

Introducción a Arduino



https://commons.wikimedia.org/wiki/File:Arduino_Uno_-_R3.jpg

Práctica 1 – Actividades (v1.3.6.1 septiembre 2022)

Software para robots

Cristian González García

gonzalezcristian@uniovi.es

Índice

Actividades principales	2
(Dado1.1) Dado electrónico (0,15 puntos)	2
(Memoria1.2) Juego de memoria (0,3 puntos)	2
Ampliaciones	3
(Zumbador1.3) Ampliar Simón con el Zumbador y LED RGB (0,25 puntos)	3
(Semáforo1.4) Simular el cruce de una calle con semáforos (0,3 puntos)	4
(Semáforo1.5) Simular el cruce de una calle con semáforos y pasos para peatones (0,25 puntos)	4
(Discoteca1.6) Sistema para DJs (0,15)	4
(Regulador1.7) Control y regulador de luz (0,15)	5
(Regulador1.8) Control de luces (0,15)	5
(Regulador1.9) Control y regulador de luz II (0,15)	6

El total de las actividades principales tienen un valor de 0,4 puntos y 1,4 puntos en ampliaciones dentro del bloque 1.

Actividades principales

(Dado1.1) Dado electrónico (0,15 puntos)

Material necesario: 3 LEDs rojos y 1 pulsador.

Habrás que simular un dado electrónicamente. Para ello, se utilizarán 3 LEDs y un pulsador. Inicialmente, todos los LEDs estarán apagados. Cuando se presione el pulsador, el sistema generará **un número aleatorio entre 1 y 3**. El número obtenido se mostrará al usuario encendiendo los LEDs. Los LEDs se deben situarse en línea, y deberán encenderse siempre de forma ordenada: primero los encendidos y luego los apagados, es decir, no mezclados.

- Número 2: Encendido, Encendido, Apagado -> Correcto
- Número 2: Encendido, Apagado, Encendido -> Incorrecto, ejercicio mal

Una vez mostrado el número, los LEDs correspondientes deben quedar encendidos hasta que se vuelva a pulsar el botón.

NOTA: el número aleatorio generado tiene que ser de verdad un aleatorio, para mirar cómo se hace, mirar la parte de teoría. En caso de no ser un aleatorio y demostrarlo en el video inicializando el ejercicio varias veces, el ejercicio no puntuará.

NOTA 2: Hay que usar un pulsador, no un sensor de final de carro.

Vídeo:

- Debe mostrar el reinicio de la placa 2 veces para comprobar que sea una secuencia totalmente aleatoria.
- También debe mostrarse la secuencia impresa por pantalla para comprobar que si se presionó el botón n veces, solamente entró en dicha función n veces, es decir, una pulsación del botón se corresponderá con que 1 llamada. Como solucionar esto está explicado en el otro PDF.

(Memoria1.2) Juego de memoria (0,3 puntos)

Material necesario: 1 LED rojo, 1 LED verde y 2 pulsadores.

Se requerirá tenerlo implementado para el opcional 1.3

En este ejercicio se creará un juego como el «Simón». Como máximo deberá tener 3 turnos, es decir, la secuencia más larga será de 5 colores. Sin embargo, esto tiene que ser modificable fácilmente a nivel de código (una variable), y mostrarlo en el vídeo.

Cuando iniciamos la aplicación se debe generar una secuencia **aleatoria** (de verdad y demostrada en el video). Esta secuencia comenzará con 3 opciones, que serán en función de la secuencia aleatoria que se obtuvo. Se recomienda utilizar una estructura de datos, como puede

ser array, para almacenar las secuencias. Por ejemplo, esta secuencia podría ser: Verde-Rojo-Verde, Verde-Verde-Verde, Rojo-Rojo-Rojo, Rojo-Verde-Verde, etc. Entre cada parpadeo se debe esperar un segundo. Tras esto, hay que esperar un par de segundos y comenzará el usuario a introducirla.

