

# Controle de temperatura On/Off ajustável

Yan Correa <sup>1</sup>

<sup>1</sup>Faculdade de Computação - Universidade Federal de Mato Grosso do Sul (UFMS)

**Resumo.** *Este relatório tem como objetivo apresentar a execução do trabalho proposto na disciplina de Microcontroladores e Aplicações, no qual faz-se a prototipação de um controlador de temperatura On/Off por meio do microcontrolador C8051F040.*

## 1. Especificações do projeto

Deverá medir continuamente a temperatura de um sensor analógico e mostrá-la no LCD gráfico. Se a temperatura baixar de um valor predefinido, deverá acionar uma saída digital (elemento de aquecimento). O valor predefinido poderá ser inserido via teclado a qualquer momento e deverá ser mantido quando a energia é desligada.

## 2. Materiais utilizados

- Placa BIG8051
- Protoboard
- Resistores de  $4.7k\Omega$  e  $100k\Omega$
- Capacitor cerâmico de  $0.01\mu F$
- Relé
- Lâmpada incandescente 127 Volts
- Jumpers

## 3. Montagem do circuito

Para o experimento, utilizou-se o sensor LM35 para ler a temperatura. Seu esquema de ligação pode ser visto na Figura 1. O módulo relé foi utilizado para acionar o aquecedor, neste caso simulado por uma lâmpada. É importante observar o filtro RC conectado ao terminal central do LM35. Tal filtro atenua as altas frequências do sinal de saída do sensor antes de entrar no conversor A/D, fazendo com que a temperatura, mostrada pelo microcontrolador através do GLCD, não fique variando excessivamente. Sua frequência de corte pode ser calculada com a equação 1. Os valores de  $R$  e  $C$  foram obtidos no datasheet do sensor [Texas ].

$$f_c = \frac{1}{2\pi RC} \quad [Hz] \quad (1)$$

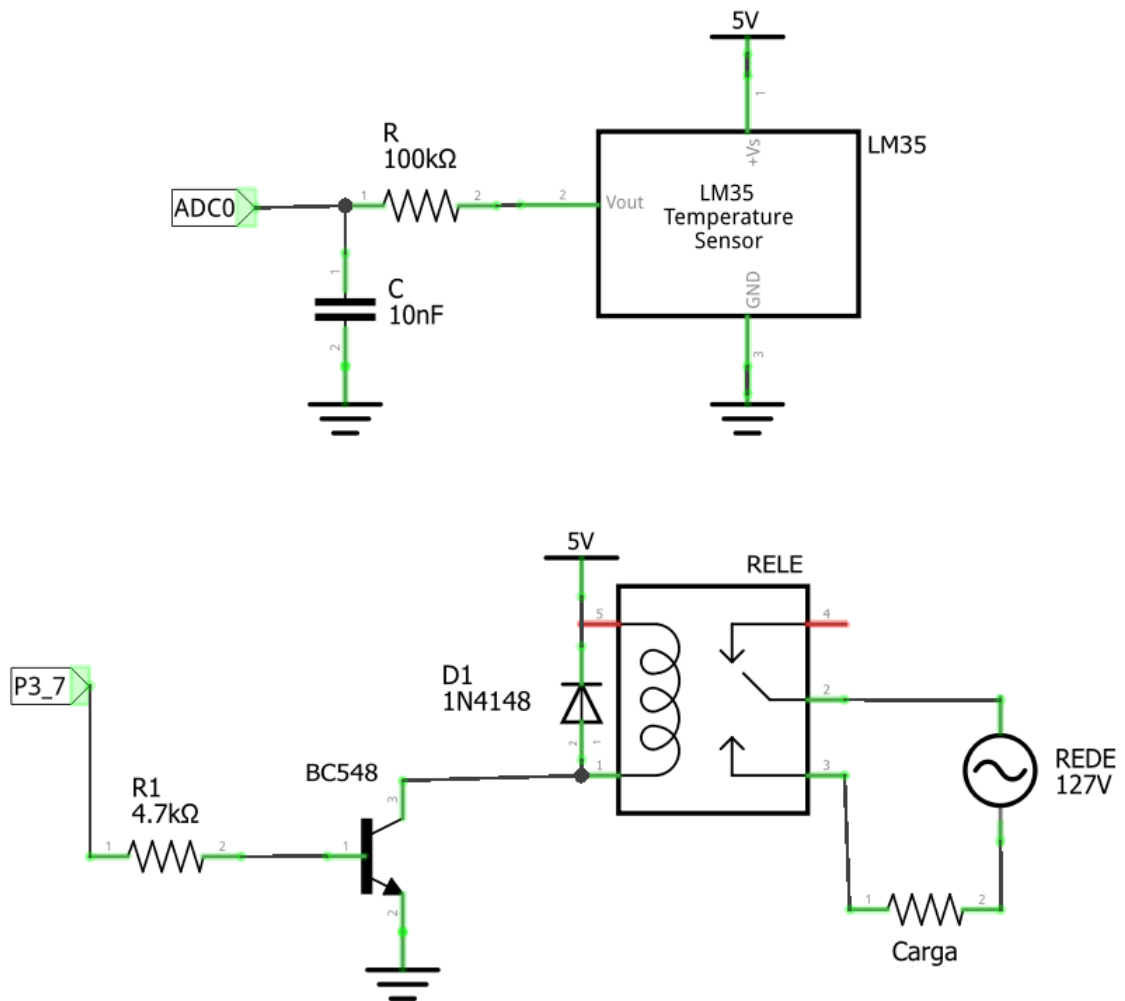
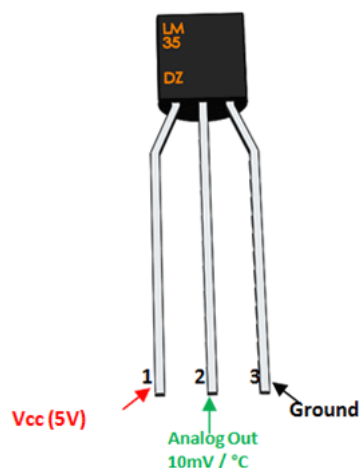


