

```

1  /*
2  Controlador de Temperatura v.1.0
3
4  Faz o controle da temperatura através de um sensor DS18B20 e exibe
5  as informações no display LCD 16x2 I2C. Possui dois módulos relê, sendo
6  um módulo de 1 canal para acionar o motor e outro de dois canais para
7  mudar a velocidade de um motor ventilador.
8
9  Circuito: O esquema está disponível para download.
10
11  Criado em 15/11/2020
12  por Leandro Rafael Celista
13
14  */
15  //Inclui as bibliotecas necessárias
16  #include <OneWire.h>
17  #include <DallasTemperature.h>
18  #include <Wire.h>
19  #include <LiquidCrystal_I2C.h>
20  //Instancia o objeto lcd
21  LiquidCrystal_I2C lcd(0x27,16,2);
22
23  #define ONE_WIRE_BUS 2
24  //Instancia o barramento e o sensor
25  OneWire oneWire(ONE_WIRE_BUS);
26  DallasTemperature sensors(&oneWire);
27
28  //Define os botões, quando pressionar liga e desliga retorna 0, quando pressionar o
29  botão
30  //menos temperatura retorna 1 e assim para os outros botões.
31  #define btLigDes 0
32  #define btMenos 1
33  #define btMais 2
34  #define btVent 3
35  #define btNenhum 4
36
37  //Define os pinos dos atuadores
38  #define buzzer 9
39  #define releBaixa 13
40  #define releAlta 12
41  #define releMotor 8
42
43  //Cria as variáveis globais, elas estarão disponíveis para o programa todo.
44  int botaoLido = 0; //Armazena o retorno do botão pressionado
45  int visor = 0; //Indica para o programa se o visor está aceso ou apagado.
46  //1=aceso 0=apagado.
47  int tempProg = 25; //Armazena o valor da temperatura programada, inicia com 25°C.
48  int tempC = 0; //Armazena o valor da temperatura lida a partir do sensor
49  int tDif = 2; //Diferença de Temperatura, inicia com 2. Evita que o compressor
50  //ligue e
51  //desligue várias vezes sempre que atingir a temperatura.
52  int velocidade = 1; //Armazena o valor da velocidade atual do ventilador. 1=Baixa
53  //2=Alta.
54
55  unsigned long tempoAtual = millis(); //Armazena o tempo desde que o Arduino foi
56  energizado.
57  unsigned long tempoDesligou = 0; //Armazena o tempo em que o compressor (motor)
58  foi desligado.
59  unsigned long intervaloMotor = 10000; //Proteção de 10 segundos (3min = 180000)
60  //Só vai ligar o motor novamente se o tempo for
61  //maior que o intervaloMotor
62  //caso o motor tenha sido desligado recentemente.
63  unsigned long tempoAnterior = 0; //Armazena o tempo anterior a leitura dos sensores.
64  unsigned long intervalo = 10000; //Ler a temperatura a cada 10 segundos
65
66  //Função para ler o botão pressionado
67  int lerBotoes(){
68      botaoLido = analogRead(0); //Faz a leitura do valor no pino A0
69      if(botaoLido > 900){ //Mudei pra 900 porque 1000 estava acionando o botao
70          btVent
71          return btNenhum; //Se o valor for maior que 900 então retorna btNenhum
72          //que é 4.
73      }
74  }

```

```

65     if(botaoLido < 150){           //Se o valor for menor que 150 então retorna btLigDes
        que é 0.
66         return btLigDes;
67     }
68     if(botaoLido < 300){
69         return btMenos;
70     }
71     if(botaoLido < 450){
72         return btMais;
73     }
74     if(botaoLido < 900){           //Aqui fica 900 também. Se você mudar o valor da condição,
        mude aqui também
75         return btVent;           //para que os valores fiquem iguais.
76     }
77     return btNenhum;             //Se nenhum botão foi pressionado, retorna btNenhum que é 4.
78 }
79
80 //Função para obter a temperatura dos sensores.
81 void obterTemp(){
82     sensors.requestTemperatures(); //Faz a requisição para o objeto sensors.
83     tempC=sensors.getTempCByIndex(0); //Armazena o retorno da função getTempCByIndex()
        na variável tempC,
84 }
85                                     //ou seja, vai devolver o valor da temperatura
        atual.
86
87 //Função para mostrar as informações no display
88 void mostrarInfo(){
89     visor=1;                       //Sinaliza para o programa que o visor está ligado.
