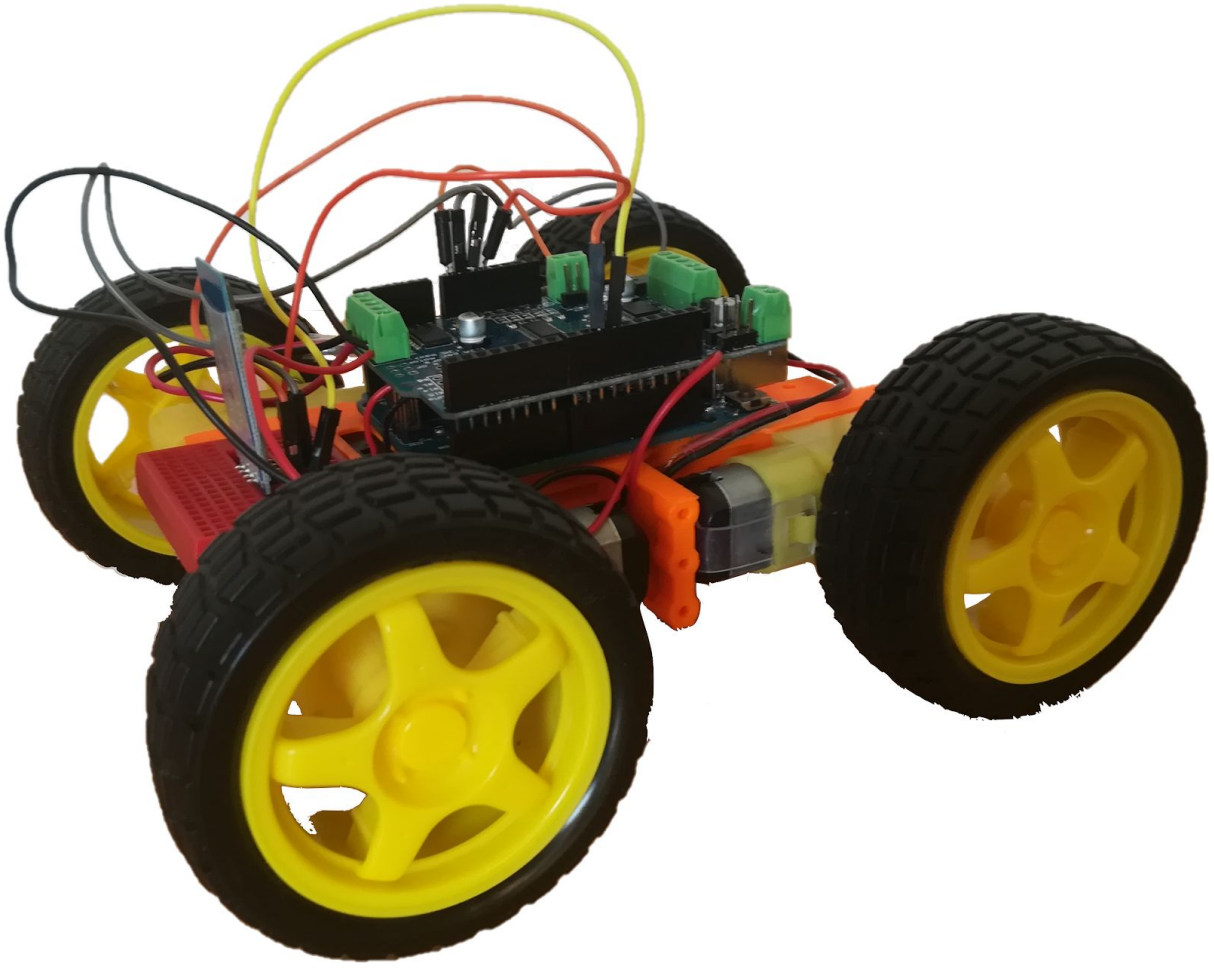
Vehículo controlado con acelerómetro del móvil



Índice:

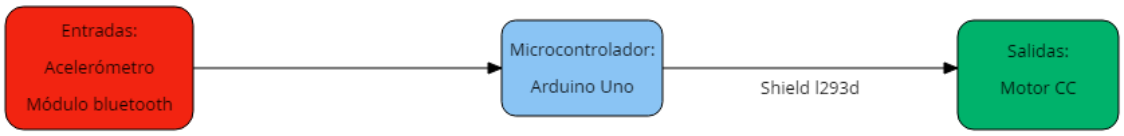
1.Finalidad del proyecto

Este proyecto se basa en el control de un coche mediante el acelerómetro del teléfono, una manera innovadora de controlar un vehículo.

Objetivo mínimo: Control con botones desde el móvil

Ampliaciones: Ajuste a cualquier móvil, sensores de proximidad para evitar colisiones.

Esquema de entradas y salidas:



2.Búsqueda de información

- Adafruit Motor Shield, controlando motores con Arduino- Programar Fácil

Artículo en el que se explica como usar el Shield L293 de Adafruit que vamos a usar en nuestro proyecto, fue nuestra base de partida

- El bus I2C en Arduino- Luis Llamas

Artículo donde explica el funcionamiento del bus I2C, lo leímos para comprender la tecnología con la funciona el shield.

