



UNIVERSIDADE ESTADUAL DE CAMPINAS

Faculdade de Tecnologia

Relatório Microcontroladores II

## **FRDMKL25Z - LCD e LED RGB**

Mateus Vall Martins 174389  
Rafael Danelon Correia 103841  
Raphaela Carvalho Cruz 157111

Prof. Dr. Talia Simões

Limeira/SP

Outubro

2019

## a. Introdução

A placa de desenvolvimento Freedom FRDM-KL25Z NXP é equipada com MCU KL25Z128 com 128KB de memória flash e 16KB de memória SRAM, que pode rodar à 48MHz.

A placa contém um acelerômetro, led RGB, sensor touch, dois conectores mini USB para sua programação e alimentação além de possibilitar instalação de conectores para acesso à GPIO.

Os códigos deste relatório são direcionados ao funcionamento do LCD e placa Freedom..

## b. Materiais

- LCD (liquid crystal display)
- *Protoboard e jumpers*
- Computador provido de Windows 7 ou 8
- Placa FRDMKL25Z e cabo USB
- Compilador Mbed armazenado na nuvem

## c. Objetivos

Evidenciar o funcionamento do componente LCD, de forma a imprimir em sua tela cada variância do LED RGB da placa Freedom KL25Z através do software para criação e simulação de programas escritos no ambiente Mbed.

## d. Desenvolvimento

Pino	Símbolo	Função
1	Vss	GND(alimentação)
2	Vdd	5V (alimentação)
3	V0	Ajuste de contraste
4	RS	Habilita/Desabilita seletor de registrador
5	R/W	Leitura/Escrita
6	E	Habilita escrita no LCD
7 a 14	DB0 a DB7	Dados
15	A	5V(backlight)
16	K	GND(backlight)

Tabela 1 - Funcionalidades do display LDC 16 x 2.

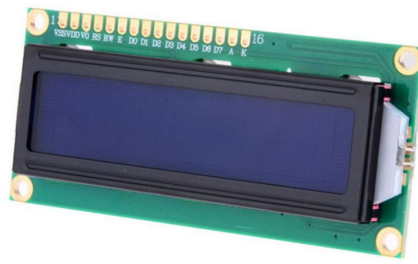


Figura 1 - Componente LCD.

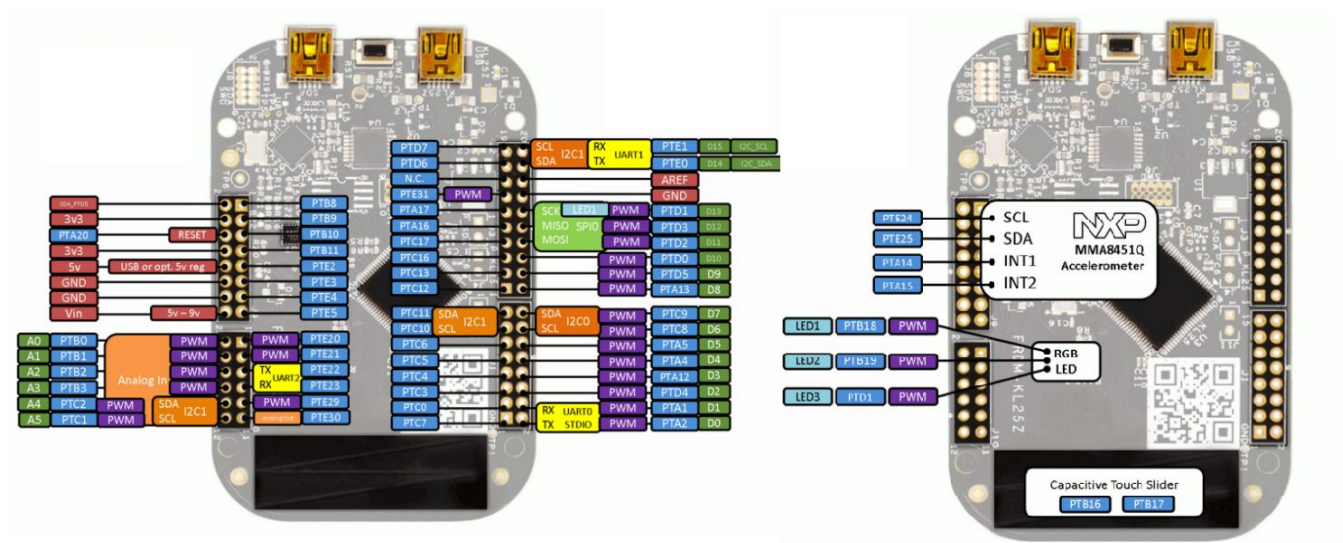


Figura 3 - Pinagem da placa FRDM-KL25Z.[4]

