



CONTROL DE TEMPERATURA AMBIENTE

Tecnología de Computadores



INGENIERÍA INFORMÁTICA
DANIEL LINFON YE LIU

49548754S

Introducción

En mi caso he elegido trabajar el proyecto 1 que consiste en un regulador de temperatura ambiental para verano e invierno con el siguiente intervalo de temperatura:

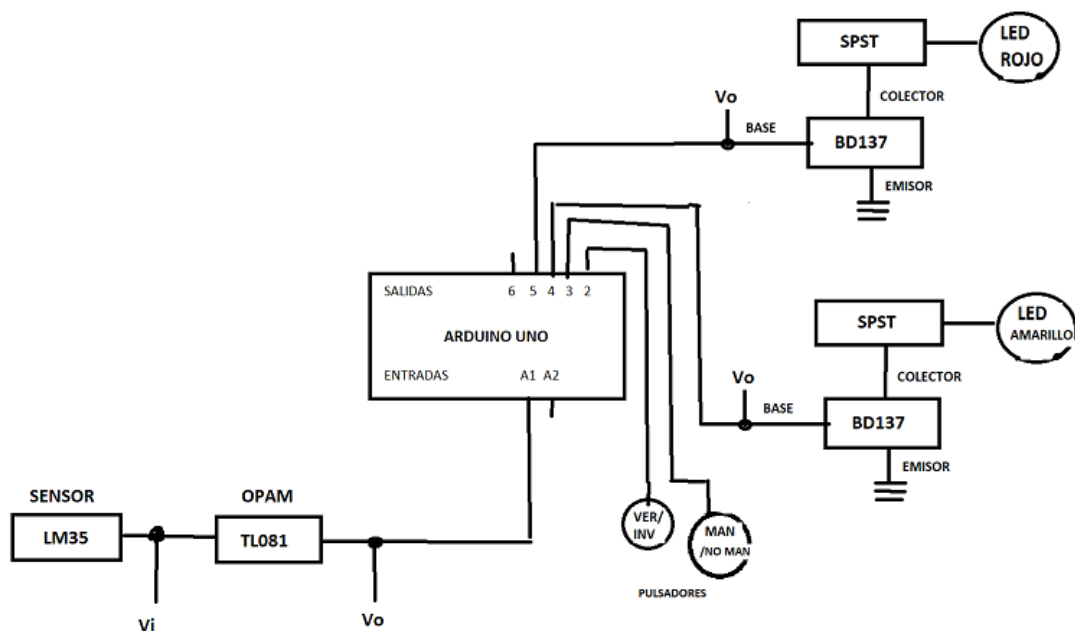
	T_{min}	T_{max}
Verano	15°C	22°C
Invierno	20°C	27°C

Para ello utilizaremos un sensor LM35 para medir la temperatura y un pulsador que intercambiara entre ambas estaciones. Si la temperatura es menor que la mínima, se activará un led rojo y si es mayor que la máxima, un led amarillo.

Posteriormente se ha implementado la capacidad de modificar los intervalos de temperaturas mínimas y máximas.

Solución propuesta

Para empezar, hice un croquis de lo que sería el diseño del circuito. Utilizaremos la placa de Arduino UNO que recibiría como entrada la equivalencia de la temperatura en voltios haciendo uso del sensor LM35 y un OPAM. Como salidas, tendríamos dos circuitos cuya única deferencia es el color del led. Estos circuitos estarán formados por un transistor que recibirá como base el voltaje obtenido en la entrada. En su colector colocaremos un relé con su respectivo led para controlar cuando debe encenderse o no. Además, se añadirán dos pulsadores: uno servirá para cambiar de verano a invierno y viceversa y el otro será para cambiar entre el modo manual (para establecer los intervalos de temperatura) y no manual. El esquema sería el siguiente:



