

TP_6

CONTROL DE TEMPERATURA MEDIANTE SENSOR TMP36

AUTOR: AHMED CHTATOU BOUGDADI

1. INTRODUCCIÓN.

Para esta práctica debemos montar un circuito de control de temperatura, consiste en apagar y encenderlo en el lugar donde precisemos controlar la temperatura para refrigeración o calefacción automáticamente de dicho lugar.

2. MATERIALES UTILIZADOS.

- Placa ARDUINO UNO
- Placa Board
- Sensor de temperatura TMP36
- Zumbados
- Pulsador
- Resistencias de 10k y 220k
- Diodos led (verde y rojo)
- Cables de conexión
- Software por ordenador ARDUINO
- Software por ordenador Fritzing

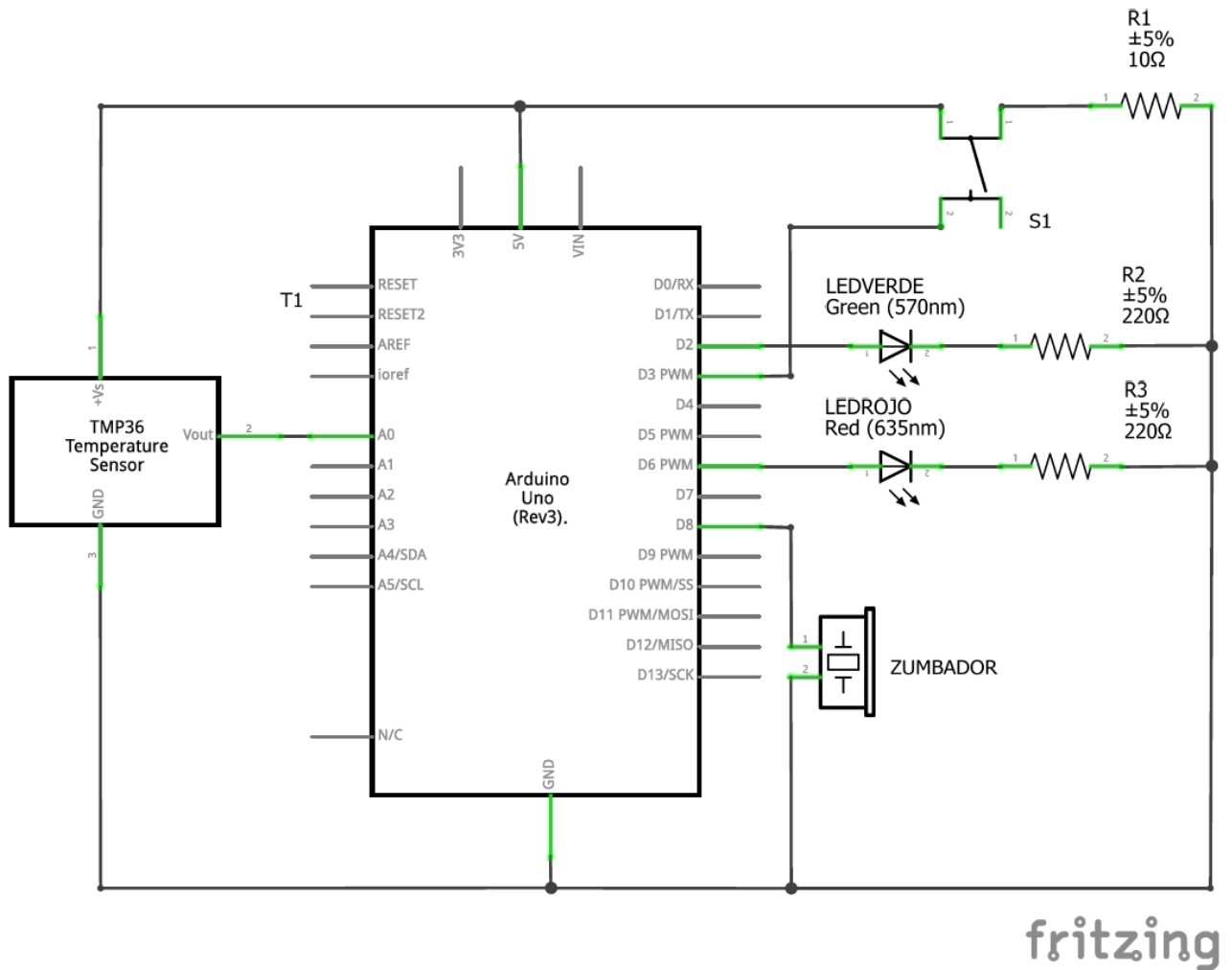
3. FUNCIONAMIENTO

Haremos uso de la placa ARDUINO para controlar este sistema, usaremos diodos led para indicar cuando se activa la calefacción y la refrigeración, un zumbador que hará la función de alarma para cuando se alcance la temperatura máxima (40°C), un pulsador para poder desactivarla, y el sensor de temperatura el cual nos indicara los datos para que el sistema actúe

