

Optativa Área Arquitectura, Sistemas Operativos y Redes:
Sistemas Embebidos



Controlador de Temperatura Programable Frio/Calor

Alumna: Johanna Yamila Tabella

E-mail: Joy.Tabella@gmail.com

Sistemas Embebidos - Profesor Ing. Diego Azcurra

Cursada 2017

Licenciatura en Sistemas

Universidad Nacional de Lanús

ÍNDICE

Resumen.	3
Introducción.....	4
Materiales y métodos.....	5
Desarrollo.....	5
Resultados	12
Conclusiones.....	13
Referencias bibliográficas.....	13
ANEXO 1 (Código fuente).....	14
ANEXO 2 (Manual de usuario).....	18

Resumen

El CTP - Controlador de Temperatura Programable - es un proyecto pensado para la comodidad del usuario. Con el dispositivo tendrá la certeza de que el entorno en cuestión se mantendrá a la temperatura deseada programada por el desarrollador que se basará en la necesidad del cliente.

Este artefacto software es un prototipo para múltiples aplicaciones cuyo objetivo se centra en automatización de un rango de temperatura deseable, donde si la temperatura es menor a la media se encenderá la calefacción, y si la temperatura es superior a la media se encenderá la refrigeración. Claro que como planteaba recién, la temperatura media o deseable será diferente en cada ocasión ya que el CTP puede ser implementado en cualquier espacio reducido, como por ejemplo habitaciones pequeñas, incubadoras, terrarios, hasta vehículos sin sistema de aire acondicionado, y por ello se debe plantear como requisito o solicitud antes de cargar el programa en la plaqueta Arduino.

En la presentación se demostraran diversos puntos importantes que hicieron falta para llevar a cabo el artefacto software. Se comienza por el Marco Teórico que nos envuelve hasta la razón de haber elegido este dispositivo para desarrollar, y no otro.

Para ayudar al lector en caso de que quiera recrear el trabajo aquí presentado, se indican materiales y métodos para llevar adelante tanto el Hardware como el Software del proyecto; Respecto a su conexionado se adjuntará gráficos ilustrativos.

Al final tendremos el resultado obtenido y la conclusión por parte de este estudio.

También se adjunta como Anexo:

- El código final necesario para programar el Arduino UNO.
- Manual de Usuario.

Introducción

En el transcurso de la asignatura “Sistemas Embebidos” con el Profesor Ing. Diego Azcurra, la cual tuvo bastante contenido práctico, compré mi propio Arduino UNO y comencé a programar. Experimente como se involucraban las partes físicas con la programación básica, voltajes, amperajes, todos conocimientos que eran básicos en mí ya que provengo de una enseñanza media de Colegio Normal. También debo nombrar a mi esposo Mario y agradecerle, ya que me ayudo con algunos conceptos de electrónica y sobretodo con el esqueleto donde está montado mi proyecto y se sostiene todas las partes necesarias del Hardware.

Lo que me motivo a realizar este dispositivo programable de control de temperatura, fue más que nada la utilización del sensor, me parece interesante, hasta casi mágico, como es que toma la información del ambiente y lo refleja en forma de datos al programa. Fue todo un reto porque al principio compre varios componentes de sensores de temperatura DS-14 que fueron un fracaso, no sé si se debía a su procedencia o si hacía mucho que los tenía el comerciante y se habrán dañado, lo cierto es que lo único que me sacó adelante fue el sensor de temperatura que es apto para sumergir en líquidos, el DS18B20. Ninguno de estos componentes salió más de treinta pesos cada uno, por lo que me arriesgue a comprar varios DS-14, hasta tres y ninguno funcionó a pesar de que estaban correctamente conectados.

