

DRIVE ART

Sprint #1

26 Abril 2019

Stefany Ariana Chóez Bolaños 1459134

Daniel Panadero Espinosa 1458674

César Valcarce Pagán 1388811

Marina Riera Velasco 1457466

Contents

| | |
|--|----|
| Descripció del projecte | 2 |
| Componentes electrónicos | 2 |
| Esquema | 3 |
| Arquitectura Software | 4 |
| Otros Esquemas: sketch del robot | 5 |
| Contribuciones | 6 |
| Componentes adicionales y piezas 3D | 7 |
| Riesgos previstos y plan de contingencia | 8 |
| Código | 9 |
| Proyecto en el cual está inspirado | 10 |

Descripción del proyecto

Drive Art es un coche robot (con cierta autonomía) que permite dibujar con los gestos que hagamos con la cabeza (dibujo en físico). Drive Art dispone de dos sensores de ultrasonidos que permitirá que el coche evite obstáculos para no chocarse. Por otra parte, teniendo en cuenta que el dibujo que se realizará con los gestos puede no ser perfecto, ya que este proyecto se ha pensado para una posible utilidad para las personas con paraplejia, se realizará un filtrado para arreglar el dibujo que se haya hecho y se mostrará por pantalla el resultado final.

Componentes electrónicos

- Arduino nano
- HC-SR04 ultrasonic sensor
- Motor shield l298n
- Motores DC y ruedas
- Baterías AA recargables
- Base para baterías
- Módulo wifi
- Micro servo SG90
- Cables

Esquema Hardware

A continuación, se muestra el esquema de conexiones de nuestro robot.

