

**CENTRO PAULA SOUZA  
ETEC ARMANDO PANNUNZIO  
Curso Técnico em Automação Industrial**

**Joel de Souza Matias  
Marcelo Thiago Henrique Meira  
Sandy Avila Inácio**

**CONTROLE TÉRMICO DE ELEMENTO LÍQUIDO COM ARDUÍNO  
EM CONTROLE ON/OFF**

**Sorocaba/ SP  
2023**

**Joel de Souza Matias**  
**Marcelo Thiago Henrique Meira**  
**Sandy Avila Inácio**

**CONTROLE TÉRMICO DE ELEMENTO LÍQUIDO COM ARDUÍNO**  
**EM CONTROLE ON/OFF**

Trabalho de Conclusão de Curso apresentado ao Curso Técnico em Automação Industrial da ETEC ARMANDO PANNUZIO, orientado pelo Professor EDISON BLAZ MARTINEZ, como requisito parcial para obtenção do Título de Técnico em Automação Industrial.

**Sorocaba/ SP**  
**2023**

## **Dedicatória**

Dedicamos esse trabalho aos nossos colegas e amigos de curso, que assim como nós encerram essa difícil e longa etapa de formação técnica, com muito estudo e aprendizado. Dedicamos esse trabalho ao curso Técnico em Automação Industrial e aos alunos que também estão em formação da Etec Armando Pannunzio, e ao corpo docentes, a quem ficamos gratos pelas contribuições de nossas formações. E as nossas famílias que sempre esteve aos nossos lados, dando-nos forças e incentivos para chegarmos até aqui, enfim ao término desta conquista que é de todos nós.

## **Agradecimentos**

Somos gratos a Deus pela saúde e por ter sustentarmos de pé diante todas as dificuldades e adversidades que encontramos durante essa etapa de formação. As nossas famílias que sempre estiveram aos nossos lados, por incentivar-nos e acreditar que seríamos capazes de concluirmos o curso, foram muitos dias que estivemos ausentes, daqueles que amamos muito para estarmos cumprindo com esse processo de formação. E aos nossos professores que com muito empenho e carinho por ensinar, estiveram à frente dos ensinamentos transmitidos, que com muita dedicação conseguimos absorver as lições, tarefas, experiências em salas de aulas e laboratórios que pudemos apresentar nosso desempenho nesse processo de formação.

## Resumo

Este projeto consiste em ser mais um elemento didático para as aulas de Controle e Automação de Processo Industrial. É um controle térmico de elemento líquido com Arduino controle ON/OFF.

Sua função é ser compatível com os Controladores lógicos programáveis (Arduíno, Esp32, PLCs e Microcontroladores) disponíveis na presente unidade escolar, em especial ao Arduino UNO utilizado nesse trabalho. Assim sendo, permitirá que os alunos façam uma experiência prática e testem suas habilidades em um projeto análogo a um industrial, enriquecendo o processo de aprendizagem e conhecimento.

Nesse projeto trabalharemos com sistema de aquecimento em malha fechada que necessitam de um controle de temperatura. Esse controle é utilizado, para manter a temperatura em um valor ajustado. Para isto, utilizamos um termostato. Por meio dele, realizamos o controle de temperatura e configuramos o valor de setpoint. O setpoint é um valor de referência para controle da temperatura do sistema.

Quando o valor de temperatura do recipiente for maior ou igual ao setpoint, o sistema desativa a resistência. Caso a temperatura do vaso seja menor com o valor desejado, o sistema liga a resistência. De acordo com seu funcionamento, desenvolveremos um termostato ON/OFF com Arduino, a fim de realizar o controle do sistema. Assim como, utilizaremos o sensor de temperatura DS18B20, para monitorar a temperatura do sistema e apresentar na tela de LCD. Portanto, de acordo com valor de temperatura lido, ativaremos ou desativaremos o relé, para controlar o aquecimento do sistema.

**PALAVRA CHAVE** – Arduino, termostato, Microcontrolador, Setpoint, Controlador e Temperatura.

## RESUME

Industrial Process Control and Automation. It is a liquid element thermal control with Arduino ON/OFF control.

Its function is to be compatible with programmable logic controllers (Arduíno, Esp32, PLCs and Microcontrollers) available in this school unit, in particular the Arduino UNO used in this work. Therefore, it will allow students to have a practical experience and test their skills in a project similar to an industrial one, enriching the learning and knowledge process.

In this project we will work with a closed-loop heating system that requires temperature control. This control is used to maintain the temperature at an adjusted value. To do this, we use a thermostat. Through it, we perform temperature control and configure the setpoint value. The setpoint is a reference value for controlling the system temperature.