Al querer usar el sensor de temperatura claramente se me vino a la mente un Controlador de Temperatura. Es un proyecto que tiene múltiples aplicaciones, desde mantener la temperatura de un ambiente pequeño, una incubadora, un terrario; A su vez, puede utilizarse para remplazar el aire acondicionado de un automóvil, sería cuestión de modificar la ficha de alimentación del dispositivo, en fin, para cualquier espacio reducido que haga falta mantener una temperatura constante; Y esto se puede lograr con la implementación del CTP (Controlador de Temperatura Programable).

El conseguir todos los materiales fue una tarea sencilla, nada que no se pueda encontrar en una casa de electrónica, en mi caso fue MARIMON en Lomas de Zamora, y por MercadoLibre conseguí la placa de Arduino UNO junto con un kit de protoboard y cables. Los peltier, quienes serán los encargados de generar calor y frio, fueron extraídos de viejas cavas de vinos que ya no funcionaban y el resto de los elementos más pequeños los conseguí en mi hogar; Como buena Gamer y reparadora de pc, fuentes de energía de computadoras me sobraban, así que pude reutilizar algunos Coolers, disipadores y también utilizar una ATX para alimentar el proyecto, más exactamente, el módulo de relés.

Materiales y Métodos

Materiales utilizados:

- Modulo Arduino UNO
- Shield Modulo de relé x 4
- Sensor de Temperatura DS18B20
- Celda peltier x 2 Unidades
- Cooler Brushless x 2 Unidades
- Disipadores x 2 Unidades
- Resistencias 4,7 Ω x 4 Unidades
- Pantalla LCD Display 128x64B V2.0
- Leds de Alta Luminosidad x 3 Unidades
- Fuente de Alimentación ATX PC.
- Cargador de Celular que suplantar  la entrada USB al PC que alimenta la placa Arduino.

Equipamiento auxiliar utilizado:

- Equipo: Intel Core I3 CPU 530 / 8 GB de Memoria RAM
- SO: Windows 7 Professional SP1 64bits

Software:

- IDE Arduino v 1.8.4

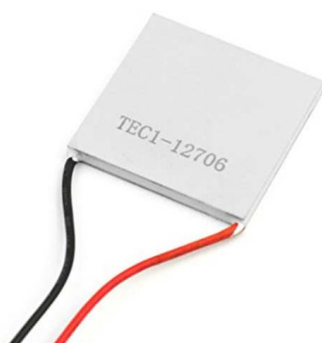
Librer as:

- OneWire y Dallas Temperature – Necesarios para el correcto funcionamiento del sensor de temperatura.
- U8glib – Necesario para el correcto funcionamiento del Display LCD 128x64.

Desarrollo

Para llevar a cabo este proyecto, fue necesario sumergirse en el mundo de Arduino, no es que sea algo imposible, pero hay que estar dispuesto a invertir bastante tiempo.

Como una breve definici n, podemos definir a Arduino como una plataforma f sica computacional open-hardware basada en una sencilla placa con entradas y salidas (E/S), anal gicas y digitales, y un entorno de desarrollo que implementa el lenguaje Processing/Wiring. Puede utilizarse en el desarrollo de objetos interactivos aut nomos o puede conectarse a la PC a trav s del puerto USB utilizando lenguajes como Flash, Processing, MaxMSP, etc.



En este dispositivo, se podr a decir que el n cleo central del CTP (Controlador de Temperatura Programable) son las c lulas peltier y su sistema de ventilaci n.

Imagen 1. C lula Peltier.

Este tipo de células funcionan por el efecto Peltier, éste se basa en que cuando una corriente pasa por los semiconductores, un cable positivo y otro negativo, se crea una transferencia de calor de una cerámica a la otra por el Efecto Peltier, provocando como resultado que una se enfríe y otra se caliente.

