

MÓDULO 4: PROYECTO 2 "Interfaz de una nave Espacial"

CURSO PROGRAMACIÓN DE PLACAS ROBÓTICAS

Introduccion

Los pins digitales de Arduino solo pueden leer dos estados. Este tipo de entrada es normalmente conocida como digital. A estos dos estados se les refiere como Alto o bajo (**HIGH** y **LOW** en inglés). Cuando le indicamos a un pin de salida un nivel **HIGH** utilizando el comando **digitalWrite()**, lo estamos activando, pero cuando le indicamos un nivel **LOW**, lo estamos desactivando.

Introduccion

Los pins digitales de Arduino pueden actuar tanto como entradas como salidas. En nuestro código, podemos configurarlos para que funcionen dependiendo de la acción que deseemos que hagan. Cuando dichos pins funcionan como salidas, podemos activar componentes tales como diodos leds. Si los configuramos como entradas, podemos comprobar por ejemplo, si un pulsador ha sido activado o no.

- Los pins 0 y 1 se utilizan para la comunicación con el pc, es mejor utilizar los pines a partir del número 2.

Introducción

- En este proyecto vamos a utilizar un sensor de temperatura para medir el calor de nuestra piel. Este componente proporciona como valor de salida una tensión que varía en función de la temperatura que detecta.
- Posee tres patillas:
 - una conectada a tierra
 - otra conectada a la alimentación
 - la tercera que proporciona la tensión variable a Arduino.

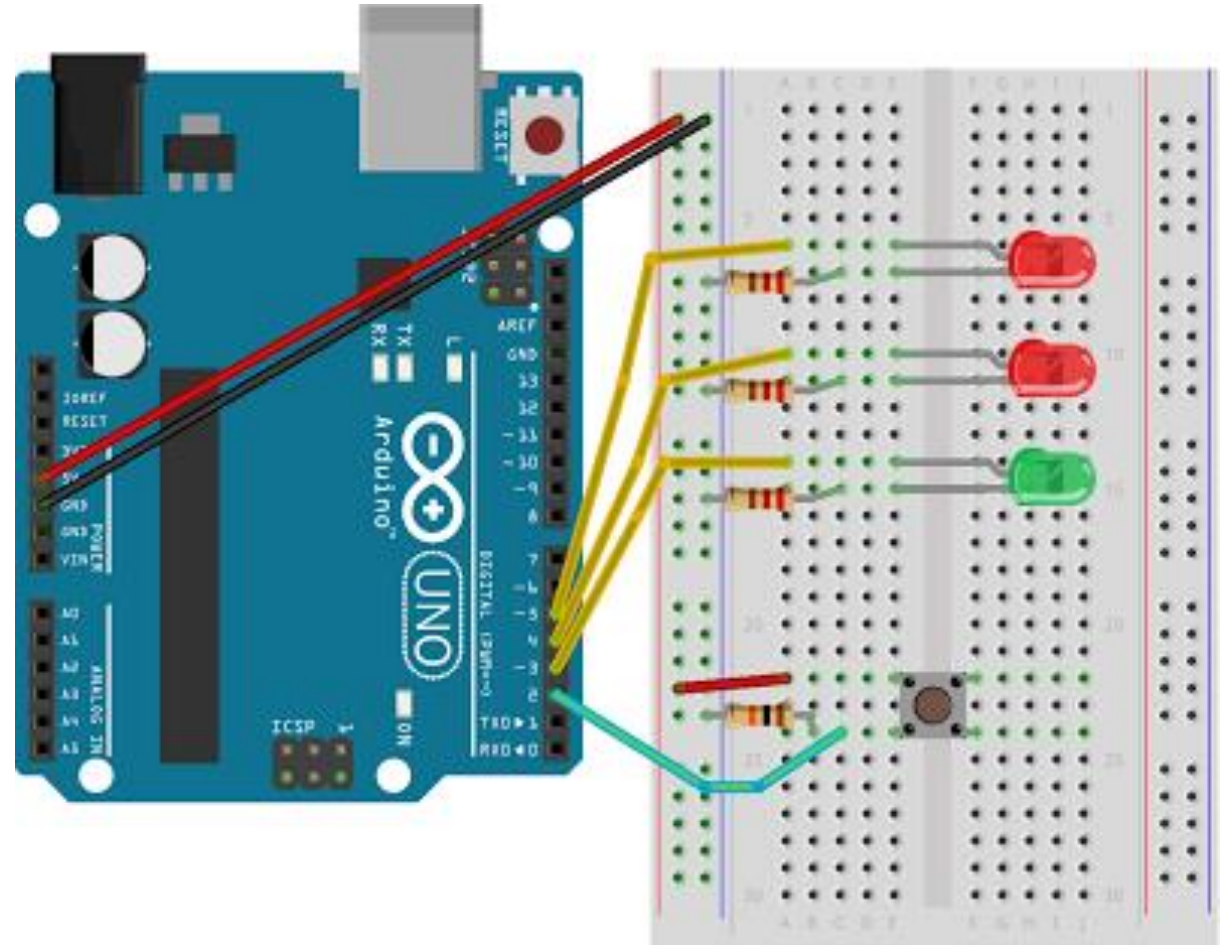
Existen multitud de diferentes sensores de temperatura. Este modelo concreto, el TMP36, nos viene de perlas porque proporciona un voltaje de salida cuya variación es directamente proporcional a la temperatura en grados Celsius.

Montando el circuito

1. Conecta la alimentación de 5v y la tierra de la placa Arduino a la protoboard.
2. Colocamos un led verde y dos rojos en dicha protoboard.
3. Cogemos una resistencia de 220Ω y conectamos uno de sus terminales al cátodo del led verde (la patilla corta) y el otro, lo conectamos a masa. Cogemos otras dos resistencias de 220Ω y repetimos el proceso para los otros dos leds.
4. A continuación, realizamos una conexión entre el ánodo del led verde (la patilla larga) y el pin 3 del Arduino.
5. Los ánodos de los leds rojos, los conectamos uno al pin 4 y el otro al pin 5.
6. Colocamos un pulsador en la protoboard, tal y como se muestra en la figura de más abajo. Conectamos un lado del pulsador a la alimentación y el otro lado al pin 2 del Arduino. Necesitaremos añadir una resistencia de $10k\Omega$ entre la tierra y la conexión del pulsador con el pin del Arduino.

Montando el circuito

Componentes	
Pulsador	1
LEDs	3
Resistencia 220 Ω	3
Resistencia 10 K Ω	1



Comentario del código

- Se declara una variable con datos **int** (*integer*, número entero); cualquier número sin coma decimal.
- La función **setup()** se ejecuta sólo una vez al encender Arduino. Aquí se configuran los pines digitales como entradas o salidas utilizando la función **pinMode**.
- La función **loop()** se ejecuta continuamente tras la finalización del **setup()**. Para comprobar el nivel de voltaje de una entrada digital, utilizamos la función **digitalRead(númeroDePin)** que comprueba el pin escogido para la tensión.
- La condición **if()** compara dos cosas o más y determina si la comparación es verdadera o falsa. Cuando se comparan dos cosas en programación, se utilizan los dos signos de igual **==**.

Comentario del código

- **digitalWrite**(*númeroDePin*, LOW/HIGH) es el comando que permite enviar 5V o 0V a una salida.
- La condición **if()** tiene una función opcional **else()** que permite que algo suceda si no se cumple la condición original.
- La función **delay**(*númeroDeMilisegundos*) permite detener Arduino para que no ejecute nada durante un periodo de tiempo determinado.