Figura 1. Esquemático do circuito com o sensor LM35

#### 4. O sensor LM35

O LM35 (Figura 2) é um sensor de temperatura que possui a faixa de medição de  $-55^{\circ}\text{C}$  à  $150^{\circ}\text{C}$ . Observando seu datasheet, obtém-se sua função de transferência, ou seja, a relação entre a temperatura [em  $^{\circ}\text{C}$ ] e a tensão de saída do sensor [em *Volts*], o que resulta em uma escala de  $10\text{mV}/^{\circ}\text{C}$ . Daí, para se medir a temperatura em graus celsius, deve-se multiplicar por 100 a tensão de saída do sensor. Sua precisão é de  $0.5^{\circ}\text{C}$  em temperaturas em torno de  $25^{\circ}\text{C}$ .



**Figura 2. Imagem do sensor LM35**

Tendo conhecimento disso, é possível obter uma equação que relaciona o valor digitalizado na leitura do ADC0, seu número de bits (resolução) e o ganho para obter a temperatura. Assim, obteve-se a Equação 2.

$$T = \frac{ValorDigitalizado * V_{ref} * 100}{2^n * Ganho} \quad [^{\circ}C] \quad (2)$$

Utilizando  $V_{ref} = 2.43V$  e  $n = 12$  (tensão de referência e resolução do ADC, respectivamente), pode-se criar a Tabela 1 que relaciona o ganho escolhido com a resolução da temperatura e a temperatura máxima que poderá ser lida pelo sensor. Para efeitos práticos, escolheu-se o ganho igual a 4, pois apresenta uma boa resolução e a temperatura máxima é aceitável para o projeto.

Ganho	Resolução [ $^{\circ}C$ ]	$T_{max}$ [ $^{\circ}C$ ]
0.5	0.12	150
1.0	0.06	150
2.0	0.03	121.47
4.0	0.015	60.73
8.0	0.0076	30.36
16.0	0.0038	15.18

**Tabela 1. Tabela relacionando ganho, resolução e temperatura máxima**

## 5. Descrição do Firmware

No laço principal, da linha 3 a 10 são feitas as inicializações básicas; a inicialização do dispositivo, configuração da página dos SFR's e, por fim, inicializa e limpa o display gráfico.

Na linha 17 é feita a leitura da memória EEPROM pra obter o valor da temperatura de referência setada na última vez em que o microcontrolador foi usado.

Dentro do while da linha 20 existe o *loop principal*. Lá é feita a leitura do sensor de temperatura através do ADC0 da placa BIG8051, na linha 24. Na linha 25 é feita a conversão do valor digital que se obteve para o valor da temperatura mensurada em graus celsius, utilizando a Equação 2. Então, é impresso no GLCD a temperatura atual, e é lido da EEPROM a temperatura de referência setada pelo usuário e impressa logo abaixo.

