

Práctica de uso de Arduino

El objetivo de esta práctica es recordar todos los conceptos vistos durante el curso y aplicarlo a un caso práctico donde la implementación es compleja.

Antes de empezar y aunque sea una práctica en grupo, se pide a todos los integrantes rellenar el siguiente [formulario](#).

La tarea está dirigida a iniciar al estudiante en el microcontrolador Arduino y consistirá en que se programe un coche robot para que siga la línea en un circuito (Simulador Oviedo).

El simulador Oviedo

Un arduino es un microcontrolador, es decir, un circuito integrado programable que incluye las tres partes funcionales de un ordenador: unidad central de procesamiento, memoria y dispositivos de entrada y salida. Los dispositivos de entrada y salida se les suele denominar **Pin**.

Tipo de Arduino	Procesador	# I/O	# entradas ADC	FLASH/RAM KB	Mhz	Voltaje (VDC)
Arduino UNO R3	ATmega328p	14	6	32 / 2	16	5
Arduino MEGA R3	ATmega2560	54	16	256 / 8	16	5
Arduino NANO	ATmega328p	14	6	32 / 2	16	5/3.3
Arduino Leonardo	ATmega32u4	14	6	28 / 2.5	16	5
Arduino DUE	AT91SAM3X8E	54	12	512 / 96	84	3.3

Figura 1: Principales modelos de arduino con sus características: tomada de hetpro-store.com

Los pines de un microcontrolador Arduino son de dos tipos:

- Digitales: La lectura o escritura una variación de voltaje entre dos valores sin tener en cuenta los intermedios. Por lo tanto, una señal digital dispone solo de dos estados. Al valor inferior de tensión -Vcc le asociamos un valor lógico LOW o '0', mientras que al valor superior +Vcc le asociamos HIGH o '1' lógico.
- Analógicos: En este caso, se puede tomar cualquier magnitud entre los valores -Vcc y +Vcc.

En esta práctica, tenemos un microcontrolador Arduino que, entre otros elementos electrónicos, consta de:

- Motores, que son los que se encargan de mover las ruedas
- Sensores, que se encargan de detectar el color negro en el suelo.

Durante la práctica, utilizaremos los dos tipos de pines. Los analógicos serán utilizados para mover los motores y los digitales se utilizarán para las lecturas de los sensores.

Empecemos por los motores, la velocidad de estos es un valor entre 0 y 180, siendo

- 90 la posición de estar parado
- 180 es velocidad máxima hacia delante en el motor derecho y hacia atrás en el izquierdo

- 0 es velocidad máxima hacia delante en el motor izquierdo y hacia atrás en el derecho

En el siguiente cuadro de código está comentado el uso y funcionamiento de los motores.

```
#include <Servo.h>
Servo servoIzq; // Rueda izquierda
Servo servoDer; // Rueda derecha

int pinServoDer = 9;
int pinServoIzq = 8;
void setup(){

    servoIzq.attach(pinServoIzq); // Escribir y leer la velocidad del motor izq
    servoDer.attach(pinServoDer); // Escribir y leer la velocidad del motor dcha
}

void stopMotor(){
    servoIzq.write(90);
    servoDer.write(90);
}

void forwardMotor(){
    servoIzq.write(0);
    servoDer.write(180);
}
```

- 1) Explique con palabras como se hace girar el coche a la izquierda y a la derecha, según el código del fichero *siguelineas.ino*. Diga como se hace para leer los pines de digitales de los sensores, explicando que valores son los que se utilizan para detectar línea en el suelo

Para la programación de un dispositivo Arduino, se utiliza el lenguaje de programación C, con algunas modificaciones y librerías que se explican por el funcionamiento del microcontrolador.

La primera modificación es la presencia de dos métodos **setup** y **loop**, y la no presencia del método **main**. El funcionamiento de un Arduino es muy parecido al de un *autómata finito* (que se vio en la asignatura de Lenguajes Formales) y por ello tiene una parte de configuración (**setup**) y la siguiente es la parte donde se repite un mismo procedimiento, a partir de las entradas (**loop**). Se podría pensar que este bucle se repite de forma infinita.

Para la programación del coche Arduino se pide utilizar el simulador incluido.

Las instrucciones de uso son las siguientes:

- Descomprime y ejecuta el simulador mediante *main.exe*
- Selecciona la opción “Robot Móvil (4 infrarrojos) y Circuito
- Selecciona *Archivo->Importar sketch* para cargar el archivo *siguelineas.ino*.

Se puede mover el coche Arduino con las teclas de movimiento “AWS D”. Para ejecutar el archivo, debes desmarcar la acción *Movimiento con el teclado*, pulsar *stop* y *play* y se iniciará la ejecución.

- 2) Quite el comentario y complete el código *siguelineas.ino*, de forma que el coche Arduino siga todo el circuito de manera autónoma.

Un problema del programa de *siguelineas.ino* es que si no empieza en una línea, no hay forma **determinista** de encontrar el camino de vuelta a la línea, ya que no se conoce la posición.

- 3) Programe el generador de números pseudoaleatorio «Linear Congruential Generator», busque su definición en Wikipedia y utilice la siguiente estrategia probabilista para encontrar la línea: generar un bit, si el bit es uno girar 90 grados a la derecha y avanzar 400 ms, si es cero girar 90 grados a la izquierda y avanzar. Seguir así hasta encontrar la línea.