



I. DESCRIÇÃO DO PROJETO

Conversor A/D, ou Conversores Analógicos - Digitais, é um circuito capaz de transformar um valor de tensão numa informação binária. Segundo o *datasheet*, o PIC18F4520, que será utilizado no projeto, apresenta um módulo conversor com uma precisão de 10 bits, ou seja, um sinal de entrada de 5V é transformado em amostras com valores entre 0 e 1023, isto significa uma sensibilidade de 4.88mV.

A conversão AD é muito utilizada para realizarmos a leitura de sensores. No projeto, o conversor será utilizado para ler o sinal de um sensor de temperatura, assim, foi escolhido um potenciômetro para simular tal sensor.



Figura 1: Sensor de temperatura LM35 e Potenciômetro.

O Display LCD 16x2 utiliza o controlador HD44780, utilizado em toda indústria de LCD's como base de interface que pode ser operado em 4 ou 8-bits paralelamente. Sua conexão é feita através de 16 pinos, sendo 12 deles para conexão básica com o microcontrolador e 11 deles pinos de entrada/saída (I/O) e os demais pinos para ajuste de contraste através de potenciômetros, trimpots e afins e para a alimentação da backlight.

No projeto o Display tem como função exibir os valores de temperatura captados pelo sensor, no qual o usuário poderá selecionar a opção de escalas disponíveis (Celsius, Fahrenheit e Kelvin).

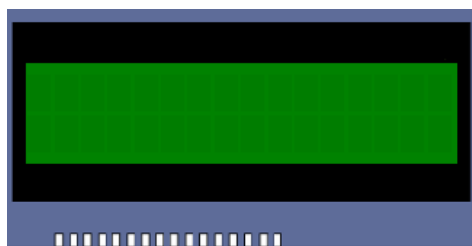


Figura 2: Display LCD do PICSimLab

Os pinos de I/O são os meios pelos quais os microcontroladores se comunicam com o mundo externo. Geralmente podem ser configurados como entradas ou saídas, dessa



forma é possível fazer a leitura de teclas (pino como entrada), acionar LEDs (pino como saída), ligar/desligar reles (pino como saída), por exemplo.

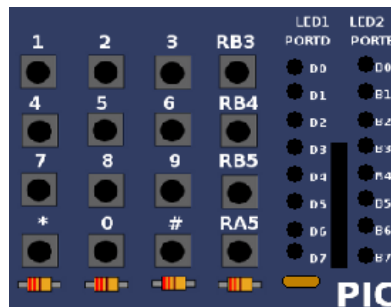


Figura 3: Botões e Led's do PICSIMLab.

Os pinos presentes no microcontrolador estão agrupados em ports, geralmente com 8 ou menos pinos, conforme a configuração do microcontrolador. Os ports são identificados como PORTA, PORTB, PORTC, etc. Cada port possui um grupo de registradores para configuração, leitura e escrita, que estudaremos mais adiante.

No projeto, serão utilizados dois botões, os quais tem como objetivo controlar o que será exibido no display e também três led's que servirão para indicar a qual escala está sendo exibida no display.

I. DIFICULDADES

No projeto, foi encontrado dificuldade na exibição dos valores de temperatura lidos pelo conversor A/D no display de LCD. Ao iniciar a simulação, apareciam caracteres especiais não representando o real valor lido, que para ser solucionado utilizou-se a função `itoa()`, a qual tem como objetivo a conversão de um valor inteiro (lido pelo conversor) para um vetor de char. Além disso, foi acrescentado o valor `0x30` em cada posição do vetor, de modo a converter para a tabela ASCII.

O código apresenta a solução encontrada.

```
120 void itoa(unsigned int val, char* str) {
121     str[0] = (val / 10000) + 0x30;
122     str[1] = ((val % 10000) / 1000) + 0x30;
123     str[2] = ((val % 1000) / 100) + 0x30;
124     str[3] = ((val % 100) / 10) + 0x30;
125     str[4] = (val % 10) + 0x30;
126     str[5] = 0 + 0x30;
127 }
```

Código 1: Função `itoa()`.

II. DESENVOLVIMENTO

Inicialmente, foram incluídas no projeto todas as bibliotecas necessárias para seu desenvolvimento, que são:



- Config.h
- Lcd.h
- Keypad.h
- Adc.h
- Bits.h
- Io.h
- Pic18f4520.h

Posteriormente, foi realizado mapeamento do hardware, utilizando a diretiva “#define” de modo a associar os pinos do microcontrolador para os botões (RIGHT_BUTTON e LEFT_BUTTON) e os led’s (CELSIUS_LED, KELVIN_LED e FAHR_LED) com constantes, facilitando o entendimento do programa. Além disso, algumas funções foram criadas para simplificar a main() do projeto. Sendo elas a função atraso_ms(), utilizada para geração de delay em milissegundos; readButton() para leitura das teclas e navegação pelo menu; itoa() para converter um int em um número da tabela ASCII; e as funções celsius(), kelvin() e fahr() para exibir o nome da escala, calcular o valor da temperatura com base na leitura do ADC e exibir o resultado. O arquivo main.c é mostrado abaixo:

```
30 void main() {
31     kpInit();
32     adcInit();
33     lcdInit();
34     while (HIGH) {
35         kpDebounce();
36         readButton();
37
38         switch (menuNum) {
39             case 1:
40                 bitSet(PORTB, CELSIUS_LED);
41                 bitClr(PORTB, FAHR_LED);
42                 bitClr(PORTB, KELVIN_LED);
43                 celsius();
44                 break;
45             case 2:
46                 bitClr(PORTB, CELSIUS_LED);
47                 bitSet(PORTB, FAHR_LED);
48                 bitClr(PORTB, KELVIN_LED);
49                 fahr();
50                 break;
51             case 3:
52                 bitClr(PORTB, CELSIUS_LED);
53                 bitClr(PORTB, FAHR_LED);
54                 bitSet(PORTB, KELVIN_LED);
55                 kelvin();
56                 break;
57         }
58     }
59 }
```

Código 2: Função main().