When the container temperature value is greater than or equal to the setpoint, the system deactivates the resistance. If the vessel temperature is lower than the desired value, the system turns on the resistance. According to its operation, we will develop an ON/OFF thermostat with Arduino, in order to control the system. Likewise, we will use the DS18B20 temperature sensor to monitor the system temperature and display it on the LCD screen. Therefore, according to the temperature value read, we will activate or deactivate the relay to control the system's heating.

KEYWORD – Arduino, thermostat, Microcontroller, Setpoint, Controller and Temperature.

## **Lista de ilustração**

Figura 1 – Esquema Eletrônico do Controle Térmico.

*Figura 2 – Tela de Reconfiguração da Temperatura do Sistema.*

*Figura 3 – Tela de Configuração de Temperatura.*

*Figura 4 – Apresentação da Temperatura do Sistema.*

Tabela 1 – Componentes eletrônico utilizados no trabalho.

Tabela 2 – Valor do Material

# **SUMÁRIO**

**1 Introdução**

**2 Desenvolvimento**

**3 Fluxograma4 Desenvolvendo o controle térmico de elemento líquido com Arduino controle ON/OFF.**

**5 O Circuito do controle térmico**

**6 Análise do Código**

**7 Declaração de Variáveis e Definição de Dados**

**8 Função void setup**

**9 Tela de menu para ajuste de temperatura**

**10 Configuração do parâmetro de temperatura**

**11 Requisição de temperatura e acionamento da resistência elétrica**

**12 Programação e configuração do Controle Térmico de Elementos Líquido com Arduino Controle ON/OFF**

**13 Lista de Materiais**

**14 Quantitativo e valor total do material**

**15 Conclusão**

**16 REFERÊNCIAS**



## 1 Introdução

Nesse trabalho de conclusão de Curso em técnico em Automação Industrial, trabalharemos com o controle térmico de elemento líquido com Arduino controle ON/OFF.

Nesse projeto trabalharemos com sistema de aquecimento em malha fechada que necessitam de um controle de temperatura. Esse controle é utilizado, para manter a temperatura em um valor ajustado. Para isto, utilizamos um termostato. Por meio dele, realizamos a manipulação da resistência elétrica que fará o aquecimento do líquido e alcançar a temperatura de setpoint. O setpoint é um valor de referência para controle da temperatura do sistema.

Utilizaremos o Arduino no controle ON/OFF de uma resistência elétrica e monitoramento através de display LCD. O experimento é composto de um recipiente para aquecimento de um elemento líquido (Água), sensor de temperatura, termostato, relés e controlador Arduino.

O tanque de aquecimento em escala de bancada foi concebido com a finalidade de monitorar e observar os efeitos do volume de elemento líquido sobre o perfil de aquecimento tendo como objetivo um set-point de 60°C.

Portanto, de acordo com valor de temperatura lido, ativaremos ou desativaremos o relé, para controlar o aquecimento do sistema.