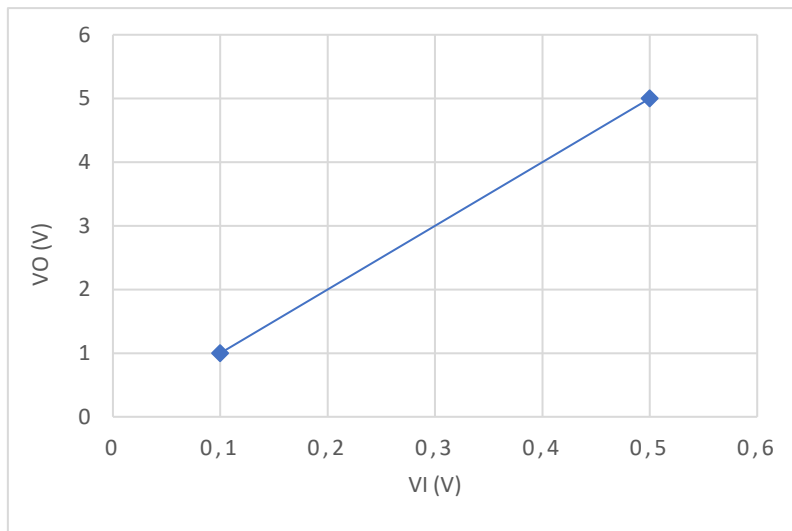
ENTRADA

Diseñaremos el circuito con el sensor de temperatura LM35 y un OPAM TL081. Para hacer los cálculos usaremos el método de las 3 columnas:

$$LM35 \rightarrow \frac{C^{\circ}}{100} = V_{Sensor} = V_i$$

Temperatura(C°)	Vi (V)	Vo (V)
10	0.1	1
50	0.5	5

Si trazamos la gráfica:



La ecuación de la recta es:

$$y = mx + b$$

Que adaptada a nuestro problema:

$$V_o = m * V_i + b$$

Donde podemos obtener la pendiente m :

$$m = \frac{5 - 1}{0.5 - 0.1}$$

$$m = 10$$

Sustituimos valores para obtener b :

$$V_o = m * V_i + b \rightarrow 1 = 10 * 0.1 + b \rightarrow b = 1 - 1 \rightarrow b = 0$$

Una vez obtenidos estos valores calculamos las K 's:

$$\begin{aligned} K_1^+ &= 10 \\ K_1^- &= 0 \\ K_0^+ &= 0 \\ K_0^- &= 9 \end{aligned}$$

$$1 + K_0^- + K_1^- = K_0^+ + K_1^+$$

$$K_0^- = 10 - 1$$

$$K_0^- = 9$$

Ahora podemos conseguir las resistencias del circuito:

Tomaremos $R_f = 18K\Omega$

$$R_0^- = \frac{R_f}{K_0^-} = \frac{18}{9} = 2K\Omega$$

$$R_1^+ = \frac{R_f}{K_1^+} = \frac{18}{10} = 1.8K\Omega$$

$$R_0^+ = 0$$

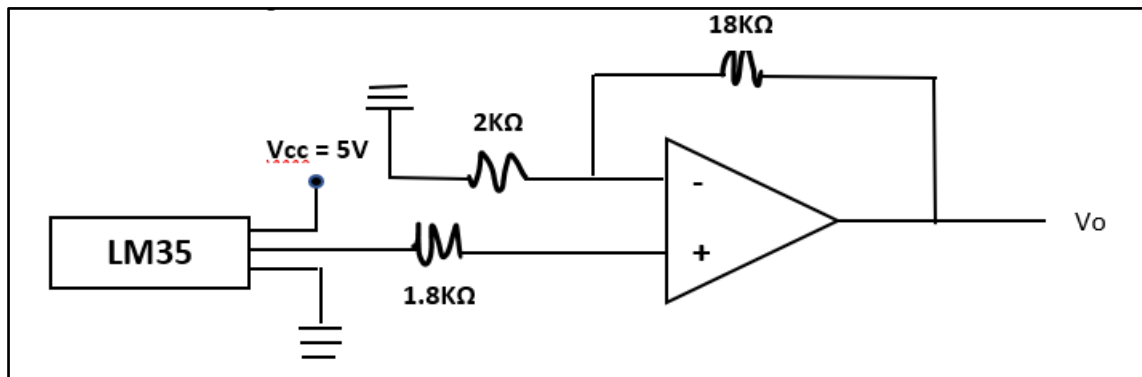
$$R_1^- = 0$$

$$R_f = 18K\Omega$$

$$R_0^- = 2K\Omega$$

$$R_1^+ = 1.8K\Omega$$

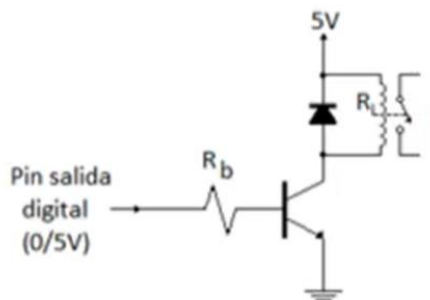
El circuito de entrada quedaría finalmente así:



SALIDAS

Como hemos mencionado antes, se trata de elaborar dos circuitos, uno para cuando la temperatura es menor que la mínima establecida y otro para cuando sea superior que la máxima. Ambos circuitos serán iguales excepto el diodo led. Para diseñar este circuito usaremos un transistor BD137, un relé SPST y un diodo led rojo o amarillo.

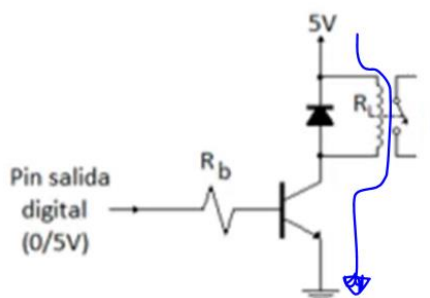
Primero calcularemos las resistencias que debemos implementar:



Según las características del relé y el transistor obtenemos que:

$$R_L = 125k\Omega$$

$$HFE_{min} = \beta = 40$$



De la malla colector – emisor calculamos la I_c

$$I_c = \frac{5V - 0.2V}{125k\Omega} \rightarrow \boxed{I_c = 38.4mA}$$

El transistor, al trabajar en zona de corte tenemos que:

$$I_B = \frac{I_c}{\beta} = \frac{38.4mA}{40} \rightarrow \boxed{I_B = 0.96}$$

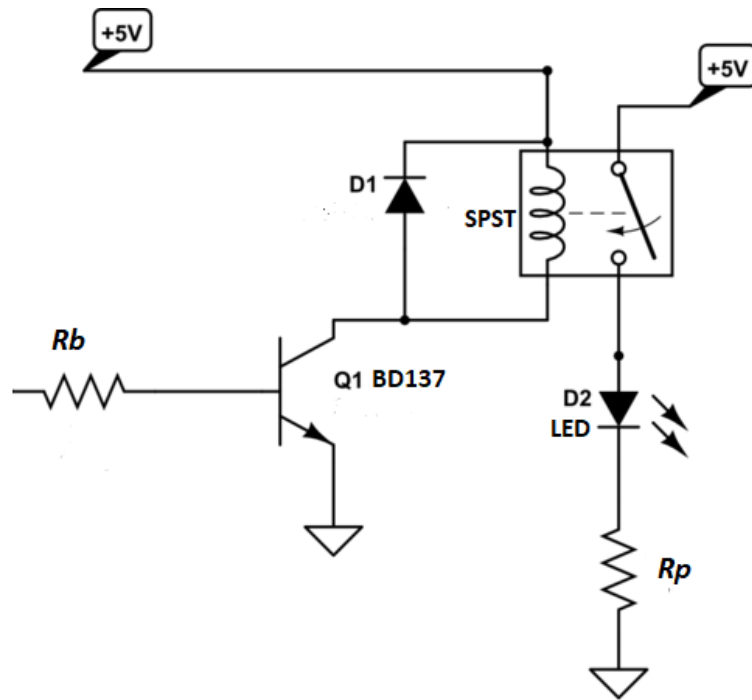
$$R_B = \frac{5V - 0.7}{0.96} \rightarrow \boxed{R_B = 4.4K\Omega}$$

Finalmente nos quedaría calcular una resistencia protectora para los leds.

