


```
escornabot_bluetooth_02
#include <escornabot.h>

escornabot mirobot;
boolean buzz = false;

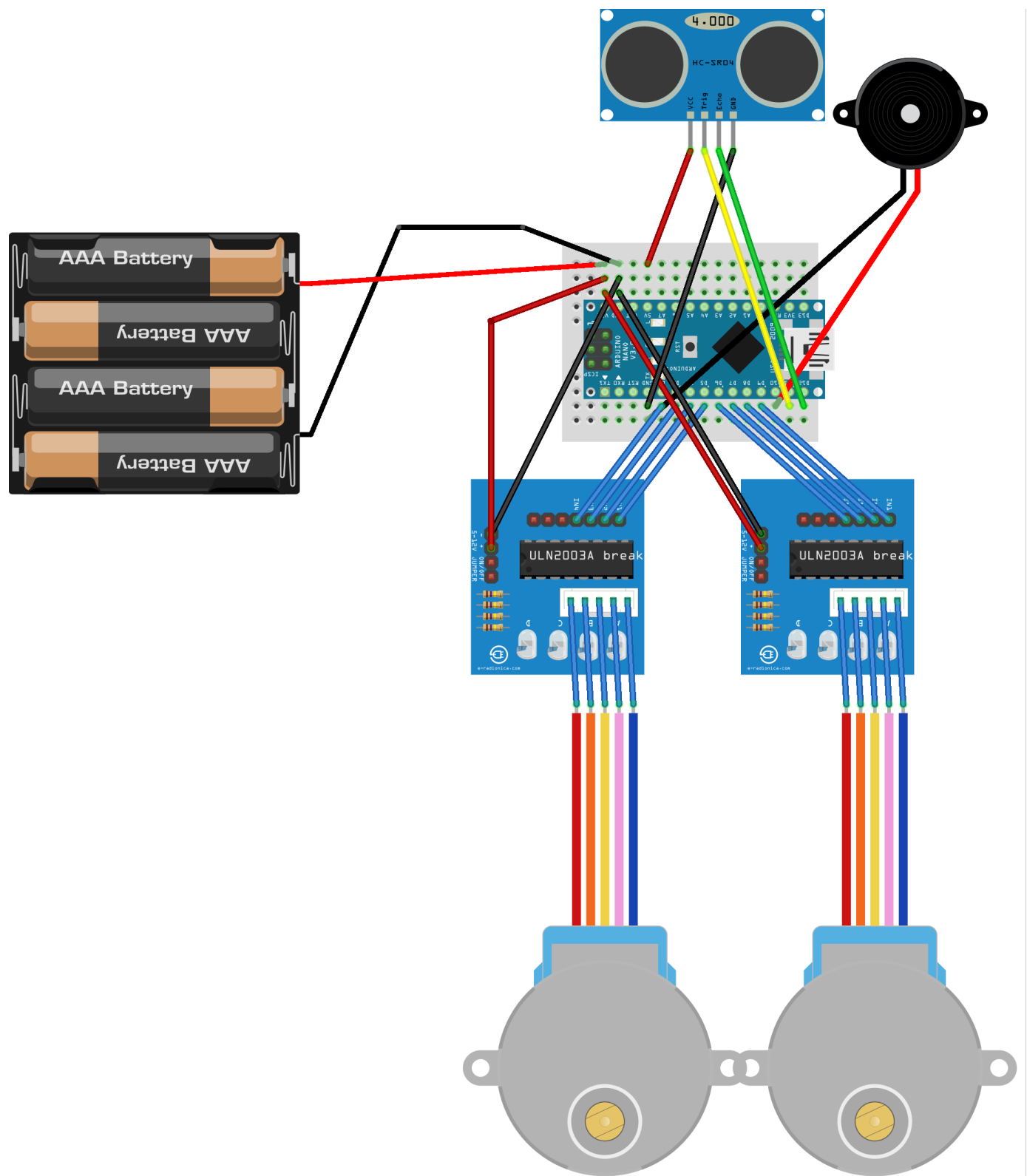
void setup() {
  Serial.begin (9600);
}

void loop() {
  //prueba de bluetooth de librería

  switch (mirobot.blueT()) { //en función del caracter emitido por bluetooth hace varias acciones
    case 'A':
      mirobot.driveD (-10, 15);
      break;
    case 'R':
      mirobot.driveD (10, 15);
      break;
    case 'D':
      mirobot.turnA (45, 15);
      break;
    case 'I':
      mirobot.turnA (-45, 15);
      break;
    case 'C':
      mirobot.turnA (360, 15);
      break;
    case '5':
      buzz = !buzz;
      if (buzz) {
        mirobot.buzzON();
      }
      else {
        mirobot.buzzOFF();
      }
      break;
    //default:
    // statements
  }
}
```

