

Sistemas Empotrados y de Tiempo Real

Desarrollo con Arduino Watchdog y Lecturas Digitales

Grado en Ingeniería de Robótica Software

Teoría de la Señal y las Comunicaciones y
Sistemas Telemáticos y Computación

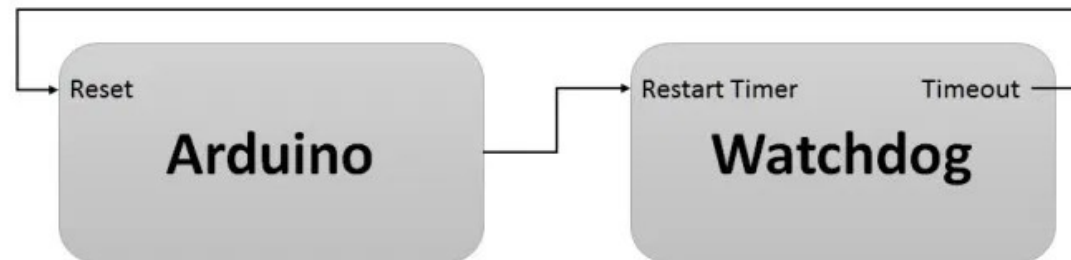
Roberto Calvo Palomino
roberto.calvo@urjc.es

Watchdog



Watchdog

- Un watchdog es un mecanismo de seguridad que provoca un reset del sistema en caso de que éste se haya bloqueado o no responde al comportamiento deseado.
- El watchdog es un timer que va dentro de la arquitectura del microcontrolador y es independiente del resto de timers y del sketch que se está ejecutando
- El watchdog utiliza una fuente de reloj interna de 128 kHz.



Watchdog

- Incluir librería en el sketch

```
#include <avr/wdt.h>
```

- Desactivar el watchdog y configurarlo

```
wdt_disable();  
wdt_enable(tiempo);
```

- Actualiza el watchdog para que no produzca el reseteo

```
wdt_reset();
```

Watchdog

- Un watchdog es un mecanismo de seguridad que provoca un reset del sistema en caso de que éste se haya bloqueado o no responde al comportamiento deseado.

| Tiempo | wtd_enable() |
|--------|--------------|
| 15 ms | WDT0_15MS |
| 30 ms | WDT0_30MS |
| 60 ms | WDT0_60MS |
| 120 ms | WDT0_120MS |
| 250 ms | WDT0_250MS |
| 500 ms | WDT0_500MS |
| 1 s | WDT0_1S |
| 2 s | WDT0_2S |
| 4 s | WDT0_4S |
| 8 s | WDT0_8S |