-Videotutorial MIT App Inventor con Arduino y Bluetooth.- Arduino

Tutorial completo de MIT App Inventor, para controlar el Arduino mediante el módulo Bluetooth HC-06 y utilizando Android.

-App inventor 2 en español- kio4.com

Tutorial de iniciación de App Inventor 2 en español

3.Planificación

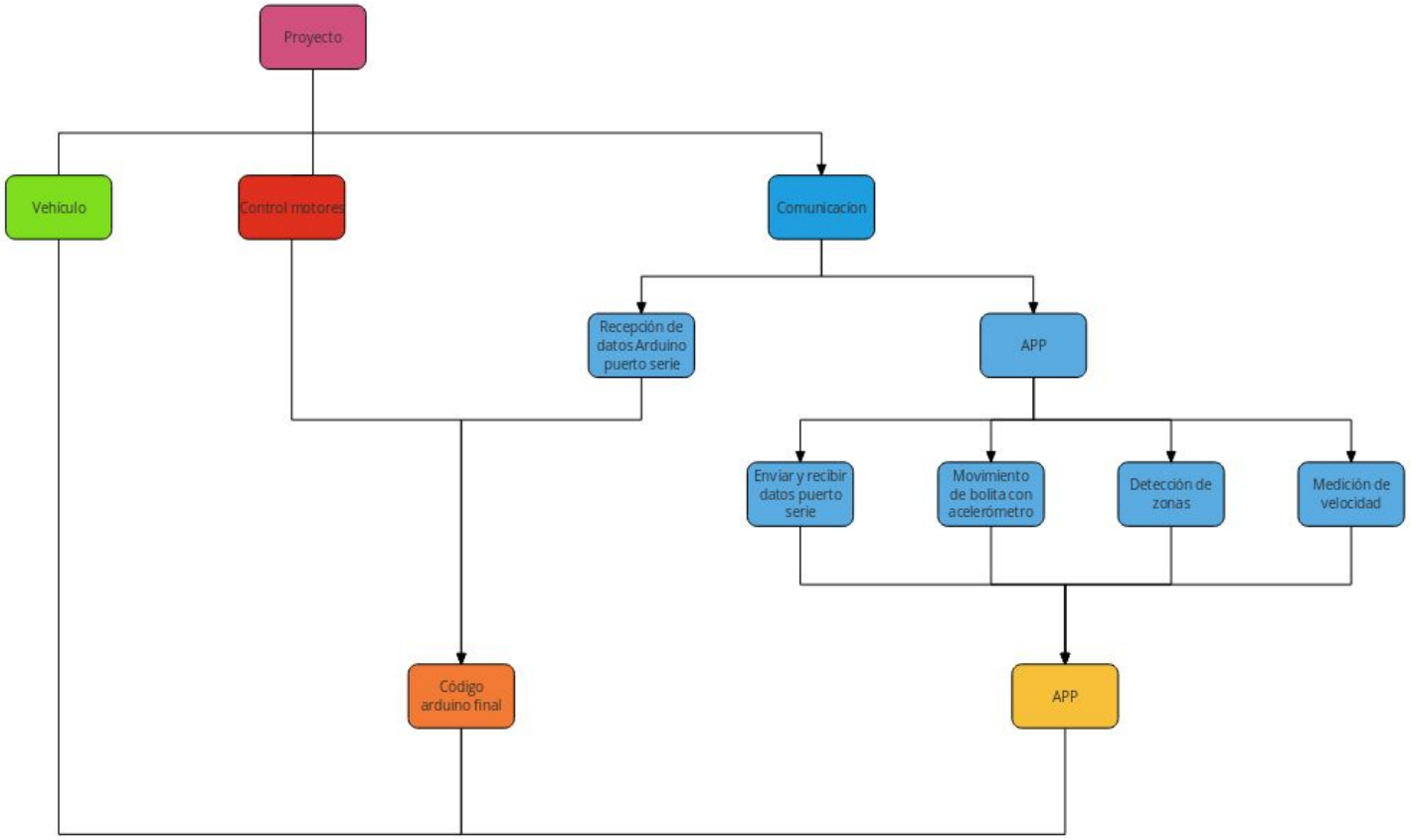
-Tabla de materiales y presupuesto:

Nº	Descripción	Cantidad	Precio Unitario	Precio Total
1	Arduino	1	8,90 €	8,90 €
2	Shield l293d	1	15,00 €	15,00 €
3	Motor CC + Rueda	4	6,20 €	24,80 €
4	Módulo bluetooth HC-06	1	6,80 €	6,80 €
5	Pilas 18650-25R	2	6,12 €	12,24 €
6	Portapilas para dos pilas 18650-25R	1	2,60 €	2,60 €
7	Chasis impreso en 3D	1	6,31 €	6,31 €
8	Pack de Cables Macho-Macho 40 unidades	1	2€	2€
9	Mini Protoboard 170 puntos	1	1,45€	1,45€
10	Cargador de pilas	1	3,45 €	3,45 €
			Total	**83,55 €**