TEMPERATURA	RESPUESTA DEL SISTEMA
T > 40°C	Activar alarma y mantener la refrigeración funcionando. Poder apagar la alarma y mantenerla apagada mientras la temperatura sea mayor que 40°C
T ≤ 40°C	<p>$T > T_r + D = 30 + 0,5 = 30,5$, teniendo en cuenta la histéresis del sensor de temperatura. Donde 30° es la T° a la cual el refrigerador se activa</p> <hr/> <p>$T_{min} < T < T_r - D$, cuando la temperatura está comprendida entre una mínima de 20° y de 29,5° se apague la refrigeración.</p> <hr/> <p>$T < T_{min}$, cuando la temperatura sea inferior a 20°C, se active la calefacción.</p> <hr/> <p>Para los demás casos que se produzca una espera de 100ms y que el sistema vuelva a leer el sensor de temperatura.</p>

4. ESQUEMA ELECTRÓNICO.

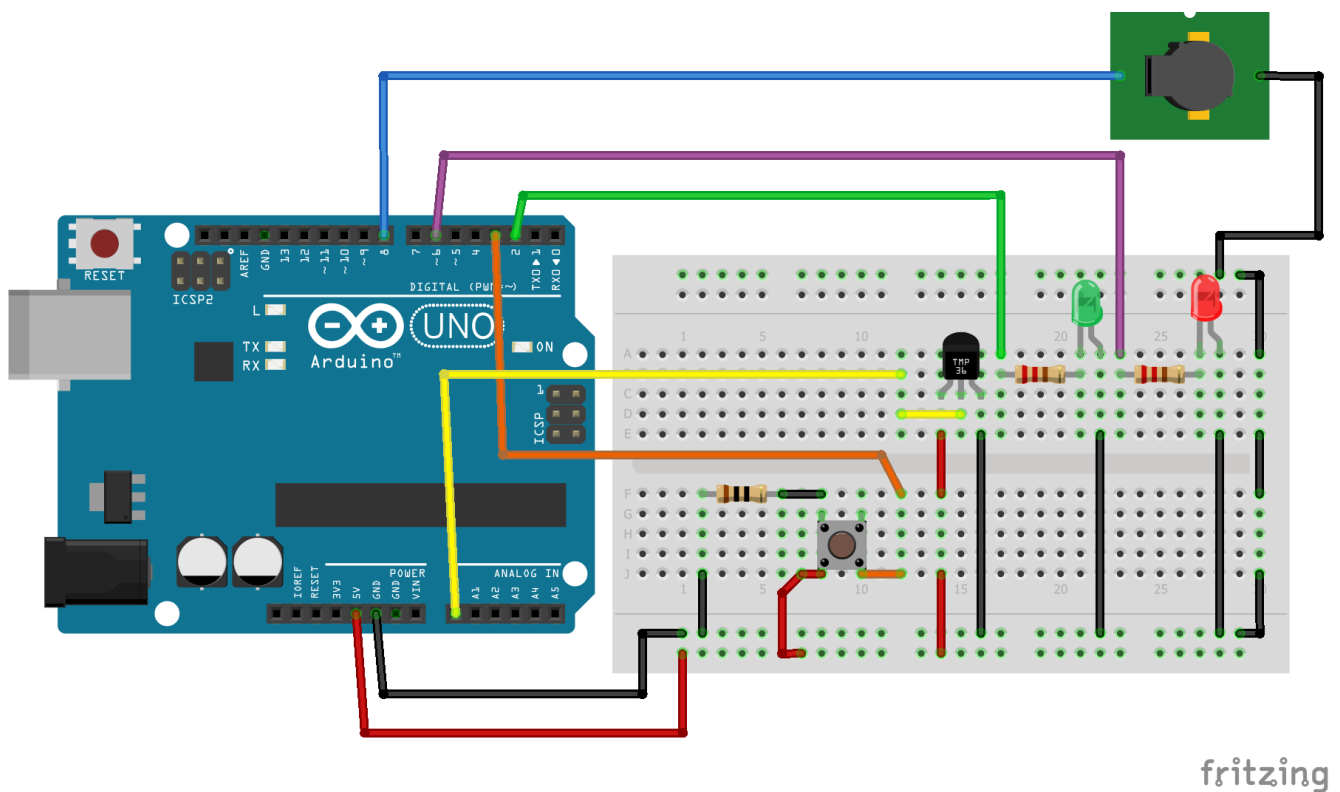
Para realizar el esquema eléctrico haremos uso de fritzing ya que nos lo ofrece hacerlo de una manera muy sencilla y gráfica.