A função `main()` é composta pela inicialização dos recursos a serem utilizados como `kplnit()`, `adclnit()` e `lcdlnit()` e, também, pelo loop infinito. O loop infinito é composto pelas funções de leitura de teclas, e por um `switch case`, implementando um sistema de menu horizontal. Em cada `case` tem-se uma escala de temperatura e a configuração de ascendimento dos led's de indicação.

Para o cálculo das temperaturas, foi utilizado o canal 0 do ADC, referente ao potenciômetro P1 da placa PICGenios do PICSimLab, de modo a simular o sensor de temperatura. Conforme calculado, cada variação na leitura representa uma variação de 4,88 mV na entrada analógica e, como o sistema foi projetado para um sensor de temperatura LM35, no qual apresenta sensibilidade de 10mV/°C, a equação utilizada para o cálculo da temperatura em celsius é:

```
tmp = (adcRead(0)* 0.0048828125 * 100);
```

Código 3: Equação para cálculo de temperatura.

A partir daí, a conversão para as escalas fahrenheit e kelvin se dá pela manipulação da variável `tmp` dentro de cada função respectiva.

Abaixo, tem-se os prints da placa PICGenius com o código em execução para a mesma leitura do ADC.

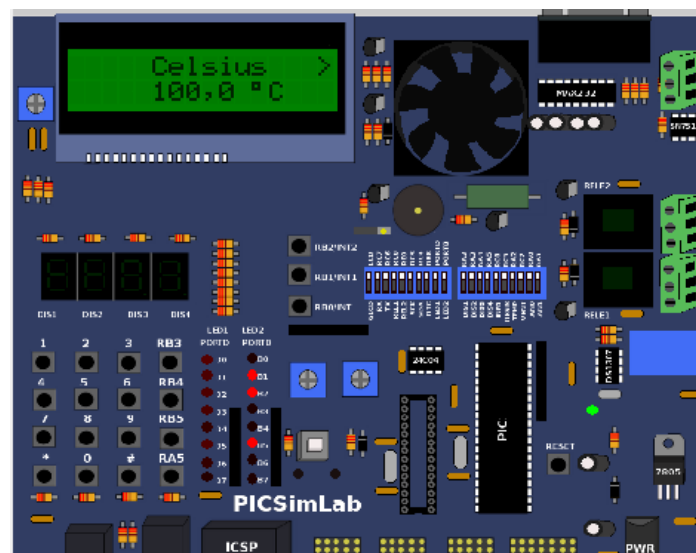


Figura 4: Tela da temperatura em Celsius.

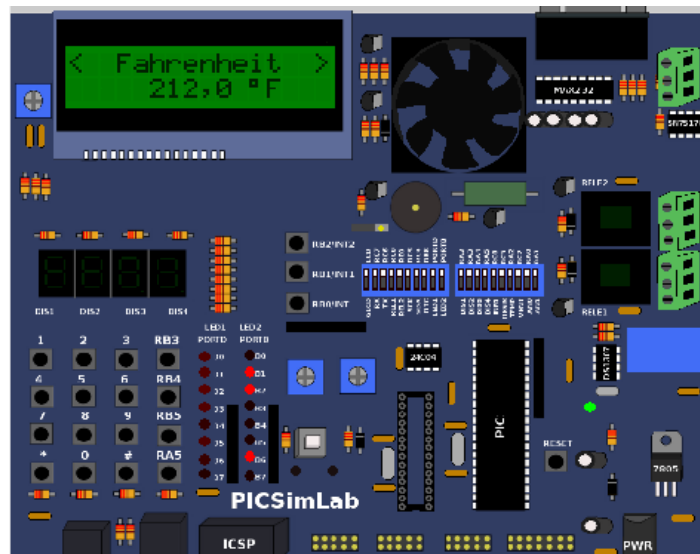


Figura 5: Tela da temperatura em Fahrenheit.

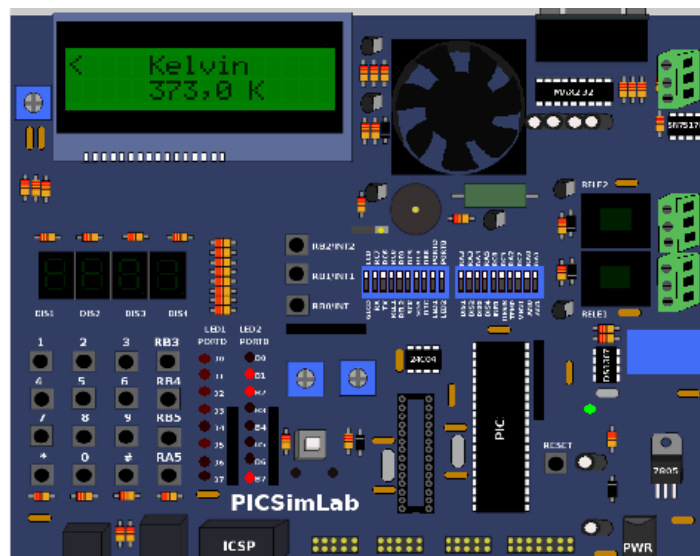


Figura 6: Tela da temperatura em Kelvin.

CONCLUSÃO

Podemos concluir que o projeto desenvolvido atende à proposta, uma vez que utilizando o PICSimLab para simulação, executa a função de termômetro digital projetada e engloba os conceitos aprendidos em aula. Sendo assim, o projeto pode ser considerado bem sucedido.