

Situación de aprendizaje 10: Sistema de aparcamiento para coches

Programación con Arduino y posterior prototipado de un sistema de aparcamiento para coches que incluya un avisador acústico que advierta al conductor del peligro de colisión por cercanía.

[sistema de aparcamiento - Búsqueda de Google.png](#)

COMPONENTES	Arduino Uno ArduinoUno.png
	Zumbador Zumbadorrecortado.png
	Sensor Ultrasonidos ultrasonido.png
	Conector 5v Recibidos (442) - raguado@juandelanuza.org - C

Trabajo de indagación: Gracias a los sensores de ultrasonidos, podemos averiguar la distancia que hay a un objeto. Estos emiten un ultrasonido, reciben el rebote y miden el tiempo que ha transcurrido desde la emisión hasta la recepción permitiéndonos calcular la distancia con la fórmula $\text{velocidad} = \text{espacio} \cdot \text{tiempo}$. Utilizando un ultrasonido en la parte trasera de un vehículo, podemos emitir un sonido cuando detectemos una distancia

prudencial al objeto y evitar de este modo chocar con él.

Diseño 2D

Es conveniente utilizar alguna herramienta para diseñar nuestro circuito antes de proceder al montaje. De este modo evitaremos cometer errores con el cableado. Existen muchas herramientas online para este propósito y algunas de ellas como Tinkercad incluso permiten su simulación, pero tienen mayor limitación en las librerías de componentes, por lo que en este caso utilizaremos EasyEDA.

Puedes encontrar un tutorial para crear tus diseños con EasyEDA pinchando [aquí](#).

Incorporamos el Arduino, buzzer y sensor ultrasónico a nuestro proyecto resultando del siguiente modo:

[EasyEDA\(Standard\) - A Simple and Powerful Electronic Circuit Design Tool \(7\).png](#)

Puedes encontrar un tutorial para crear tus diseños con EasyEDA pinchando [aquí](#). Como podemos observar, en nuestro esquemático, vamos a conectar un **actuador** (buzzer) y un **sensor** (sensor ultrasónico). Ambos tienen un pin a **GND** y otro a **5V**, El zumbador tiene un pin de salida que conectaremos al número 2 de nuestra placa y en cambio, el sensor de ultrasonidos, tiene 2 pines, uno de salida o Trigger (le pide al sensor que envíe un tono ultrasónico) que conectaremos al pin 3, y otro de entrada o echo (por el que se recibe el aviso de haber recibido el rebote del disparo inicial) que conectaremos al pin 4 de nuestra placa.

Para abordar la programación de este proyecto, en primer lugar **inicializaremos** el puerto serie que nos servirá para ver en pantalla las medidas del sensor, y crearemos 3 funciones "lejos", "cerca" y "muy_cerca". De este modo será tan sencillo como llamar a cada una de ellas en el bucle principal en función de los rangos de distancia que determinemos para cada uno de estos supuestos.

Construcción

Arduino tiene únicamente 1 pin a 5v y 3 a Gnd.

En esta práctica, utilizamos el Zumbador que tiene 3 pines (5v, Gnd y Señal) y el sensor de ultrasonidos, que tiene 4 pines (2 de datos, uno para Gnd y el otro para 5v).

Para alimentar a 5v ambos, necesitaremos utilizar el conector multiplicador (1 a 4). Utilizaremos cualquiera de los 3 pines Gnd del Arduino para conectar los Gnd del sensor y del zumbador. El pin de datos del zumbador lo conectaremos al número 2 de nuestra placa y los dos pines (trigger y echo) del sensor de ultrasonidos irán respectivamente al 3 y 4 de la placa de Arduino.

[20221119_084032.jpg](#)

Programación - Inicialización del puerto serie y asignación de pines

Como venimos haciendo hasta ahora, representamos un sencillo diagrama de flujo que nos ayude a comprender la ejecución del bucle:

[Bitbloq Robotics - Documento sin título \(1\).png](#)

El siguiente paso, una vez hemos diseñado nuestro circuito es programarlo. Para ello abrimos el editor web de ArduinoBlocks y nos conectamos con nuestro usuario. Partiendo de un proyecto en blanco eligiendo como placa **Arduino Uno**, lo primero que haremos será sacar en pantalla nuestros componentes.

En primer lugar arrastraremos al área de trabajo el Zumbador que lo encontramos dentro del bloque de "actuadores".

[Zumbador.png](#)

y haremos lo mismo con el sensor ultrasónico.

[ArduinoBlocks.png](#)

. Ahora debemos asignarle a cada uno de ellos los Pines que hemos planteado en nuestro esquema electrónico. Para el **zumbador** tendremos que asignarle el **Pin2**, el **Pin3** y **Pin4** respectivamente para los pines Trig y Echo del **sensor de ultrasonidos**.