5. MONTAJE Y DISEÑO EN FRITZING

- Para tener una idea de cómo montar nuestro circuito lo que haremos es utilizar la herramienta por software fritzing la cual nos permite hacer el diseño de nuestro circuito

En las imágenes veremos el montaje del circuito en una protoboard tanto como físico y esquemáticamente.



En la imagen podemos observar todas las conexiones del circuito las cuales se resumen en lo siguiente.

Una salida analógica que ira a la patilla central del sensor de temperatura (Cable amarillo).

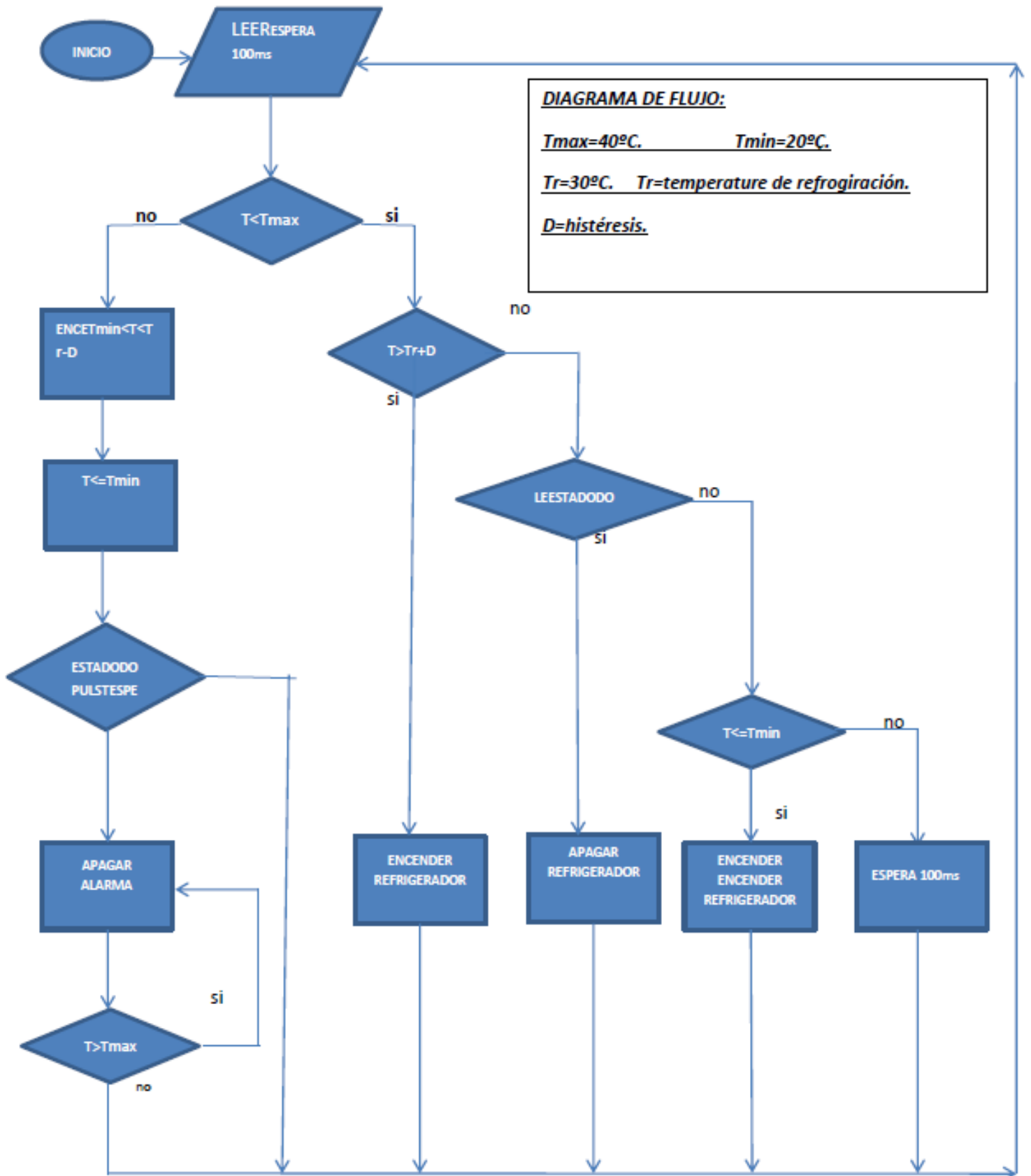
La salida del Pin 6 va conectada al led verde en la patilla del positivo mediante una resistencia de 220k, y la otra patilla directamente a tierra (GND)