Watchdog

- Código ejemplo:

```
#include <avr/wdt.h>

int loops = 0;

void setup()
{
  Serial.begin(9600);
  wdt_disable();
  wdt_enable(WDTO_8S);
  Serial.println("iniciando sketch");
  delay(1000);
}

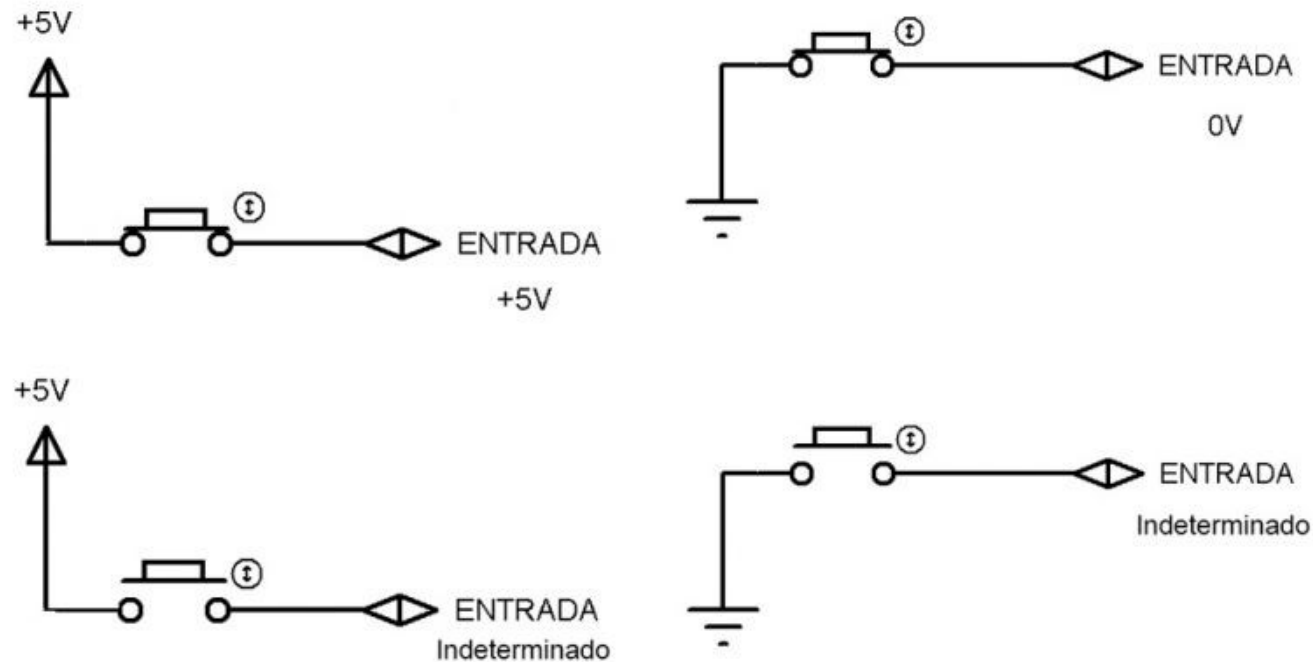
void loop()
{
  Serial.println("Iteracion: " + String(loops));
  delay(1000);
  if (loops >= 10) {
    Serial.println("Dentro de bucle largo");
    for (int i = 0; i < 100; i++) {
      Serial.println(i);
      delay(1000);
    }
  }

  wdt_reset();
  loops++;
}
```

Lecturas Digitales

Lecturas Digitales

- Lecturas confiables desde un botón o switch
- Cuando una entrada digital no está conectada a nada se define que tiene un estado de *alta impedancia* o *flotante*.
- Un estado flotante nos puede llevar a valores no esperados en las entradas digitales.



Lecturas Digitales

- Código ejemplo:

```
const int inputPin = 2;

void setup() {
  Serial.begin(9600);
  pinMode(LED_BUILTIN, OUTPUT);
  pinMode(inputPin, INPUT);
}

void loop(){
  int val = digitalRead(inputPin);
  Serial.println(val);
  digitalWrite(LED_BUILTIN, val);

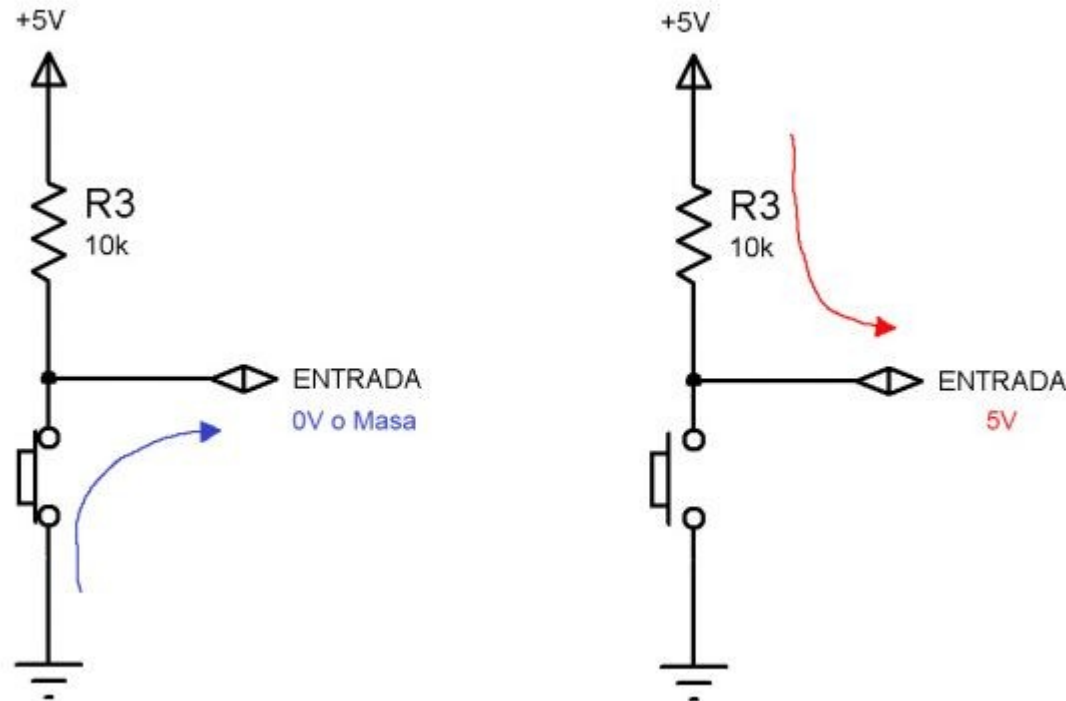
  delay(500);
}
```