## 2 Desenvolvimento

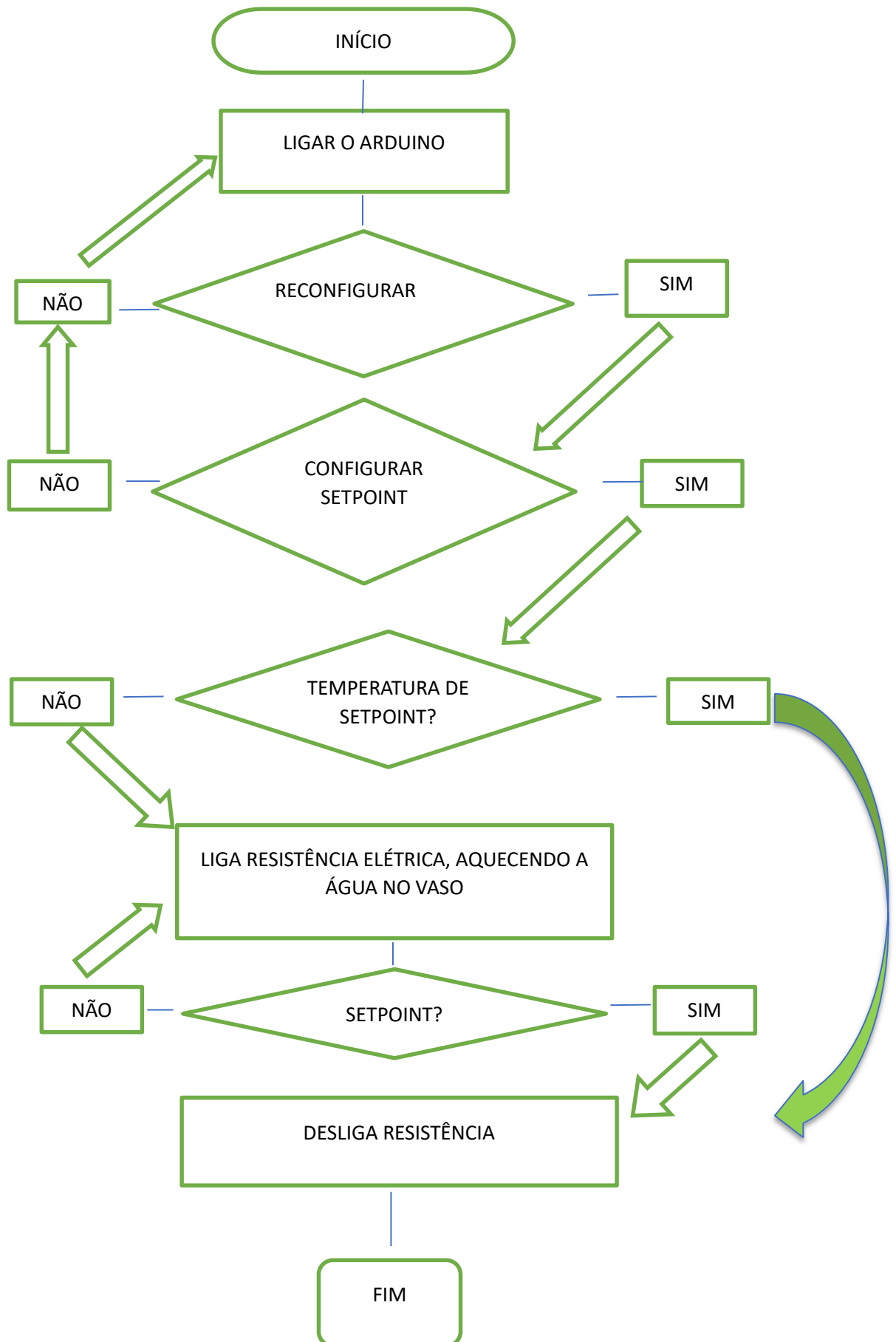
Nesse projeto trabalharemos com sistema de aquecimento em malha fechada que necessitam de um controle de temperatura. Esse controle é utilizado, para manter a temperatura em um valor ajustado. Para isto, utilizamos um termostato. Por meio dele, realizamos a manipulação da resistência elétrica que fará o aquecimento do líquido e alcançar a temperatura de setpoint

Utilizaremos o Arduino no controle ON/OFF de uma resistência elétrica e monitoramento através de display LCD. O experimento é composto de um recipiente para aquecimento de um elemento líquido (Água), sensor de temperatura, termostato, relés e controlador Arduino.

O tanque de aquecimento em escala de bancada foi concebido com a finalidade de monitorar e observar os efeitos do volume de elemento líquido sobre o perfil de aquecimento tendo como objetivo um set-point de 60°C.

Portanto, de acordo com valor de temperatura lido, ativaremos ou desativaremos o relé, para controlar o aquecimento do sistema.

## 3 Fluxograma



## **4 Desenvolvendo o controle térmico de elemento líquido com Arduino controle ON/OFF.**

De acordo com o princípio de funcionamento do dispositivo, serão implementadas as seguintes funcionalidades:

Criação de tela de confirmação, para alteração do valor da temperatura;

Implementar função para salvar o valor de temperatura ajustada pelo usuário, mesmo que o sistema desligue e ligue novamente;

Criação de um teclado para configurar o valor da temperatura, de modo a utilizar apenas 2 botões;

Trabalhar com o sensor de temperatura DS18B20;

Criação de Menus no LCD 16 x 2.

## **5 O Circuito do controle térmico**

No circuito da Figura 1, temos 3 push buttons. O 1 (+) e o 2 (-), respectivamente, são utilizados para incremento e decremento do valor de ajuste da temperatura

O botão 3 (set) é utilizado para entrar nas configurações de temperatura ou salvar a temperatura ajustada pelo usuário.

O LCD será utilizado para receber e apresentar as telas e informações de temperatura do sensor DS18B20 Arduino (4).

Quando a temperatura do sistema for menor que o valor ajustado pelo usuário, o relé será acionado. Dessa forma, a resistência ligará, para aquecer a temperatura do sistema.

