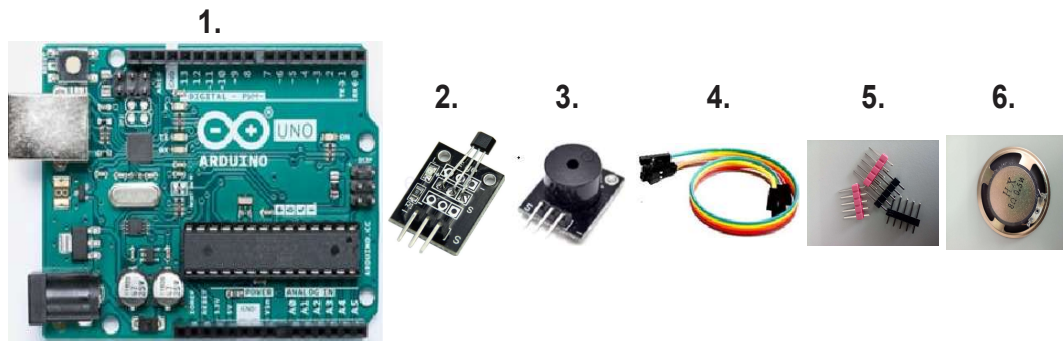


Práctica 13: Sensor magnético

Para esta práctica se necesita:

1. Placa Arduino UNO
2. Sensor magnético K7-003
3. Módulo buzzer KY-006 (zumbador)
4. Cables para realizar las conexiones
5. Puentes
6. Imán (puede ser el de un parlante pequeño)



Introducción

En esta práctica utilizaremos el módulo KY-003, el cual integra un sensor que utiliza el principio del efecto Hall para detectar la presencia de un campo magnético. Este módulo permite detectar campos magnéticos cerca de manera fácil y rápida. Si no hay campo magnético presente, la línea de señal del sensor es ALTA (5V). Cuando se le aproxima un objeto magnético indica la detección cerrando a tierra el pin marcado como “S” (pin de la señal de detección), de esta manera la línea de señal pasa a un estado BAJO (0V).

Este módulo es útil en aplicaciones de electrónica como robótica o proyectos de medición como contadores de ciclos, sistemas de alarmas, etc. El dispositivo

Antes de entrar en detalle de la práctica a desarrollar, es importante conocer un poco del efecto Hall. Por ahora mencionaremos que fue descubierto por el físico Edwin Herbert Hall quien encontró que si se aplica un campo magnético elevado a una fina lámina de oro por la que circula corriente, se produce un voltaje en la lámina transversalmente a como fluye la corriente, este voltaje se llama voltaje Hall (https://www.ecured.cu/Efecto_Hall).

En esta práctica veremos cómo crear un sistema de alarma mínimo y simple con el módulo KY-003, la placa Arduino Uno y un buzzer (módulo KY-006).

