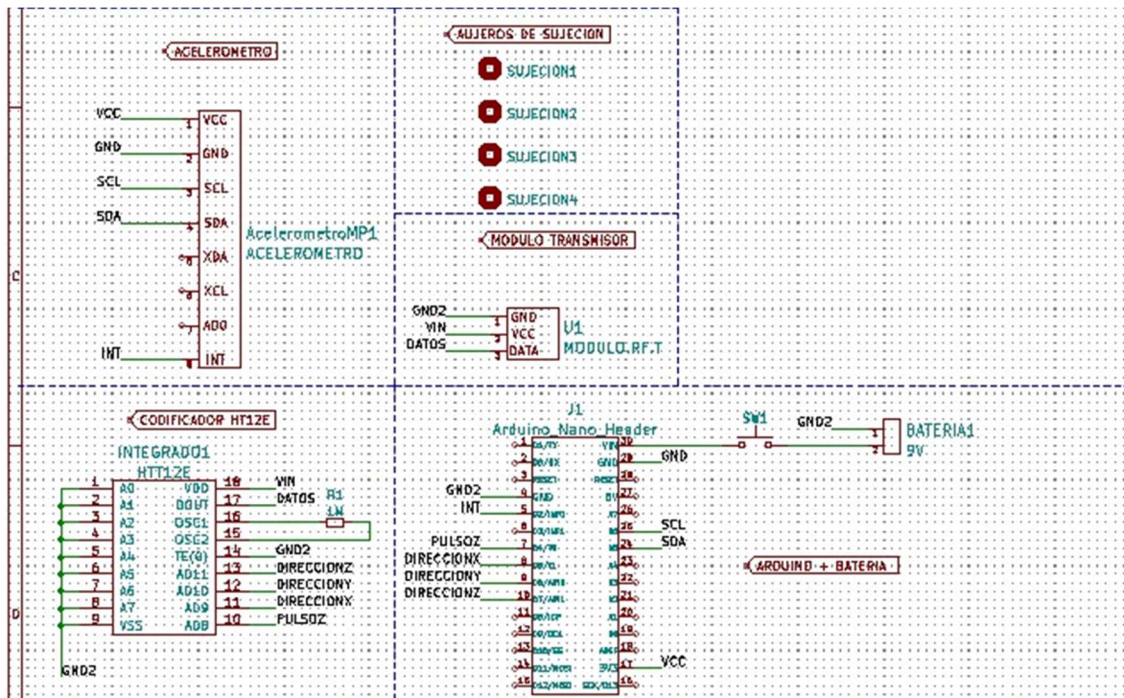
La etapa que cuenta con el acelerómetro y la etapa que cuenta con los motores.

## Etapas acelerómetro:

Circuito:



## Schematic:

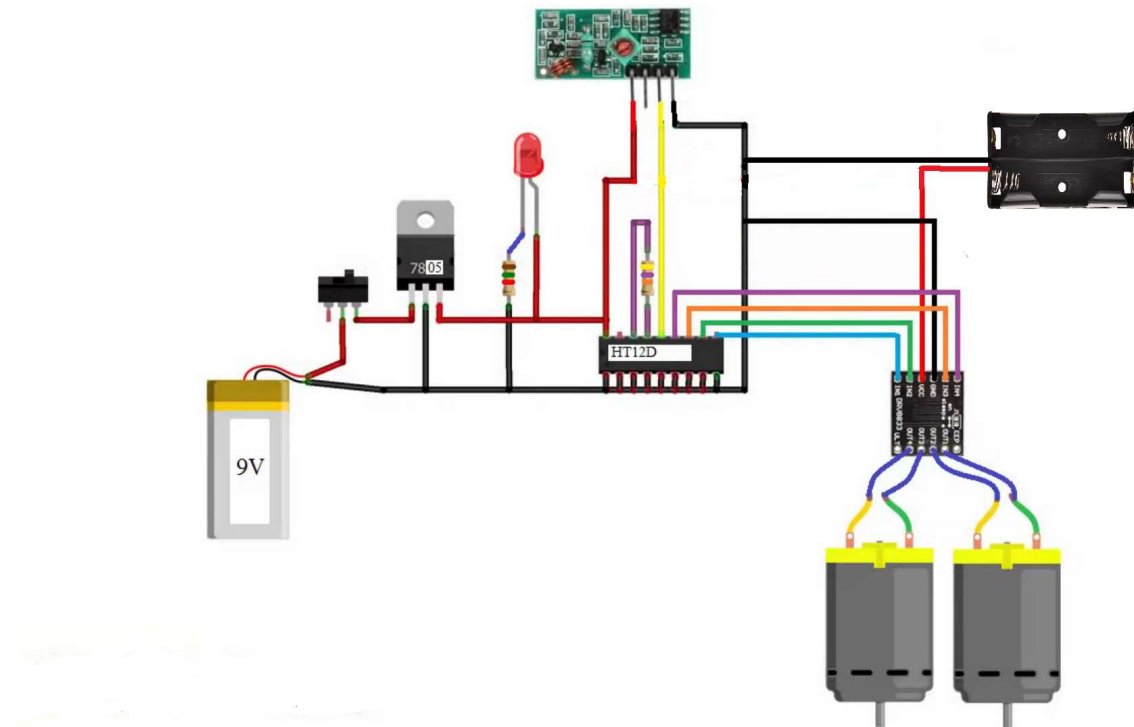


The schematic shows a custom PCB layout for a sensor module. The board is populated with various components including a central microcontroller (HTT12E), a sensor (HTT12E), a resistor (R1), and a battery (BATERIA1). It features four mounting holes (SUJECION1, SUJECION2, SUJECION3, SUJECION4) and a connector (CON1). Red lines indicate the routing of signals and power across the board.

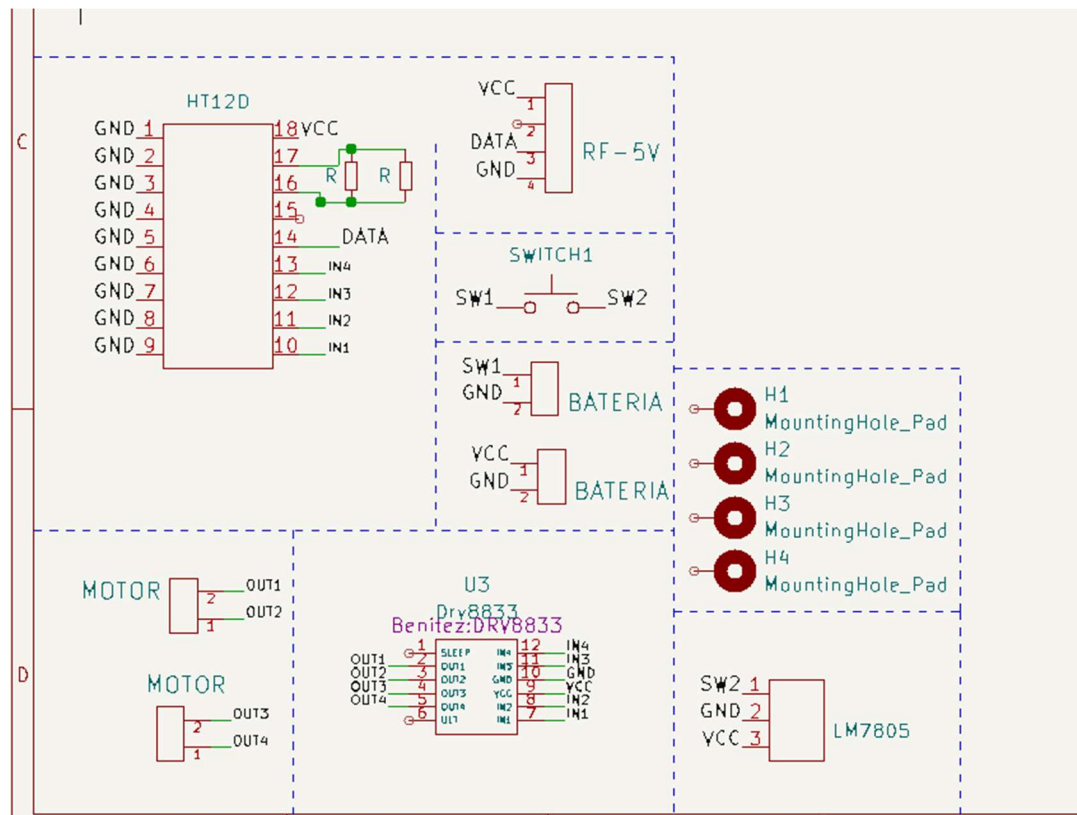
En esta etapa se encuentra el acelerómetro que va a ser el encargado de tomar y capturar la información de los movimientos de la mano (arriba, abajo, izquierda y derecha), midiendo ángulos mediante un giroscopio, y enviará dicha información hacia el arduino quien va a tomar y procesar esta información, y posteriormente le enviará señales de pwm al HT12E para que este la codifique y envíe a la siguiente etapa a través del módulo RF emisor.