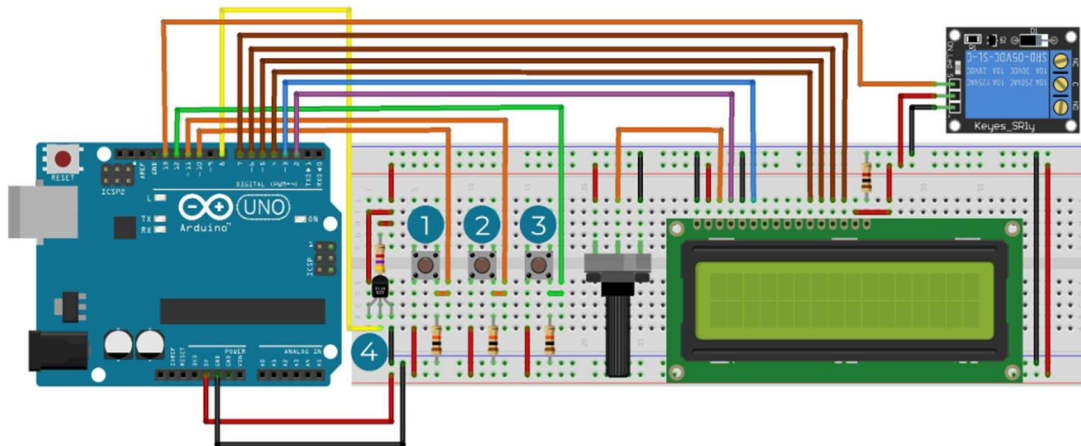


Figura 1 – Esquema Eletrônico do Controle Térmico.

## 6 Análise do Código

Para este projeto, incluímos as bibliotecas dos periféricos e dispositivos utilizados no projeto, conforme apresentado nas linhas 001 a 004.

Os pinos do LCD foram configurados (Linhas 006 e 007) e iniciamos o processo de configuração do sensor DS18B20.

O Sensor DS18B20 ArduinoO Sensor DS18B20 é um sensor digital que utiliza o protocolo 1-Wire. Portanto, é um sensor digital com boa precisão, confiável e com característica parasite power. Nas linhas 009 a 017, fizemos a configuração de funcionamento do sensor DS18B20.

E observe que na linha 10, o valor 8 é definido com o nome ONE\_WIRE\_BUS. Isto serve para usarmos esse nome como referência ao pino digital 8, onde conectaremos o sensor de temperatura.

Criamos uma instância chamada oneWire e passamos o pino digital que o nosso sensor está conectado (ONE\_WIRE\_BUS ).

No protocolo 1-Wire, o dispositivo se comunica por um único fio e permite a comunicação de diversos dispositivos no mesmo barramento. Por fim, passaremos oneWire como referência para que o objeto sensors possa saber que utilizaremos o sensor DS18B20 conectado ao pino digital 8. Logo, todos os dados do sensor serão lidos e manipulados pelas funções da biblioteca DallasTemperatura. Conseqüentemente, para que possamos trabalhar com esses dados e dispositivos de entradas, necessitamos criar diversos tipos de variáveis para manipular os dados de projeto.

## 7 Declaração de Variáveis e Definição de Dados

Foram declaradas diferentes variáveis para armazenar os dados dos botões, estado dos botões e sensor, de acordo como comentado em cada variável. E além da variável, fizemos 2 definições de extrema importância no código. Definimos o valor 60 como EndVerif e o valor 59 como EndTemp (Linhas 035 e 038).

Esses dois valores serão utilizados para salvarmos dados nas posições 120 e 125 da memória EEPROM, referente ao nosso sensor de temperatura. Mas antes de iniciarmos a apresentação da lógica de gravação dos dados na memória, devemos compreender o processo de configuração de cada dispositivo e sensor na função void setup.

## 8 Função void setup

Na função void setup, inicializamos o nosso LCD como 16 x 2 (Linha 043) e aguardamos um tempo de 500 ms para a sua inicialização (Linha 045). Em seguida, imprimimos a mensagem Automação por 3 segundos (Linhas 047 a 052).

Em seguida, realizamos 4 comandos de configuração:

1. Inicialização do sensor DS18B20 Arduino (Linha 055);
2. Limpeza da tela do LCD 16 x 2 (Linha 057).
3. Configuração do pino de conexão do relé como saída digital (Linha 059);
4. Leitura do valor de temperatura salvo na memória (Linha 061).

## 9 Tela de menu para ajuste de temperatura

Ao pressionar o botão (Linha 070), o sistema apresentará uma mensagem. Então, aparecerá a mensagem Reconfigurar o Sistema com duas opções Sim e Não (Linhas 073 a 078). A tela de reconfiguração é apresentada na Figura 3.



*Figura 2 – Tela de Reconfiguração da Temperatura do Sistema.*

Portanto, podemos configurar a temperatura ou sair do menu, caso tenha pressionado o botão de forma acidental.

Após pressionar o botão e a mensagem for impressa, o sistema fica em loop no laço (Linhas 080 a 093). Nesse momento, realizará o processo de leituras dos botões incremento e decremento.