4.Diseño de la maqueta

Diseño maqueta: Arduino Robot Tank Chassis - dan24129

5.Subsistemas



5.1 Motores

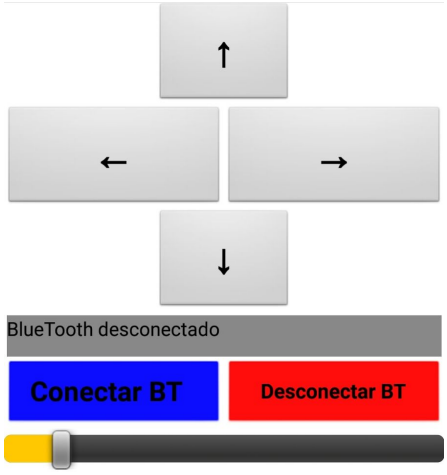
Comenzamos a saber como funciona el shield y los motores probandolo con este código, que es el código de ejemplo que viene acompañando a la librería que hemos instalado. Vamos a *Archivos/Ejemplos/Adafruit Motor Shield V2 Library/DCMotorTest.*:

[Adafruit Motor Shield, controlando motores con Arduino]

5.2 Comunicación

5.2.1 Enviar y recibir datos puerto serie:

Hicimos una app para controlar el coche mediante una botonera y la velocidad mediante un deslizador.



5.2.2 Movimiento de bolita con acelerómetro:

Aquí conseguimos que una bolita se moviera según las lecturas del acelerómetro por la pantalla.

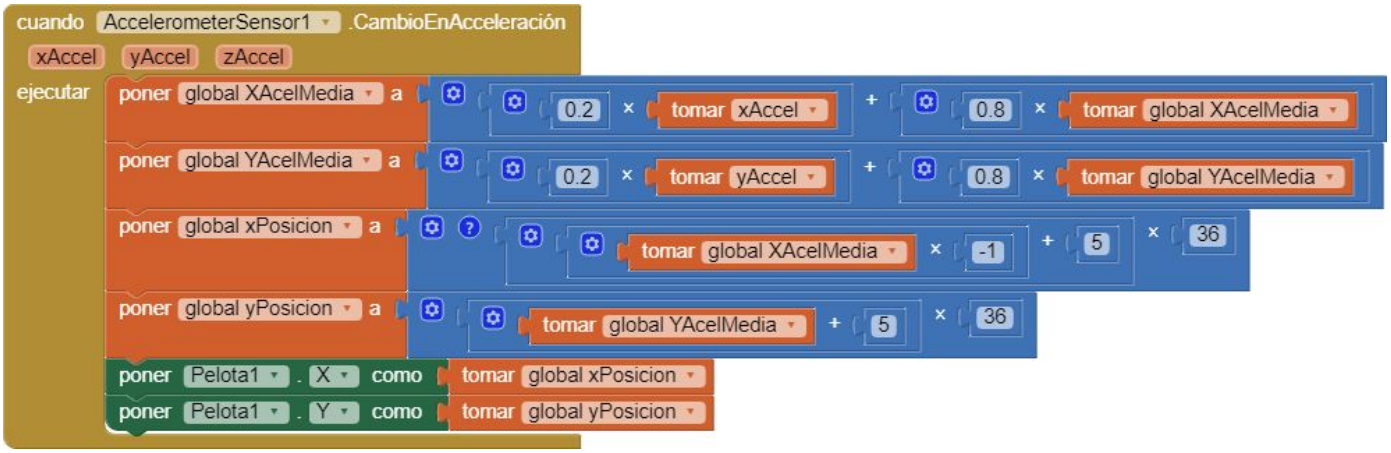
Eliminación del ruido del sensor:
Como el sensor era muy sensible y tenía mucho ruido lo tuvimos que estabilizar realizando medias. La ecuación toma el valor medio y el actual para hacer una media de estos dos valores sumando un 20% del valor actual y el 80% del medio.

```
ValorMedio= Valoractual*0,2+ ValorMedio*0,8
```

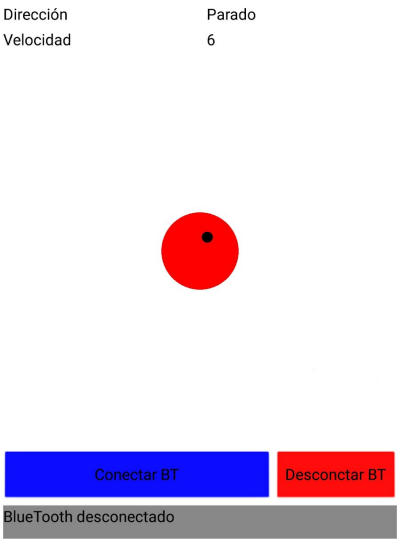
Posición de la bola en pantalla:
Aquí queda como se transforma las medidas de XaccelMedia e YaccelMedia en Xposicion e Yposición
LargoPantalla:360
AltoPantalla:598
Valores máximos y mínimos del acelerómetro:0-9.8258
x e y y valores en el centro(centro del lienzo):180,180

Xposicion=-XaccelMedia+5*36(La variable se pone en negativo ya que los ejes están invertidos)
Yposicion=YaccelMedia+5*36
Estas dos fórmulas salen de multiplicar un número(que según la sensibilidad del movimiento de la bolita será uno mayor o menor,contra menor se el número más sensibilidad tendrá la bolita) y de dividir 180 (que es el centro en el caso de mi dispositivo) entre el número utilizado anteriormente.

