

SISTEMAS DIGITALES (3º GISIT)

PRÁCTICA 3: PROYECTO ARDUINO ROBOT OTTO



Javier Mateos Carrasco
jmateosaj@alumnos.unex.es
María Guadalupe Pañero
mguadalu@alumnos.unex.es
Curso 2020/21

ÍNDICE

- Motivación
- Técnicas y materiales utilizados
- Esquemas gráficos y código
- Resultado obtenido
- Futuros desarrollos posibles
- Conclusión
- Bibliografía

1. Motivación:

Queríamos hacer un proyecto original y divertido para esta parte de la asignatura. Después de barajar varias opciones, nos decidimos por recrear una versión del robot "Otto". Otra de nuestras alternativas era por ejemplo crear unas tiras de luces led que se iluminaban al ritmo de la música. A raíz de esto encontramos a Otto, un robot que baila. Y al ver un vídeo de como lo hacía, nos dimos cuenta de que este era el tipo de proyecto que queríamos hacer.

Vimos que Otto podía realizar muchas acciones, tales como: andar, bailar, emitir sonidos y evitar obstáculos. La que nos pareció más interesante fue el baile. Así que nos pusimos manos a la obra.

2. Técnicas y materiales utilizados:

Al ver los materiales que se necesitaban para construir a Otto, preguntamos al profesor Antonio Gordillo que si disponíamos de dichos materiales en el laboratorio de la asignatura:

- Diferentes piezas de la carcasa de Otto
- 4 x Servomotores
- Arduino ESP 8266 (WEMOS D1 mini pro)
- Cable USB
- Sensor de ultrasonidos
- Zumbador (Buzzer)
- Cables
- Compartimento para pilas
- Pilas 6V
- Interruptor ON/OFF
- Tornillos

El objetivo principal de esta práctica era iniciarse en la programación con Arduino Uno. Pero en nuestro caso, aunque la web oficial de Otto recomendaba utilizar la placa Nano ATmega328, nosotros hemos utilizado la ESP 8266 al no disponer de la otra en el laboratorio. Esta placa también es compatible para algunas funciones.

Una vez que conseguimos que funcionasen todos los materiales, nos dispusimos a realizar el montaje. En primer lugar, gracias a una impresora 3D, obtuvimos las diferentes partes de Otto: cabeza, cuerpo, piernas y pies. Las medidas de los agujeros donde iban los componentes no eran las correctas, así que con ayuda de un soldador y limas pudimos arreglarlo. Después atornillamos los servomotores a las piernas y los pies, y unimos estas partes al resto del cuerpo. Estos dispositivos son los que permiten el movimiento.

Seguidamente, colocamos dentro de la parte central del robot el buzzer, el interruptor y la caja de las pilas. Antes de nada, soldamos los cables que van unidos a estos componentes. El buzzer y el interruptor van colocados en unos agujeros que van en la carcasa del cuerpo. El primero sirve para emitir un sonido mientras Otto baila, mientras que el interruptor se utiliza para encender o apagar el circuito. Por otro lado, las pilas nos sirven para tener en funcionamiento al robot sin necesidad de tenerlo enchufado a la corriente.

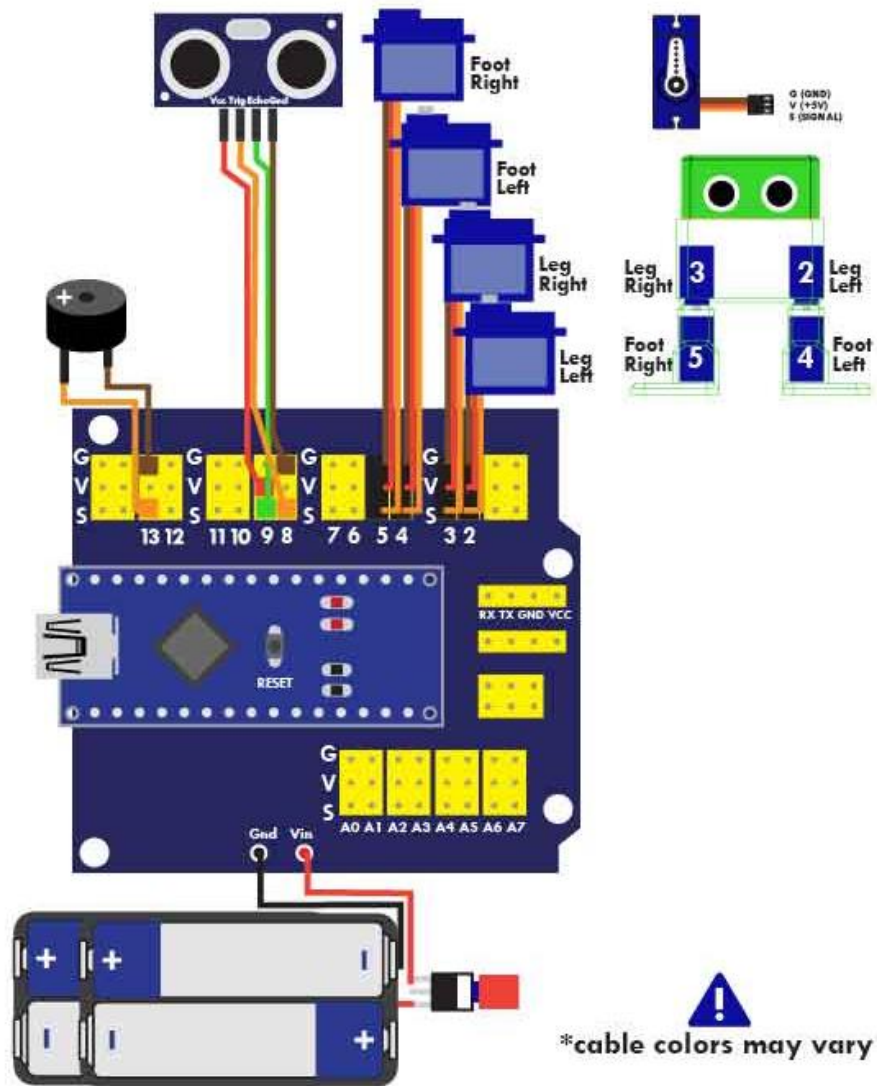
Una vez hecho esto, colocamos los pines necesarios en la placa WEMOS para poder enchufar los cables y los soldamos con estaño, 16 en total. Además, soldamos a otra placa distinta varios pines. En esta, colocamos por un lado todos los cables de los componentes que necesitaban corriente (6V) y por otro todos los que iban a tierra.

