

# Prácticas con snap4arduino: Entradas y salidas digitales

Manuel Romero Fraidía  
Departamento de Tecnología

IES Virgen de Valme



## Finalidad de la práctica

En esta segunda práctica vamos a ver cómo maneja Arduino las entradas y salidas digitales. para ello haremos uso de un pulsador (entrada) que hará que se encienda un led (salida).

## Búsqueda de información

Antes de empezar con la práctica es conveniente visitar los siguientes enlaces:

<https://aprendiendoarduino.wordpress.com/2015/03/30/entradas-y-salidas-digitales/>

<http://snap4arduino.rocks/>

<https://github.com/bromagosa/Snap4Arduino/wiki/Sensors-and-Actuators>

<https://programarfacil.com/blog/arduino-blog/resistencia-pull-up-y-pull-down/>

<http://ardupineda.blogspot.com.es/2017/03/turning-led-on-and-off-using-push.html>

## Hardware

Los componentes y materiales utilizados son los siguientes:

1 placa Arduino UNO

1 placa protoboard

1 LED

1 resistencia de valor  $220\Omega$

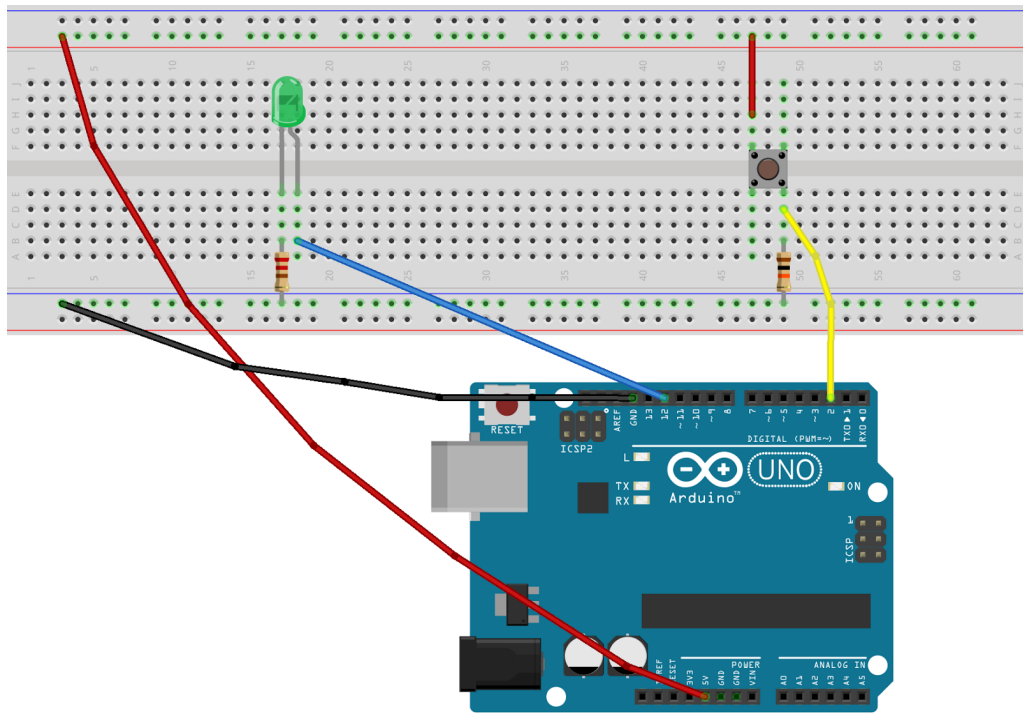
1 pulsador

1 resistencia de valor  $10k\Omega$

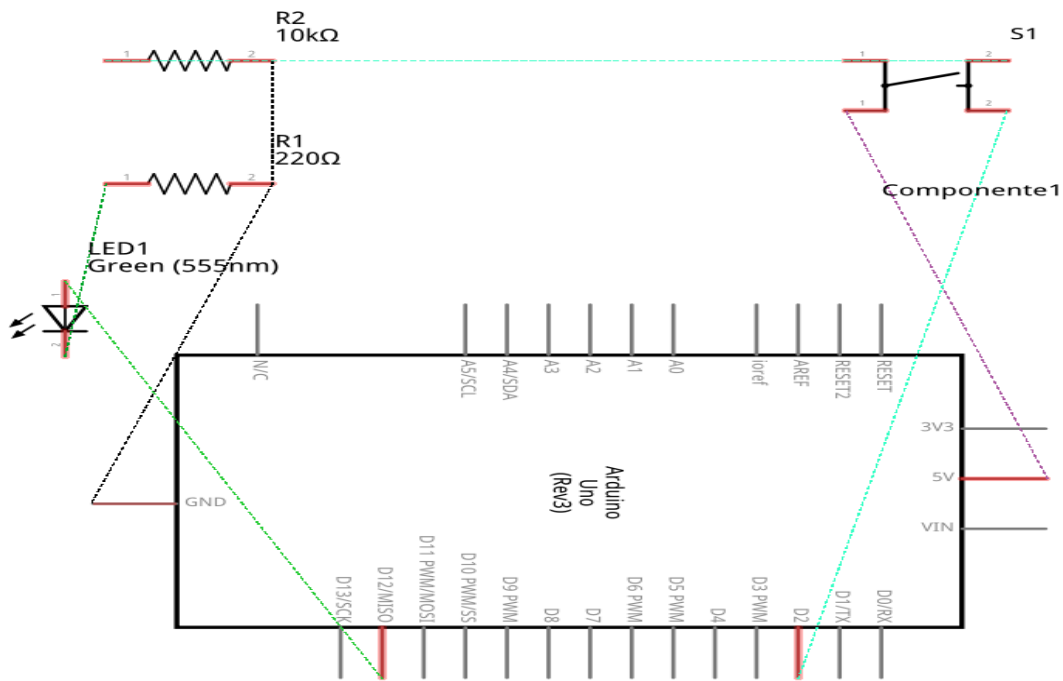
Cables de conexión

## Esquemas de conexión

A continuación se adjuntan tanto el esquema de conexión en protoboard como el esquema eléctrico, ambos generados con el programa Fritzing.