Montaje

Las partes principales del módulo se puede ver en la figura 1

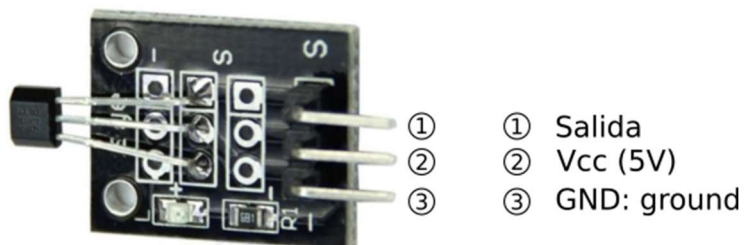


Fig. 1 Pin out módulo KY-003

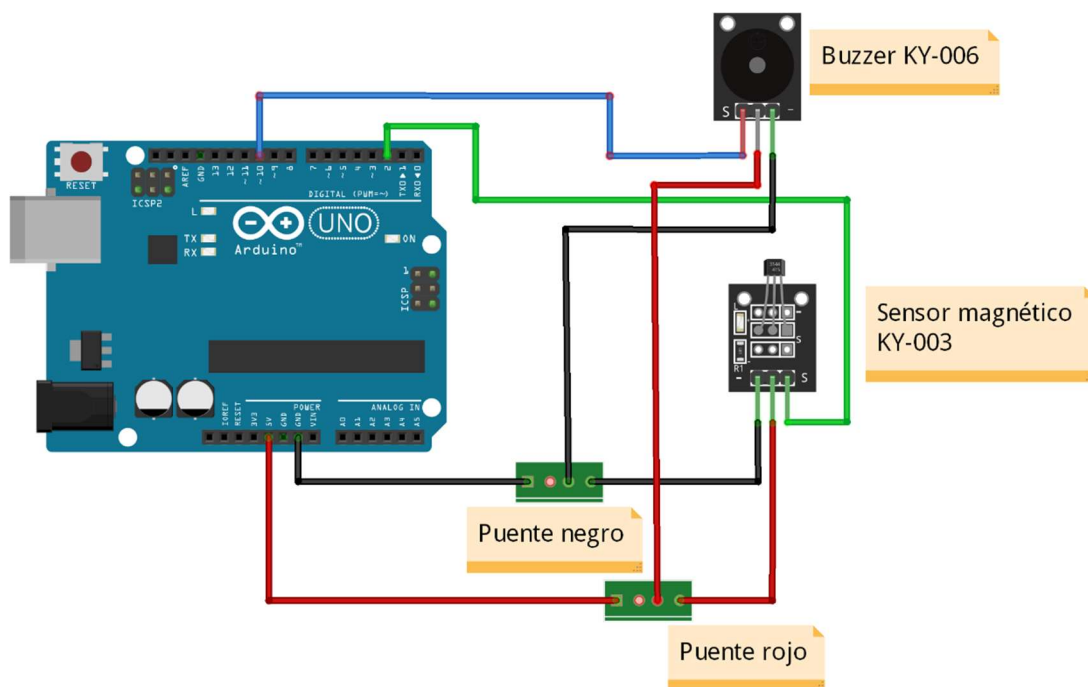
Realizar la conexión del módulo **KY-003** cableando el terminal **(-)** al puente negro (**GND**), el terminal del centro al puente rojo de **5V (Vcc)** y el terminal **"S"** al pin **2**. El módulo del zumbador realizamos el cableado conectando el terminal **"-"** al puente negro (**GND**), el el terminal del centro al puente rojo de **5V (Vcc)** y el terminal **"S"** al pin **10** de la placa Arduino.

Sensor magnético

KY-003	Pin Arduino
-	Puente Negro
Centro	Puente rojo (5V)
S	PIN 2

Zumbador

KY-006	Pin Arduino
-	Puente Negro
Centro	Puente rojo (5V)
S	PIN 10



Programación

El escenario planteado es que tenemos una puerta cerrada, la cual cuenta con el sensor asegurado en la pared y en la parte superior de la puerta va a estar ubicado un objeto magnético (imán u otro elemento con propiedades magnéticas). La idea es que se active una alarma sonora (zumbador) cuando la puerta sea abierta y se apague al momento de cerrarla nuevamente.

Para el desarrollo de nuestra alarma con sensor magnético, vamos a plantear la lógica que debe tener nuestro programa.

En la introducción mencionamos que el PIN “S” va a estar en estado **ALTO (5V)** sino hay campo magnético cerca y en estado **BAJO (0V)** si hay campo magnético cerca del módulo. Por lo tanto, vamos a necesitar un pin de entrada digital en nuestro Arduino para hacer la lectura del estado. Vamos a seleccionar el PIN “2” y vamos a colocar una condición para verificar el estado del pin. Si la entrada está en 5V es porque la puerta se encuentra abierta (no hay campo magnético cerca).

El siguiente paso es activar la alarma sonora cuando el pin 2 esté en 5V. Para ello vamos a utilizar una salida digital para activar el buzzer (zumbador). Vamos a seleccionar el PIN 10 para este fin. Cuando la puerta esté abierta, el PIN 10 debe colocarse en estado ALTO (5V) para encender el buzzer y en estado BAJO (0V) para apagar la alarma. Adicionalmente para generar el sonido vamos a estar alternando entre encendido/apagado la salida para el zumbador. Un valor de tiempo utilizado para esta practica es de 100 us (microsegundos). Pueden probar con otros valores para obtener diferentes sonidos. Para facilitar la modificación del tiempo utilizaremos una variable de tipo entero para guardar el valor de tiempo establecido.