90     lcd.backlight();               //Acende a luz de fundo do display.
91     lcd.setCursor(0,0);            //Coloca o cursor na coluna zero e linha zero.
92     lcd.print("Temperatura: ");    //Escreve Temperatura:
93     lcd.setCursor(0,1);            //Coloca o cursor na coluna zero e linha um.
94     lcd.print(tempC);               //Escreve a temperatura ambiente através do valor da
        variável tempC.
95     lcd.print(char(223));           //Imprime a bolinha do grau
96     lcd.print("C");                //Escreve C
97     lcd.setCursor(14,1);           //Coloca o cursor na coluna quatorze e linha um.
98     lcd.print(tempProg);           //Escreve a temperatura programada através do valor da
        variável tempProg.
99     lcd.display();                 //Exibe todas as informações na tela do display.
100 }
101
102 //Função para ocultar as informações do display
103 void ocultarInfo(){
104     visor=0;                       //Sinaliza para o programa que o visor está desligado.
105     lcd.noBacklight();              //Apaga a luz de fundo do display.
106     lcd.clear();                    //Limpa todas as informações que foram escritas no display.
107     lcd.noDisplay();               //Apaga o display.
108 }
109
110 //Função para bipar
111 void bipar(int tempo, int freq){ //Recebe como parâmetro o tempo em milisegundos e a
        frequência.
112     tone(buzzer,freq);             //Emite o bipe no pino 9, veja que esse valor vem da
        variável buzzer.
113     delay(tempo);                  //Aguarda o tempo em milisegundos
114     noTone(buzzer);                //Corta o bipe do buzzer.
115 }
116
117 //Função que troca a velocidade do ventilador
118 void ajustarVelocidade(){
119     if(velocidade==1){              //Se a velocidade atual é baixa=1 então muda pra
        alta, portanto
120         digitalWrite(releBaixa,LOW); //Desliga o relê da baixa
121         delay(100);                 //Aguarda 100ms
122         digitalWrite(releAlta,HIGH); //Liga o relê da alta
123         velocidade=2;               //e sinaliza pro programa que agora a velocidade
        atual é alta=2.
124     }else{                          //Caso contrário, se velocidade não for igual a 1,
125         digitalWrite(releAlta,LOW); //significa que a velocidade atual é alta, então
        desliga o relê,
126         delay(100);                 //Aguarda 100ms

```

```

127     digitalWrite(releBaixa,HIGH); //Liga o rele da baixa
128     velocidade=1;                //sinaliza que agora a velocidade é baixa.
129 }
130 }
131
132 //Função para verificar a temperatura de forma a ligar ou cortar o relê do
compressor (motor).
133 void verificarTemp(){
134     tempoAtual=millis();          //Atualiza o valor de tempoAtual.
135     if(tempC>=tempProg+tDif){     //Se a temperatura atual for maior ou igual a
temperatura programada
136                                     //mais a diferença de temperatura então liga o motor
mas antes...
137     if(tempoAtual - tempoDesligou >= intervaloMotor){ //verifique se o tempo desde
que desligou é
138         digitalWrite(releMotor,HIGH);                //maior ou igual ao
intervaloMotor, se for, liga.
139     }
140 }
141 if(tempC<=tempProg-tDif){ //Se a temperatura atual for menor ou igual a
temperatura programada
142                                     //menos a diferença de temperatura então
143     tempoDesligou=tempoAtual;        //atualiza o valor da variavel tempoDesligou e
144     digitalWrite(releMotor,LOW);    //desliga o motor.
145 }
146 }
147
148 //Função para desligar todos os relês
149 void desligarReles(){
150     digitalWrite(releBaixa,LOW); //Desliga o relê da baixa.
151     digitalWrite(releAlta,LOW);  //Desliga o relê da alta.
152     digitalWrite(releMotor,LOW); //Desliga o relê do motor.
153     tempoAtual=millis();         //Atualiza o valor de tempoAtual.
154     tempoDesligou=tempoAtual;    //Atualiza o valor de tempoDesligou.
155 }
156
157 void setup() {
158     sensors.begin();              //Inicializa o sensor.
159     lcd.init();                  //Inicializa o display.
160     pinMode(buzzer,OUTPUT);      //Define como saída o pino do buzzer.
161     pinMode(releBaixa,OUTPUT);   //Define como saída o pino do relê da baixa.