ESCORNABOT CON ULTRASONIDOS

Conexionado



fritzing

Código Evita-obstáculos

```
escornabot_ultrasonidos_evita
#include <escornabot.h>

escornabot mirobot;

const int echoPin = 12;
const int triggerPin = 11;
boolean funciona = false;

void setup() {
  //Serial.begin(9600);
  pinMode(triggerPin, OUTPUT);
  pinMode(echoPin, INPUT);
}

void loop() {

  compruebaBoton();

  if (funciona == true) {
    int cm = ping (triggerPin, echoPin);
    //Serial.print("Distancia: ");
    //Serial.println(cm);

    if (cm > 15) {
      mirobot.driveD(-5, 15);
    }
    else if (cm <= 15) {

      mirobot.driveD (5, 15);
      mirobot.turnA (-45, 15);
    }

  }

  else if (funciona == false) {
    mirobot.stop();
  }
}

void compruebaBoton () {
  if (mirobot.pushButton() == right) {
    funciona = !funciona;
    delay (300);
  }
}

int ping(int triggerPin, int echoPin) {
  long duration, distanceCm;

  digitalWrite(triggerPin, LOW); //para generar un pulso limpio ponemos a LOW 4us
  delayMicroseconds(4);
  digitalWrite(triggerPin, HIGH); //generamos Trigger (disparo) de 10us
  delayMicroseconds(10);
  digitalWrite(triggerPin, LOW);

  duration = pulseIn(echoPin, HIGH, 15000); //medimos el tiempo entre pulsos, en microsegundos

  distanceCm = duration * 10 / 292 / 2; //convertimos a distancia, en cm
  return distanceCm;
}
```