Após isso, é feito o controle do sensor de temperatura, a partir da linha 33, onde é verificado se a temperatura atual é menor que a temperatura de referência. Se confirmado, liga-se o aquecedor. A linha 39 verifica se a temperatura atual é mais alta que a temperatura de referência mais uma constante para que, assim, o sensor não fique desligando e ligando de forma intermitente, criando um laço de histerese. Observe também que as leituras são feitas aproximadamente a cada 500ms, conforme o delay na linha 47.

```
1 int main()
2 {
3     Init_Device();
4     SFRPAGE= LEGACY_PAGE;
5
6     inicia_lcd();
7     limpa_glcd(ESQ);
8     limpa_glcd(DIR);
9     conf_pag(0, ESQ);
10    conf_pag(0, DIR);
11
12    //reseta o barramento SMBus
13    SMB0CN &= ~0x40;
14    SMB0CN |= 0x40;
15
16    IE=0x80;
17    temp_ref=le_eeprom(0xA0,0);
18    IE=0xA0;
19
20    while(1)
21    {
22
23        //le sensor
24        ladc=le_adc0(AIN00, GAIN4);
25        temp_atual = ladc*0.05932617/4; //ldc*100*(Tref
           /4096)/GAIN
26
27        //imprime temperatura atual
28        printf_fast_f("\x02 T Atual: %2.1f%c",
           temp_atual,128);
29        IE=0x20;
30        temp_ref=le_eeprom(0xA0,0);
31        printf_fast_f("\x03 T Refer: %2d.0%c", temp_ref
           ,128);
32        IE=0xA0;
33        if(temp_atual<temp_ref)
34        {
```

```

35         if(AQUECEDOR==OFF)
36             AQUECEDOR=ON;
37         printf_fast_f("\x06 Aquecedor ON ");
38     }
39     else if(temp_atual>=(temp_ref+DIFTEMP))
40     {
41         if(AQUECEDOR==ON)
42             AQUECEDOR=OFF;
43         printf_fast_f("\x06 Aquecedor OFF");
44     }
45
46
47     delay_ms(500);
48 }
49
50
51 return 0;
52 }

```

Abaixo, na função `interrupt_tc2`, é feita a leitura dos botões da matriz de teclas. A cada 20ms ocorre uma interrupção no TC2 e esta função é chamada. Nela verifica-se se alguma tecla da porta 0 (P0) está sendo pressionada, utilizando *polling*. Observe que a variável `temp_ref`, foi declarada como *volatile* para não sofrer nenhuma otimização do compilador, visto que ela é usada na *main* e também em funções de interrupção. As funções chamadas por uma rotina de interrupção, por exemplo a *esc\_eeprom* e *le\_eeprom* foram declaradas como *reentrant* pois compartilham uma variável global (*tem\_ref*).

```

1 void interrupt_tc2() __interrupt 5
2 {
3     if(P0==0xFF && P1==0xFF)
4     {
5         ant=0;
6     }
7     else
8     {
9         if(ant==0)
10        {
11            ant=1;
12
13            if(le_tecla()==6 && temp_ref<TEMPMAX)
14            {
15                temp_ref++;
16                esc_eeprom(0xA0, 0, temp_ref);
17            }
18            else if(le_tecla()==7 && temp_ref>
19                TEMPMIN)
20            {
21                temp_ref--;
22                esc_eeprom(0xA0, 0, temp_ref);

```

```

23         }
24
25     }
26 }
27     TF2=0; //zera o flag de interrupcao do TC2
28 }

```

Abaixo, a função *le\_tecla* utilizando *polling* para retornar qual a porta (PO\_X), de 0 a 7, está sendo pressionada.

```

1 unsigned char le_tecla()
2 {
3     unsigned char pp0, kp0;
4     pp0=~P0;
5     kp0=0;
6     if(pp0!=0x00)
7     {
8         while(pp0!=0x01)
9         {
10             pp0=pp0>>1;
11             kp0++;
12         }
13         return kp0;
14     }
15     return 50;
16 }

```

## Conclusão

Conclui-se que foi possível desenvolver o projeto conforme as especificações requeridas, utilizando diversas técnicas aprendidas durante a disciplina de *Microcontroladores*. O tipo de controle utilizado foi o mais simples possível, no caso o *Controlador On-Off*, porém o projeto pode ser melhorado substituindo o sistema de potência e utilizando *PWM* em conjunto com um *Controlador PID* visando atenuar a oscilação da temperatura em torno do setpoint desejado.

## Referências

Texas, I. <http://pdf1.alldatasheet.com/datasheet-pdf/view/517588/ti1/lm35.html>, acessado em: 28 de maio. de 2018.