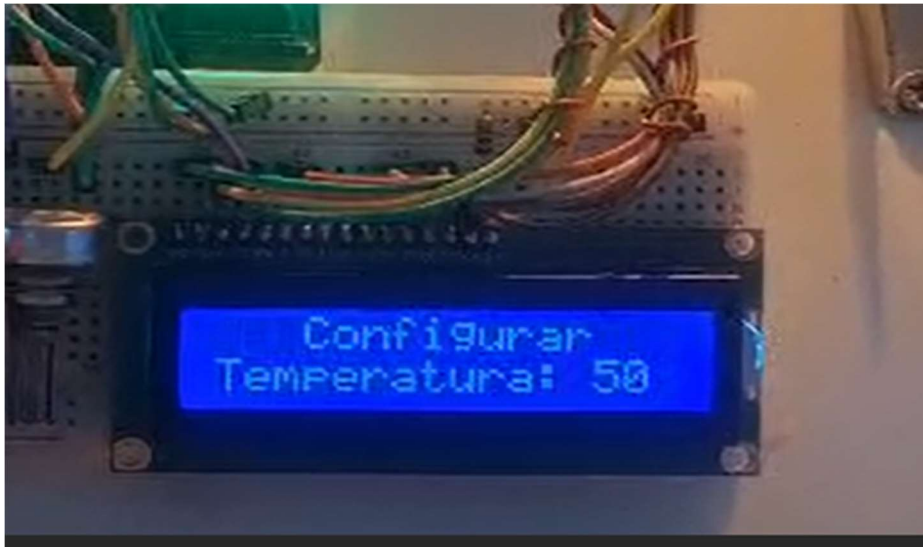
Logo após a leitura dos botões (Linhas 083 e 084), realizamos o tratamento de debounce com um delay (Linha 086). Por fim, analisamos uma condição (Linha 088), para verificar se o botão incremento está pressionado e o decremento não pressionado.

Portanto, caso a condição seja verdade, o valor 0 será salvo na posição EndVerif (60) da memória EEPROM. Em seguida, sairá do laço, já que a condição de permanência será falsa. A outra forma de saída do laço é quando o botão decremento (Não) é pressionado (Linha 093).

## 10 Configuração do parâmetro de temperatura

Após a entrada na condição (Linha 100), aparecerá a mensagem Configurar Temperatura (Linhas 103 a 106). Em seguida, é realizada a leitura do valor de temperatura armazenada na posição EndTemp (60) da memória EEPROM.

Logo após a leitura, o valor lido será 0, pois a memória EEPROM estará com todas as posições zeradas inicialmente. Em seguida, o valor será apresentado na tela, conforme apresentado na Figura 3.



*Figura 3 – Tela de Configuração de Temperatura.*

Caso a temperatura já tenha sido configurada outra vez, aparecerá o último valor ajustado.

Então, o sistema ficará em loop realizando a leitura dos botões. A lógica de leitura verifica o botão pressionado e armazena seu estado anterior, para evitar incrementar mais de uma vez. Portanto, essa lógica simula um contador com incremento e decremento utilizando botões (Linhas 118 a 138).

A partir da modificação dos valores, o sistema apresenta o valor na tela.

Quando pressionar o botão set, a condição será falsa e a execução sairá do loop (Linha 145). Após isto, a tela do LCD será limpa e o valor 60 será salvo na posição EndVerif da EEPROM. Logo, esse valor será utilizado para sinalizar que a temperatura está ajustada no sistema.

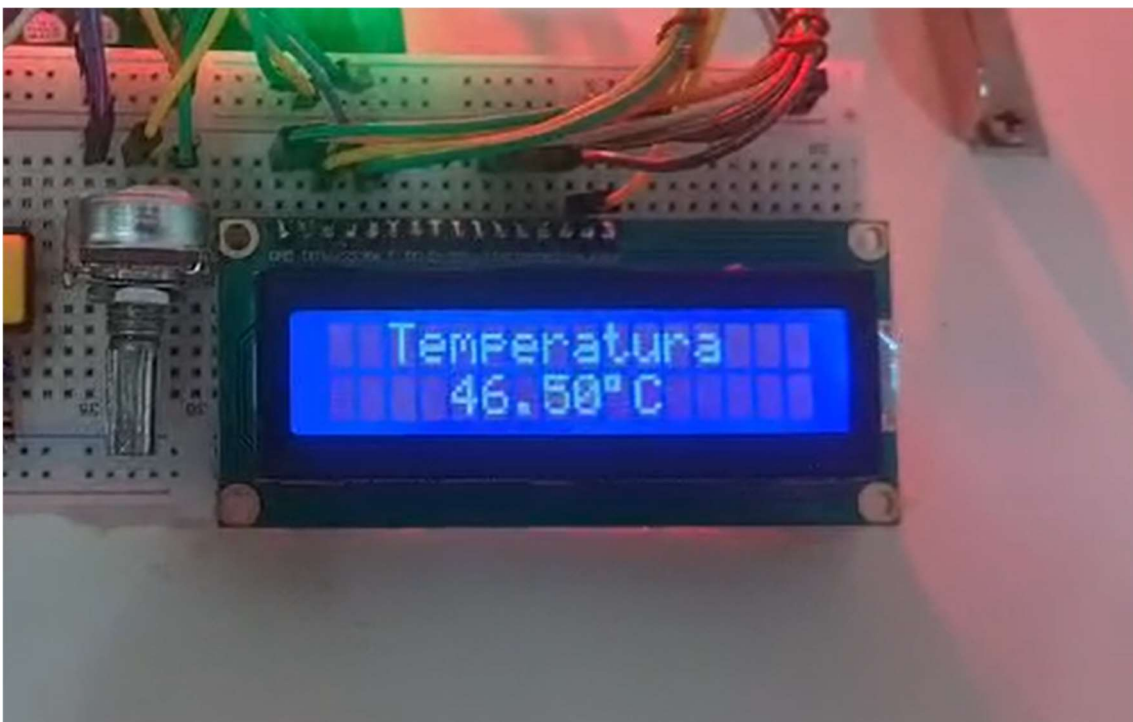
Em seguida, o valor de ajuste da temperatura é salvo na posição EndTemp (Linha 149). Isto serve para manter o valor salvo em caso de falta de energia. Portanto, evitará a perda da informação, quando o sistema for ligado novamente. O valor salvo será lido novamente na inicialização do sistema (Linha 061).