Lecturas Digitales

- Resistencias *pull up* o *pull down*
- Resistencias normales pero que debido a su disposición en un circuito electrónico establecen el estado lógico de un pin cuando se encuentra en estado reposo.
- Normalmente se utilizan resistencias de $10K\Omega$

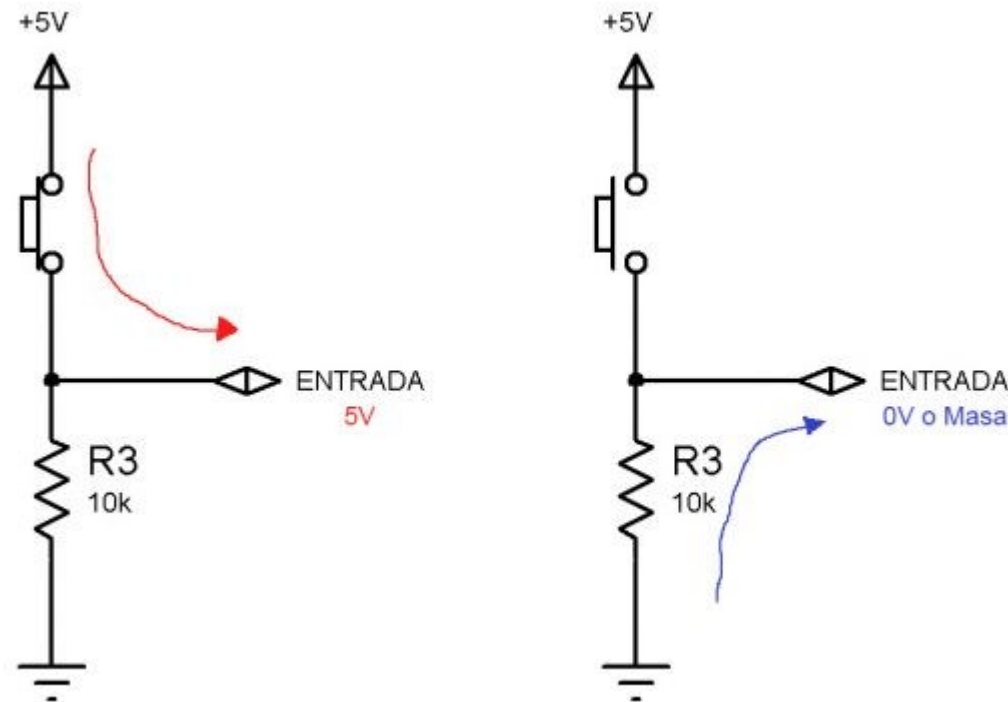
Resistencias Pull-up

- Nos aseguramos que la entrada digital va a tener un valor HIGH hasta que se produzca la pulsación del pulsador, en ese momento el valor se establece a LOW



Resistencias Pull-down

- Nos aseguramos que la entrada digital va a tener un valor LOW hasta que se produzca la pulsación del pulsador, en ese momento el valor se establece a HIGH