En este momento es cuando el usuario debe reproducir esa secuencia utilizando los pulsadores. Un pulsador se corresponderá al rojo y el otro al verde. Es decir, para «acertar» la secuencia Verde, Rojo, Verde el usuario deberá pulsar el pulsador asociado al verde una vez, después el pulsador asociado al rojo y tras esto el pulsador asociado al verde una vez. **Cuando se pulse un pulsador, el LED asociado a ese pulsador se deberá iluminar.**

Si el usuario realiza la acción correctamente, el juego sube de nivel. El incremento de nivel implica añadir un nuevo color generado aleatoriamente a la secuencia previa. Cada vez que el usuario acierta la combinación se vuelve a repetir la secuencia anterior añadiendo un parpadeo adicional para así ir aumentando la dificultad. **¡Ojo! Se añade un nuevo color a la secuencia previa, no hay que generar una nueva secuencia entera.** Cuando llegué al final y el usuario obtenga la victoria, se mostrará un efecto de luz especial que indicará la victoria del usuario.

Si el usuario se equivoca y realiza una pulsación incorrecta, el juego finalizará, se mostrará un efecto de luz que indicará el fin de la partida (diferente al de la victoria) y se volverá a iniciar desde el principio. Se debe volver a **generar una secuencia aleatoria inicial nueva.**

Se debe dividir la funcionalidad del sistema en funciones. Si toda la lógica se encuentra en el método «loop», se restará la mitad de puntos, aunque el ejercicio esté bien.

¡Cuidado! Los pulsadores necesitan de una resistencia, así que hay que añadirla, pues, en el caso de que quedaran conectadas directamente las corrientes positivas y negativa se haría un cortocircuito y se podría quemar la placa.

Atención: cuando se grabe el video, hay que mostrar **tres secuencias/rondas, en dos partidas**, una en la que funcione bien y otra partida en la que se falle en la segunda secuencia, para ver cómo se reinicia el juego. Habrá que mostrar varios inicios del primer turno para **enseñar que el aleatorio se hizo correctamente.** Si no, se descontarán puntos.

Ampliaciones

(Zumbador1.3) Ampliar Simón con el Zumbador y LED RGB (0,25 puntos)

Material necesario: 1 LED RGB, 2 pulsadores, y 1 zumbador.

Requiere tener implementado el ejercicio obligatorio 1.2

Hay que ampliar el ejercicio anterior y agregar un sonido característico a cada uno de los LEDs utilizando el zumbador y sustituir los LEDs por un LED RGB. **En lugar de utilizar 2 LEDs normales habrá que utilizar un LED RGB.**

El **sonido** debe reproducirse cuando los LEDs parpadeen o cuando el usuario pulse el pulsador asociado a cada LED.

Cuando la partida finalice, y sea debido a que el usuario ha perdido hay que mostrar el color azul durante unos segundos. En caso de haber ganado, deberá de haber una «explosión» de colores.

(Semáforo1.4) Simular el cruce de una calle con semáforos (0,3 puntos)

Material necesario: 2 LEDs rojos, 2 LEDs verde y 2 LEDs amarillos.

Se requerirá tenerlo implementado para el opcional 1.4

En este ejercicio hay que implementar un cruce con **dos semáforos**. Para ello, cuando un semáforo esté en verde, el otro estará en rojo. Después de un tiempo, el semáforo que está en verde tendrá que pasar a amarillo, y, tras unos segundos, a rojo. Tras una breve pausa para esperar a que los coches terminen de cruzar, el otro deberá de ponerse en verde y repetir el mismo proceso.

En el video deberá verse el proceso un par de veces, pues deberá de ser un bucle infinito.

(Semáforo1.5) Simular el cruce de una calle con semáforos y pasos para peatones (0,25 puntos)

Material necesario: 2 LEDs RGB, 1 LEDs rojo, 1 LEDs verde y 1 zumbador.

Requiere tener implementado el ejercicio opcional 1.4

Este ejercicio es una ampliación sobre el ejercicio anterior, cambiando algún componente y añadiendo un paso para peatones.