## 11 Requisição de temperatura e acionamento da resistência elétrica

Após isto, o sistema voltará para execução normal. Desse modo, fará a requisição de temperatura e receberá o valor pelo index 0. Este index representa o único sensor utilizado (Linha 154).

Por fim, apresentamos o valor da temperatura quando a condição (Linhas 156) é verdadeira, conforme apresentado na Figura 4 . Esta condição é utilizada para evitar que o sistema apresente o mesmo valor anterior de temperatura.



*Figura 4 – Apresentação da Temperatura do Sistema.*

Em seguida, será verificado se o valor é diferente do anterior. Portanto, se for verdade, apresentará a temperatura na tela do LCD 16 x 2. Em seguida, pegará o valor atual e guardará na variável ValorAnterior para ser utilizada como referência na próxima comparação.

Após isto, o sistema compara a temperatura do ambiente com o valor de ajuste. Com base na comparação, ele acionará ou desligará o relé do aquecedor do sistema (Linhas 167 a 175).

Após esta etapa, iniciaremos a apresentação de toda a lógica e configuração de parâmetros do termostato ON/OFF.

## 12 Programação e configuração do Controle Térmico de Elementos Líquido com Arduino Controle ON/OFF

```

#include <LiquidCrystal.h> //Inclusao da biblioteca do LCD
#include <EEPROM.h> //Inclusao da biblioteca para manipulacao das funcoes
da memoria EEPROM
#include <OneWire.h> //Inclusao da biblioteca para manipulacao das funcoes
de dispositivos OneWire
#include <DallasTemperature.h> //Inclusao da biblioteca para manipulacao das
funcoes do sensor DS18B20

const int rs = 2, en = 3, d4 = 4, d5 = 5, d6 = 6, d7 = 7;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

//Fio de dados de temperatura é conectado ao pino 11 no Arduino
#define ONE_WIRE_BUS 8
/*****/
//Configura uma instância oneWire para se comunicar com qualquer dispositivo
OneWire
OneWire oneWire(ONE_WIRE_BUS);
/*****/

//Endereçamento do dispositivo sensor DS18B20
DeviceAddress sensor1;

//Passar nossa referência oneWire para Dallas Temperature.
DallasTemperature sensors(&oneWire);

//Variavel para verificar se o usuário já configurou a temperatura durante a
inicialização do sistema
unsigned int verificador = 0;

//Variaveis utilizadas para realizar a leitura dos botoes de configuracao do
sistema
bool incremento, decremento, set = 0;

//Variaveis utilizadas para armazenar o estado de acionamento dos botoes
bool EstadoIncremento = 0, EstadoDecremento = 0, EstadoSet = 0;

//Variavel para armazenar o valor de ajuste de temperatura
unsigned int temperatura = 0;

//Variavel utilizada para armazenar o tempo de execucao atual do codigo
unsigned long int tempo_atual = 0;

//Variavel utilizada para armazenar o ultimo valor armazenado na variavel
tempo_atual
unsigned long ultimo_tempo = 0;

