

Universidade de São Paulo-USP
Escola de Engenharia de São Carlos-EESC
SEL0629 - Aplicações de Microprocessadores I
Relatório Prática III

Vinicius William da Silva - Número USP: 11233842
Prof. Marcelo Andrade da Costa Vieira

São Carlos, 2023

1 Introdução

Nesta prática utilizei o microcontrolador PIC18F45k22 por meio do Kit de Desenvolvimento EasyPIC v.7 da MikroElektronika juntamente com um termômetro digital LM35 e *display* LCD (2x16).

O objetivo da prática consiste em aprender a utilizar o conversor AD embutido no kit para ler o valor de tensão em um potenciômetro e mostrar no *display* de LCD e utilizar o conversor AD para ler o valor de temperatura utilizando o CI LM35 e mostrar o valor lido no *display* de LCD.

A prática foi validada utilizando um osciloscópio para comparar os valores de tensão lidos do conversor AD com o valor real de tensão no potenciômetro e no termômetro.

2 Resultados Experimentais

2.1 Esquemático

O esquemático gerado está a seguir.

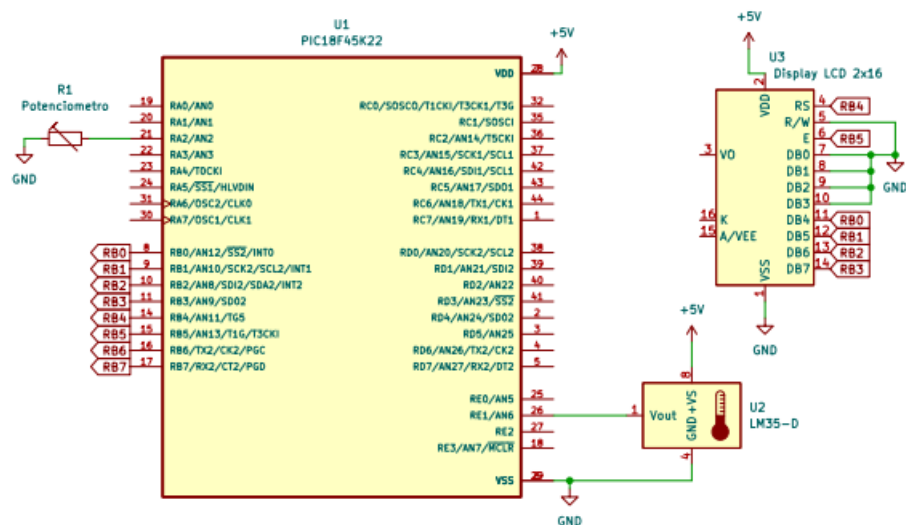


Figura 1: Esquemático do circuito.

2.2 Códigos e Simulação

Para esta prática, foi utilizada a biblioteca que auxilia o uso do LCD e do conversor analógico-digital (ADC) do MikroC. Assim, começo meu código definindo alguns elementos que são utilizados pela biblioteca de LCD:

```
// LCD module connections
sbit LCD_RS at RB4_bit;
sbit LCD_EN at RB5_bit;
sbit LCD_D4 at RB0_bit;
sbit LCD_D5 at RB1_bit;
sbit LCD_D6 at RB2_bit;
sbit LCD_D7 at RB3_bit;

sbit LCD_RS_Direction at TRISB4_bit;
sbit LCD_EN_Direction at TRISB5_bit;
sbit LCD_D4_Direction at TRISB0_bit;
sbit LCD_D5_Direction at TRISB1_bit;
```

```
sbit LCD_D6_Direction at TRISB2_bit;
sbit LCD_D7_Direction at TRISB3_bit;
// End LCD module connections
}
```

Feito isso, posso realizar a configuração inicial do LCD dentro do procedimento principal do programa (main). Basta apenas definir a saída da porta B como digital e chamar a biblioteca.

```
// Inicializa o LCD sem cursor
anselb = 0;
Lcd_Init();
Lcd_Cmd(_LCD_CLEAR);
Lcd_Cmd(_LCD_CURSOR_OFF);
```

No que tange ao potenciômetro e ao termômetro, basta apenas definir suas entradas (RA2 e RE1, respectivamente) como analógicas e definir o tri-state como entrada:

```
// Porta A (ligada ao potenciometro) como entrada analogica
ansela = 0b00000100;
trisa = 0xff;

// Porta E (ligada ao termometro) como entrada analogica
ansele = 0b00000010;
trise = 0xff;
```

Além disso, posso configurar o ADC utilizando a biblioteca e configurando alguns registradores do PIC. No registrador ADCON1, não são necessárias mudanças. Para definir a escala do conversor, utilizo o gerador de tensão interna ao PIC habilitando o FVR no registrador VREFCON0. Por último, opto por utilizar o modo de 10 bits do conversor AD.

```
// Inicializa o conversor AD
ADC_Init();
```

```
adcon1 = 0;
vrefcon0.f7 = 1;    // Enable FVR
adfm_bit = 1;       // 10 bit mode
```

Finalmente, com os dispositivos todos configurados, posso entrar no *loop* principal do programa. Neste *loop*, vou ler o valor do potenciômetro, convertê-lo para digital e exibi-lo na linha 1 do LCD. Após isso, farei o mesmo para o termômetro, exibindo sua medida na linha 2 do LCD.

Note que o potenciômetro gera tensões entre 0 V e 5 V, enquanto o termômetro gera tensões entre 0 V e 1 V, assumindo uma variação de temperatura entre 0 °C e 100 °C. Dessa forma, é necessário alternar a tensão de referência do ADC quando formos ler o potenciômetro (5 V) e quando formos ler o termômetro (1.024 V).