## ETAPA MOTOR:

### Circuito:

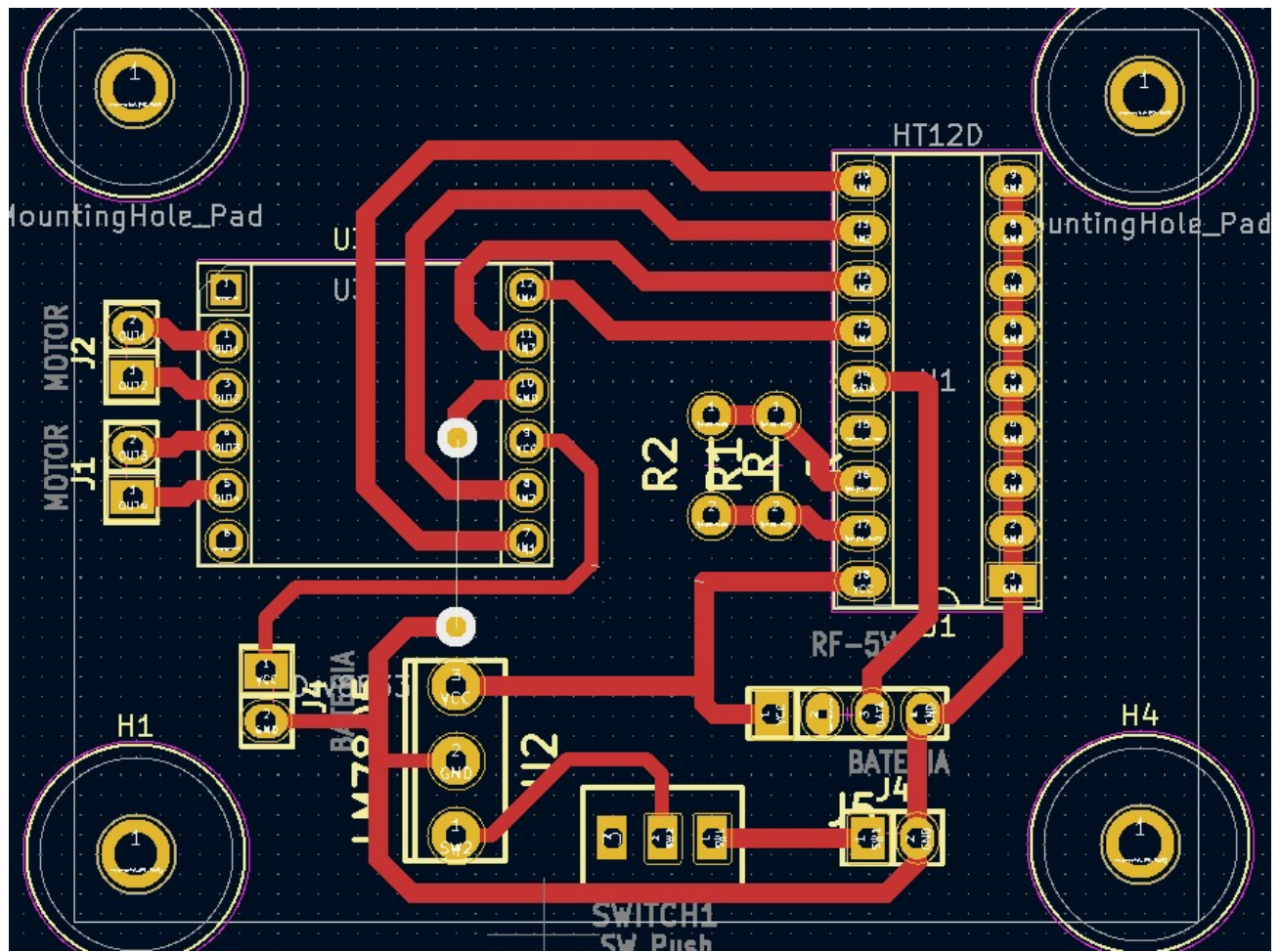


### Schematic:





PCB:



## RESUMEN

A esta etapa llegará la información codificada desde el transmisor, la recibirá el módulo RF receptor y la enviará al HT12E para que este la decodifique y la envíe al driver de motor DRV8833 el cual enviará los pulsos para que los motores se muevan a la posición establecida por el acelerómetro al momento de moverlo hacia la posición determinada.

## Codigo Arduino:

```
1  #include<MPU6050_tockn.h>
2  #include<Wire.h>
3
4  MPU6050 mpu6050(Wire);
5
6  long timer = 0;
7
8  void setup() {
9    pinMode(4, OUTPUT);
10   pinMode(5, OUTPUT);
11   pinMode(6, OUTPUT);
12   pinMode(7, OUTPUT);
13
14   Serial.begin(9600);
15   Wire.begin();
16   mpu6050.begin();
17
18 }
19
20 void loop() {
21   mpu6050.update();
22
23   if(mpu6050.getAccX()>= 0.70){
24     Serial.println("adelante");
25     digitalWrite(4,HIGH);
26     digitalWrite(5,LOW);
27     digitalWrite(6,LOW);
28     digitalWrite(7,HIGH);
29   }
30 }
```

```