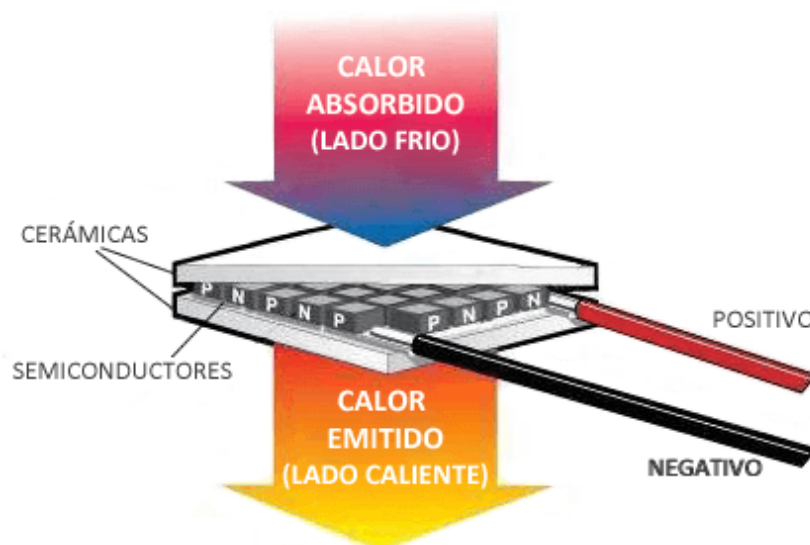
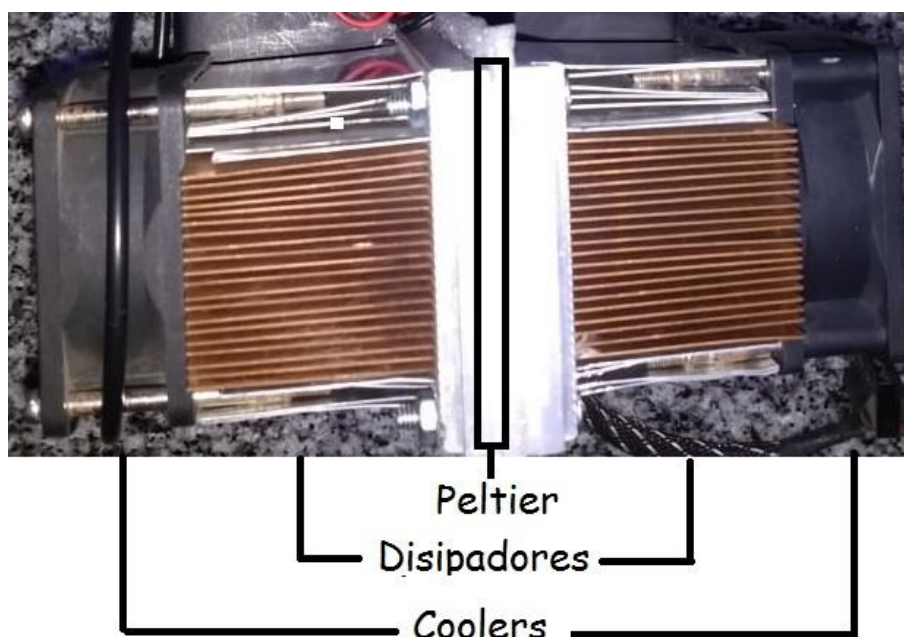


Imagen 2. Funcionamiento de Célula Peltier.

<https://www.luisllamas.es/arduino-peltier>

Si caso contrario, la polaridad se invierte, ahora el lado que generaba frio, generará calor y el lado caliente se volverá frio. De esta forma nos permite enviar la corriente de una polaridad a la otra, según lo requiera. Para evitar que se queme la celda, cosa que puede suceder en cuestión de minutos si no se provee de la ventilación adecuada, es obligatorio el uso de disipadores y Coolers en ambos extremos del peltier. Éstos se pueden extraer de viejas computadoras o fuentes que ya no se utilicen. Se puede agregar grasa siliconada en ambas caras para favorecer la disipación homogénea, tanto del frio como del calor.