Além disso, vale notar que o termômetro aumenta sua tensão em 10 mV para cada °C, de forma que a conversão da tensão lida pelo ADC para graus Celsius é feita dividindo a tensão lida por 0.01 V, que é equivalente a multiplicar por 100.

```
while(1) {
    adcon1.f3 = 0;
    vrefcon0.f4 = 0;
    // Obtem dados do conversor AD
    potenciometro = ADC_Get_Sample(2);
    potenciometroFlt = 5.0f/1023.0f * potenciometro;
    sprintf(potenciometroStr, "Pot: %5.2fV",
potenciometroFlt);
    // Escreve na linha 1 do LCD
    Lcd_Out(1, 1, potenciometroStr);

    adcon1.f3 = 1;
```

```
vrefcon0.f4 = 1;
termometro = ADC_Get_Sample(6);
termometroFlt = 1.024f/1023.0f * termometro * 100.0f;
sprintf(termometroStr, "Ter: %4.1f°C", termometroFlt);
// Escreve na linha 2 do LCD
Lcd_Out(2, 1, termometroStr);

delay_ms(250);
Lcd_Cmd(_LCD_CLEAR);
}
```

3 Resultados e Discussão

Para verificar a precisão das medições e manipulações dos dados do conversor AD, realizei a leitura dos valores no osciloscópio. A voltagem no termômetro foi medida a partir do seu terminal central, enquanto a voltagem do potenciômetro foi medida a partir do pino RA2. Também capturei uma foto dos dados exibidos no *display* LCD. Os resultados obtidos são apresentados a seguir:

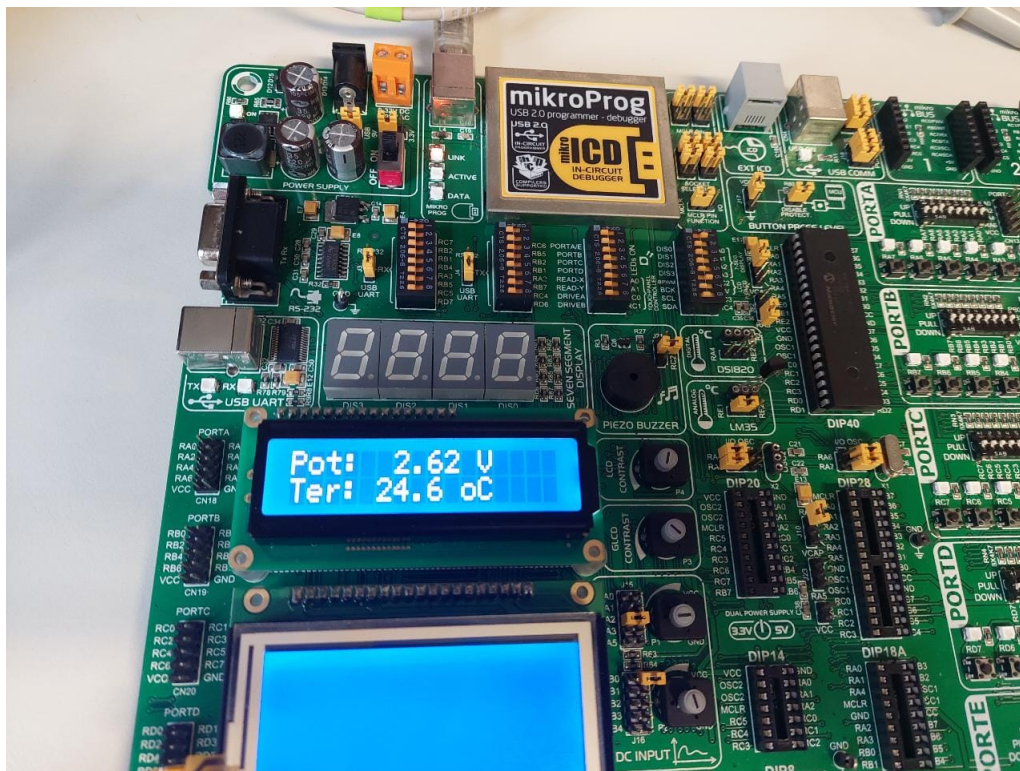


Figura 2: Medições apresentadas no LCD.

Sabendo que o manual do LM35 indica que sua curva teórica possui uma taxa de variação de $10 \text{ mV}/^{\circ}\text{C}$, pode-se montar a tabela abaixo.

Tabela 1: Resultados obtidos por osciloscópio.

Medida	LCD	Osciloscópio	Desvio (%)
Temperatura Ambiente	24.6 °C	245.37 mV	0.25
Tensão no Potenciômetro	2.62 V	2.55 V	2.67

A diferença observada na temperatura ambiente, provavelmente é devido ao erro de conversão analógico-digital, que possui uma resolução de $1.024\text{V}/2^{10} \approx 1 \text{ mV}$.

Em relação à diferença observada na medição da voltagem do potenciômetro, provavelmente, cometi algum erro grosseiro durante a etapa de medição, seja na configuração ou manipulação do osciloscópio.

4 Conclusão

Ao comparar as temperaturas obtidas diretamente do conversor AD com as esperadas ao medir a voltagem no termômetro através do osciloscópio, constatei que obtive resultados satisfatórios, pois o desvio foi de 0.25%. Embora as diferenças nas medições do potenciômetro não sejam muito baixas, elas são aceitáveis considerando que provavelmente cometi algum erro durante a medição.

Dessa forma, verifiquei que consegui utilizar o conversor AD de forma satisfatória com o PIC18F45K22 para ler a voltagem tanto do potenciômetro quanto do termômetro.