31  else if (mpu6050.getAccX()<= -0.60){
32     Serial.println("atras");
33     digitalWrite(4,LOW);
34     digitalWrite (5,HIGH);
35     digitalWrite(6,HIGH);
36     digitalWrite(7,LOW);
37   }
38
39  else if (mpu6050.getAccY()>= 0.60){
40     Serial.println("izquierda");
41     digitalWrite(4,HIGH);
42     digitalWrite(5,LOW);
43     digitalWrite(6,LOW);
44     digitalWrite(7,LOW);
45   }
46  else if (mpu6050.getAccY()<= -0.60){
47     Serial.println("derecha");
48     digitalWrite(4,LOW);
49     digitalWrite(5,LOW);
50     digitalWrite(6,LOW);
51     digitalWrite(7,HIGH);
52   }
53
54  else
55  {
56     Serial.println("parado");
57     digitalWrite(4,LOW);
58     digitalWrite(5,LOW);
59     digitalWrite(6,LOW);
60     digitalWrite(7,LOW);
61   }
62 }
```



## EXPLICACION CODIGO:

Primero se incluyen las librerías que utilizamos ("Wire.h" y "MPU6050\_tockn.h")

- La librería Wire sirve para comunicarse con dispositivos I2C/TWI.
- La librería MPU6050 facilita la comunicación con el MPU6050. Puede obtener datos de aceleración, giroscopio y ángulo.