Por último, conectamos el sensor de ultrasonidos en la parte de la cabeza, que a su vez simula los ojos del robot. Este componente tiene cuatro cables, dos de ellos van a la placa principal, uno a tierra y otro a la placa donde están los cables que se alimentan con 6V.

Una vez conectados todos los cables, hicimos coincidir la salida del micro USB de la placa con el agujero de la carcasa. Y cerramos nuestro robot.

3. Esquemas gráficos y código:

Uno de los esquemas principales que hemos utilizado para guiarnos a la hora del montaje ha sido:



	Nano Atmega 328	WEMOS D1 mini pro
Right Leg	3	D2
Right Foot	2	D1
Left Leg	4	D3
Left Foot	5	D0
Buzzer	13	D6
Ultrasoner (Naranja)	8	D7
Ultrasoner (Amarillo)	9	D8

Aquí podemos ver como se conectan todos nuestros componentes, desde los servomotores hasta las pilas. Si nos fijamos la placa que se utiliza aquí no es la misma que la nuestra, por eso, tuvimos que buscar los pines equivalentes.

Tras hacer varias pruebas en el código y con ayuda de Antonio, nos dimos cuenta de que estos pines no se correspondían con los que poníamos en el código. Entonces mapeamos estos pines, quedando de la siguiente forma:

D0	16
D1	5
D2	4
D3	0
D6	12
D7	13
D8	15

Por lo tanto, el código que hemos utilizado es:

https://github.com/OttoDIY/OttoDIYESP/blob/master/examples/Otto_smooth_criminal/Otto_smooth_criminal.ino

Como es muy largo, hemos decidido insertar en enlace de la página donde lo hemos obtenido. Las modificaciones que realizamos son:

```
#include <Oscillator.h>
#include <Servo.h>
#include <Oscillator.h>
#include <EEPROM.h>

#define N_SERVOS 4
//-- First step: Configure the pins where the servos are attached
/*
    -----
    |  O  O  |
    |-----|
YR 3==> |      | <== YL 2
    -----
    ||  ||
    ||  ||
RR 5==> ----- <== RL 4
    |-----|
*/
#define EEPROM_TRIM false
// Activate to take callibration data from internal memory
#define TRIM_RR 7
#define TRIM_RL 4
#define TRIM_YR 4
#define TRIM_YL -7
//OTTO.setTrims(-7,-4,-4,7);

#define PIN_RR 16
#define PIN_RL 5
#define PIN_YR 0
```

```
#define PIN_YL 4
```

```
#define INTERVALTIME 10.0
```

```
Oscillator servo[N_SERVOS];
```

4. Resultado obtenido:

El resultado de aplicar este código a nuestra placa ha sido que Otto consiga reproducir el paso de baile “smooth_criminal”, basado en los pasos de baile de Michael Jackson y adaptado a nuestra versión del robot.

5. Futuros desarrollos posibles:

Varias mejoras que podríamos incluir a nuestro proyecto serían: hacer que Otto evitase obstáculos, realizase otro tipo de baile o incluso poder controlarlo mediante una aplicación vía bluetooth. Serían opciones muy interesantes que podríamos implementar en un futuro.

6. Conclusión:

En definitiva, hemos disfrutado realizando este proyecto, ya que hemos llevado a cabo diferentes actividades a lo largo del proceso que nos han hecho aprender cosas que no sabíamos antes. Desde comprender los fundamentos de arduino hasta soldar componentes a una placa. Gracias a esto, nos hemos dado cuenta de que nos hubiese gustado que la asignatura tratara más a fondo estos temas, ya que desde la parte práctica, en nuestra opinión, es la mejor forma de conocer la asignatura.

7. Bibliografía:

<https://create.arduino.cc/projecthub/cparrapa/otto-diy-build-your-own-robot-in-one-hour-5f2a1c>

<https://github.com/OttoDIY/OttoDIYESP>