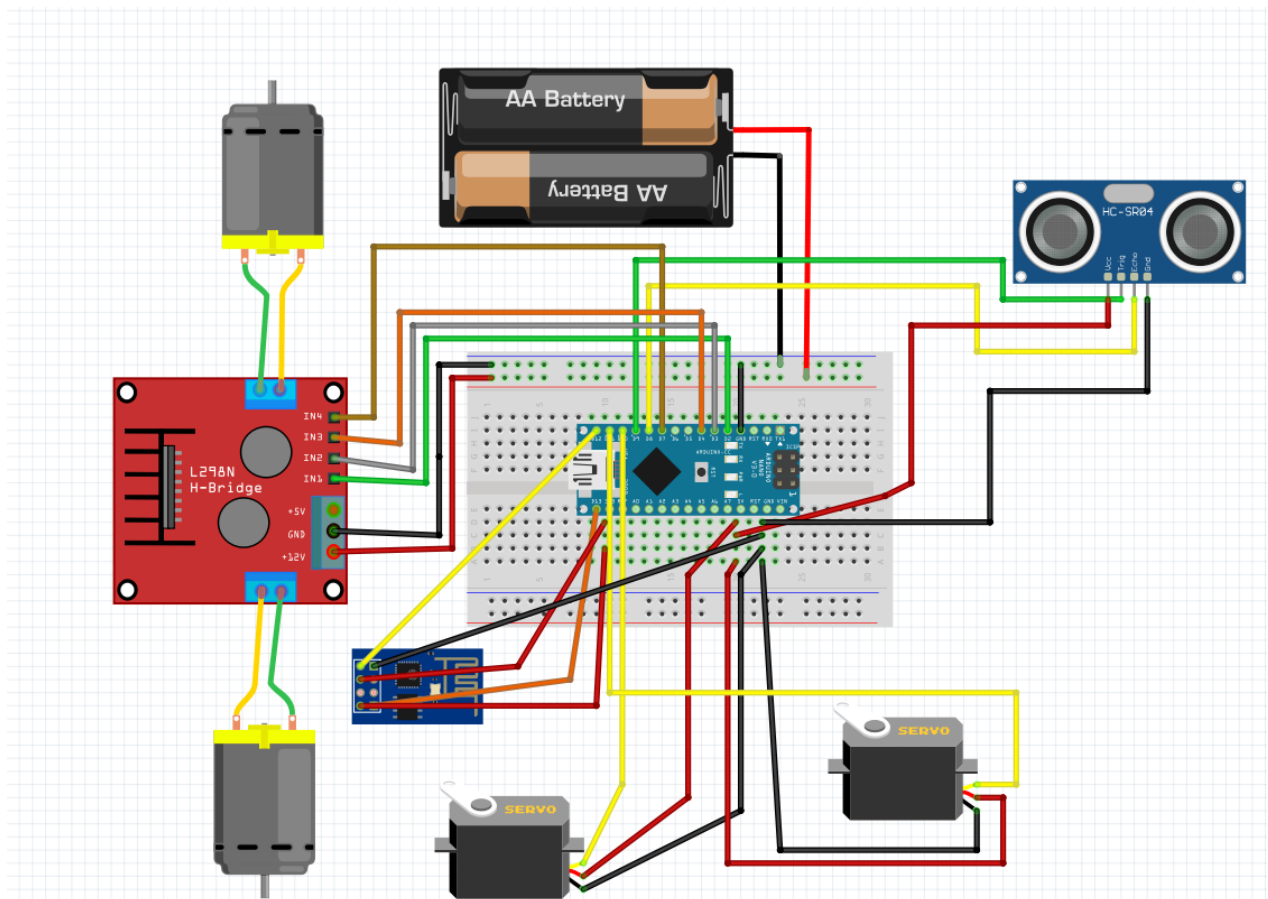


Figure 1: Esquema de connexiones de los componentes.

Arquitectura Software

A continuación, se muestran los distintos módulos software que usaremos y cómo se comunican entre ellos y con los distintos dispositivos.