Nos debería quedar de este modo:



A fin de que las celdas peltier no sean dañadas por las variables de temperaturas y los cambios de polaridad, tomadas en muestras que se realizan cada dos segundos, se agregó la siguiente porción de código que permite al programa esperar 5 segundos si ocurrió un cambio de Estado, y como Estado definimos a los comportamientos del sistema: Calefaccionando – Relajado – Refrigerado.

```
void loop() {  
    ....  
    int estadoAnterior = 0;  
    int estado = 0;  
  
    ...  
  
    if (estadoAnterior != estado) {  
        delay(5000);  
    }  
  
    if (temperatura >= tempMaxima) { // pregunta si la temperatura esta mayor o  
    igual que X grados  
        digitalWrite(6, HIGH); // apaga relay 2  
        digitalWrite(7, LOW); // enciende relay 1  
        estado = 1;  
    }  
    if (temperatura <= tempMinima) { // pregunta si la temperatura es menor o  
    igual a Y grados  
        digitalWrite(6, LOW); // enciende relay 2  
        digitalWrite(7, HIGH); // apaga relay 1  
        estado = 2;  
    }  
    if (temperatura > tempMinima && temperatura < tempMaxima) { // pregunta  
    si la temperatura es mayor a Y y menor a X grados  
        digitalWrite(6, HIGH); // apaga relay 2  
        digitalWrite(7, HIGH); // apaga relay 1  
        estado = 3;  
    }  
  
    estadoAnterior = estado;  
  
}
```

Su conexionado es simple, pero como todo este conjunto necesita de mayor voltaje del que proveen los pines de INPUT/OUTPUT del Arduino de 5 volt, se debe utilizar Relees y una fuente de alimentación de PC, que provea de 12 volt. (IMPORTANTE: Para encender una fuente de computadora hay que hacer puente entre los cables verde y negro del conector Power del Motherboard, más adelante se podría mejorar con la instalación de una ficha de encendido.

En mi caso, adquiriré un Shield de Modulo de relees que trae 4, más que nada por practicidad de simplificar las salidas GND y VCC.

Luego podemos ver los pines IN1, IN2, IN3, y IN4, que corresponden cada uno a un relee con sus respectivos pines de datos, donde según la señal que reciba desde el Arduino, será su comportamiento. En este proyecto lo utilizaremos para generar frío o calor con la polaridad del peltier. En el caso de los Coolers no será necesario prenderlos y apagarlos, ya que por un tema de costo de voltajes y a fin de evitar sobrecalentamientos en las celdas, se decidió dejarlo siempre encendidos.



Imagen 3. Shield Modulo de Relees

Finalizando con lo que es la parte de refrigeración y calefacción del sistema, nos encontramos con la necesidad de obtener un sensor de Temperatura que nos indique los datos necesarios para hacer nuestro relevo. Como comente en la Introducción de este documento, se pueden adquirir diversos tipos de Sensores de Temperatura, pero el que a mí me resulto fue el DS18B20, esto no quiere decir que no se pueda utilizar otro.



Imagen 4. Sensor de Temperatura DS18B20.

Su forma de conexionado es muy simple, vemos que tenemos un GND (Cable Negro), VCC (Cable Rojo), y DQ (Cable Amarillo), que será nuestro cable receptor de los datos y que va conectado a través de una resistencia de $4,7\ \Omega$ al positivo y al pin digital del Arduino, en mi caso está conectado en el pin 13, pero podría ser cualquier otro. No olvidar que para su correcta utilización necesitaremos descargar dos librerías al IDE del Arduino que son OneWire y DallasTemperature. Ambos son necesarios para el correcto funcionamiento del sensor. Estas librerías nos permiten captar y censar los datos de temperatura de forma eficaz.

Por último, y no menos importante, tenemos la aplicación de una Pantalla Display LCD que nos proveerá de la información necesaria para saber qué es lo que sucede en el interior de nuestro artefacto. Nos indica la Temperatura actual con un refresh cada 2 segundos, y además nos indica en que instancia se encuentra el sistema, ya sea en el Estado de Refrigeración, Calefacción, o simplemente se encuentra en modo Relajado ya que alcanzo la temperatura deseada.

