D)- Como implementar un sensor inteligente de temperatura.

En un sistema de ambientación automático, donde se busca controlar varios elementos electrónicos en una habitación como; ventiladores, aire acondicionado calefacción etc. Mediante un sensor de temperatura que es el que nos permite saber la temperatura de la habitación, clasificamos 3 estados; frio, ambiente caliente y así poder dar los parámetros de preferencia, es decir, a mi gusto 20°c es frio, 25°c es normal y 30° es caliente, dando estos parámetros de ejemplo podemos decir que cuando la temperatura sea 30° (caliente) entonces automáticamente se enciende el aire acondicionado y así definirlo de acuerdo a nuestras necesidades.

Materiales:

Lista de componentes

- 1 Arduino.
- 1 Servemotor.
- 1 LED RGb.
- 1 Modulo rele.
- 1 sensor de temperatura lm35.
- 3 Resistencia 220 ohm.
- 1 Resistencia 2.2k ohm.
- 1 Buzzer pasivo.
- 1 Motor dc o un ventilador de computadora.
- 1 Cargador de celular 5v.
- Display LCD 16x2.

Sensor de temperatura lm35

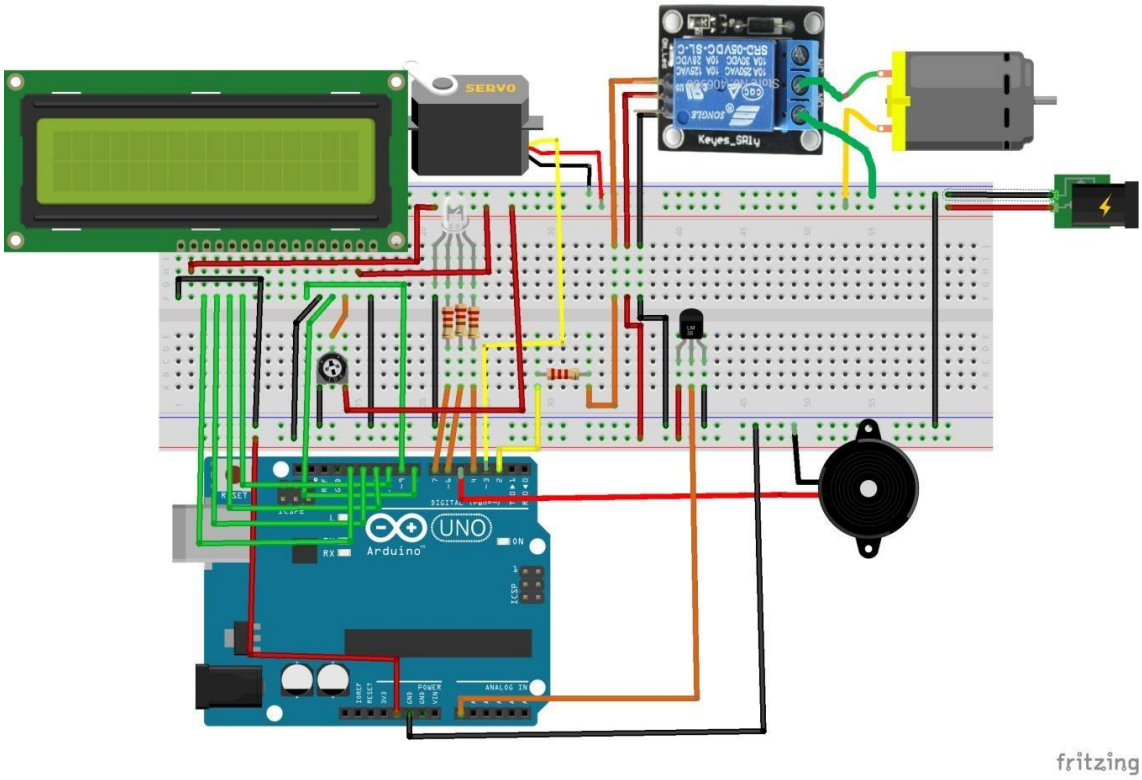
El sensor LM35, es un sensor de temperatura integrado de precisión cuyos rangos de operación oscilan desde los -55°C hasta los 150°C, teniendo en cuenta de este tipo de sensores ofrecen en una precisión de +- 1.4°C a temperatura ambiente. Además de ellos, son del tipo lineal; es decir, que no es necesario forzar al usuario a realizar conversaciones debido a que otros sensores están en grados kelvin.