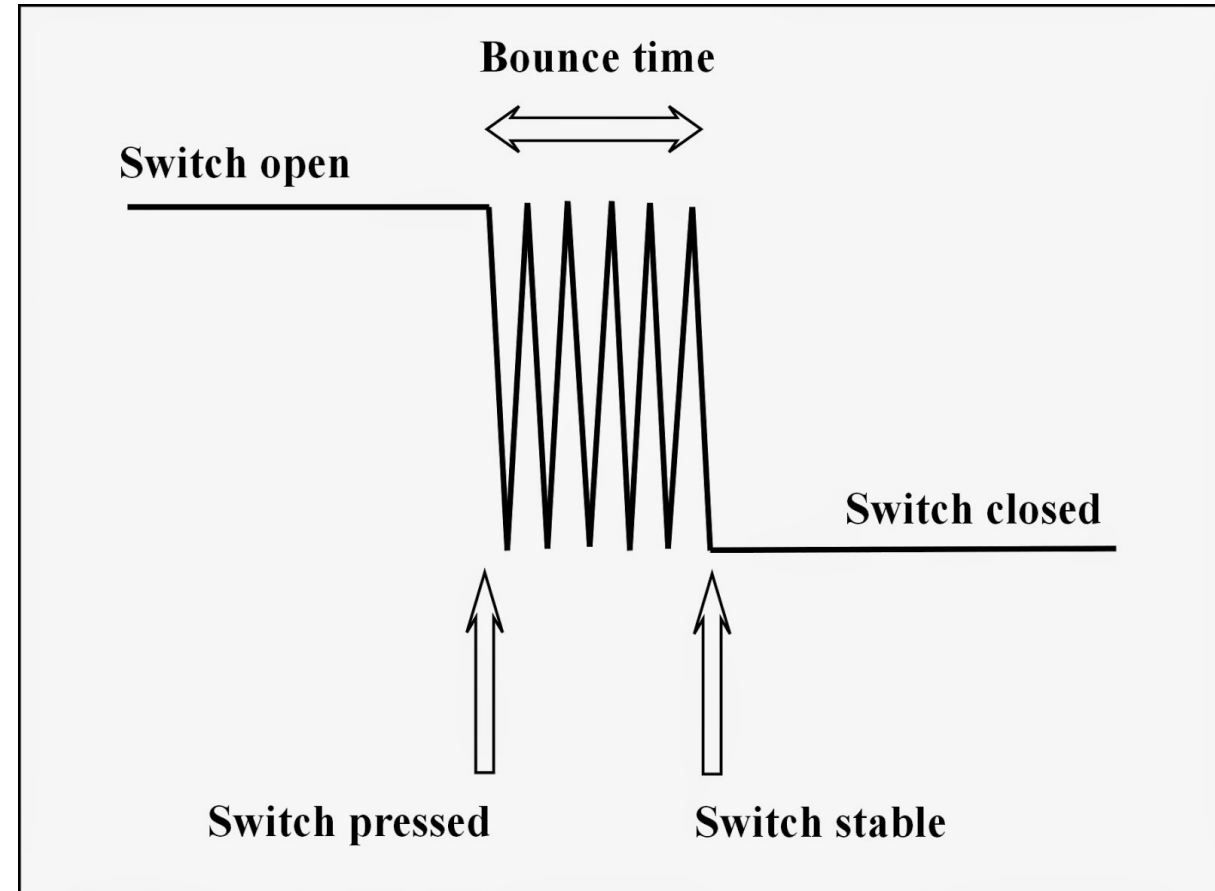
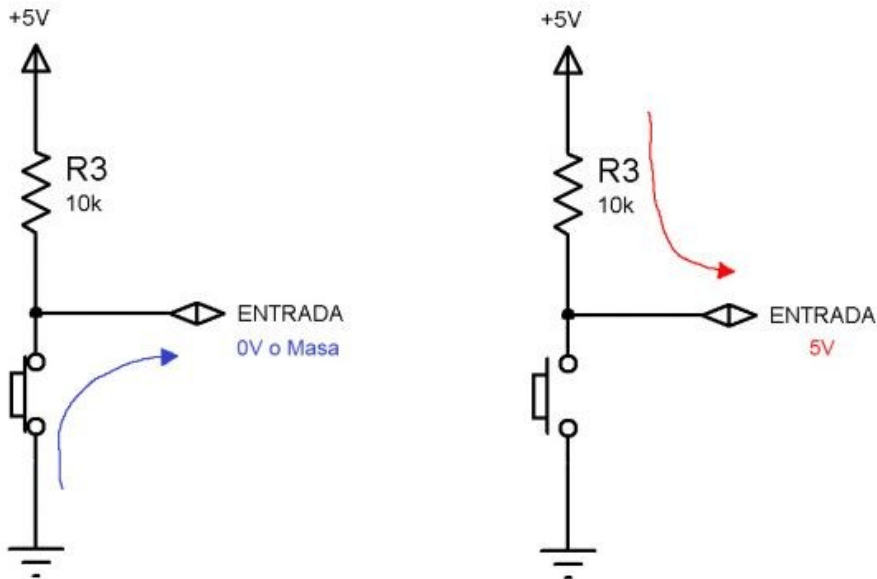
Resistencia pull-up con Arduino