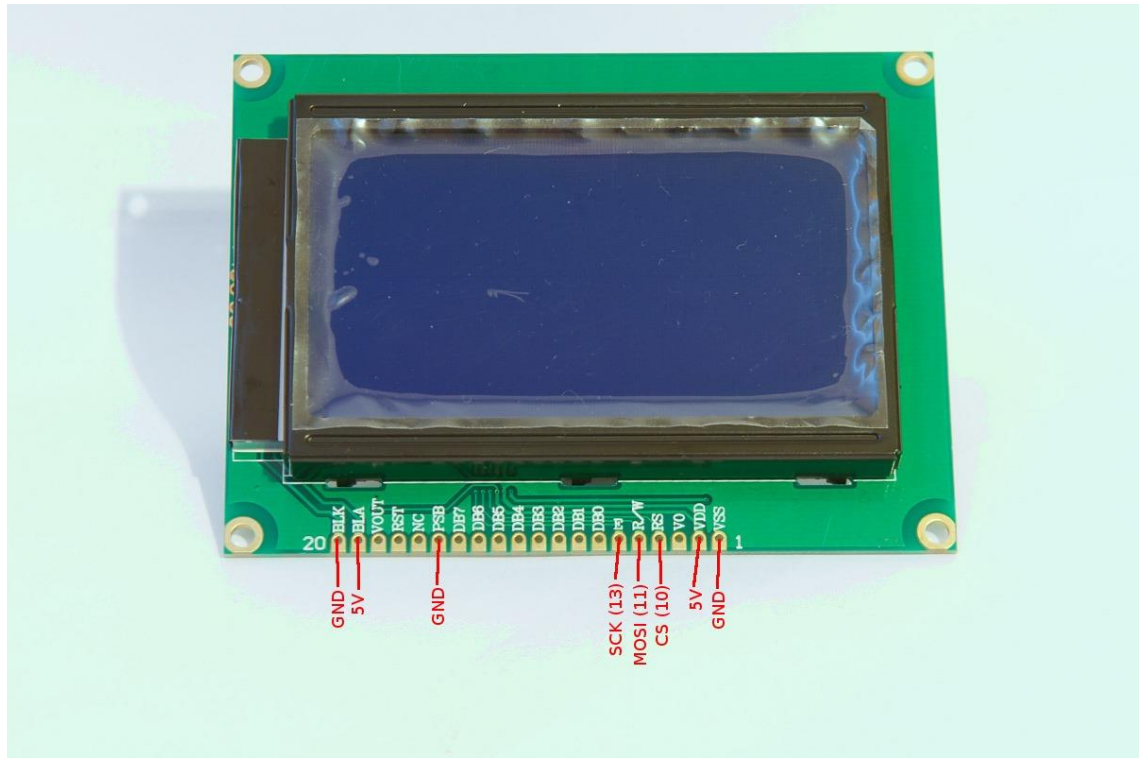


Imagen 5. Pantalla Display LCD 128x64B.

En la imagen () nos encontramos con la información necesaria para lograr un conexionado exitoso. Tener en cuenta que los pines SCK/E, MOSI/RW, y CS/RS, van conectados directamente a pines digitales del Arduino. En mi caso opte por los 10, 11 y 12, diferente a como se encuentran es esta ilustración. Luego, para el manejo óptimo de la pantalla, es necesario descargarse una librería que trabaje con displays. Tener en cuenta que existen diversos tipos de LCD, algunos más pequeños, otros a color, lo ideal es encontrar el que mejor se ajuste a nuestras necesidades. Lo mismo pasa con las librerías; No existe una única librería para la implementación de texto en el Display, pero a mí el que mejor me resulto fue u8glib.