Seguido a esto se crea el objeto mpu6050 mediante un constructor y se le asigna Wire como parámetro:

**MPU6050 mpu6050(Wire);**

En el setup primero establecemos los pines que utilizaremos para enviar pwm a los motores como salidas (4,5,6 y 7). Luego se establece la velocidad de **comunicación serial** en **9600** y con la función **begin** se inicializa la biblioteca Wire para que se una al bus I2C como controlador o periférico. Posteriormente se hace lo mismo con el mpu6050 para que este se pueda comunicar.

- **NOTA:** Para poder obtener datos del MPU6050, se debe ejecutar el método **update()**. Este método obtendrá todos los datos de MPU6050 y calculará el ángulo por acelerómetro y giroscopio usando un filtro complementario.

Esos dos coeficientes son determinados por el constructor.

El coeficiente predeterminado del acelerómetro es 0,02, el giroscopio es 0,98.

**Si se desea cambiar el coeficiente del acelerómetro y el coeficiente del giroscopio, es necesario agregarlo en la línea del constructor para aclararlo:**

**" MPU6050 mpu6050(Wire, A, B); "**

Siendo A el coeficiente del acelerómetro y B el coeficiente del giroscopio.

En nuestro caso utilizamos los coeficientes estándar así que omitimos esos valores y solo escribimos **(Wire)** en los parámetros del objeto.

Ya en el loop primero se ejecuta el método **update()** mencionado anteriormente.

Por último colocamos una serie de **If** que se ejecutarán dependiendo los ángulos medidos. Los comandos **"getAccX"** (Para el eje X) y **"getAccY"** (Para el eje Y) obtendrán las medidas del mpu6050. En nuestro caso colocamos valores menores a **-0.60** y mayores a **0.60** para derecha e izquierda y, valores menores a **-0.60** y mayores **0.70** para atrás y adelante.

El último **if** se ejecutará solo si se detectan medidas dentro de los valores asignados (entre -0.60 **R** y 0.60 **L**) (entre -0.60 **B** y 0.70 **A**) para dejar el auto estático

Dentro de cada **If** se establecen los estados para cada una de las salidas que enviarán la señal pwm correspondiente hacia el integrado **HT12E**.

### Costos:

#### COMPONENTE:

Precio \$:

1. Arduino nano:-----	1200\$
2. Acelerómetro MP6050:-----	500\$
3. Modulo RF 433 emisor y receptor:-----	500\$
4. HT12E Y HT12D:-----	700\$
5. Driver motor DRV8833:-----	600\$
6: Motor y ruedas BO:-----	1300\$
7. 2x Batería 9v:-----	1000\$
8. Lm7805:-----	150\$
9. Interruptor:-----	150\$
TOTAL APROXIMADO: 6600\$	

### Componentes:

1. Arduino nano
2. Acelerómetro MP6050
3. Modulo RF 433 emisor y receptor
4. HT12E Y HT12D
5. Driver motor DRV8833
- 6: Motor y ruedas BO
7. Batería 9v
8. Lm7805
9. Interruptor

Recursos disponibles: Equipamiento, componentes, instrumental, lugar de trabajo, etc.

Equipamiento:

\*Pinzas y alicates.(plaqueta)

\*Soldador (plaqueta)

\*Protoboard (para pruebas)

\*PC (Software)

\*Pendrive (guardado del software)

Bibliografía.

DATASHEETS COMPONENTES:

Acelerómetro MP6050:

<https://nomada-e.com/descargas/datasheet/Aceler%c3%b3metro%20y%20Giroscopio.pdf>

Modulo RF 433 emisor y receptor:

<https://store.robodacta.mx/Material/RFMODULOS433MHZ001.pdf>

HT12E Y HT12D

<https://www.farnell.com/datasheets/1899539.pdf>

Driver motor DRV8833:

<https://www.ti.com/lit/ds/symlink/drv8833.pdf?ts=1649298681391>

Lm7805:

<https://www.sparkfun.com/datasheets/Components/LM7805.pdf>

Diagrama de conexiones de referencia:

<https://www.facebook.com/jfetronic/photos/a.1696966023653045/2569543313061974/?type=3&theater>