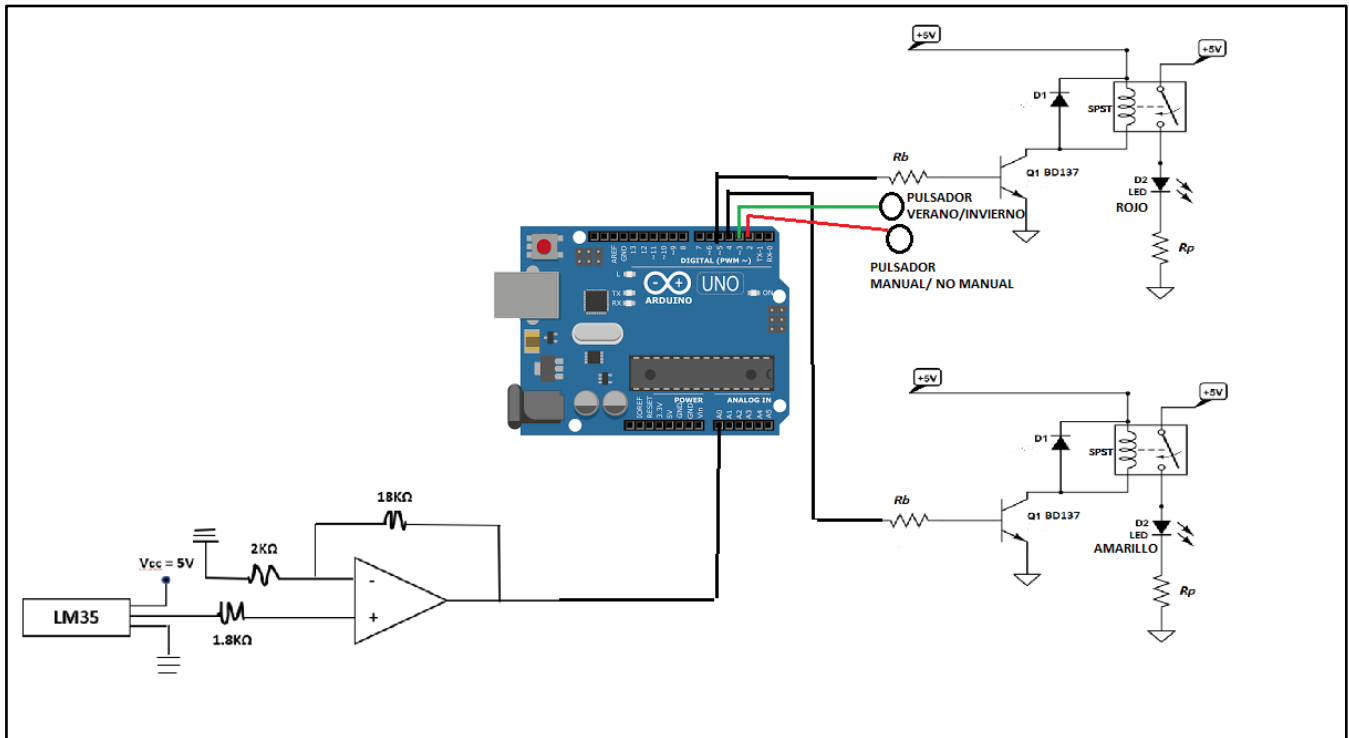
Led Rojo y Amarillo: 2.2V para 20mA

$$R_p = \frac{5V - 2.2V}{20mA} \rightarrow \boxed{R_p = 140K\Omega}$$

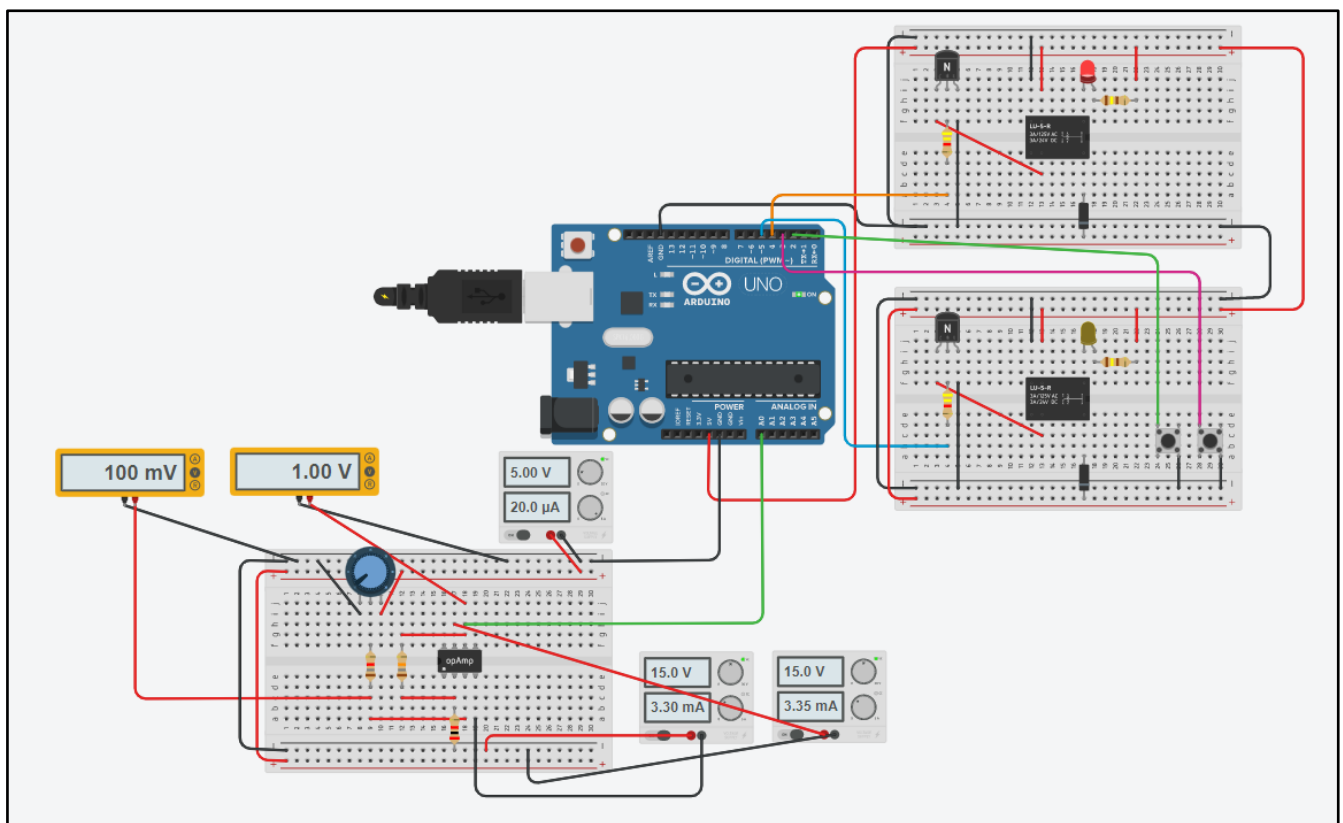
El circuito de salida quedaría así:



DISEÑO FINAL

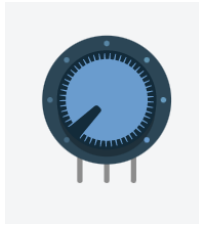


Para realizar la simulación hemos utilizado el programa online, Tinkercad. El diseño en dicho programa es el siguiente:



COMPONENTES ADICIONALES

Al tratar de construir el circuito en un simulador, carecemos de algunos componentes, por lo que los hemos sustituido por otros que cumplen la misma función:

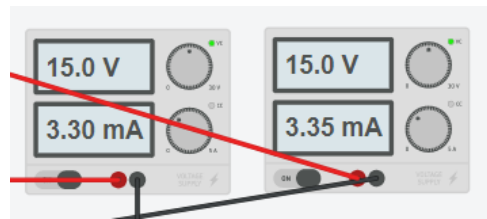


Potenciómetro. Realiza la función del sensor LM35, suministrando la cantidad de voltaje deseado.