A continuación se facilitan las figuras de conexionado, tanto en vista Protoboard como en Esquemático. Ambos desarrollados con el Software Fritzing.



Resultados

Podemos observar que el proyecto terminado cuenta con un sinfín de aplicaciones en el mundo real. Solo basta con determinar para qué universo estaría orientado y el resto es cambiar dos variables, temperatura mínima y temperatura máxima.

Como puntos interesantes, podemos destacar algunas señales que nos da el artefacto. Una señal auditiva, un pitido al momento de encender el equipo y una señal visual, con el Led blanco que queda encendido durante la utilización del dispositivo. Aún faltan algunas mejoras para transformar el artefacto de uso universal que pueda ser manipulado por cualquier persona, más allá de que cuente con alguna discapacidad, pero sé que este es el camino, y puede ser mejorado.

Hay muchos trabajos similares en la web, pero pocos que incluyan un Display para reflejar la información. Fue todo un desafío porque no daba con la librería exacta para manejar el LCD que había comprado. Pero me alegra haber resuelto el problema y seguir adelante finalizando el trabajo.



Imagen 6. Proyecto terminado y funcionando.

Conclusiones

- Al enchufar el proyecto a la fuente de alimentación de PC y al cable USB del Arduino, el sistema se enciende de forma tal, que lanza un pitido de 300 milisegundos y se enciende un indicador de Led Blanco, hasta que el sistema se desenchufe.
- El Sistema permite saber la Temperatura actual, información que toma del entorno cada 2 segundos.
- Datos como la Temperatura actual y el estado del sistema, están reflejados en la pantalla LCD; Se van refrescando en tiempo real.
- Cuando el sistema censa la temperatura, y está dentro de los rangos permitidos como máximo y mínimo, el sistema se relaja, los peltier no se prenden y los Coolers giran a modo de ventilación.
- Cuando el sistema censa la temperatura, y detecta una temperatura menor a la mínima permitida, el programa acciona los peltier y en consecuencia la parte frontal comienza a lanzar aire caliente. Cuando esto sucede, también se activa un indicador de Led Rojo.
- Cuando el sistema censa la temperatura y detecta una temperatura mayor a la máxima permitida, el programa acciona los peltier y en consecuencia la parte frontal comienza a lanzar aire frio. Cuando esto sucede, también se activa un indicador de Led Azul.

Referencias Bibliográficas

BricoGeek - Tutorial Arduino: Pantalla Serial LCD 128x64.

<https://blog.bricogeek.com/noticias/tutoriales/tutorial-arduino-pantalla-serial-lcd-128x64/>

Nergiza - Cómo hacer un aire acondicionado casero con una célula de peltier.

<https://nergiza.com/como-hacer-un-aire-acondicionado-casero-con-una-celula-de-peltier>

OpenWebinaris - Tutorial Arduino: Sonidos con Arduino.

<https://openwebinars.net/blog/tutorial-de-arduino-sonidos-con-arduino/>

PatagoniaTec – LCD SPI 128x64. https://saber.patagoniatec.com/2014/11/lcd-spi-128x64-serie-arduino-argentina-ptec-12864-grafico-glcd/?fbclid=IwAR1Fni4wL1tbDXnA4_kIJVacZNn8D3eSQLzBGTIWKJ1UaPuH0u0NIajpXKU

ProgramarFacil - DS18B20 sensor de temperatura para líquidos con Arduino.