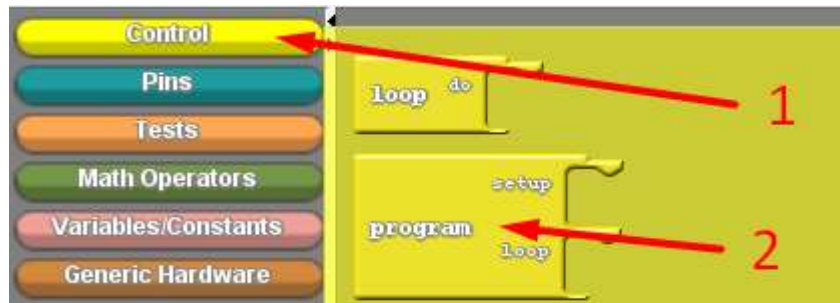
La secuencia del programa nos quedaría de la siguiente manera:

1. Definir variable de tipo entero para guardar el tiempo entre encendido/apagado para generar el sonido en el zumbador.
2. Leer el estado del PIN 2. Si la entrada está en estado **ALTO**, la puerta está abierta. Encender ALARMA (PIN 10) alternadamente cada 100 us.
3. Leer el estado del PIN 2. Si la entrada está en estado **BAJO**, la puerta está cerrada. ALARMA (PIN 10) apagada.

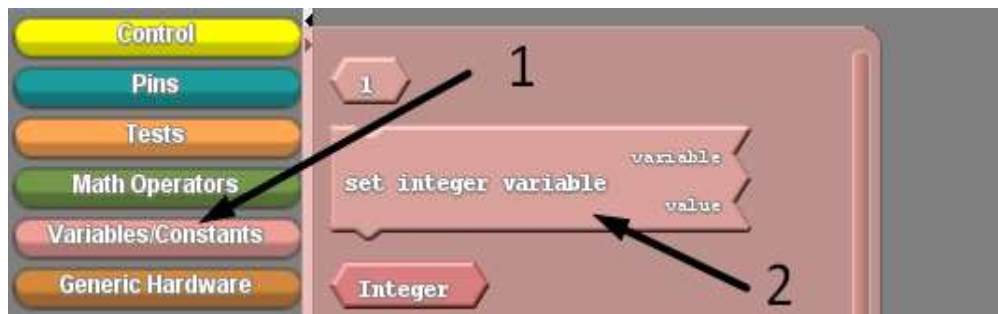
Con la secuencia del proceso pasamos a construir el código en Ardublock.

1. Quitamos el bloque “**loop**” de nuestro espacio de trabajo arrastrando y soltando en la parte inferior izquierda.

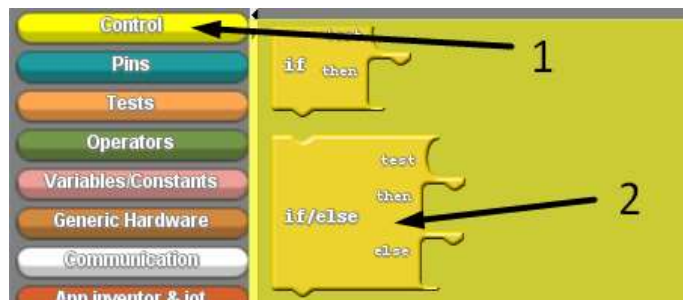
- Colocamos el bloque para tener la opción de la configuración inicial (setup). Ir al botón **“Control”** del panel izquierdo y seleccionar el bloque **program**.