[ArduinoBlocks \(3\).png](#)

El puerto serie del **Arduino** sirve para poder comunicarse con el ordenador y de este modo, además de poder grabar en su procesador nuestros programas, nos sirve para visualizar en pantalla los valores de entradas y salidas que configuremos.

En el caso que nos afecta, dentro del bucle principal, crearemos una variable que contendrá en cada momento el valor del sensor de distancia, que se verá modificada en función de la distancia a la que se encuentre un objeto de este. Aprovecharemos que este valor se actualiza en esta variable para mostrar su valor por pantalla gracias a la comunicación del puerto serie.

Para **inicializar el puerto serie**, lo haremos directamente sobre el bloque de *Inicializar*, de ArduinoBlocks del siguiente modo:

[ArduinoBlocks \(2\).png](#)

[ArduinoBlocks \(1\).png](#)

Para crear funciones, debemos abrir el menú de funciones y arrastra el bloque al área de trabajo:

[funciones.png](#)

Programación de las funciones

En el planteamiento de nuestra solución, decidimos implementar 3 rangos de actuación determinados por la distancia a la que se encuentra el objeto. Cuando el objeto esta en la zona "lejos" el zumbador emitirá un tono grave y con una pausa de un segundo, si se encuentra en la zona "cerca", aumentará la frecuencia del tono y disminuirá la pausa a medio segundo y por último, si el objeto se encuentra "muy cerca", el zumbador deberá emitir tonos sin apenas pausa (50 milisegundos) y un tono lo suficientemente agudo como para advertir del peligro.

Por ello configuramos **3 funciones** que llamaremos desde el programa principal en función de que se cumplan dichas condiciones y que quedarán del siguiente modo:

[ArduinoBlocks \(4\).png](#)

Antes de incluir las funciones en nuestro programa principal, vamos a configurar la lógica de actuación del sensor. Para ello, como indicábamos anteriormente, definimos una variable que contenga el valor de las medidas del sensor y la mostramos por el puerto serial para ver su valor.

Para crear variables, seleccionamos en el menú lateral "**Variables**" y seleccionamos el bloque "Crear variable". Estas pueden ser numéricas, de texto o lógicas (verdadera/falsa).

[ArduinoBlocks \(17\).png](#)

Seleccionaremos "Crear variable...(Número)" y le introduciremos por nombre "distancia". A partir de este momento, en el mismo menú lateral, al elegir la opción Variables, podremos ver bloques para establecer o tomar el valor de nuestra variable recién creada.

Para asignarle un valor a nuestra variable "distancia" elegimos el bloque "Establecer [distancia]" y lo asociamos al sensor de ultrasonidos que tenemos definido con los pines 3 Trigger y 4 Echo. De este modo, la información del sensor quedará almacenada en la variable y al estar dentro del bucle principal, cada vez que haya un cambio, quedará almacenado el nuevo valor.

[ArduinoBlocks \(7\).png](#)

En estos momentos podemos subir el programa a nuestro Arduino y conectar el puerto serial para ver los resultados del siguiente modo:

[ArduinoBlocks \(8\).png](#)

[ArduinoBlocks \(9\).png](#)

En estos momentos podemos observar como alejando y acercando un objeto al sensor, los valores de medida cambian. Estos sensores no son muy precisos y es aconsejable determinar cual es la máxima y la mínima distancias en las que es capaz de medir correctamente, para determinar los rangos de actuación.

En el desarrollo de esta practica, con los valores obtenidos se definieron los siguientes rangos:

muy_cerca--> entre 0 y 10 centímetros

cerca--> de 11 a 30 centímetros

lejos--> de 31 a 60 centímetros

Por último solo nos falta programar las condiciones para cada rango y llamar a cada función según proceda.

Para crear las condiciones, utilizamos los bloques que encontramos en **Lógica**.

[ArduinoBlocks \(12\).png](#)

Si el valor del sensor (**distancia**) es **mayor que cero** y además es **menor o igual que 10**, debe activarse la función "**muy_cerca**":

Si **distancia es mayor que 10 y menor o igual que 30** entonces debe activarse la función "**cerca**".

Si **distancia es mayor que 30 y menor o igual que 60** entonces deberá advertirnos de la existencia de un objeto **lejano**.

[ArduinoBlocks \(13\).png](#)

y finalmente nuestro programa completo quedaría del siguiente modo:

[ArduinoBlocks \(14\).png](#)

<https://www.youtube.com/embed/w0Hd0M3owI0>

Financiado por el Ministerio de Educación y Formación Profesional y por la Unión Europea - NextGenerationEU

[logo.png](#)

Revision #23

Created 2022-10-15 09:13:46 CEST by Ricardo Aguado Vallejo

Updated 2023-01-17 15:53:27 CET by Equipo CATEDU