Código :

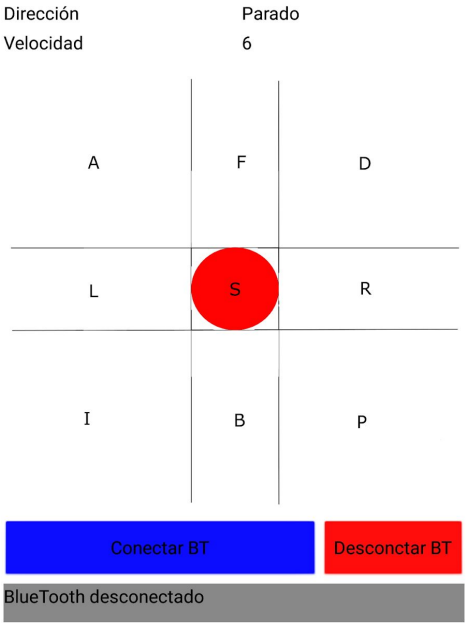


Diseño de la App:



5.2.3 Detección de zonas:

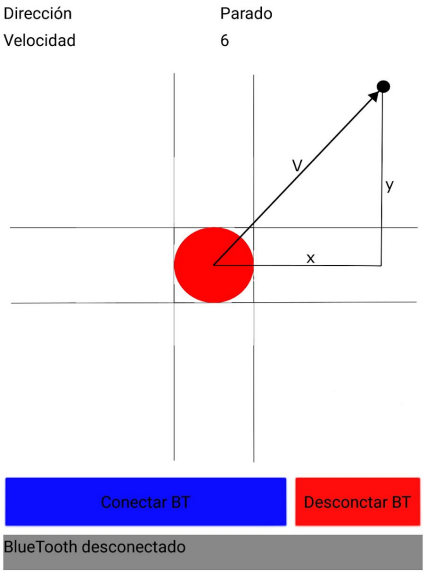
Nos basamos en un cuadrante para que cuando la bolita “tocase” una zona enviara una orden, para el control del vehículo.Se divide en 9 zonas y en cada zona los motores tienen un comportamiento diferente: Recto, diagonal, lado

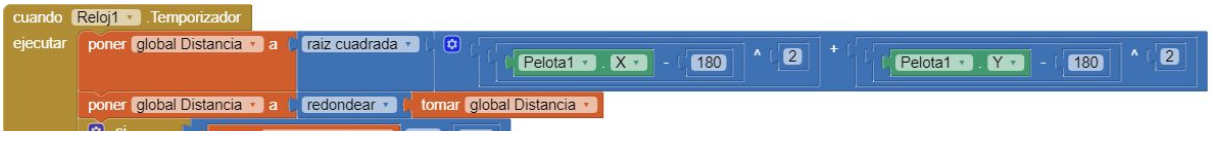


5.2.4 Medición de velocidad:

Este fue el último paso para terminar la app, que era que según la variación de la posición de la bolita se midiera el radio desde el centro de la pantalla hasta la posición de la bolita y que este radio se transformara en la potencia de los motores.