La salida del Pin 2 va conectada al led rojo en la patilla del positivo mediante una resistencia de 220k y la otra patilla directamente a tierra (GND)

En la salida del Pin 8 conectaremos nuestro zumbador que es el que hará la función de alarma y nos indicará el cambio que realizará el sistema en ese momento.

Por ultimo vemos las tomas de alimentación +5V y tierra (GND), que van conectadas en la parte inferior de la protoboard eso es esencial ya que desde este punto podemos tomar cuando necesitemos hacer una toma algún dispositivo.

6. FLUJO GRAMA



7. PSEUDOCÓDIGO

Leer temperatura

Si $T < 40^{\circ}\text{C}$

Si la temperatura es mayor que temperatura de refrigeración más histéresis

Entonces: Activar refrigeración.

Apagar la calefacción

Sino y si la temperatura esta entre la temperatura mínima y la de refrigeración menos histéresis:

Entonces: Apagar refrigeración.

Sino y si la temperatura es menor que la temperatura mínima

Entonces: Activar calefacción

Sino

Entonces: espera y vuelva a leer la temperatura

Sino (temperatura mayor o igual que 40°)

Leer pulsador

Si (pulsador on)

Entonces: desactivar alarma Activar la refrigeración

Mantener alarma apagada mientras la temperatura sea menor mayor o igual que 40°

=Sino vuelve a leer temperatura.