- El microcontrolador Atmega del Arduino tiene resistencias pull-up internas
- Evitamos mas componentes en nuestro circuito
- Solo podemos usar PULLUP.

```
pinMode(inputPin, INPUT_PULLUP);
```

Rebotes (Bounce)

- En el instante de abrir o cerrar un interruptor, se pueden producir rebotes hasta que se estabilice la señal.



Rebotes (Bounce)

- Soluciones:
 - Uso de condensadores en la entrada
 - Añadir código para detectar los “rebotes”
- Ideas sobre código a añadir
- Revisar
 - Capitulo 5.4 de Arduino Cookbook, 3rd Edition
 - Determining How Long a Switch Is Pressed

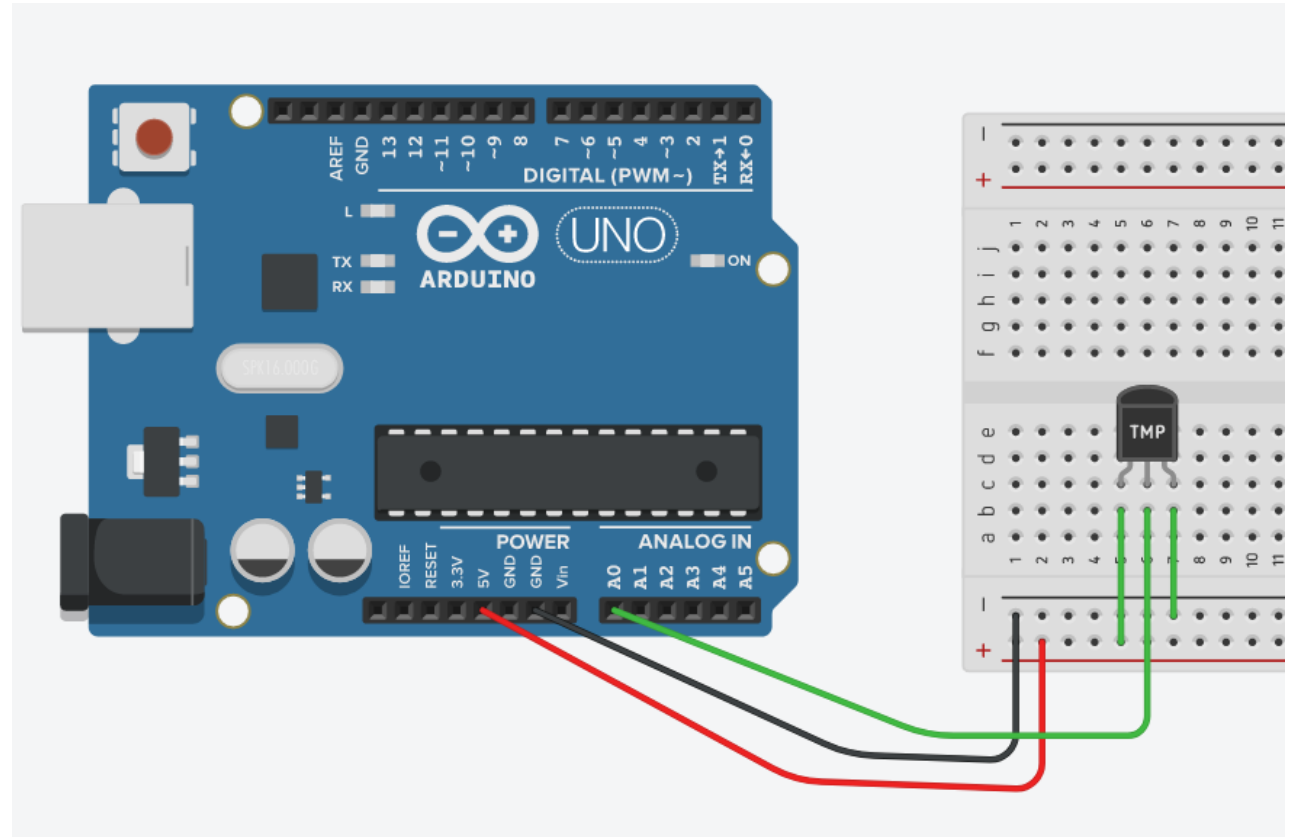


Sensores



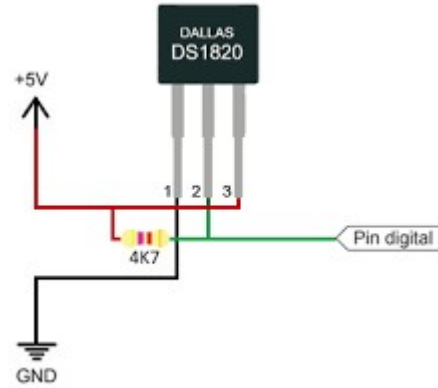
Sensores: TMP36

- Sensor Temperatura (TMP36)
- Sensor Analógico