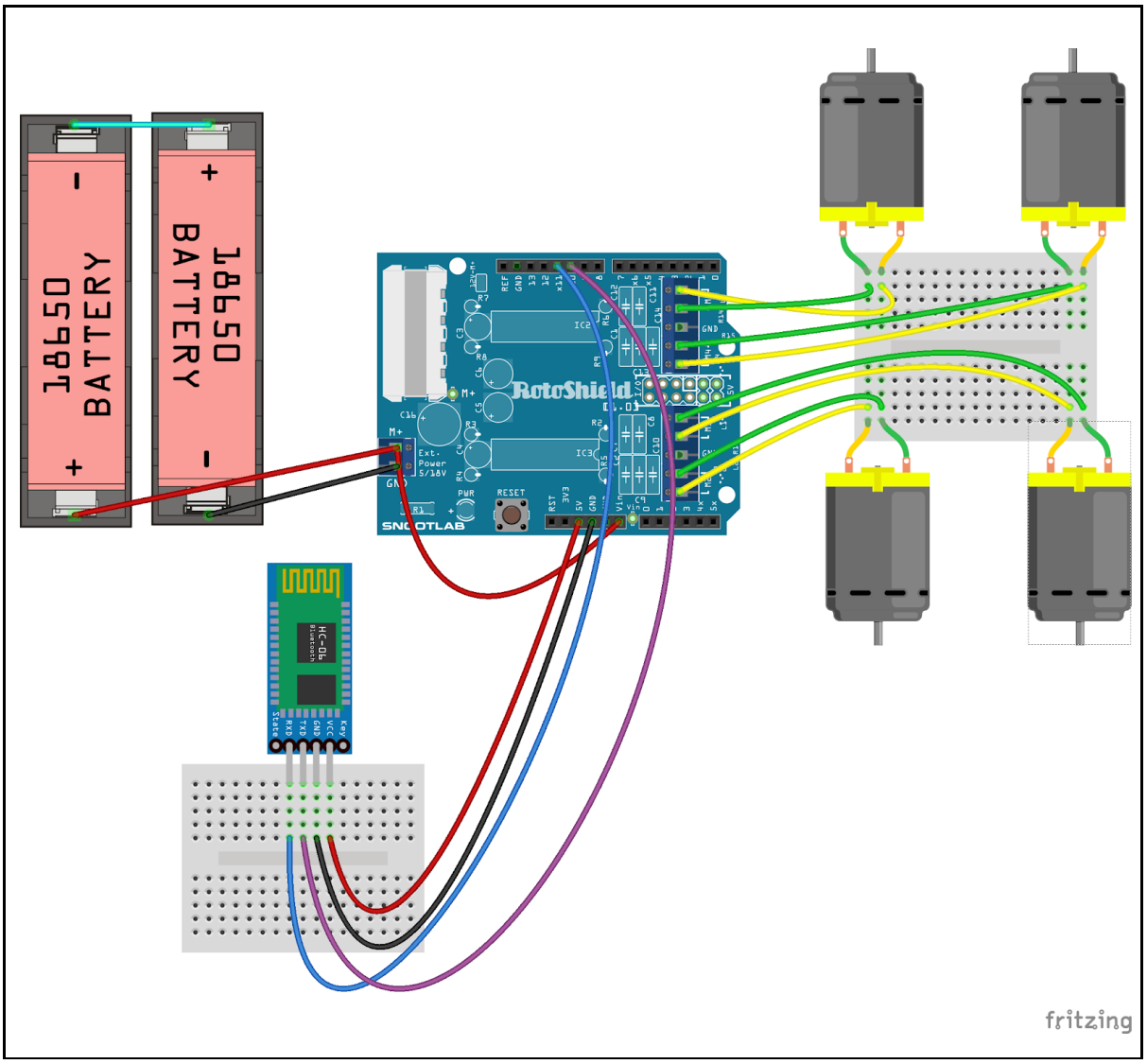
Ecuación: $V = \sqrt{x^2 + y^2}$



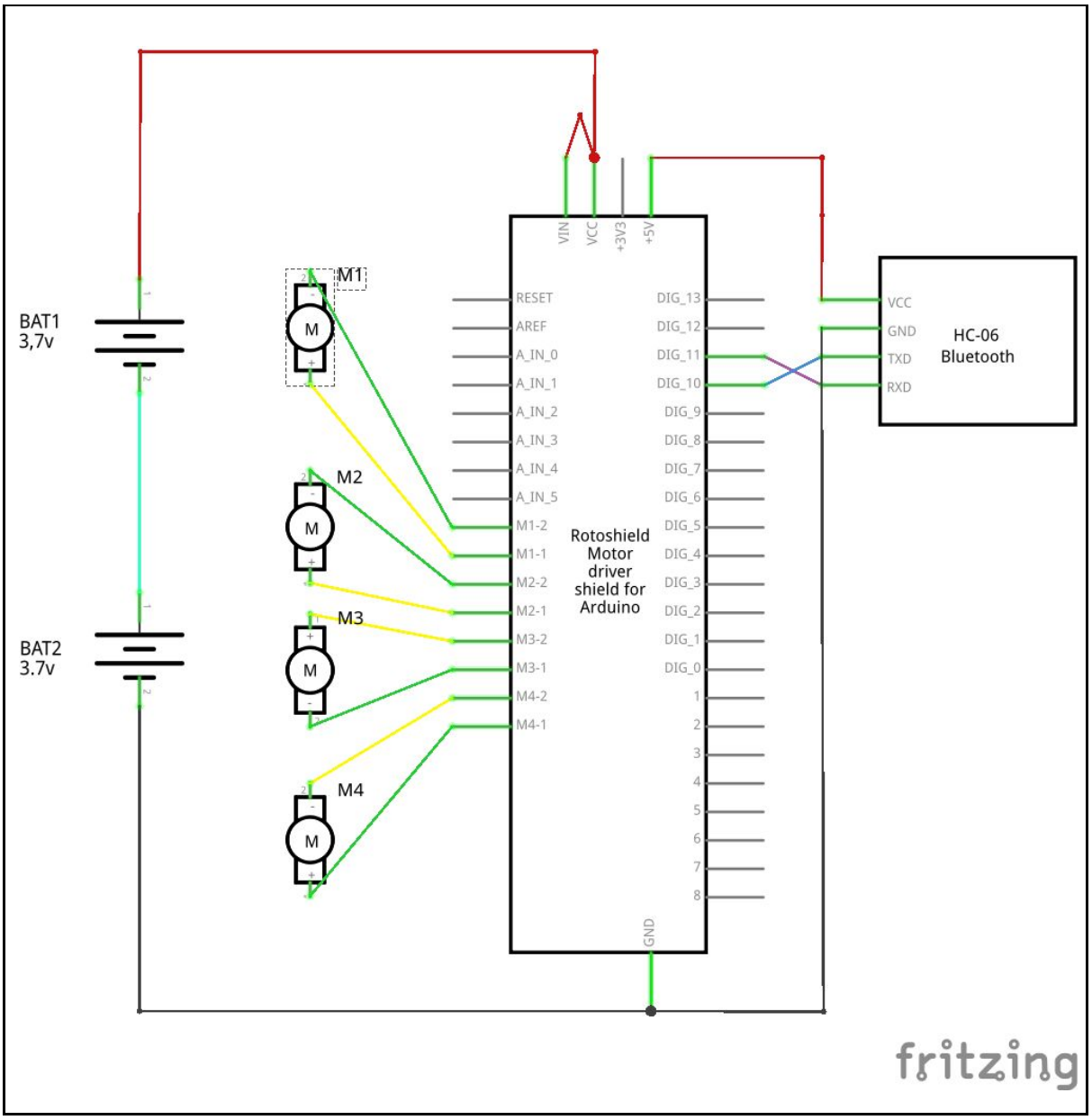


6.Hardware

- Esquema protoboard:



- Esquema electrónico:



7. Código

7.1 Código arduino

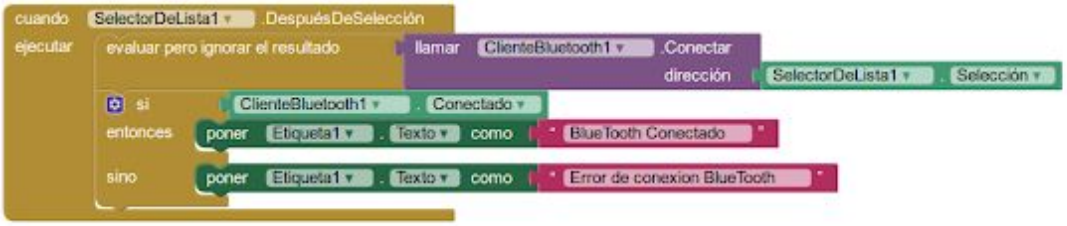
```
#include <SoftwareSerial.h>
SoftwareSerial I2CBT(10, 11);// El TX del módulo BT va al pin 10 del Arduino
// El RX del módulo BT va al pin 11 del Arduino
int incomingByte;      // variable para leer los bytes de entrada
int value = 0;  // variable para almacenar el valor numerico
int velocity = 255;
int velocity2 = 255;
#include <Wire.h>
#include <Adafruit_MotorShield.h>
#include "utility/Adafruit_MS_PWMServoDriver.h"
Adafruit_MotorShield AFMS = Adafruit_MotorShield();
// Seleccione qué 'puerto' M1, M2, M3 or M4
Adafruit_DCMotor *myMotor = AFMS.getMotor(1);
Adafruit_DCMotor *myMotor2 = AFMS.getMotor(2);
Adafruit_DCMotor *myMotor3 = AFMS.getMotor(3);
Adafruit_DCMotor *myMotor4 = AFMS.getMotor(4);

void setup() {
  Serial.begin(9600);          // configurar la biblioteca Serial en 9600 bps
  I2CBT.begin(9600);
  AFMS.begin(1000);  // crear con la frecuencia por defecto 1.6KHz
  // Establezca la velocidad de inicio, de 0 (apagado) a 255 (velocidad máxima)
  velocidadMotores(velocity, velocity, velocity, velocity);
  sentidoMotores(FORWARD, FORWARD, FORWARD, FORWARD);
  sentidoMotores(RELEASE, RELEASE, RELEASE, RELEASE);
}

void loop() {
  if (I2CBT.available() > 0) {
    incomingByte = I2CBT.read();
    Serial.print("Caracter=");
    Serial.println(incomingByte);
    if (incomingByte >= '0' && incomingByte <= '9') {
      //Acumula los datos numericos multiplicando por 10 el valor acumulado
      value = (value * 10) + (incomingByte - '0'); // Resta 48 que es el valor decimal del 0
ASCII
    }
    else if (incomingByte == '>') // uso > como finalizador
    {
      velocity = value;   // Guarda el valor en la variable pwmValue
      velocity2 = velocity - velocity * 0.35;
```

```
      Serial.print("Velocidad=");
      Serial.println(velocity);     // Lo imprime por monitor serie
      value = 0;
    }
    else if (incomingByte == 'F') {   //Adelante
      velocidadMotores(velocity, velocity, velocity, velocity);
      sentidoMotores(FORWARD, FORWARD, FORWARD, FORWARD);
    }
    else if (incomingByte == 'D') {   //Adelante en diagonal hacia la derecha
     velocidadMotores(velocity, velocity2, velocity2, velocity);
      sentidoMotores(FORWARD, FORWARD, FORWARD, FORWARD);
    }
    else if (incomingByte == 'A') {   //Adelante en diagonal hacia la izquierda
      velocidadMotores(velocity2, velocity, velocity, velocity2);
      sentidoMotores(FORWARD, FORWARD, FORWARD, FORWARD);
    }
    else if (incomingByte == 'B') {   //Atrás
      velocidadMotores(velocity, velocity, velocity, velocity);
      sentidoMotores(BACKWARD, BACKWARD, BACKWARD, BACKWARD);
    }
    else if (incomingByte == 'I') {   //Atrás en diagonal hacia la izquierda
      velocidadMotores(velocity2, velocity, velocity, velocity2);
      sentidoMotores(BACKWARD, BACKWARD, BACKWARD, BACKWARD);
    }
    else if (incomingByte == 'P') {   //Atrás en diagonal hacia la derecha
      Serial.println("AtrasD");
      velocidadMotores(velocity, velocity2, velocity2, velocity);
      sentidoMotores(BACKWARD, BACKWARD, BACKWARD, BACKWARD);
    }
    else if (incomingByte == 'S') {   //Parado
      sentidoMotores(RELEASE, RELEASE, RELEASE, RELEASE);
    }
    else if (incomingByte == 'R') {   //Derecha(sobre su eje)
      velocidadMotores(velocity, velocity, velocity, velocity);
      sentidoMotores(FORWARD, BACKWARD, BACKWARD, FORWARD);
    }
    else if (incomingByte == 'L') {   //Izquierda(sobre su eje)
      velocidadMotores(velocity, velocity, velocity, velocity);
      sentidoMotores(BACKWARD, FORWARD, FORWARD, BACKWARD);
    }
  }
}
//Ahora van las funciones para acortar el código


void velocidadMotores(int vel1, int vel2, int vel3, int vel4) {
```

```
  //Se utiliza esta función para establecer la velocidad de los motores
myMotor->setSpeed(vel1);//Establece velocidad del motor
  myMotor2->setSpeed(vel2);
  myMotor3->setSpeed(vel3);
  myMotor4->setSpeed(vel4);
}

void sentidoMotores(int sent1, int sent2, int sent3, int sent4) {
  //Se utiliza esta función para establecer el sentido de giro de los motores
  myMotor->run(sent1);//Establece sentido de giro del motor
  myMotor2->run(sent2);
  myMotor3->run(sent3);
  myMotor4->run(sent4);
}
```