```

```

byte ValorAnterior = 0;

//Definicao da palavra memoria para representar o valor 120
#define EndVerif 120
#define EndTemp 125

void setup()
{
    Serial.begin(9600);
    //put your setup code here, to run once:
    lcd.begin(16, 2);

    delay(500);

    lcd.setCursor(4,0);
    lcd.print("TCC 2023");
    lcd.setCursor(3,1);
    lcd.print("AUTOMACAO");

    delay(3000);

    //Inicialização do funcionamento do Sensor DS18B20
    sensors.begin();
    lcd.clear();

    pinMode(13, OUTPUT);
}

void loop()
{
    set = digitalRead(12);

    if(set == 1)
    {
        lcd.clear();
        lcd.setCursor(2,0);
        lcd.print("Reconfigurar?");

        lcd.setCursor(0,1);
        lcd.print("Sim (+) Nao (-)");

        do
        {
            incremento = digitalRead(10); //Leitura do estado do pino digital do botao
            incremento

```

```

    decremento = digitalRead(11); //Leitura do estado do pino digital do
botao decremento

    delay(50);

    if(incremento == 1 && decremento == 0)
    {
        verificador = 0;
        EEPROM.write(EndVerif,0);
    }

    }while(incremento == 0 && decremento == 0);

    delay(200);
    lcd.clear();
}

verificador = EEPROM.read(EndVerif);

if(verificador != 75)
{

    lcd.setCursor(3,0);
    lcd.print("Configurar");
    lcd.setCursor(0,1);
    lcd.print("Temperatura:");

    temperatura = EEPROM.read(EndTemp); //Leitura do valor da posicao
EndTemp

    do
    {
        incremento = digitalRead(10); //Leitura do estado do pino digital do
botao incremento
        decremento = digitalRead(11); //Leitura do estado do pino digital do
botao decremento
        set = digitalRead(12); //Leitura do estado do pino digital do botao
set

        delay(50);

        if(incremento == 1 && EstadoIncremento == 0)
        {
            temperatura++; //Incrementa o valor da temperatura
            EstadoIncremento = 1; //Armazena 1 na variavel incremento
            lcd.setCursor(13,1); //Configura o cursor na posição da segunda
linha e coluna 13
            lcd.print(" ");
        }
    }
}

```

```

        if(incremento == 0 && EstadoIncremento == 1)
        {
            EstadoIncremento = 0; //Armazena 0 na variavel decremento para
indicar o estado atual do botao incremento - desacionado
        }

        if(decremento == 1 && EstadoDecremento == 0)
        {
            temperatura--; //Decrementa o valor da temperatura
            EstadoDecremento = 1; //Armazena 0 na variavel decremento para
indicar o estado atual do botao decremento
            lcd.setCursor(13,1); //Configura o cursor na posição da segunda
linha e coluna 13
            lcd.print(" ");
        }

        if(decremento == 0 && EstadoDecremento == 1)
        {
            EstadoDecremento = 0; //Armazena 0 na variavel decremento para
indicar o estado atual do botao decremento - desacionado
        }

        lcd.setCursor(13,1); //Configura o cursor na posição da segunda linha e
coluna 13
        lcd.print(temperatura); //Imprime o valor da temperatura na tela

    }while(set != 1);

    lcd.clear();
    EEPROM.write(EndVerif,75); //Salva o valor 75 na posição EndVerif
    EEPROM.write(EndTemp, temperatura); //Salva o valor da temperatura
configurada na posição EndTemp
}

//lcd.clear();//Limpar a tela do LCD

//Configurar o sensor DS18B20 e fazer a leitura do seu valor digital
Serial.print(" Solicitando temperatura...");
sensors.requestTemperatures(); // Envia o comando para obter leituras
de temperatura
Serial.println("Leitura realizada!");
/*****/
Serial.print("A Temperatura e:");
Serial.println(sensors.getTempCByIndex(0));
//O Index e utilizado, pois pode existir mais de um sensor conectado ao
barramento

float TempSensor = sensors.getTempCByIndex(0);
//float temperatura = sensors.getTempC(sensor1);

```

```
if(TempSensor != ValorAnterior)
{
  lcd.setCursor(2,0);
  lcd.print("Temperatura");
  lcd.setCursor(4,1);
  lcd.print(TempSensor);
  lcd.print((char)223);
  lcd.print("C");
```

```
ValorAnterior = TempSensor;
```

```
if(TempSensor >= temperatura)
{
  digitalWrite(13, HIGH);
}
```

```
if(TempSensor <= temperatura)
{
  digitalWrite(13, LOW);
}
}
```

```
}
```

### 13 Lista de Materiais

Todos os componentes eletrônicos utilizados na Tabela 1 abaixo:

ITEM	DESCRIÇÃO	QTDA	V. UNI.	V. TOTAL
1	Placa Arduino Uno	R\$ 80,00	1	R\$ 80,00
2	Sensor de temperatura DS18B20	R\$ 10,70	1	R\$ 10,70
3	Tela LCD 16X2	R\$ 15,50	1	R\$ 15,50
4	Barra de pinos macho	R\$ 1,17	1	R\$ 1,17
5	Potenciômetro 10K	R\$ 2,24	1	R\$ 2,24
6	Resistor 4,7K ohm	R\$ 0,50	1	R\$ 0,50
7	Protoboard 830 pontos	R\$ 14,50	1	R\$ 14,50
8	Botão 12x12x7,3mm	R\$ 0,82	2	R\$ 1,64
9	Capa Verde	R\$ 0,55	1	R\$ 0,55
10	Capa Vermelha	R\$ 0,55	1	R\$ 0,55
11	Jumper macho 20cm	R\$ 0,20	10	R\$ 2,00
12	Jumer macho x fêmea 20cm	R\$ 0,20	10	R\$ 2,00
13	Módulo réle 4 canais 12v	R\$ 45,50	1	R\$ 45,50
14	Tomada Sobrepor 2+T	R\$ 15,00	1	R\$ 15,00
15	Resistor 1k	R\$ 0,50	1	R\$ 0,50
16	lampada bolinha 220V Vermelha	R\$ 5,50	1	R\$ 5,50
17	lampada bolinha 220V Amarela	R\$ 5,50	1	R\$ 5,50
18	Receptáculo de porcelanato	R\$ 3,00	2	R\$ 6,00
19	Disjuntor bipolar 20A curva C	R\$ 35,00	1	R\$ 35,00
20	Mini contator 220V	R\$ 60,00	1	R\$ 60,00
21	Resistência Blidada 1000W 220V	R\$ 195,00	1	R\$ 195,00
22	Tubo Acrilico 3mm Interno X 50cm	R\$ 14,00	1	R\$ 14,00
23	Terminal cocector sindal	R\$ 10,99	1	R\$ 10,99
24	Mdf 15mm 500x500	R\$ 40,00	1	R\$ 40,00
25	Cabo PP 3x1,5mm 500V	R\$ 4,00	1	R\$ 4,00

Tabela 1 – Componentes eletônico utilizados no trabalho.

## 14 Quantitativo e valor total do material

Abaixo na tabela 2 que consta valor e quantidade de item:

Coluna1	Coluna2	
NÚMERO TOTAL DE ITENS		R\$ 25,00
VALOR TOTAL DE QUANTIDADE		R\$ 516,92
VALOR TOTAL DO PEDIDO		R\$ 568,34

Tabela 2 – Valor do Material



## 15 Conclusão

Finalmente concluímos que através desse trabalho de controle térmico de elemento líquido com Arduino controle ON/OFF, foi possível obter o resultado esperado e implementar diversas funcionalidades. Portanto, entre as funcionalidades, foi possível criar um dispositivo com os seguintes recursos:

1. Criação de tela de confirmação, para alteração do valor da temperatura;
2. Função para salvar o valor de temperatura ajustada pelo usuário, mesmo que o sistema desligue e ligue novamente;
3. Criação de um teclado para configurar o valor da temperatura, de modo a utilizar apenas 2 botões;
4. Trabalhar com o sensor de temperatura DS18B20;
5. Criação de Menus no LCD 16 x 2.

Concluímos que através desse trabalho, vem fechar com chave de ouro, esse período de aprendizagem e conhecimento. Somos gratos por essa conquista e chegar ao fim desse curso.

## 16 REFERÊNCIAS

Afonso, Antonio Pereira Eletrônica: circuitos elétricos / Antonio Pereira Afonso, Enio Filoni (autores); Tsuyoshi Okihiro (revisor); Jun Suzuki (coordenador). -- São Paulo: Fundação Padre Anchieta, 2011 (Coleção Técnica Interativa. Série Eletrônica, v. 1)

Pinto, Luiz Fernando Teixeira Eletrônica: eletrônica analógica / Luiz Fernando Teixeira Pinto, Rômulo Oliveira Albuquerque (autores); Luiz Tetsuharu Saito (revisor); Jun Suzuki (coordenador). -- São Paulo: Fundação Padre Anchieta, 2011 (Coleção Técnica Interativa. Série Eletrônica, v. 2)

Diago, Ronaldo Eletrônica: eletrônica digital / Ronaldo Diago, Valder Moreira Amaral (autores); Edson Horta (coautor); Marcos Vagner Zamboni (revisor); Jun Suzuki (coordenador). -- São Paulo: Fundação Padre Anchieta, 2011. (Coleção Técnica Interativa. Série Eletrônica, v. 4)

Parede, Ismael Moura Eletrônica: automação industrial / Ismael Moura Parede, Luiz Eduardo Lemes Gomes (autores); Edson Horta (coautor), Luiz Carlos da Cunha e Silva (revisor); Jun Suzuki (coordenador). -- São Paulo: Fundação Padre Anchieta, 2011 (Coleção Técnica Interativa. Série Eletrônica, v. 6)