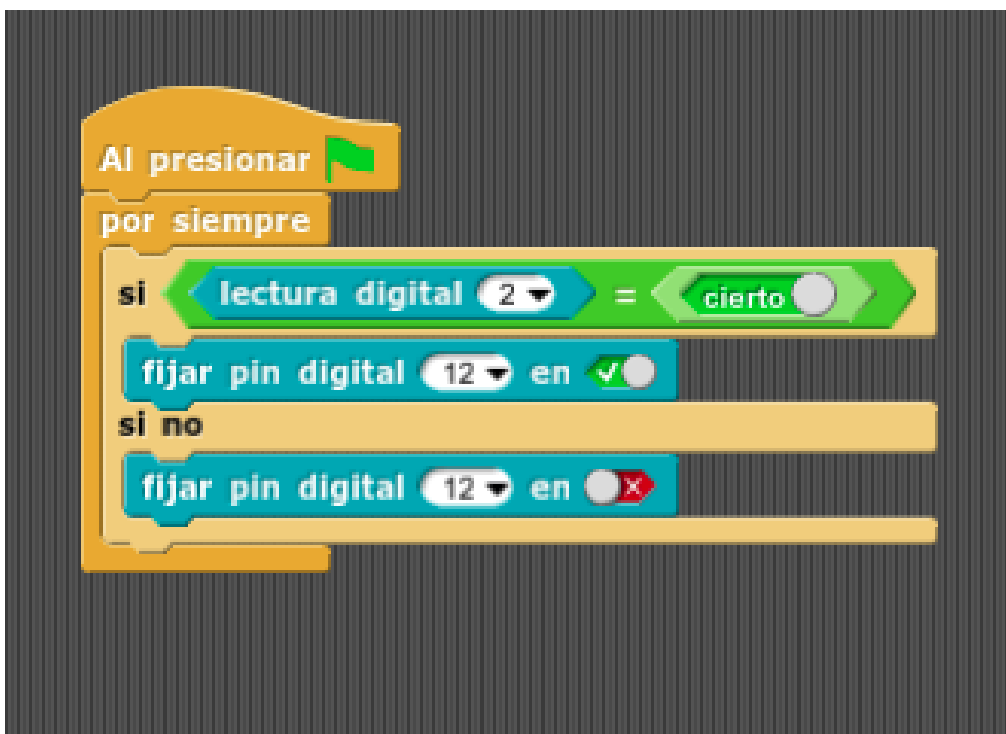
fritzing



fritzing

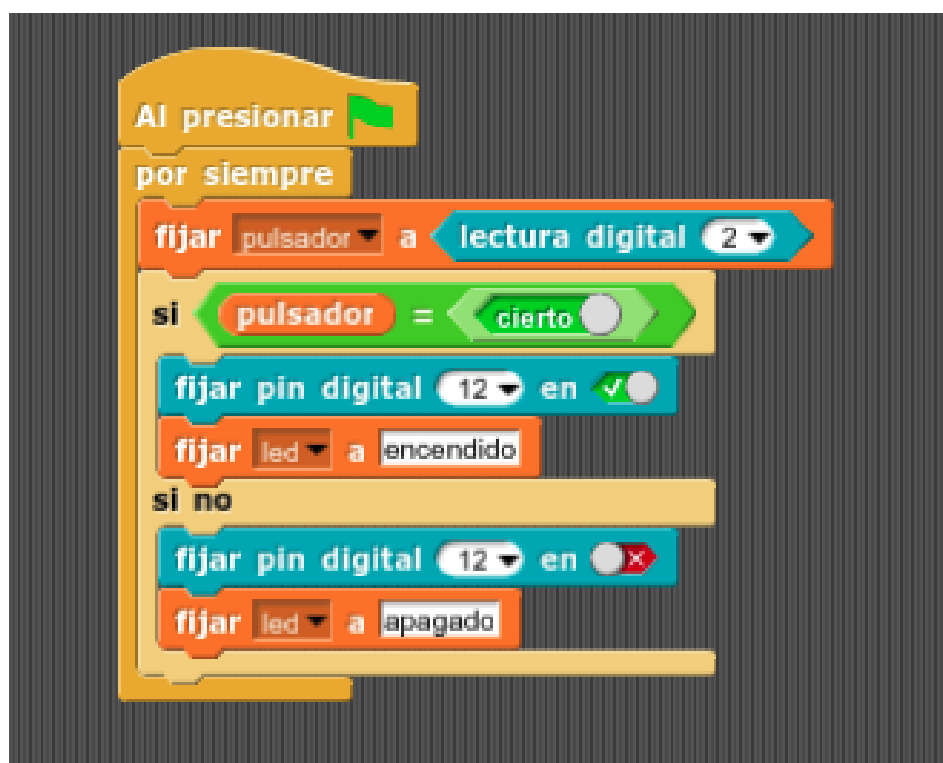
## Software

Los bloques que debemos arrastrar al área de programas son los siguientes:

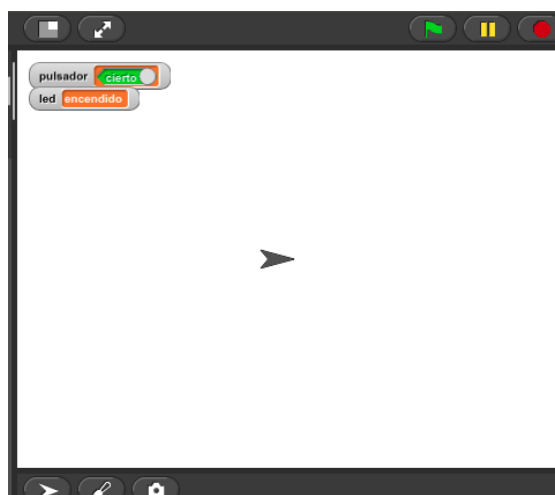
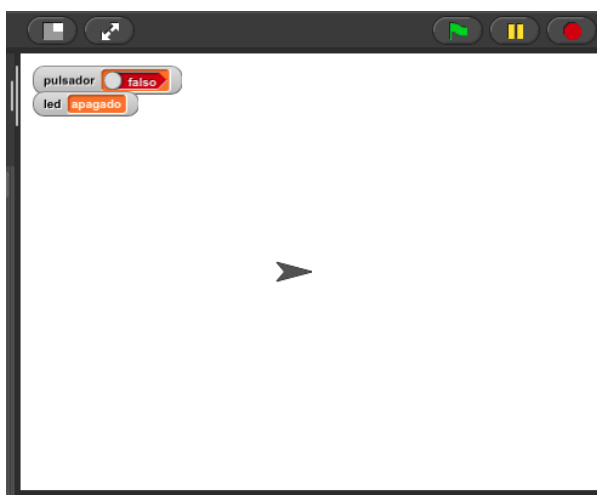


Como vemos, al presionar la bandera verde se le ordena a arduino que encienda la salida digital 12 solo si la lectura en la entrada digital 2 es verdadera, es decir, solo mientras se mantenga presionado el pulsador.

Vamos a hacer lo mismo pero de una manera más correcta, más “profesional”, introduciendo variables. Crearemos dos, una llamada “pulsador”, en la que se guardará la lectura digital de este, y otra llamada “led”, que nos mostrará en todo momento si el led está encendido o apagado. Para ello, nos vamos al apartado de bloques “Variables” y pulsamos en el botón “crear variable”, automáticamente nos deben aparecer en pantalla. Veamos cómo quedarían ahora los bloques:



Vemos como al iniciar el programa lo primero que hacemos es fijar la variable pulsador a la entrada digital a la que tenemos conectado el mismo, en este caso, la 2. A continuación, introducimos la condición: "si el pulsador está presionado... si no..." para que encienda o apague el led y muestre en pantalla la palabra "encendido" o "apagado, respectivamente. En las siguientes capturas podemos apreciarlo más claramente:



Para terminar, veamos el código que habría que introducir en el IDE de Arduino para que nuestro montaje hiciera exactamente lo mismo:

```
// Práctica encender y apagar un LED a través de botón pulsador
const int LED=12;
const int BOTON=2;
int val;
void setup(){
  pinMode(LED,OUTPUT);
  pinMode(BOTON,INPUT);
}
void loop(){
  val=digitalRead(BOTON);
  if (val==HIGH){
    digitalWrite(LED,HIGH);
  }
  else { digitalWrite(LED,LOW);
  }
}
```

## Actividades y propuestas de mejora

1. Compara los bloques con el código escrito, ¿eres capaz de relacionar cada comando con lo cada bloque?
2. ¿Cómo podríamos hacer que el led se mantuviera encendido sin necesidad de estar presionando continuamente el pulsador? Es decir, presionamos una vez y el led se enciende, otra vez y se apaga. Seguro que eres capaz de hacerlo pensado un poco. (Pista: juega con la variable "led" de manera que esta almacene el estado previo del led (1-led))
3. Añade otro pulsador al montaje. ¿Cómo podríamos hacer que el led solo se encendiera si mantenemos los dos pulsados a la vez? ¿y que uno encendiera y otro apagara?