

MÓDULO 4:

PROYECTO 3 "Medidor de amor"

CURSO PROGRAMACIÓN DE PLACAS ROBÓTICAS

A solid orange horizontal bar at the bottom of the slide.

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.

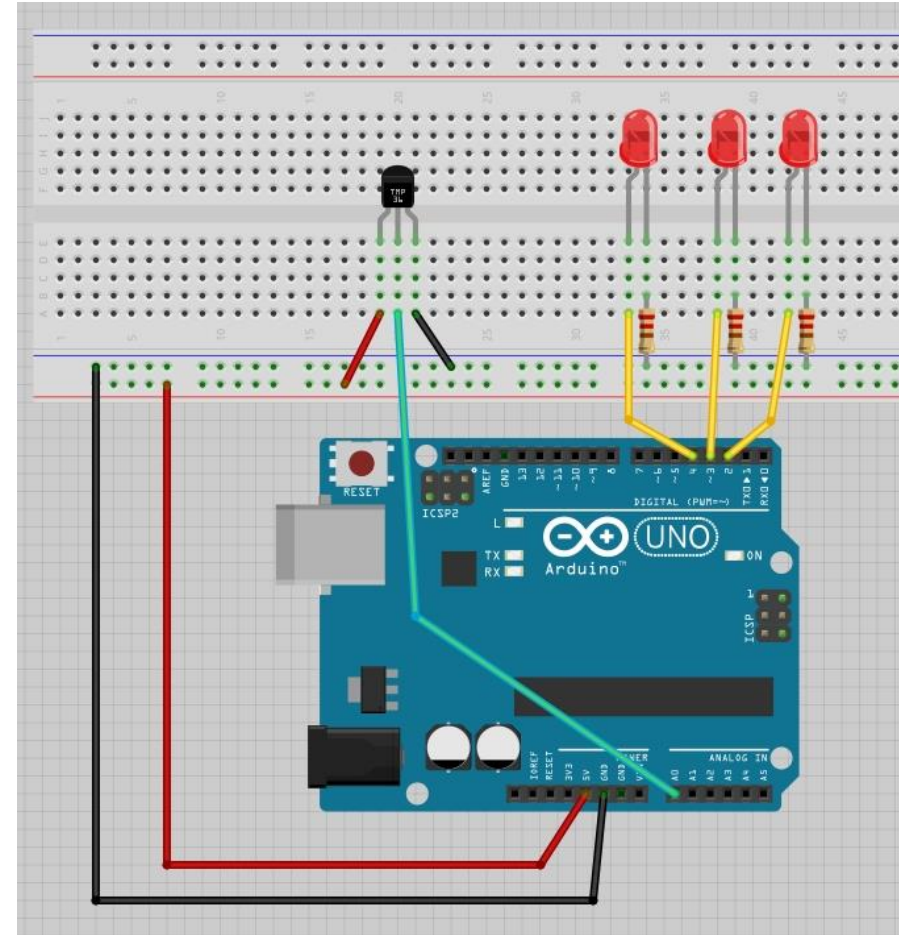
Existen multitud de diferentes sensores de temperatura. Este modelo concreto, el TMP36, nos viene de perlas porque proporciona un voltaje de salida cuya variación es directamente proporcional a la temperatura en grados Celsius.

Montando el circuito

1. Tal y como hemos hecho en los proyectos anteriores, conectamos la alimentación y la tierra en nuestra protoboard.
2. Conectamos el cátodo de cada led (la patilla corta) a tierra utilizando una resistencia de 220Ω . Los ánodos de cada led irán conectados a los pins del 2 al 4 del Arduino.
3. Colocamos en el sensor TMP36 en la protoboard con la cara redondeada mirando en dirección contraria a nuestro Arduino (el orden de los pins del sensor es importante) tal y como se muestra en la figura de más abajo.
4. Seguidamente, conectaremos el pin izquierdo del sensor (mirando el sensor por su lado plano) a la alimentación y el pin derecho a tierra.
5. A continuación, el pin central irá conectado al pin A0 del Arduino, el cual, es el pin 0 de entrada analógica.

Montando el circuito

Componentes	
sensor de temperatura TMP36	1
LEDs	3
Resistencia 220 Ω	3



Comentario del código

- Se declara la constante `const` de la entrada analógica para tener una fácil referencia y se crea otra para mantener la temperatura base.
- En la función **`setup()`** se abre una conexión entre Arduino y el ordenador a través del puerto serie con **`Serial.begin(9600)`**. El argumento (9600) indica la velocidad con la que Arduino se comunicará, 9600 bits por segundo.
- El bucle **`for()`** establece algunos pines como salida. El método que utiliza es pasar desde el pin 2 hasta el 4 en secuencias. Este es un truco muy útil si tienes un gran número de cosas similares que necesitan ser ejecutadas en un programa.
- En la función **`loop()`** se utiliza una variable `sensorVal` para almacenar la lectura del sensor. Para conseguir el valor del sensor se utiliza `analogRead(pin)`. El valor entre 0 y 1023 no es más que la representación del voltaje.

Comentario del código

- La función **Serial.print()** envía información al monitor serie del ordenador. Si el argumento está entre comillas, se imprimirá en la salida el texto introducido.
- Con un poco de matemáticas es posible calcular el voltaje real del pin. Si dividimos `sensorVal` entre 1024.0 y lo multiplicamos por 5.0 obtenemos el resultado y lo almacenamos en una variable. Luego, se imprime la información en el monitor serie.
- Si al voltaje le restamos 0.5 y lo multiplicamos por 100, conseguiremos la temperatura en grados ([arduino.cc/kitsdatasheets](https://www.arduino.cc/kitsdatasheets)).
- Se configuran una serie de condiciones `if()` y `else if` para encender los LEDs. Si la temperatura es más baja que la temperatura base no se encenderá ningún LED.

Comentario del código

- El operador `&&` significa "y" en un sentido lógico. Si la temperatura es de 2 grados más que la temperatura base y a la vez es menos de 4 grados más que la temperatura base, se encenderá un LED. Igualmente, las siguientes secuencias encienden o apagan LEDs dependiendo de los cambios de temperatura.
- La función `delay()` realiza una pequeña pausa al final del bucle del programa, para dar tiempo a la lectura del sensor y ver la información más clara.