Código Sigue-Objeto

```
escornabot_ultrasonidos_siguel_buzz
#include <escornabot.h>

escornabot mirobot;

const int echoPin = 12;
const int triggerPin = 11;
boolean funciona = false;

void setup() {
  //Serial.begin(9600);
  pinMode(triggerPin, OUTPUT);
  pinMode(echoPin, INPUT);
}

void loop() {

  compruebaBoton();

  if (funciona == true) {
    int cm = ping (triggerPin, echoPin);
    //Serial.print("Distancia: ");
    //Serial.println(cm);

    if (cm <= 15) {
      mirobot.buzzOFF();
      mirobot.driveD(-5, 15);
    }
    else {
      mirobot.buzzON();
      mirobot.driveD(0, 0); //si no lo ponemos falla |
      mirobot.turnA (45, 15);
    }

  }

  else if (funciona == false) {
    mirobot.stop();
  }

}

void compruebaBoton () {
  if (mirobot.pushButton() == right) {
    funciona = !funciona;
    delay (300);
  }
}

int ping(int triggerPin, int echoPin) {
  long duration, distanceCm;

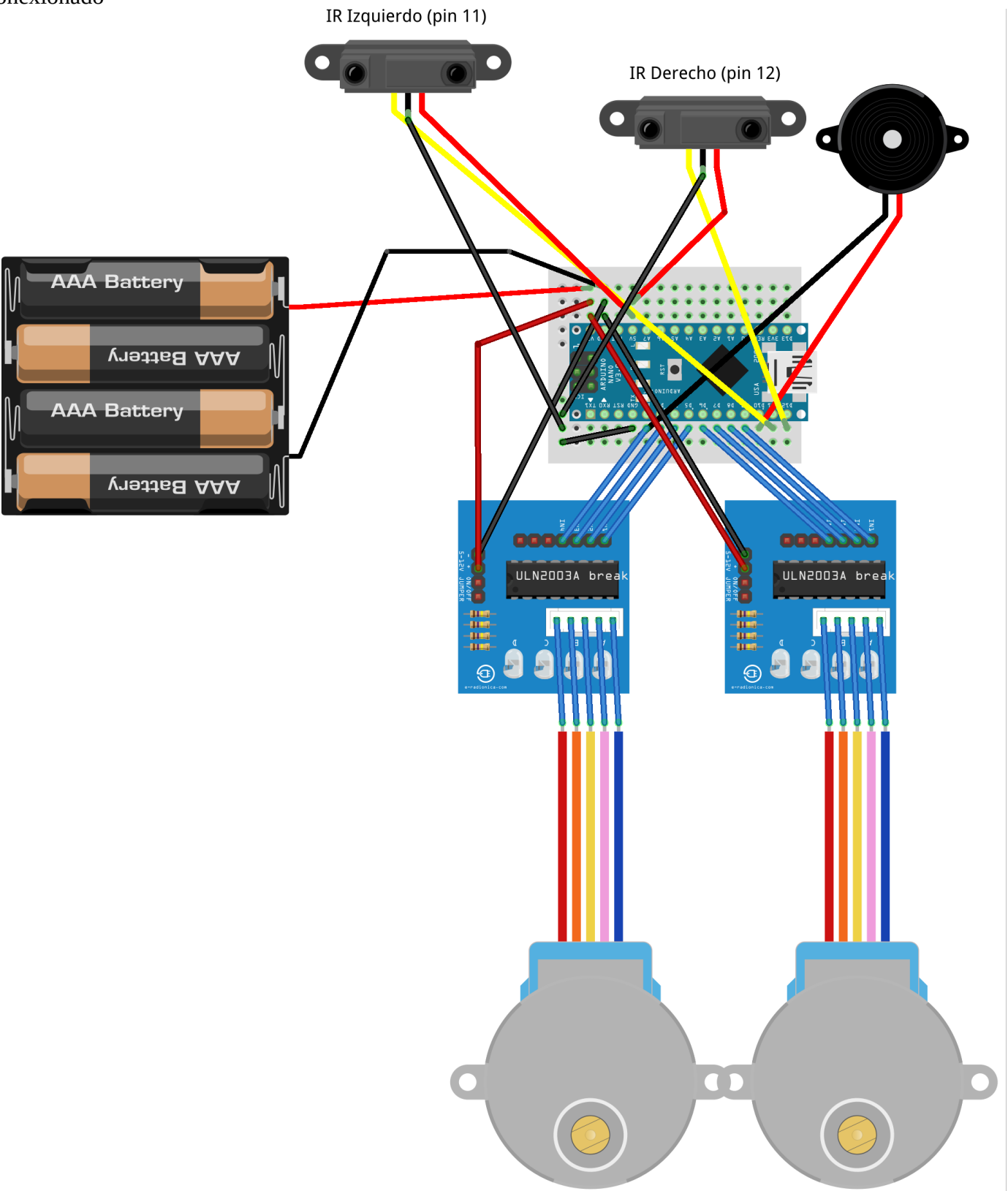
  digitalWrite(triggerPin, LOW); //para generar un pulso limpio ponemos a LOW 4us
  delayMicroseconds(4);
  digitalWrite(triggerPin, HIGH); //generamos Trigger (disparo) de 10us
  delayMicroseconds(10);
  digitalWrite(triggerPin, LOW);

  duration = pulseIn(echoPin, HIGH, 15000); //medimos el tiempo entre pulsos, en microsegundos

  distanceCm = duration * 10 / 292 / 2; //convertimos a distancia, en cm
  return distanceCm;
  //delay (100);
}
```