7.2 APP INVENTOR

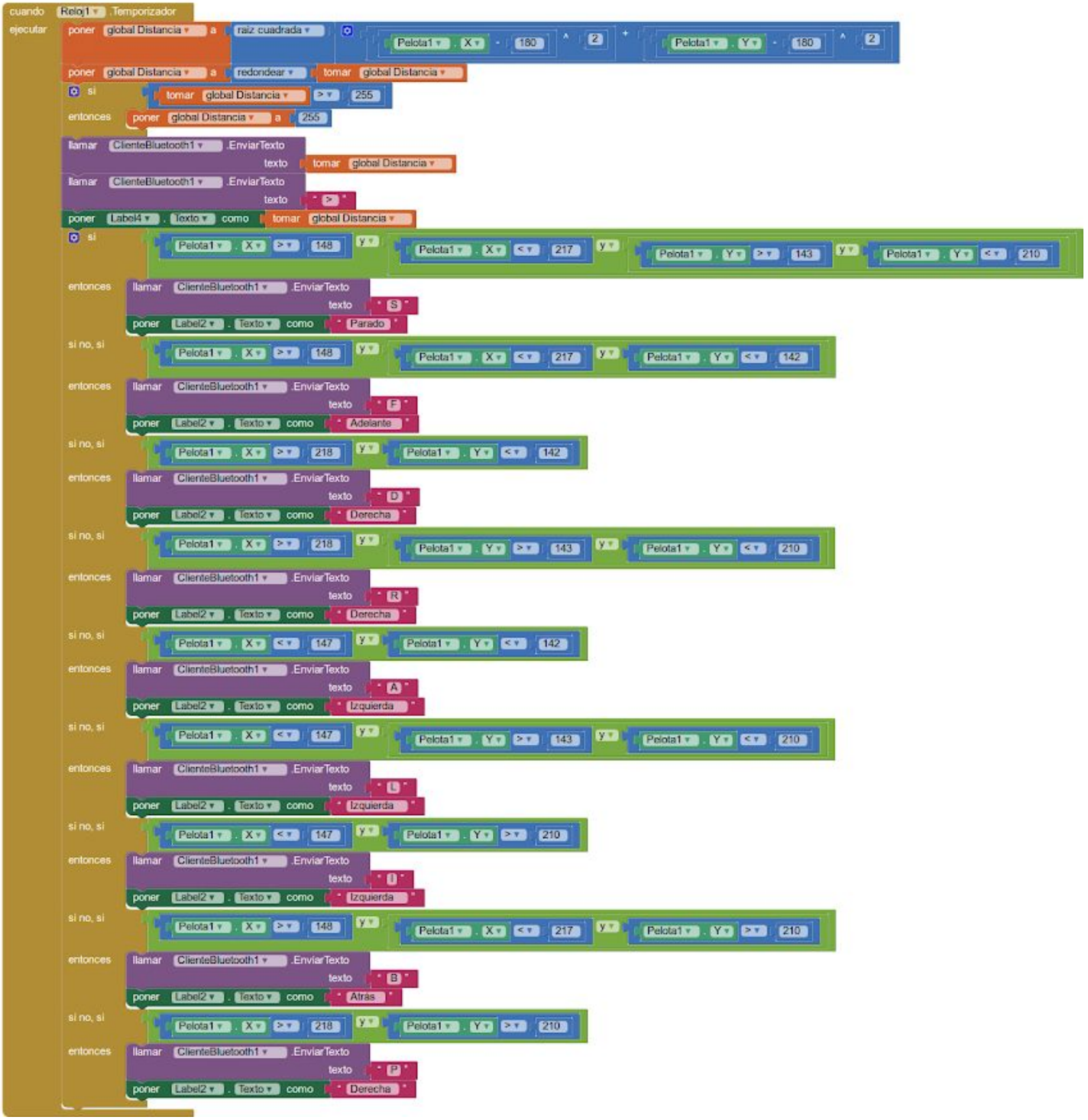
Conexión bluetooth y ubicación de objetos al inicializar