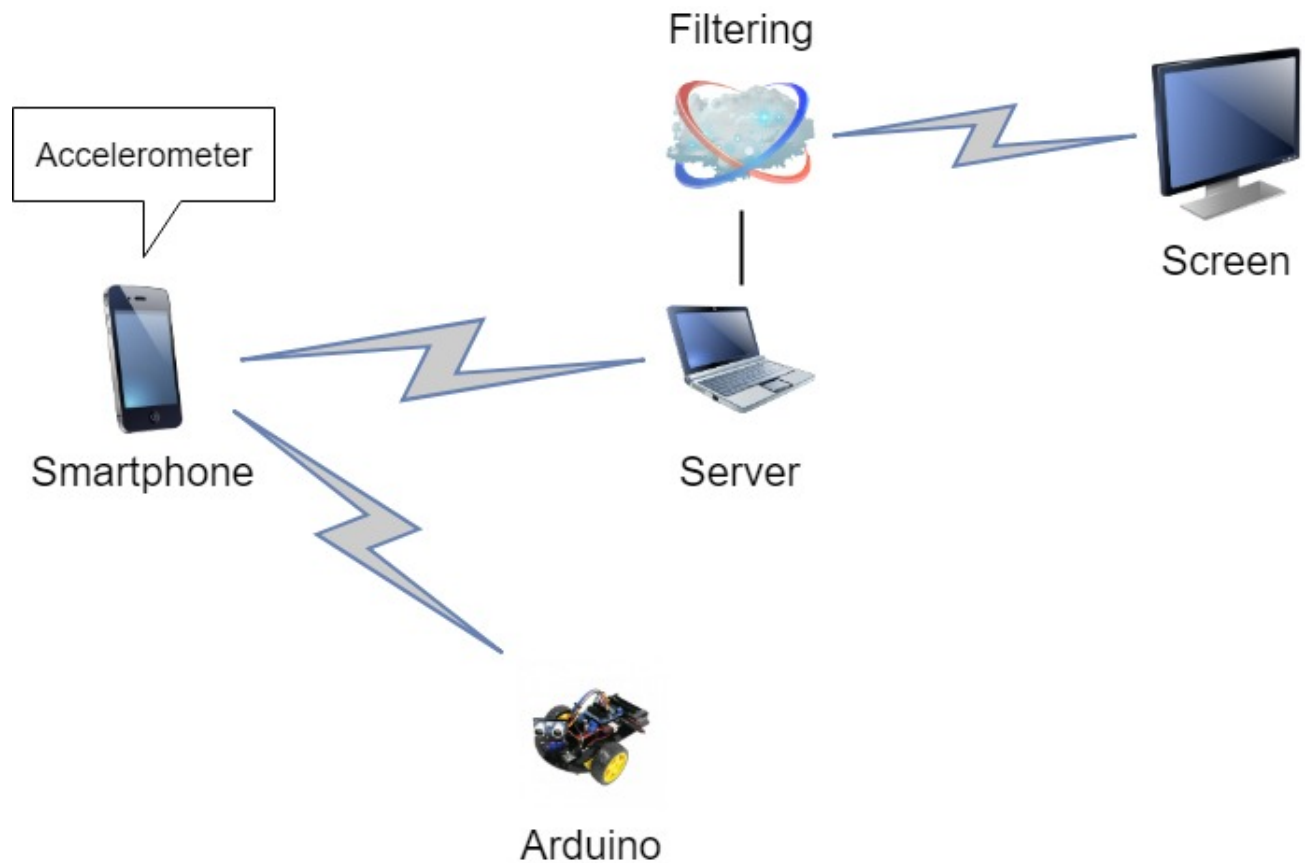


Figure 2: Esquema de módulos software.

Otros Esquemas: sketch del robot

A continuación, se muestra un sketch inicial del concepto de nuestro robot, con las medidas de los componentes.

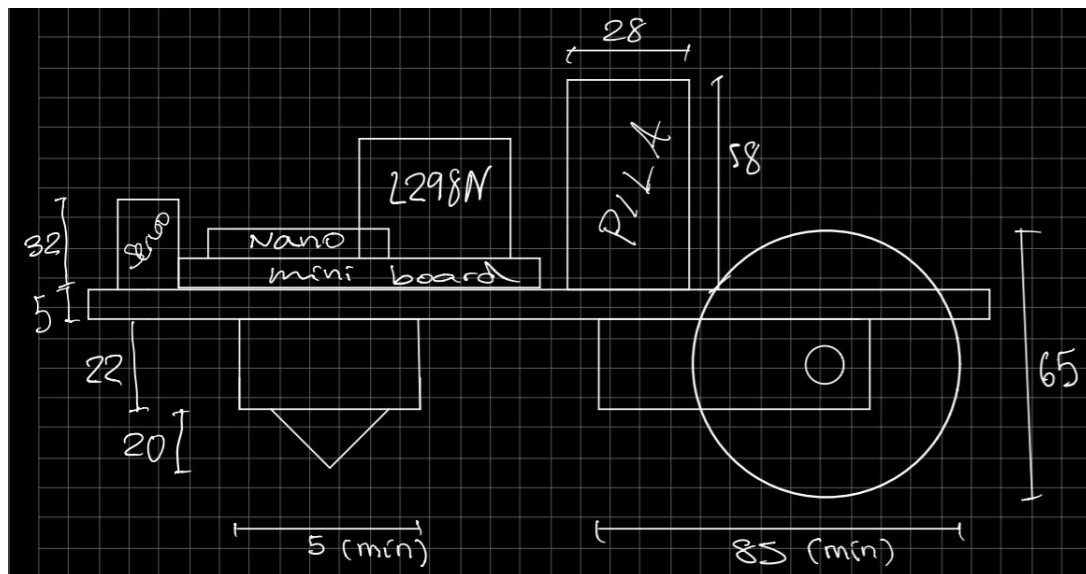


Figure 3: Sketch inicial del robot (cotas en milímetros).

Contribuciones

Como se ha comentado en la descripción del proyecto, DriveArt consiste en un coche robot que permita dibujar, y que se controla mediante los gestos de la cabeza. Está basado no en un sólo proyecto, sino en una combinación de varios (consultar sección referencias). De modo que nuestra principal contribución ha sido la de unificar todos estos proyectos en uno solo, y al mismo tiempo permitir que el robot dibuje en físico.

Por otra parte, otra contribución notable será la de aplicar un filtrado al dibujo resultante (similar al proyecto "Fishes Can Drive Too", de RLP 2018) y mostrarlo por pantalla. El filtrado puede consistir en eliminación de ruido (especialmente al ser para personas con paraplejía, que pueden realizar algunos movimientos involuntarios) y suavizar las líneas del dibujo.