Un LM35 puede funcionar a partir de los 5v (corriente continua), sea por alimentación simple o por doble alimentación (+/-). Sus características mas importantes se describen a continuación:

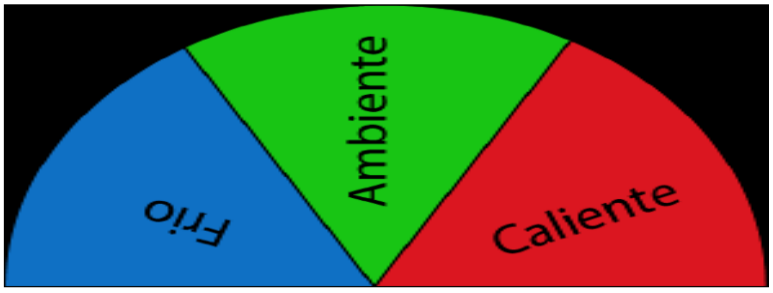
- Configuración para ser leído en grados Celsius.
- Factor de escala lineal de +10mv/°C.
- Rango de trabajo entre -55 °C hasta 150 °C.
- Apropiado para aplicaciones remotas.
- Bajo Costo.
- Funciona con tensiones entre 4V hasta 30V.
- Baja impedancia de salida, 0.1W, para cargas de 1mV.

Por lo tanto, este tipo de sensores son los mas usados en la practica debido a su reducido costo y escasa opciones de mantenimientos, lo que hace un elemento viable en la mayoría de sus aplicaciones.

Conexión.



En este proyecto la dificultad de conexión es mayor, ya que usaremos muchos componentes de entrada y salida, por lo que es necesario organizar la conexión de la mejor manera, limpia y ordenada, para no equivocarnos, y debemos fijarnos muy bien en como conectamos cada componente, ya que la mayoría de ellos son polarizados. Se aclara en este proyecto se debe usar fuente externa para poder alimentar cada dispositivo, por lo que podemos conectar una fuente a arduino, o alimentarlo con una batería u otra fuente fuera del circuito de arduino. En mi caso utilizare un cargador de celular OUTPUT: 5V, 0.75A



Antes de comenzar el código debemos analizar lo que haremos.

Dividiremos la función del Smart sensor en tres etapas, frio, ambiente y caliente. Y le asignaremos a cada una su indicador visual y auditivo, la lcd imprimirá el valor de la temperatura en tiempo real siempre.

Frio: De los 30°C hacia a abajo el led encenderá Azul, el servo se posicionará en 60°, el buzzer en una nota grave y lento.

Ambiente: De los 30°C hasta los 35°C el led encenderá color Verde, el servo se posicionará en 90°, el buzzer en una nota media y a velocidad media.

Caliente: De los 35°C en adelante el led encenderá color Rojo, el servo se posicionará en 150°, y el sonará en una nota más aguda y a velocidad más rápida además el ventilador se encenderá como autoprotección para regular la temperatura.

Ahora que hemos analizado a lo que queremos llegar, podemos comenzar a escribir el código.