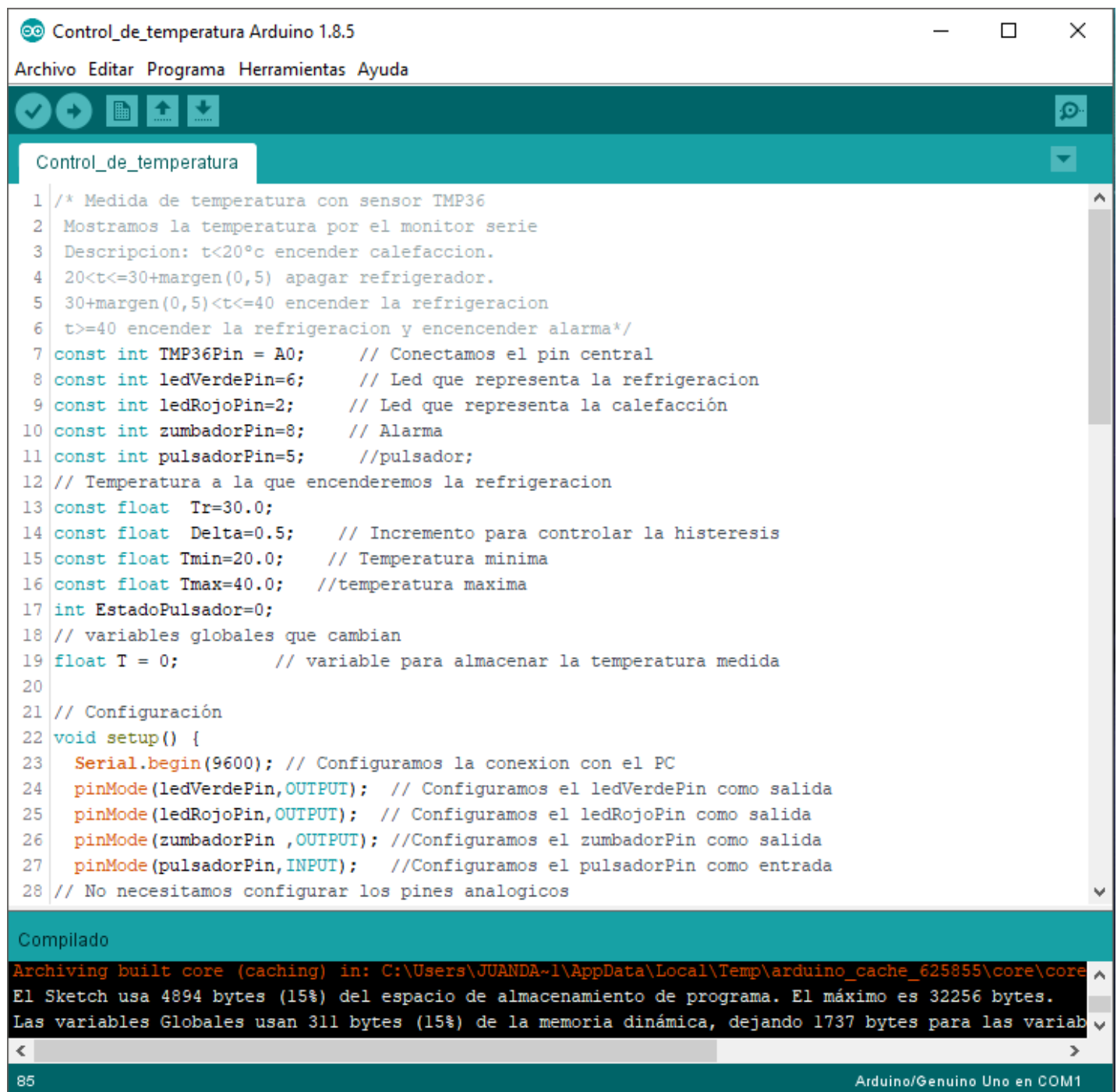
$T_{\text{max}}=40^{\circ}$. Histéresis= 0.5°

$T_{\text{min}}=20^{\circ}$. $T_r=30^{\circ}$.

Se observa en este pseudocódigo que hay dos variables, en fi.