Para el control del robot, se necesita comunicar el robot utilizando un módulo WIFI:

- Por una parte, con una aplicación móvil que le enviará al robot los datos de los gestos realizados con la cabeza, recogidos mediante un acelerómetro interno.
- Por otra parte, con un servidor que es el que recogerá el dibujo que realiza el robot, y que aplicará el filtrado descrito.

Finalmente, la nota a la cual aspiramos, cumpliendo con todos los objetivos y funcionalidades que nos hemos planteado, es el 10, porque consideramos que el proyecto es muy completo tanto a nivel hardware como a nivel software, y además se han propuesto varias contribuciones muy significativas, que añaden valor al proyecto, haciendo que destaque y se distinja de los que se pueden encontrar en internet.

También hemos relacionado el proyecto con otra asignatura de nuestra mención (APC, Aprendizaje Computacional), de la cual aprovecharemos algunos conocimientos para realizar algún filtraje tipo "smoothing" o de regresión (ya sea lineal o polinómica). Asimismo, también se aprovecharán conocimientos de otra asignatura fuera de la mención (Web) para desarrollar la aplicación móvil y configurar el servidor.

Extra components and 3D pieces

- Chasis del coche (impreso en 3D)

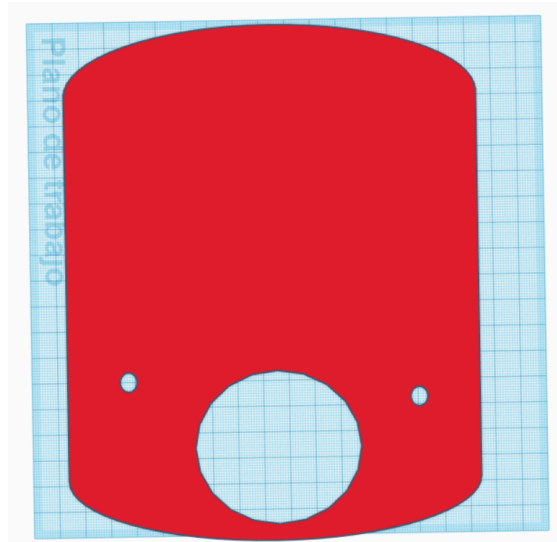


Figure 4: Modelo 3D del chasis del coche robot

- Estructura de engranajes para realizar la funcionalidad del cambio de color (tomados del laboratorio)
- Rueda loca (comprada)
- Mini breadboard para realizar todas las connexiones

Riesgos previstos y planes de contingencia

A continuación, analizamos los riesgos que se prevé que podrían ocurrir en el proyecto, recogidos en la siguiente tabla:

| Riesgo | Descripción | Probabilidad | Impacto | Plan contingencia |
|--------|--|--------------|---------|---|
| 1 | Desviación del coche provocada por diferente rendimiento entre los dos motores | Media | Alto | Habrà que calibrar los motores previamente, adaptando la velocidad de cada uno para que el coche se mueva en línea recta. |
| 2 | Cambio color no funcional (a causa de la transmisión de los engranajes o por falta de potencia del servo) | Baja | Medio | Simular el cambio de color sólo en pantalla y no en físico (rebajando ligeramente la ambición inicial). |
| 3 | En caso de que se necesite un pin adicional para RST del módulo WiFi, nos faltaria un pin | Media/Alta | Bajo | Eliminar algún componente no esencial, como el sensor de ultrasonidos secundario (el de detrás). |
| 4 | Falta de precisión al realizar el dibujo (ya sea por cómo se recogen los gestos con el acelerómetro, o por falta de precisión del robot) | Media | Bajo | Aplicar un filtrado más potente/agresivo para suavizar, aunque el dibujo no quedará tan bien. |
| 5 | No poder realizar el filtrado a tiempo real | Media/Alta | Bajo | El dibujo en físico es más visual, de modo que no supone un inconveniente si el filtrado se realiza una vez acabado el dibujo y se puestra por pantalla al final. |

Table 1: Tabla de riesgos y plan de contingencia.