El código:

```
#include <LiquidCrystal.h>
#include <Servo.h>

// Puertos LCD -> RS  E   D4  D5  D6  D7
LiquidCrystal lcd(8, 9, 10, 11, 12, 13);
Servo servo1;
int red=4;
int green=7;
int blue=6;
int buzzer=5;
int ventilador=2;
int sensorTemperatura=0;
float temperaturaValor;
void setup(){
  lcd.begin(16, 2);
  lcd.setCursor(0, 0);
  Serial.begin(9600);
  pinMode(red,OUTPUT);
  pinMode(green,OUTPUT);
  pinMode(blue,OUTPUT);
  pinMode(buzzer,OUTPUT);
  pinMode(ventilador,OUTPUT);
  servo1.attach(3);
}
```

Primero importaremos las librerías de lcd y de servos, luego procederemos a declarar todas las variables y a nombrar los pines. Luego en el void setup haremos cada configuración necesaria de los pines de salida, el tamaño de la lcd su inicio, el servomotor enganchado a un pin demás.

```
void loop(){
  temperaturaValor=analogRead(sensorTemperatura);
  temperaturaValor=(5*(temperaturaValor)*100)/1024;
  Serial.println(temperaturaValor);
  lcd.setCursor(0,0);
  lcd.print("Temp= ");
  lcd.print(temperaturaValor);
  lcd.print(" C");
  delay(100);
  if(temperaturaValor > 35){
    tempCalor();
    buzzerCalor();
  } else if(temperaturaValor > 30) {
    tempNormal();
    buzzerNormal();
  }
   else if(temperaturaValor < 30) {
    tempFrio();
    buzzerFrio();
  }
}
```

Luego en el void loop, vamos a almacenar el valor leído por el sensor, lo vamos a convertir a grados centígrados, luego vamos a configurar la lcd para que imprima el mensaje “Temp” y seguido que imprima el valor en grados centígrados del sensor, una vez terminado vamos a seguir con las condiciones para los rangos de nuestro Smart sensor, lo configuraremos con funciones externas que las programaremos adelante en esta sección solo las llamaremos. Primero condicionamos con un If, cuando el valor leído sea mayor a 35 entonces ejecuta la función tempCalor y BuzzerCalor. Y así sucesivamente con los otros 2 estados de nuestro smartSensor.

```
void tempCalor(){
  digitalWrite(red,HIGH);
  digitalWrite(green,LOW);
  digitalWrite(blue,LOW);
  digitalWrite(ventilador,HIGH);
  servo1.write(150);
  lcd.setCursor(0, 1);
  lcd.print("Estado=Caliente");
}
void buzzerCalor(){
 tone(5,1000, 200);
 delay(500);
 tone(5,1000, 200);
 delay(500);
}
void tempNormal(){
  digitalWrite(red,LOW);
  digitalWrite(green,HIGH);
  digitalWrite(blue,LOW);
  digitalWrite(ventilador,LOW);
  servo1.write(90);
  lcd.setCursor(0, 1);
  lcd.print("Estado=Ambiente");
}
void buzzerNormal(){
 tone(5,200, 400);
 delay(1000);
 tone(5,200, 400);
 delay(1000);
}
```

Procedemos a crear cada función para cada estado y lo separaremos en dos funciones por estado, una donde se configuren los led, el ventilador, el servo y la lcd, y el otro donde solo se configure el buzzer, para que este no afecte con sus delay a las demás funciones, de igual forma configuramos cada estado.

```
}
void tempNormal(){
  digitalWrite(red,LOW);
  digitalWrite(green,HIGH);
  digitalWrite(blue,LOW);
  digitalWrite(ventilador,LOW);
  servo1.write(90);
  lcd.setCursor(0, 1);
  lcd.print("Estado=Ambiente");
}
void buzzerNormal(){
 tone(5,200, 400);
 delay(1000);
 tone(5,200, 400);
 delay(1000);
}
void tempFrio(){
  digitalWrite(red,LOW);
  digitalWrite(green,LOW);
  digitalWrite(blue,HIGH);
  digitalWrite(ventilador,LOW);
  servo1.write(40);
  lcd.setCursor(0, 1);
  lcd.print("Estado=Frio");
}
void buzzerFrio(){
 tone(5,100, 600);
 delay(1000);
 tone(5,100, 600);
 delay(1000);
}
```

Al finalizar cargamos el programa al Arduino y veremos cómo funciona nuestro Sensor.

Conclusión los sensores de temperatura tiene una infinidad de uso tanto a nivel domestico como industrial o agropecuario etc.