

```

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     }

```

```

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

```

```

127     digitalWrite(releBaixa,HIGH); //Liga o relé 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
133 compressor (motor).
134 void verificarTemp(){
135     tempoAtual=millis();      //Atualiza o valor de tempoAtual.
136     if(tempC>=tempProg+tDif){ //Se a temperatura atual for maior ou igual a
137         temperatura programada
138             //mais a diferença de temperatura então liga o motor
139             //mas antes...
140             if(tempoAtual - tempoDesligou >= intervaloMotor){ //verifique se o tempo desde
141                 que desligou é
142                     digitalWrite(releMotor,HIGH);           //maior ou igual ao
143                     intervaloMotor, se for, liga.
144             }
145     }
146 }
147
148 //Função para desligar todos os relês
149 void desligarReles(){
150     digitalWrite(releBaixa,LOW); //Desliga o relé 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
177             Desliga
178             {
179                 if(visor==1)    //Se o visor estiver ligado, então a intenção do usuário é
180                 desligar
181                 {
182                     //o sistema, então
183                     ocultarInfo(); //oculta as informações
184                     desligarReles(); //e desliga os relês.
185                 }else{          //Caso contrário
186                     mostrarInfo(); //Mostra as informações
187                     digitalWrite(releAlta,LOW); //Desliga o relê da alta.
188                     delay(100);        //Aguarda 100ms
189                     digitalWrite(releBaixa,HIGH); //Liga o relê da baixa.
190                     velocidade=1;       //Sinaliza a velocidade atual = 1, baixa.
191                 }
192             bipar(150,1000); //Emite um bipe durante 150ms em 1000hz.
193             break;           //Sinaliza para interromper a condição.
194             //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
196     temp.
197     {
198         if(visor==1) //Se o visor estiver ligado e...
199         {
200             if(tempProg>=18) //se a temperatura programada for maior ou igual a 18
201             entao...
202             {
203                 tempProg=tempProg-1; //diminui uma unidade do valor atual de tempProg.
204             }
205             bipar(150,1000); //Bipa
206             mostrarInfo(); //Mostra as informações
207         }
208         break;
209     }
210
211     case 2: //Pressionou Mais Temp
212     {
213         if(visor==1) //Se o visor estiver ligado...
214         {
215             if(tempProg<=29) //e se a temperatura programada for menor ou igual a 29
216             {
217                 tempProg=tempProg+1; //aumenta o valor atual de tempProg em uma unidade.
218             }
219             bipar(150,1000); //Bipa
220             mostrarInfo(); //Mostra as informações no display
221         }
222         break;
223     }
224
225     case 3: //Pressionou Mudar Velocidade
226     {
227         if(visor==1)
228         {
229             ajustarVelocidade(); //Muda a velocidade de ventilação
230             bipar(150,1000); //Bipa
231             mostrarInfo(); //Mostra as informações
232         }
233         break;
234     }
235
236     case 4: //Nao apertou nenhum botão, então não faz nada.
237     {
238         break;
239     }
240     delay(50); //Aguarda 50ms que é o tempo necessário pra dar tempo do programa
241     //verificar se algum botão está pressionado. Você pode mudar esse
242     //valor para
243     //150 ou um pouco mais se notar que a mudança das informações está
244     //muito rápida.
245     //Por exemplo você aperta pra desligar e ele liga novamente em
246     //seguida.
247 }
248
249 //Ok fez a leitura dos botões então continua...
250 if(visor==1) //Se o visor estiver ligado
251 {
252     tempoAtual=millis(); //Atualiza a variável tempoAtual, aqui é o temporizador da
253     //leitura de temperatura.
254     if(tempoAtual-tempoAnterior >= intervalo) //Se a última leitura foi há mais de
255     //10 segundos...
256     {
257         tempoAnterior=tempoAtual; //Atualiza a variável de tempoAnterior
258         obterTemp(); //Solicita ao sensor a temperatura atual e armazena na
259         //variável tempC
260         mostrarInfo(); //Mostra as informações na tela
261         verificarTemp(); //Verifica a Temperatura para acionar ou cortar o relé do motor.
262     } //Fim do programa, retorna para o começo.
263 }

```