- Rango: -40°C to $+125^{\circ}\text{C}$
- Resolución: 0.5°C
- $10\text{ mV}/^{\circ}\text{C}$



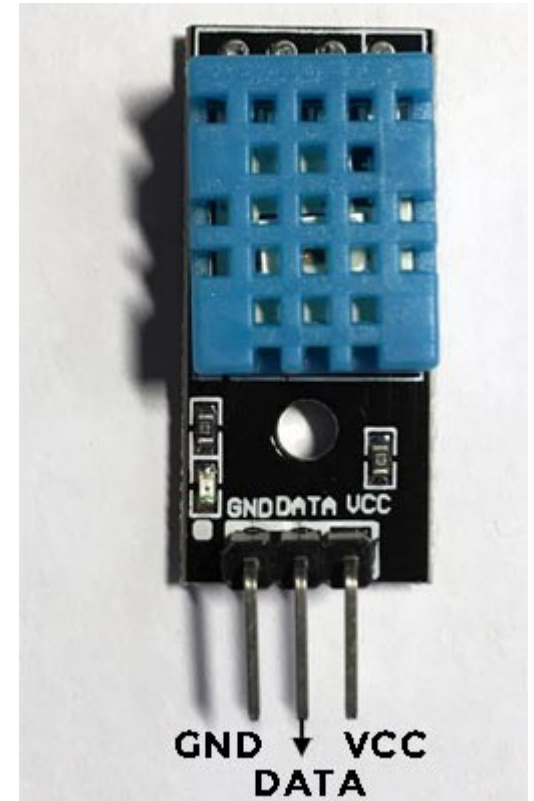
Sensores: TMP36

- TEMP36, DS1820, ...



Sensores DHT-11/20

- Sensor de temperatura y humedad
- Alta fiabilidad y estabilidad debido a su señal digital calibrada
- Se trata de un sensor analógico, pero dentro de la PCB se realiza la conexión digital.
- La salida digital siempre va conectada en modo pull-up



0011 0101
8 bits humedad

0000 0000
8 bits humedad

0001 1000
8 bits temperatura

0000 0000
8 bits temperatura

0100 1001
bits de paridad

Sensores DHT11

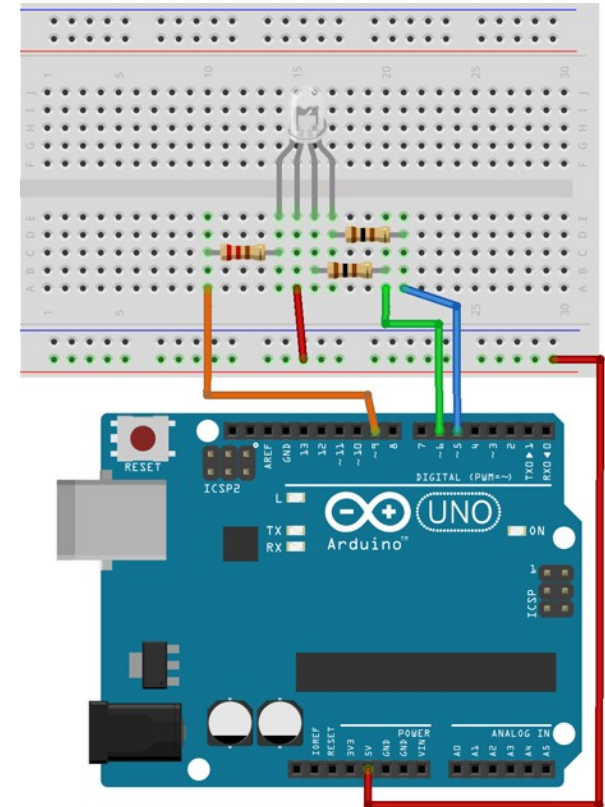
- Necesidad de instalar librerías adicionales desde el IDE de arduino.