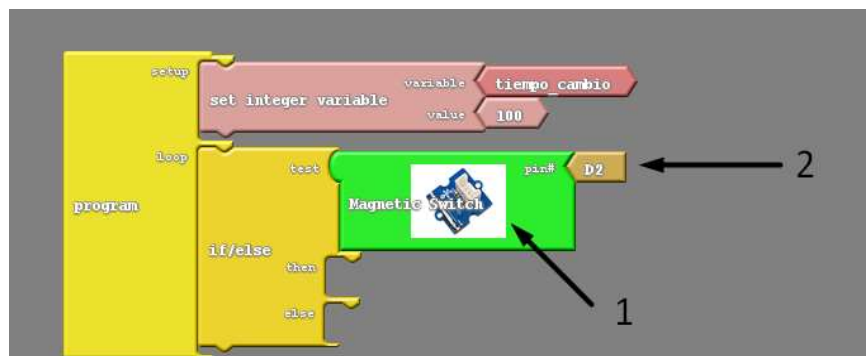
- Colocar el bloque de variable entera para el retardo. Asignar el nombre **“tiempo_cambio”** y el valor en 100. El bloque debe ir en la sección **setup** para que unicamente se configure cada vez que se prende la tarjeta Arduino Uno. Recordemos que el **setup** se ejecuta una vez y el **loop** se mantiene ejecutando mientras la tarjeta esté encendida.



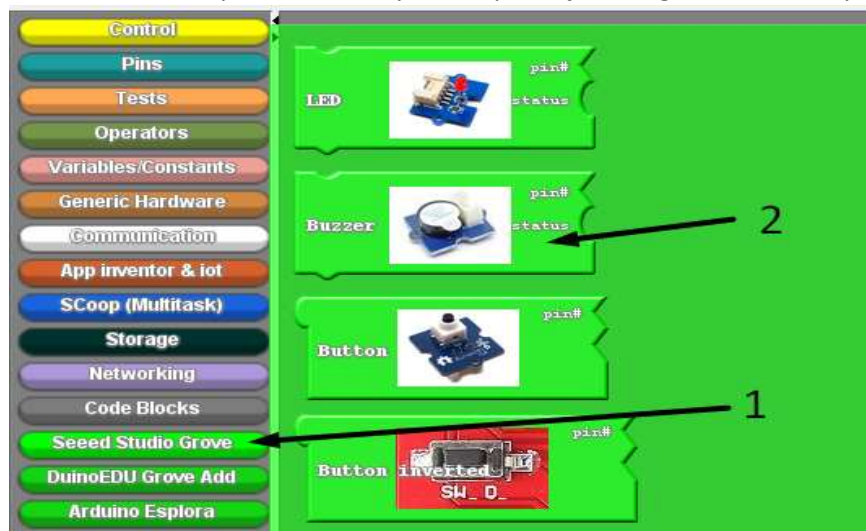
- Colocar el bloque condicional **if/else** (si/sino) para leer el estado del PIN 2. Si la puerta está abierta (HIGH) encender ALARMA, si no, ALARMA debe estar apagada.

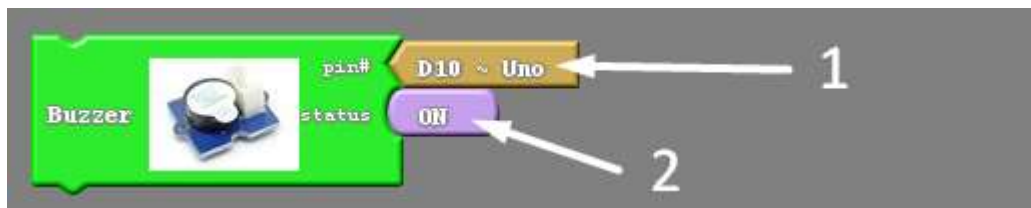


5. Colocar el bloque Magnetic switch y asignar el PIN 2:

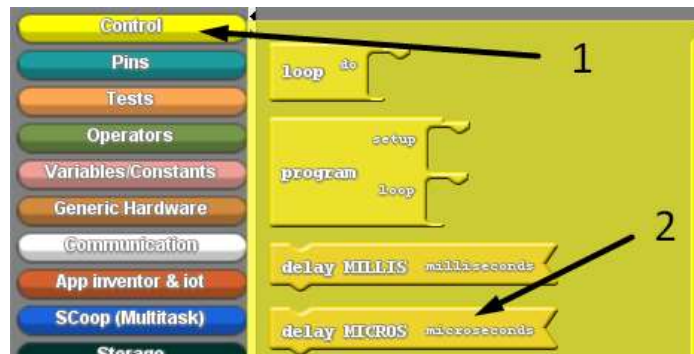


6. Colocamos el bloque del Buzzer y en la opción pin# asignamos el D10 y el status en ON





7. Colocamos un bloque para un retardo de 100 microsegundos. Aquí utilizaremos la variable que hemos creado antes “**tiempo_cambio**”.