<https://programarfacil.com/blog/arduino-blog/ds18b20-sensor-temperatura-arduino/>

ANEXO 1

CÓDIGO FUENTE


```

#include <OneWire.h>
#include <DallasTemperature.h>
#include "U8glib.h"

//DISPLAY
//U8GLIB_ST7920_128X64_1X u8g(10); //este es para uno
U8GLIB_ST7920_128X64_1X u8g(12, 11, 10); // SPI Com: SCK = en = 18,
MOSI = rw = 16, CS = di = 17

//rs 10 , rw 11 , e 12

// Pin donde está conectado dq
const byte pinDatosDQ = 13;
float temperatura = 0;
float tempMinima = 23;
float tempMaxima = 25;

// instancia a las clases OneWire y DallasTemperature
OneWire oneWireObjeto(pinDatosDQ);
DallasTemperature sensorDS18B20(&oneWireObjeto);

void setup() {
  //inicio la comunicación serie
  Serial.begin(9600);
  pinMode(3, OUTPUT); // pin 3 como salida Led azul ... antes estaba en el 2
  pinMode(4, OUTPUT); // pin 4 como salida Led rojo. ... antes estaba en el 3
  pinMode(2, OUTPUT); // pin 2 como salida Led testigo ON-OFF ... antes
estaba en el 4 ..
  pinMode(5, OUTPUT); // pin 5 Coolers x2
  pinMode(6, OUTPUT); // pin 6 Peltier negativo
  pinMode(7, OUTPUT); // pin 7 Peltier positivo
  pinMode(8, OUTPUT);
  digitalWrite(2, HIGH); // Inicializo el pin 2 en cero
  digitalWrite(3, LOW); // Inicializo el pin 3 en cero
  digitalWrite(4, LOW); // Inicializo el pin 4 en uno.
  digitalWrite(5, LOW); // Inicializo pin 5 en uno, siempre prendidos los Coolers.
  relay 1
  digitalWrite(6, HIGH); // peltier negativo relay 2
  digitalWrite(7, HIGH); // peltier positivo relay 3

  // Sonido
  digitalWrite(8, HIGH);
  delay(300);
  digitalWrite(8, LOW);

  // Inicio el bus OneWire
  sensorDS18B20.begin();

```

```

// Display
u8g.setColorIndex(3);    // definir intensidad, 3 es máximo
u8g.setColorIndex(1);    // pixeles on, poniendo 0 no salen
}

void loop() {
  float tempAnterior = 0;
  temperatura = sensorDS18B20.getTempCByIndex(0);
  int estadoAnterior = 0;
  int estado = 0;

  u8g.firstPage();
  do {
    draw();
  } while ( u8g.nextPage() );

  // mando el comando para tomar la temperatura del sensor
  Serial.println("Mandando comando al sensor");
  sensorDS18B20.requestTemperatures(); // clase de DallasTemperature

  //leo y muestro los datos del sensor
  Serial.print("Temperatura sensor: ");
  Serial.println(temperatura);
  // clase de dallas temperature.
  delay(2000);

  if (estadoAnterior != estado) {
    delay(5000);
  }

  if (temperatura >= tempMaxima) { // pregunta si la temperatura esta mayor o
igual que X grados
    digitalWrite(6, HIGH); // apaga relay 2
    digitalWrite(7, LOW); // enciende relay 1
    estado = 1;
  }

  if (temperatura <= tempMinima) { // pregunta si la temperatura es menor o
igual a Y grados
    digitalWrite(6, LOW); // enciende relay 2
    digitalWrite(7, HIGH); // apaga relay 1
    estado = 2;
  }

  if (temperatura > tempMinima && temperatura < tempMaxima) { // pregunta
si la temperatura es mayor a Y y menor a X grados
    digitalWrite(6, HIGH); // apaga relay 2
    digitalWrite(7, HIGH); // apaga relay 1
    estado = 3;
  }
}