- Documentación
 - <https://github.com/adafruit/DHT-sensor-library>

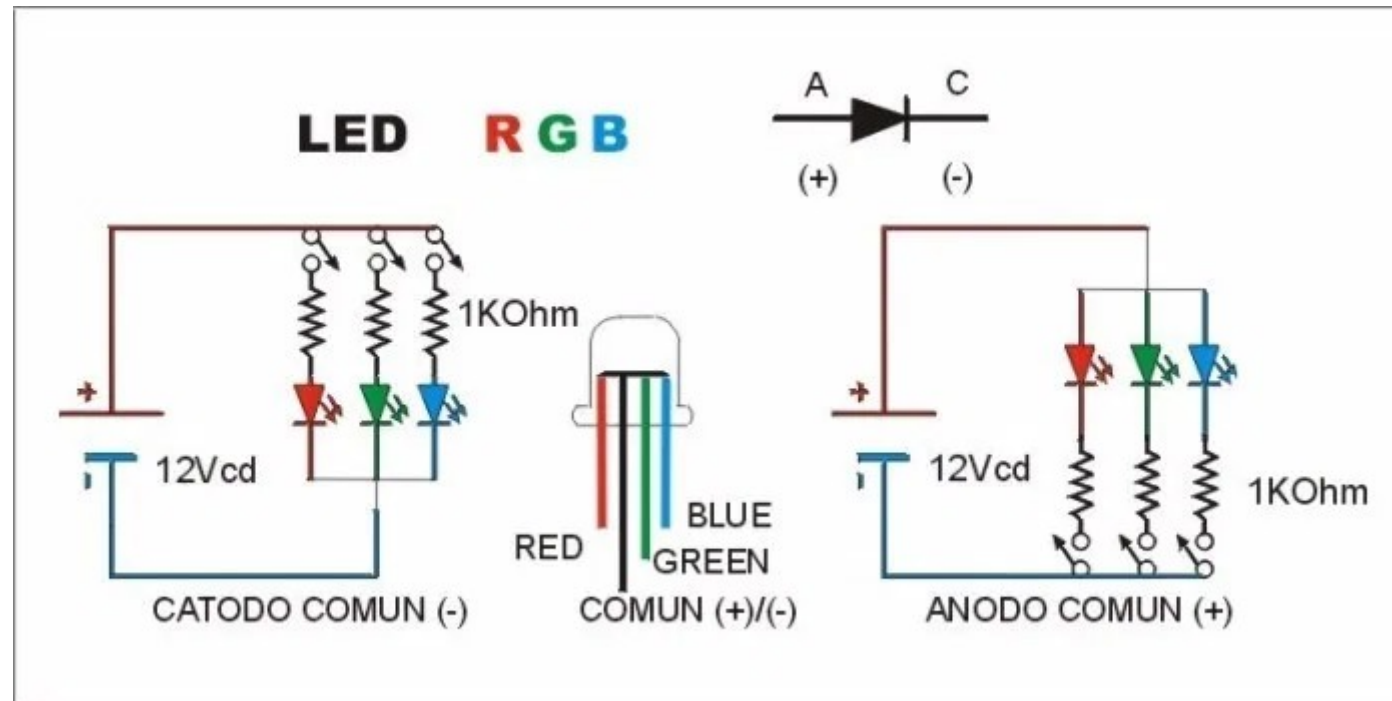
Sensores: LED RGB

- LED RGB
- Utilizan los pines “analógicos” PWM
- Basado en el concepto “duty cycle”