8. Colocamos el bloque del **Buzzer** y en la opción **pin#** asignamos el **D10** y el **status** en **OFF**



9. Colocamos un bloque para un retardo de 100 microsegundos.

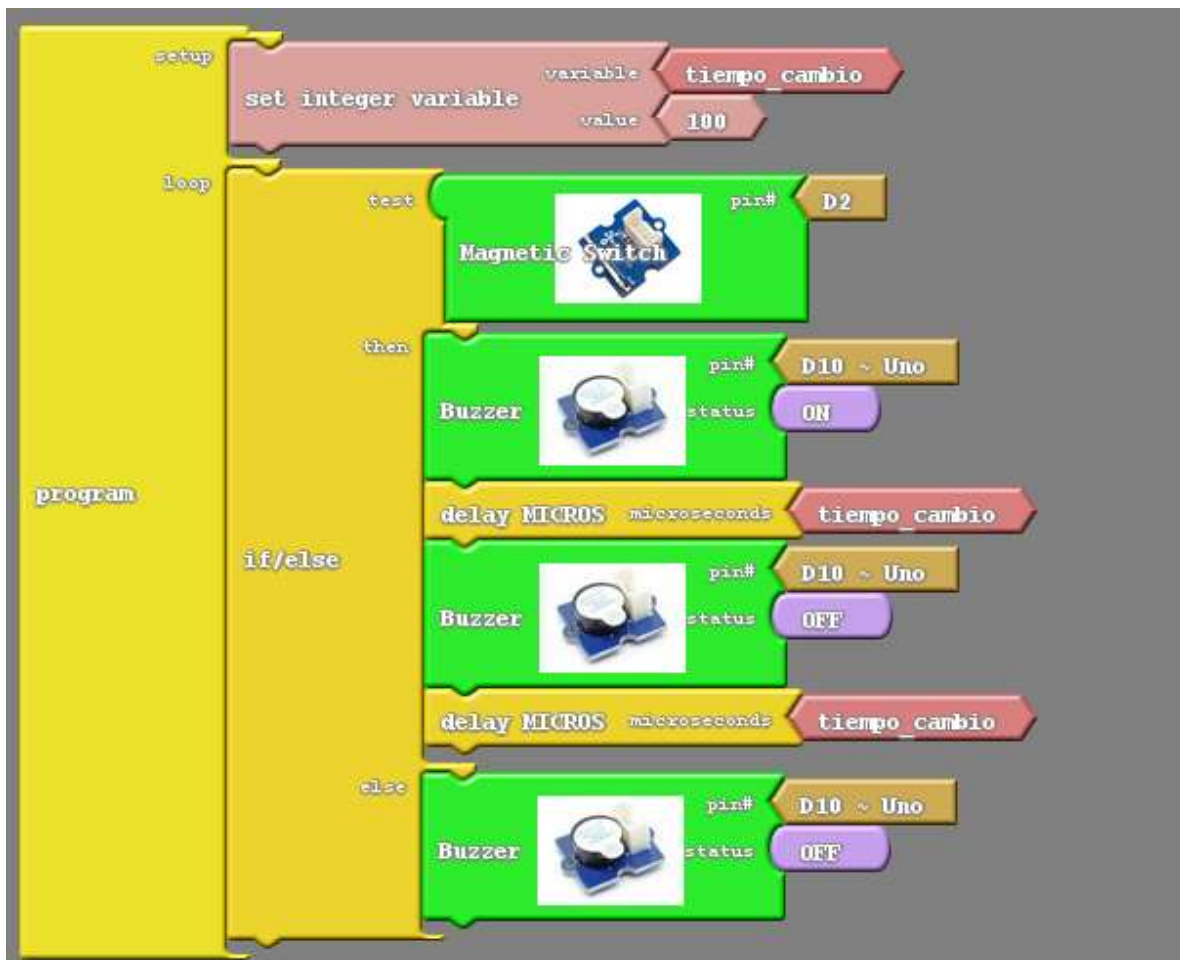


10. Para finalizar colocamos el bloque del **Buzzer** en la sección **else** del bloque **if/else**. Aquí la ALARMA se mantiene apagada para cuando la puerta esté cerrada.