```



```

    }

    estadoAnterior = estado;

}

void draw(void) {
    // Otros tipos de letras.
    //u8g.setFont(u8g_font_unifont);
    //u8g.setFont(u8g_font_osb21);
    u8g.setFont(u8g_font_5x8r);
    u8g.setColorIndex(3);      // definir intensidad, 3 es máximo
    u8g.setColorIndex(1);      // pixeles on, poniendo 0 no salen
    u8g.drawStr( 25, 15, "CONTROLADOR DE");
    u8g.drawStr( 30, 25, "TEMPERATURA");
    u8g.drawStr( 6, 40 , "Temperatura:");
    u8g.enableCursor();
    u8g.setPrintPos(75, 40);
    u8g.print(temperatura);

    if (temperatura <= tempMinima) {
        u8g.drawStr( 3, 55, "Sistema Calefaccionando *");
        digitalWrite(3, LOW);    // Led Azul Apaga
        digitalWrite(4, HIGH);   // Led Rojo Enciende
    } else if (temperatura >= tempMaxima) {
        u8g.drawStr(3, 55, "Sistema Refrigerando %");
        digitalWrite(3, HIGH);   // Led azul Enciende
        digitalWrite(4, LOW);    // Led Rojo Apaga
    } else {
        u8g.drawStr( 3, 55, "Sistema Relajado");
        digitalWrite(3, LOW);    // Led Azul Apaga
        digitalWrite(4, LOW);    // Led Rojo Apaga
    }
}
}

```

ANEXO 2

MANUAL DE USUARIO

Como primer paso, se debe indicar al programador cual es la temperatura ideal necesaria para mantener el ambiente deseado.

Una vez que el Cliente indicó su franja de temperaturas media, como la mínima y máxima temperatura soportables, se modifican los valores de las variables “tempMinima” y “tempMaxima” del código y ya está listo para cargar el Software al PIC de Arduino a través de su IDE.

Solo resta la instalación del dispositivo, procurando que la parte posterior del sistema de ventilación quede por fuera del ambiente para un mejor resultado.

Para su utilización es necesario conectar el Artefacto con la fuente ATX a través de su ficha de entrada de Disco Rígido IDE, que se encuentra en la parte lateral derecha. Esta alimentación extra se encarga de proveer de energía a los peltiers y Coolers; Si no se conecta no funcionará esta parte, y por ende, no habrá ningún cambio de temperatura.

No olvidarse de conectar la placa Arduino a través de su cable USB, que puede alimentarse con cualquier cargador de celular.



Imagen 1. Conector IDE.

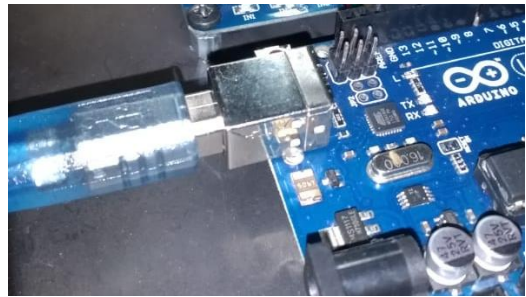


Imagen 2. Conector USB Arduino

Una vez conectado correctamente el dispositivo, comenzarán a funcionar los Coolers y hará una señal sonora de un pitido que indica el encendido exitoso, también cuenta con un Led Blanco que funciona como testigo e indica que esta se encuentra encendido.

El Display o Pantalla Led, le brindará al usuario, información respecto a la temperatura actual, y al Estado en que se encuentre el sistema en ese momento, y será respaldado con Leds frontales, uno de color Azul cuando el sistema este Refrigerando, y un led Rojo cuando el sistema este Calefaccionando.

Estos Estados son:

- Sistema Refrigerando % : Se refiere a que la temperatura sobrepaso la temperatura Máxima de la media y debe reducirse.
- Sistema Relajado : Se refiere a que se alcanzó una temperatura media deseable.
- Sistema Calefaccionando * : Se refiere a que la temperatura es inferior a la temperatura Mínima de la media y debe aumentarse.

Ahora sí, a disfrutar!

Link de la video presentación

<https://drive.google.com/open?id=1kJd255CeRuvD4L7Zdde3RLS0yIXMquUb>

Link del Proyecto

<https://drive.google.com/open?id=1f-Hk6rbFyPfECTygb7wrFco7zrBB62K->