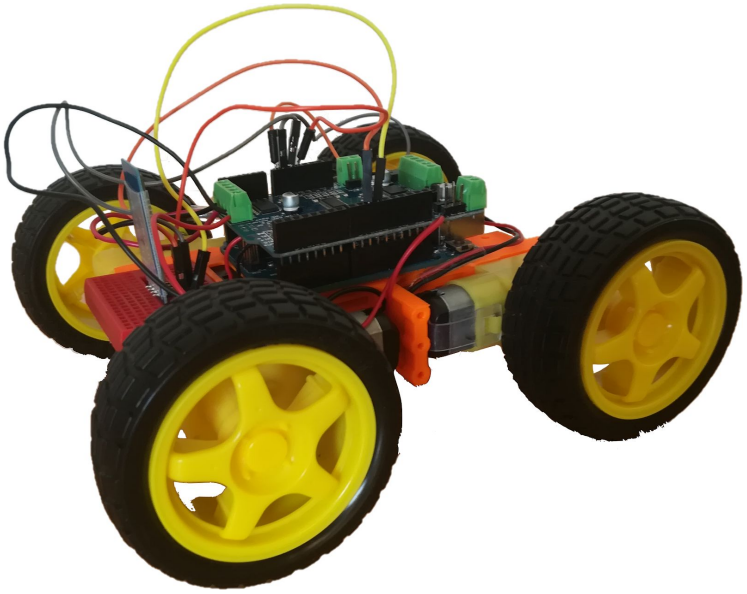
Variables

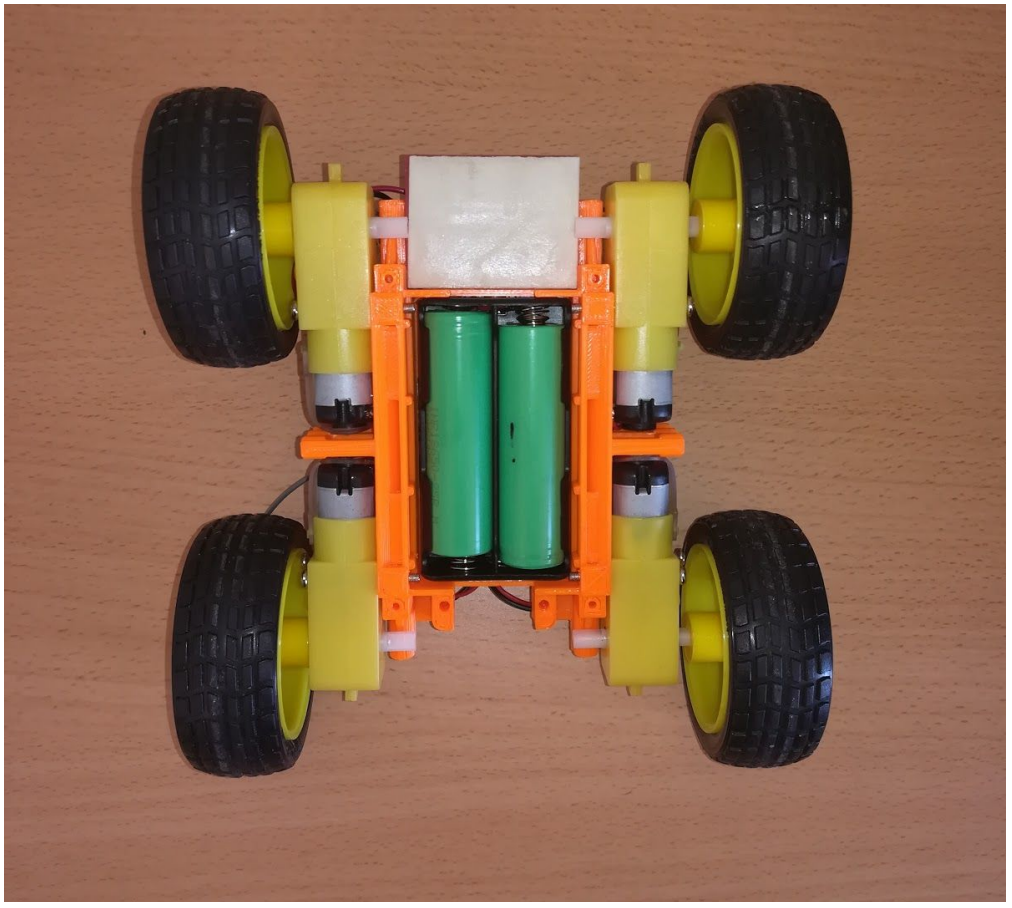
Posicionamiento de la bolita según el acelerómetro

Detección de zonas y Cálculo de la velocidad



8.Fotos





9.Análisis

- Análisis del funcionamiento del proyecto
    - El proyecto funciona perfectamente y sin ningún fallo.
    - Mejoras:
        - Sujeción móvil mejorar usabilidad
        - Calibración del acelerómetro que permita que la app funcione bien en cualquier teléfono
    - Limitaciones: cada teléfono da valores diferentes del acelerómetro por lo que sería necesario hacer ajustes en el código
- Ampliaciones del proyecto: Hacer un modelo de coche algo más grande y que utilice servomotores para la dirección.
- Problemas y soluciones: El problema más grande que hemos tenido es que no hay en internet ningún proyecto por lo que ha sido empezar desde cero con un gran trabajo de documentación.