Voltímetro. Nos ayuda a comprobar la tensión que pasa por algún punto del circuito. En nuestro caso usamos dos, uno para comprobar la tensión del potenciómetro y otro para comprobar la tensión que sale por el OPAM.



Fuentes de alimentación. Hemos utilizado dos para suministrar $\pm 15V$ al OPAM y pueda funcionar.



FUNCIONAMIENTO DEL CIRCUITO

Para programar el Arduino hemos utilizado lenguaje C++ con los siguientes métodos:

- `void cambiar()`. Este método va asociada al pulsador que permite cambiar entre las estaciones.
- `void cambiar2()`. Este método va asociada al pulsador que permite cambiar entre el modo manual y no manual.
- `float convertir(long valor)`. Convierte en entero el integer recibido por parámetro, es decir, devuelve °C.
- `void controlar_temp(int valor)`. Es la función que nos permite controlar que se enciendan los leds cuando se den las condiciones.
Para plantear las condiciones usamos sentencias if para comparar el valor recibido por parámetro con los límites de temperatura (en integer).

Las funciones principales son:

```
void setup()
{
  pinMode(2, INPUT_PULLUP);
  pinMode(3, INPUT_PULLUP);
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);
  attachInterrupt(digitalPinToInterrupt(2), cambiar, FALLING);
  attachInterrupt(digitalPinToInterrupt(3), cambiar2, FALLING);
  Serial.begin(9600);
}

void loop()
{
  temp = analogRead(Amplificador);

  controlar_temp(temp);

  if(millis() >= t_mensaje + 5000)
  {
    t_mensaje = millis();
    temp_float = convertir(temp);
    Serial.print("Temperatura actual: ");
    Serial.print(temp_float);
    Serial.println(" grados Celsius\n");
  }

  while(millis() < t_inicial + T)
  {
  }
}
```

En la primera se declaran las entradas y salidas de la placas así como los pulsadores.

En la función principal, primero se le asigna a la variable `temp` el valor que sale por el OPAM (en integer). Luego se llama al `controlar_temp` para ver que led encender.

Finalmente con la función `convertir` obtenemos la temperatura para mostrarlo por el monitor en serie.

CONCLUSIÓN

El circuito cumple los objetivos del proyecto. Mediante el potenciómetro elegimos la temperatura que deseamos, equivalente a V. En base a la temperatura que elijamos, se encenderá o no algún led si esta se sale de los límites establecidos. De manera predeterminada estamos en verano. Si presionamos el pulsador de verano/invierno, cambiaremos a invierno y los límites cambian.

Adicionalmente, se ha implementado la posibilidad de cambiar los límites de temperatura. Si pulsamos el pulsador correspondiente, se nos pedirá por el monitor en serie que introduzcamos la nueva temperatura máxima y mínima. Una vez establecidos los nuevos límites, Si apretamos nuevamente el pulsador, volveríamos al modo predeterminado.

Como posible mejora, se podría implementar la posibilidad de una vez en el modo manual, establecer unos límites para verano y otros para invierno. De modo que si apretamos el pulsador de verano/invierno (estando en modo manual), introduciríamos un nuevo intervalo para invierno. Así, tendríamos un verano e invierno editado. Para cambiar de nuevo estos límites, habría que volver al modo no manual y volver a pulsar el pulsador para activar el modo manual.

REFERENCIAS

[Enlace al proyecto de Tinkercad](#)