Habrà que **cambiar los LEDs de los semáforos a LEDs RGB. Seguirà habiendo dos semáforos, pero solo uno tendrá paso para peatones.**

Para el paso de peatones, habrá que utilizar el LED verde, el LED rojo, y el zumbador. El LED rojo estará encendido cuando el semáforo para coches correspondiente esté en verde. Cuando este esté en rojo, y después de un tiempo, para evitar atropellos, se encenderá el LED verde para peatones y el zumbador hará pitidos para que las personas ciegas sepan cuando cruzar. Cuando quede poco tiempo, el LED verde deberá de parpadear y los sonidos ir un poco más rápido. Una vez se ponga en rojo y tras unos segundos, se abrirá el tráfico a los coches en la vía correspondiente.

En el video deberá verse el proceso un par de veces, pues deberá ser un bucle infinito.

(Discoteca1.6) Sistema para DJs (0,15)

Material necesario: 2 LEDs RGB y 1 Zumbador.

Se tratará de hacer un sistema para que el DJ pueda cambiar las tonalidades de las luces y el volumen a su gusto. Para ello, se usarán dos potenciómetros.

Un potenciómetro controlará las luces, en nuestro caso, son LEDs RGB. Cuando esté en su valor mínimo, un LED RGB estará apagado y el otro estará iluminando con todos los colores a

tope. Según se vaya creciendo el valor del potenciómetro, el LED RGB apagado irá encendiéndose, haciendo una escala de colores (arcoíris y de forma libre y comenzando en 000 y terminando los colores en 255) y el segundo LED hará justamente lo contrario. En el valor central del potenciómetro ambos LEDs RGB deberán de tener la misma tonalidad.

El segundo potenciómetro controlará las notas musicales. El valor mínimo no hará ruido y el máximo hará una de las notas más altas en la escala musical.

Como se necesitarán 7 salidas digitales PWM y en el Arduino UNO solo hay 6, hay tres posibilidades para hacer este ejercicio:

- Pedir prestada la placa Arduino Mega.
- Usar un transistor.
- «Trampearlo» usando una salida para dos LEDs.

(Regulador1.7) Control y regulador de luz (0,15)

Material necesario: 3 LEDs rojos y 1 potenciómetro.

Conectar 3 LEDs y un potenciómetro. En función del valor de entrada del potenciómetro se tendrán que encender 0, 1, 2 o los 3 LEDs de forma secuencial, empezando por el 0 y siguiendo el orden. En este caso, todos los LEDs deben encenderse siempre con la misma intensidad, es decir, estarán apagados o encendidos.

La secuencia de encendido es: se encienda primero el A, después el B, después el C, después A y B, después B y C, después A y C, y cuando esté en el máximo valor los 3.

(Regulador1.8) Control de luces (0,15)

Material necesario: 3 LEDs rojos y 1 potenciómetro.

Conectar 3 LEDs y un potenciómetro. En función del valor de entrada del potenciómetro se tendrán que encender 0, 1, 2 o los 3 LEDs. No obstante, en este caso, cuando más cercano esté del límite inferior, menos luminosidad tendrán los LED y cuantos más cercanos al límite superior tendrán más luminosidad. Utilizar las salidas PWM para encenderlos.

Ejemplo de entradas y salidas:

Potenciómetro	LED 1	LED 2	LED 3
0	0	0	0
1	1	0	0
32	160	0	0
63	255	0	0
64	0	1	0
100	0	160	0
127	0	255	0
128	0	0	1
160	0	0	160
191	0	0	255
192	1	1	1

210	160	160	160
255	255	255	255

(Regulador1.9) Control y regulador de luz II (0,15)

Material necesario: 3 LEDs rojos y 2 potenciómetro.

Requiere tener implementado los ejercicios opcionales 1.7 y 1.8

En este ejercicio habrá que juntar los ejercicios 1.7 y 1.8, haciendo, que un potenciómetro se encargue de seleccionar los LED a encender como en el ejercicio 1.7 y el otro potenciómetro de aplicar la intensidad de luz entre 0 y 255 a los LEDs que estén encendidos según el primer potenciómetro.