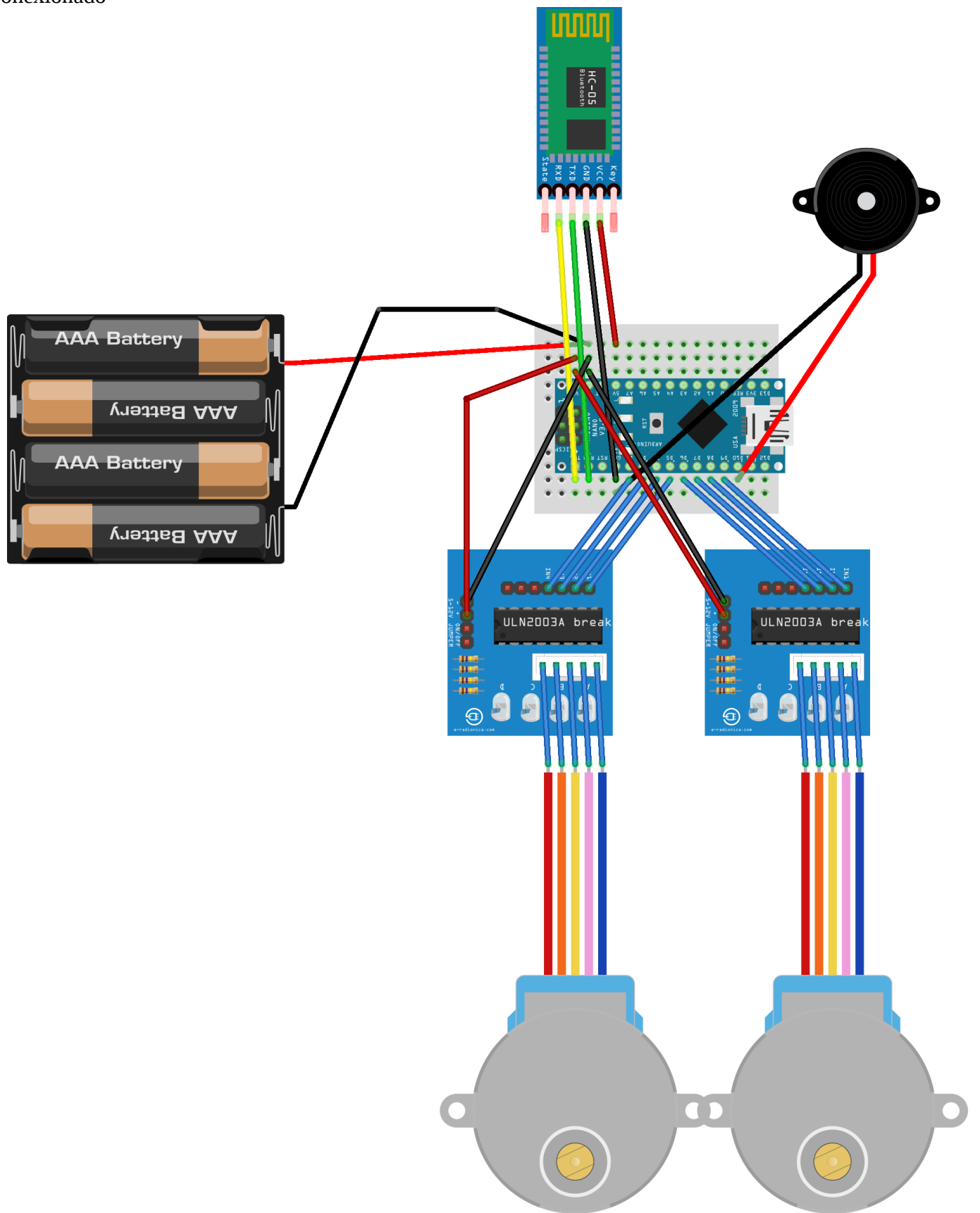


# ESCORNABOT CON LIBRERIA PARA ARDUINO Y SENSORES

## ESCORNABOT CONTROLADO CON BLUETOOTH

Conexionado



fritzing

```
escornabot_bluetooth_02