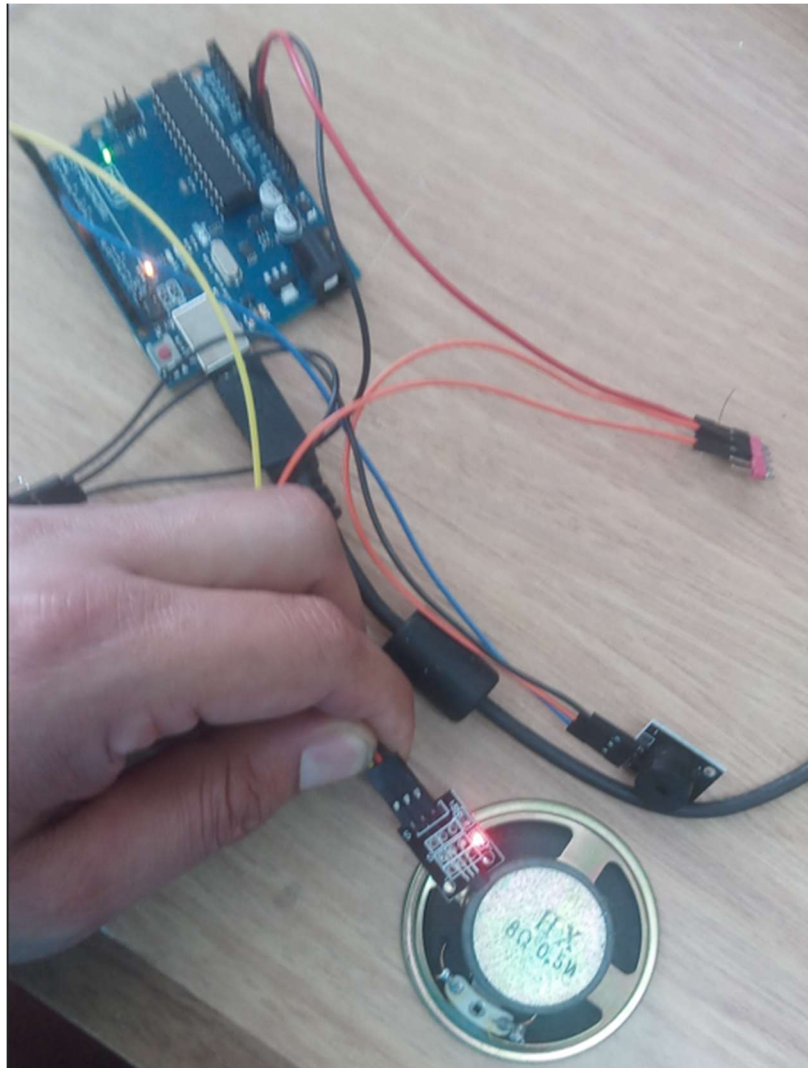


Con esto hemos finalizado la inserción de bloques para nuestro semáforo. El uso de la variable **Retardo** nos facilita la modificación del tiempo que un color permanece encendido, porque solo debemos modificar el valor en el inicio del programa.

Finalmente, nuestro programa completo nos debe quedar de la siguiente manera:



Para probar, solo debemos acercar el parlante al sensor de tal manera que veamos que el LED del módulo magnético se encienda. Al alejar el imán se debe apagar el LED y sonar la alarma.



Ampliación

- ¿Cómo podríamos mejorar el programa para que nuestra alarma esté disponible cuando el usuario decida habilitarla? Ejemplo sale de casa o en horas de la noche cuando está durmiendo.
- Implemente los cambios necesarios en el programa para que la alarma dure encendida un tiempo determinado, ejemplo 30 segundos.