162     pinMode(releAlta,OUTPUT);    //Define como saída o pino do relê da alta.
163     pinMode(releMotor,OUTPUT);   //Define como saída o pino do relê do motor.
164     digitalWrite(releBaixa,LOW); //Coloca como desligado o relê da baixa.
165     digitalWrite(releAlta,LOW);  //Coloca como desligado o relê da alta.
166     digitalWrite(releMotor,LOW); //Coloca como desligado o relê do motor.
167                                     //Ou seja começa tudo desligado.
168     obterTemp();                 //Faz a leitura do sensor
169     bipar(200,2000);             //Bipa durante 200ms em 2000hz.
170 }
171
172 //Programa Principal
173 void loop() {
174     switch(lerBotoes())           //Faz a leitura dos botões e verifica se as condições abaixo
175     {                             //correspondem ao retorno do valor de lerBotoes().
176         case 0:                   //Caso lerBotoes tenha retornado zero, pressionou Liga
Desliga
177         {
178             if(visor==1)          //Se o visor estiver ligado, então a intenção do usuário é
desligar
179             {
180                 //o sistema, então
181                 ocultarInfo();    //oculta as informações
182                 desligarReles(); //e desliga os relês.
183             }else{                //Caso contrário
184                 mostrarInfo();    //Mostra as informações
185                 digitalWrite(releAlta,LOW); //Desliga o relê da alta.
186                 delay(100);        //Aguarda 100ms
187                 digitalWrite(releBaixa,HIGH); //Liga o relê da baixa.
188                 velocidade=1;      //Sinaliza a velocidade atual = 1, baixa.
189             }
190             bipar(150,1000);       //Emite um bipe durante 150ms em 1000hz.
191             break;                //Sinaliza para interromper a condição.
192                                     //Se você não colocar o break o programa continua a executar

```

```

192                                     //o que estiver abaixo.
193     }
194
195     case 1:          //Caso o retorno de ler botoes seja igual a 1, pressionou menos
temp.
196     {
197         if(visor==1) //Se o visor estiver ligado e...
198         {
199             if(tempProg>=18) //se a temperatura programada for maior ou igual a 18
entao...
200             {
201                 tempProg=tempProg-1; //diminui uma unidade do valor atual de tempProg.
202             }
203             bipar(150,1000); //Bipa
204             mostrarInfo();    //Mostra as informações
205         }
206         break;
207     }
208
209     case 2: //Pressionou Mais Temp
210     {
211         if(visor==1) //Se o visor estiver ligado...
212         {
213             if(tempProg<=29) //e se a temperatura programada for menor ou igual a 29
214             {
215                 tempProg=tempProg+1; //aumenta o valor atual de tempProg em uma unidade.
216             }
217             bipar(150,1000); //Bipa
218             mostrarInfo();    //Mostra as informações no display
219         }
220         break;
221     }
222
223     case 3: //Pressionou Mudar Velocidade
224     {
225         if(visor==1)
226         {
227             ajustarVelocidade(); //Muda a velocidade de ventilação
228             bipar(150,1000); //Bipa
229             mostrarInfo();    //Mostra as informações
230         }
231         break;
232     }
233
234     case 4: //Nao apertou nenhum botão, então não faz nada.
235     {
236         break;
237     }
238     delay(50); //Aguarda 50ms que é o tempo necessário pra dar tempo do programa
239                //verificar se algum botão está pressionado. Você pode mudar esse
                //valor para
240                //150 ou um pouco mais se notar que a mudança das informações está
                //muito rápida.
241                //Por exemplo você aperta pra desligar e ele liga novamente em
                //seguida.
242 }
243
244 //Ok fez a leitura dos botões então continua...
245 if(visor==1) //Se o visor estiver ligado
246 {
247     tempoAtual=millis(); //Atualiza a variável tempoAtual, aqui é o temporizador da
leitura de temperatura.
248     if(tempoAtual-tempoAnterior >= intervalo) //Se a última leitura foi ha mais de
10 segundos...
249     {
250         tempoAnterior=tempoAtual; //Atualiza a variável de tempoAnterior
251         obterTemp();    //Solicita ao sensor a temperatura atual e armazena na
variável tempC
252         mostrarInfo();    //Mostra as informações na tela
253         verificarTemp(); //Verifica a Temperatura para acionar ou cortar o relê do motor.
254     } //Fim do programa, retorna para o começo.
255 }
256

```