#include <escornabot.h>

escornabot mirobot;
boolean buzz = false;

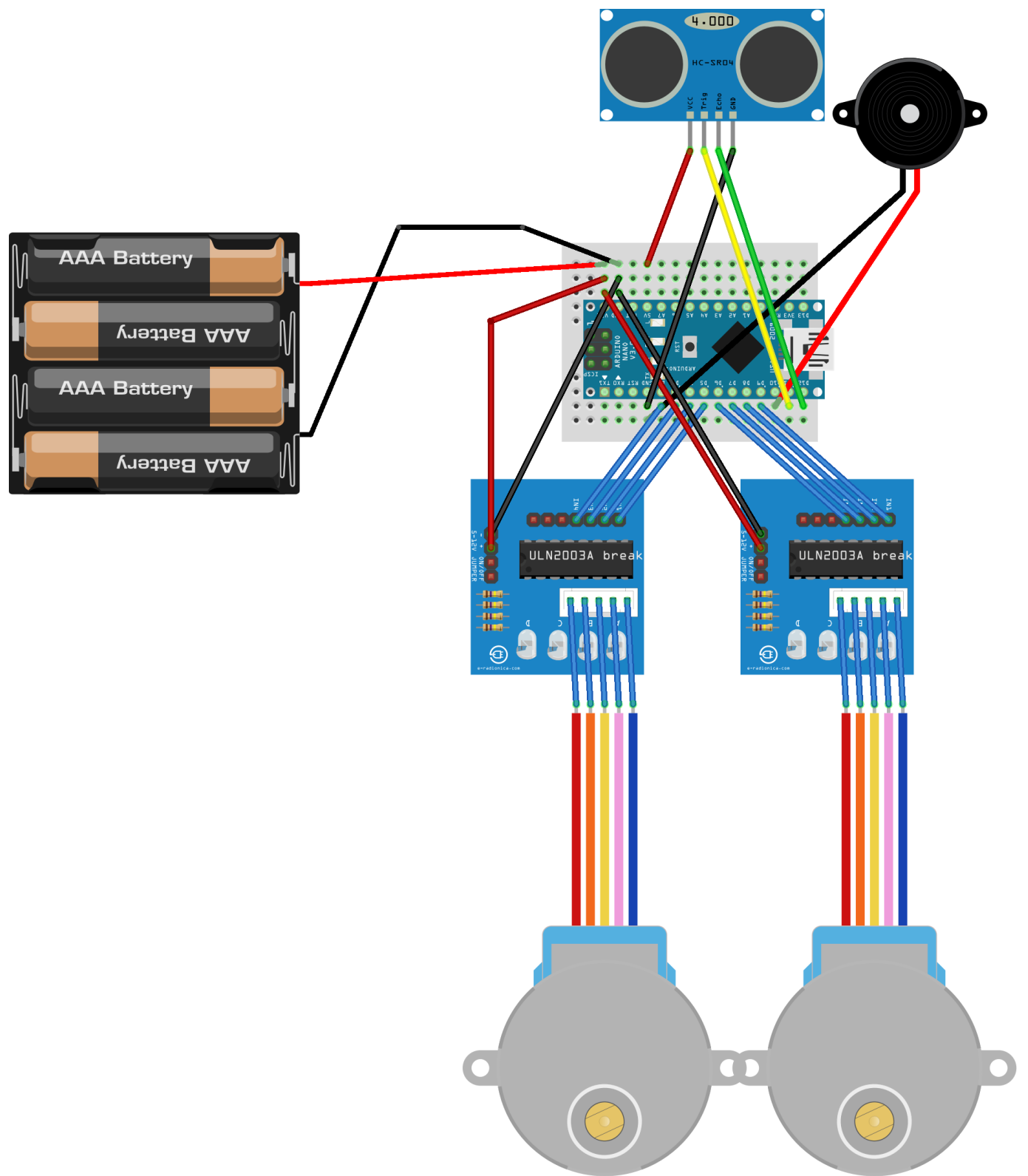
void setup() {
  Serial.begin (9600);
}

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);
}
```