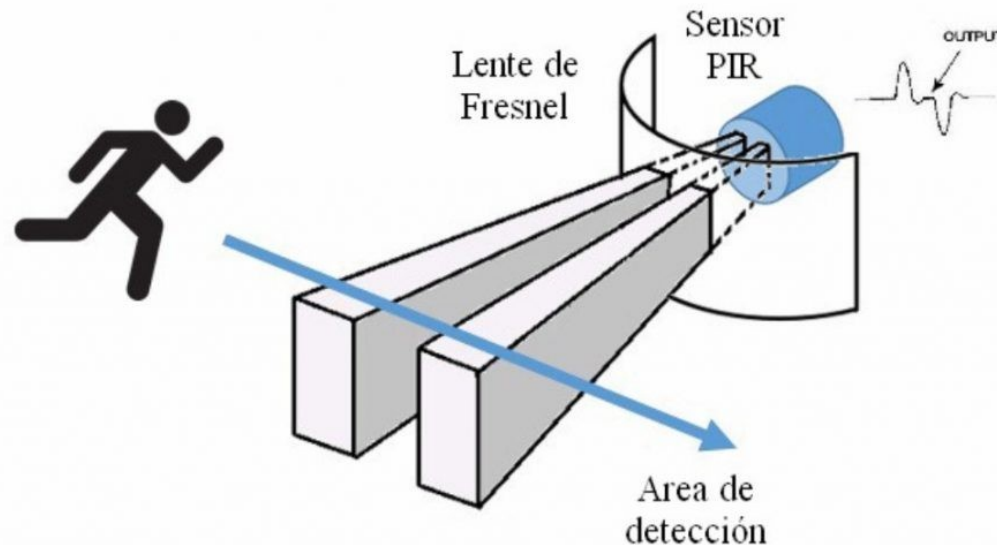
Sensores: LED RGB

- Ánodo o Cátodo común
- 3 pines (R,G,B)
- Uso de analogWrite



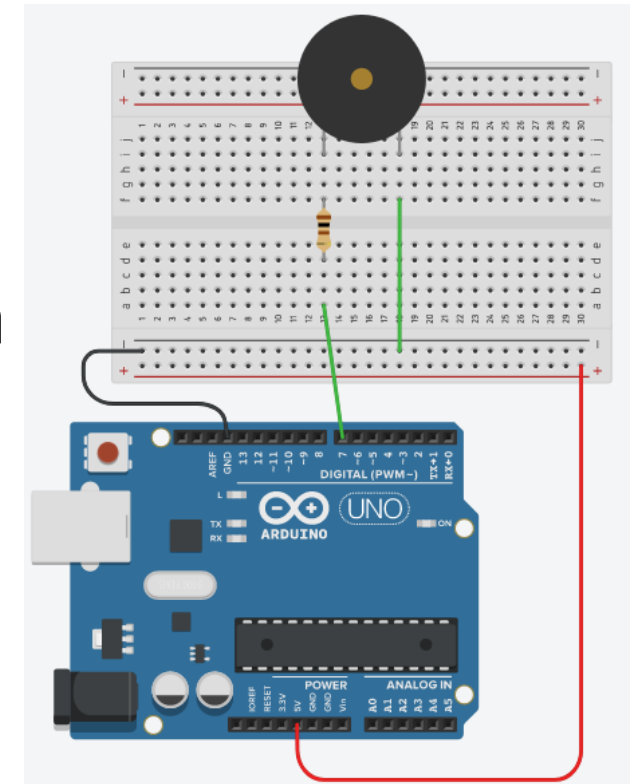
Sensores: PIR

- Los sensores infrarrojos pasivos (PIR) son dispositivos usados para la detección de movimiento.
- Se basan en la medición (recepción) de la radiación infrarroja
- Lectura digital (HIGH presencia, LOW no-presencia)



Sensores: Zumbador

- Los Zumbadores (buzzer), en ocasiones denominados zumbadores, son dispositivos que generan un sonido de una frecuencia determinada y fija cuando son conectados a tensión.
- Arduino dispone de 2 funciones para hacer funcionar estos dispositivos.
 - `Tone()` y `noTone()`
- Genera una onda cuadrada a la frecuencia especificada.



<https://www.arduino.cc/reference/en/language/functions/advanced-io/tone/>