8. CÓDIGO

Código compilado a través del software de arduino.



```
Control_de_temperatura Arduino 1.8.5
Archivo Editar Programa Herramientas Ayuda
Control_de_temperatura
1 /* Medida de temperatura con sensor TMP36
2 Mostramos la temperatura por el monitor serie
3 Descripcion: t<20°c encender calefaccion.
4 20<t<=30+margen(0,5) apagar refrigerador.
5 30+margen(0,5)<t<=40 encender la refrigeracion
6 t>=40 encender la refrigeracion y encencender alarma*/
7 const int TMP36Pin = A0; // Conectamos el pin central
8 const int ledVerdePin=6; // Led que representa la refrigeracion
9 const int ledRojoPin=2; // Led que representa la calefacción
10 const int zumbadorPin=8; // Alarma
11 const int pulsadorPin=5; //pulsador;
12 // Temperatura a la que encenderemos la refrigeracion
13 const float Tr=30.0;
14 const float Delta=0.5; // Incremento para controlar la histeresis
15 const float Tmin=20.0; // Temperatura minima
16 const float Tmax=40.0; //temperatura maxima
17 int EstadoPulsador=0;
18 // variables globales que cambian
19 float T = 0; // variable para almacenar la temperatura medida
20
21 // Configuración
22 void setup() {
23 Serial.begin(9600); // Configuramos la conexion con el PC
24 pinMode(ledVerdePin,OUTPUT); // Configuramos el ledVerdePin como salida
25 pinMode(ledRojoPin,OUTPUT); // Configuramos el ledRojoPin como salida
26 pinMode(zumbadorPin ,OUTPUT); //Configuramos el zumbadorPin como salida
27 pinMode(pulsadorPin,INPUT); //Configuramos el pulsadorPin como entrada
28 // No necesitamos configurar los pines analogicos
Compilado
Archiving built core (caching) in: C:\Users\JUANDA-1\AppData\Local\Temp\arduino_cache_625855\core\core
El Sketch usa 4894 bytes (15%) del espacio de almacenamiento de programa. El máximo es 32256 bytes.
Las variables Globales usan 311 bytes (15%) de la memoria dinámica, dejando 1737 bytes para las variab
85 Arduino/Genuino Uno en COM1
```

```

/* Medida de temperatura con sensor TMP36
Mostramos la temperatura por el monitor serie
Descripcion: t<20°C encender calefaccion.
20<t<=30+margen(0,5) apagar refrigerador.
30+margen(0,5)<t<=40 encender la refrigeracion
t>=40 encender la refrigeracion y encender alarma*/
const int TMP36Pin = A0;      // Conectamos el pin central
const int ledVerdePin=6;     // Led que representa la refrigeracion
const int ledRojoPin=2;      // Led que representa la calefacción
const int zumbadorPin=8;     // Alarma
const int pulsadorPin=5;     //pulsador;
// Temperatura a la que encenderemos la refrigeracion
const float Tr=30.0;
const float Delta=0.5;      // Incremento para controlar la histeresis
const float Tmin=20.0;     // Temperatura minima
const float Tmax=40.0;     //temperatura maxima
int EstadoPulsador=0;
// variables globales que cambian
float T = 0;                // variable para almacenar la temperatura medida

// Configuración
void setup() {
  Serial.begin(9600); // Configuramos la conexión con el PC
  pinMode(ledVerdePin,OUTPUT); // Configuramos el ledVerdePin como
salida
  pinMode(ledRojoPin,OUTPUT); // Configuramos el ledRojoPin como
salida
  pinMode(zumbadorPin ,OUTPUT); //Configuramos el zumbadorPin como
salida
  pinMode(pulsadorPin,INPUT); //Configuramos el pulsadorPin como
entrada
// No necesitamos configurar los pines analogicos
}

// Bucle que se repite
void loop() {
  int lectura = analogRead(TMP36Pin); // Valor entre 0 y 1023

  float voltaje = lectura * 5.0 / 1024.0; // Voltaje entre 0 y 5V

  T = (voltaje - 0.5) * 100; // Temperatura en ° Celsius según
dataheet del fabricante

  Serial.print("Temperatura: \t"); // Mostramos la palabra
temperatura
  Serial.print(T); // Mostramos el valor de la temperatura
  Serial.println("C");

  if (T<Tmax){ //Aquí se determinara el comportamiento de
los leds //segun el valor de la temperatura asignado*/

  if(T>=Tr+Delta){
    digitalWrite(ledVerdePin,HIGH);
    Serial.println("Encendemos Refrigerador");
    digitalWrite(ledRojoPin,LOW);
  }
  else if((Tmin<T)&&(T<=Tr-Delta)){

```

```

    digitalWrite(ledVerdePin, LOW);
    digitalWrite(ledRojoPin, LOW);
    Serial.println("Apagamos Refrigerador");
  }
  else if (T <= Tmin) {
    digitalWrite(ledRojoPin, HIGH);
    Serial.println("Encendemos calefactor");
    digitalWrite(ledVerdePin, LOW);
  }
  else {
delay(200);
  }
}
else {
  digitalWrite(ledVerdePin, HIGH);
  Serial.println("Encendemos Refrigerador");
  digitalWrite(zumbadorPin, HIGH);
  tone(zumbadorPin, 500);

delay(100);

}
  int EstadoPulsador = digitalRead(pulsadorPin);
  delay(50);

// si el pulsador esta presionada apaga la alarma mientras la
// temperatura
// sea mayor que 40°C
if (EstadoPulsador == 1) {
  do {
    digitalWrite(zumbadorPin, LOW);
    noTone(zumbadorPin);
  } while (T > Tmax);
}
}
}

```


Como podemos observar el circuito que hemos realizado funciona correctamente.

Como podemos observar en la pantalla monitor nos marca el proceso que está ejecutando nuestro sistema de control de temperatura.