ESCORNABOT CON SENSORES DE INFRARROJOS

Conexionado



fritzing

Código Encerrado en recinto con líneas negras

```
escornabot_ir_encerrado
#include <escornabot.h>

escornabot mirobot;

const int ir_izquierdo = 11;//izquierdo en sentido de los sensores mirando hacia delante
const int ir_derecho = 12;//derecho en sentido de los sensores mirando hacia delante
boolean funciona = false;

void setup() {
  //Serial.begin(9600);
  pinMode(ir_derecho, INPUT);
  pinMode(ir_izquierdo, INPUT);
}

void loop() {

  compruebaBoton();

  if (funciona == true) {

    if (digitalRead (ir_derecho) && !digitalRead (ir_izquierdo)) {//si sólo sensor derecho encuentra línea negra
      mirobot.turnA(90, 10);//gira hacia izquierda en el sentido contrario a la marcha
    }
    if (!digitalRead (ir_derecho) && digitalRead (ir_izquierdo)) {//si sólo sensor izquierdo encuentra línea negra
      mirobot.turnA(-90, 10);//gira hacia derecha en el sentido contrario a la marcha
    }

    if (digitalRead (ir_derecho) && digitalRead (ir_izquierdo)) {//si los dos sensores encuentran línea negra
      mirobot.driveD(2, 13);//da marcha atrás en el sentido contrario a la marcha
    }
    if (!digitalRead (ir_derecho) && !digitalRead (ir_izquierdo)) {//si los dos sensores no encuentran línea negra
      mirobot.driveD(-2, 13);//va hacia delante en el sentido contrario a la marcha
    }

  }

  else if (funciona == false) {
    mirobot.stop();
  }

}

void compruebaBoton () {
  if (mirobot.pushButton() == right) {
    funciona = !funciona;
    delay (300);
  }
}
```

Código Siguelineas

escornabot_ir_siguelineas

```
#include <escornabot.h>
```

```
escornabot miescorni;
```

```
const int ir_izquierdo = 11;//izquierdo en sentido de los sensores mirando hacia delante
```

```
const int ir_derecho = 12;//derecho en sentido de los sensores mirando hacia delante
```

```
boolean funciona = false;
```

```
void setup() {
```

```
  //Serial.begin(9600);
```

```
  pinMode(ir_derecho, INPUT);
```

```
  pinMode(ir_izquierdo, INPUT);
```

```
}
```

```
void loop() {
```

```
  compruebaBoton();
```

```
  if (funciona == true) {
```

```
    if (digitalRead (ir_derecho) && !digitalRead (ir_izquierdo)) {//si sensor izquierdo encuentra blanco  
      miescorni.turnA(-5, 10);//gira hacia la derecha en el sentido contrario a la marcha
```

```
    }
```

```
    if (!digitalRead (ir_derecho) && digitalRead (ir_izquierdo)) {//si sensor derecho encuentra blanco  
      miescorni.turnA(5, 10);//gira hacia la izquierda en el sentido contrario a la marcha
```

```
    }
```

```
    if (digitalRead (ir_derecho) && digitalRead (ir_izquierdo)) {//si los dos sensores encuentran negro  
      miescorni.driveD(-2, 13);//se mueve hacia delante en el sentido contrario a la marcha
```

```
    }
```

```
    if (!digitalRead (ir_derecho) && !digitalRead (ir_izquierdo)) {//si los dos sensores encuentran blanco  
      miescorni.driveD(2, 13);//se mueve hacia detrás en el sentido contrario a la marcha
```

```
    }
```

```
  }
```

```
  else if (funciona == false) {
```

```
    miescorni.stop();
```

```
  }
```

```
}
```

```
void compruebaBoton () {
```

```
  if (miescorni.pushButton() == right) {
```

```
    funciona = !funciona;
```

```
    delay (300);
```

```
  }
```

```
}
```