Códigos prueba

En este apartado se mostrarán todos los códigos que se han utilizado para probar cada una de las partes del robot.

| Motores DC | Servo SG90 |
|---|---|
| <pre> int IN1 = 2; int IN2 = 3; int IN3 = 4; int IN4 = 5; int motor1_ena = 6; int motor2_ena = 7; void setup() { pinMode(LED_BUILTIN, OUTPUT); pinMode(motor1_ena , OUTPUT); pinMode(motor2_ena , OUTPUT); pinMode (IN1, OUTPUT); pinMode (IN2, OUTPUT); pinMode (IN4, OUTPUT); pinMode (IN3, OUTPUT); digitalWrite(IN1, LOW); digitalWrite(IN2, HIGH); digitalWrite(IN3, HIGH); digitalWrite(IN4, LOW); analogWrite(motor1_ena , 200); analogWrite(motor2_ena , 130); } </pre> | <pre> #include <Servo.h> int servoPin = 4; Servo servo; int servoAngle = 0; // servo position in degrees void setup() { Serial.begin(9600); servo.attach(servoPin); } void loop() { //control the servo's direction and the position of the servo.write(45); // Turn SG90 servo Left to 45 delay(1000); // Wait 1 second servo.write(90); // Turn SG90 servo back to 90 delay(1000); // Wait 1 second servo.write(135); // Turn SG90 servo Right to 135 delay(1000); // Wait 1 second servo.write(90); // Turn SG90 servo back to 90 delay(1000); //end control the servo's direction and the position of the //control the servo's speed //if you change the delay value (from example change it) for(servoAngle = 0; servoAngle < 180; servoAngle++) { servo.write(servoAngle); delay(50); } for(servoAngle = 180; servoAngle > 0; servoAngle--) { servo.write(servoAngle); delay(10); } //end control the servo's speed } </pre> |
| Sensores HC-SR04 | Módulo Wifi |
| <pre> int trigPin = 2; //Trig - green Jumper int echoPin = 3; //Echo - yellow Jumper long duration, cm, inches; void setup() { Serial.begin (9600); pinMode(trigPin, OUTPUT); pinMode(echoPin, INPUT); } void loop() { digitalWrite(trigPin, LOW); delayMicroseconds(5); digitalWrite(trigPin, HIGH); delayMicroseconds(10); digitalWrite(trigPin, LOW); pinMode(echoPin, INPUT); duration = pulseIn(echoPin, HIGH); cm = (duration/2) / 29.1; inches = (duration/2) / 74; Serial.print(cm); Serial.println(); delay(350); } </pre> | <pre> #include <SoftwareSerial.h> SoftwareSerial wifiSerial(2, 3); // RX, TX for ESP8266 bool DEBUG = true; //show more logs int responseTime = 10; //communication timeout void setup() { pinMode(LED_BUILTIN,OUTPUT); //set build in led as output // Open serial communications and wait for port to open esp8266: Serial.begin(115200); while (!Serial) { ; // wait for serial port to connect. Needed for Leonardo only } wifiSerial.begin(115200); while (!wifiSerial) { ; // wait for serial port to connect. Needed for Leonardo only } sendToWifi("AT+CWMODE=2",responseTime,DEBUG); // configure as access point sendToWifi("AT+CIFSR",responseTime,DEBUG); // get ip address sendToWifi("AT+CIPMUX=1",responseTime,DEBUG); // configure for multiple connections sendToWifi("AT+CIPSERVER=1,80",responseTime,DEBUG); // turn on server on port 80 sendToUno("Done!",responseTime,DEBUG); } </pre> |

Referencias

El proyecto ha estado inspirado, principalmente, como una combinación de los siguientes proyectos de internet:

1. Human Head to Robot Head: https://create.arduino.cc/projecthub/jegatheesan/human-head-to-robot-head-364bfd?ref=\search&ref_id=head%20control&offset=5
2. Obstacle Avoiding Robot:
<https://www.instructables.com/id/Arduino-Ultimate-Obstacle-Avoiding-Robot/>
3. CNC Drawing Arm: <https://www.instructables.com/id/CNC-Drawing-Arm/>
4. Fishes Can Drive Too: <https://rlpengineeringschooluab2018.wordpress.com/2018/05/29/fishes-can-drive-too/>