Códigos para simulação de mudança de cores dos leds de acordo com a angulação formada.

```
main.cpp
1 #include "mbed.h"
2 #include "TextLCD.h"
3
4 TextLCD lcd(PTE0, PTE1, PTE2, PTE3, PTE4, PTE5);
5
6 int main()
7 {
8     lcd.printf("Bom dia!\n");
9 }
10
```

Primeiramente, em todos os códigos, deve-se incluir a biblioteca oficial mbed, neste caso, além dela, declara-se a biblioteca do LCD. Em seguida, define-se a parte física do LCD. Já no *int main()*, define-se a frase que o LCD imprimirá em sua tela, neste caso “Bom dia!”.

```
main.cpp x
1 #include "mbed.h"
2 #include "TextLCD.h"
3
4 DigitalOut led1(LED1); //declarando led vermelho
5 DigitalOut led2(LED2); //declarando led verde
6 TextLCD lcd(PTE0, PTE1, PTE2, PTE3, PTE4, PTE5);
7
8 int main()
9 {
10     while(true)
11     {
12         led1 = 1; //apaga led vermelho
13         led2 = 1; //apaga led verde
14         led1 = 0; //acende led vermelho
15         lcd.cls();
16         lcd.printf("Vermelho \n");
17         wait(10); //espera 10s
18         led1 = 1; //apaga led vermelho
19         led2 = 0; //acende led verde
20         lcd.cls();
21         lcd.printf("\n Verde");
22         wait(7); //espera 7s
23         led1 = 0; //acende led vermelho
24         led2 = 0; //acende led verde
25         lcd.cls();
26         lcd.printf("Amarelo \n");
27         wait(3); //espera 3s
28     }
29 }
```

Primeiramente, em todos os códigos, deve-se incluir a biblioteca oficial mbed, neste caso, além dela, declara-se a biblioteca do LCD. Após, os dois pinos de saída tipo digital, referentes às cores vermelho e verde e por último define-se a parte física do LCD. Já no *int main()*, ocorre a configuração da lógica do programa a operar. Enquanto for verdade, laço *while(true)*, inicia-se ambos os leds apagados para depois acender somente o desejado, neste caso o vermelho. A função *lcd.cls()* apaga o conteúdo do LCD e imprime “Vermelho” na primeira linha da tela do LCD. Ocorre a espera de 10 segundos antes de apagar novamente o led vermelho e posteriormente acender o led verde. Novamente, o conteúdo do LCD é apagado e “Verde” é impresso na segunda linha, devido ao acréscimo do “\n” no início da frase, fazendo com que seja escrito na segunda linha. Ocorre a espera de 7 segundos antes de acender novamente o led vermelho adicionalmente ao verde, formando a cor amarela. Novamente, o conteúdo do LCD é apagado e “Amarelo” é impresso na primeira linha.

## e. Resultados

O trabalho introduziu o uso do LCD e posteriormente o combinou com o funcionamento de um semáforo, já simulado em aula no passado. Houve pequenas diferenças entre os códigos, sendo que o primeiro é apenas a impressão de uma frase e o segundo opera conforme variação de tempo e variância de cores do LED RGB.

## **f. Conclusão**

As simulações na placa trazem a possibilidade de fazer projetos diferentes utilizando ferramentas, físicas e em nuvem, de forma fácil e acessível a fim de testar funcionamento de lógicas e projetos, e neste caso, os códigos operam de forma a evidenciar a possibilidade de combinar e sincronizar diferentes componentes em um mesmo projeto de forma bastante acurada.

## **g.Referências**

- [1] Documentação Mbed OS5. Disponível em:<<https://os.mbed.com/docs/mbed-os/v5.9/introduction/index.html>>Acesso em 24 de setembro de 2019.
- [2] Apresentando a Freedom KL25Z. Disponível em:<<https://www.filipeflop.com/blog/apresentando-frdm-kl25z/>>Acesso em 27 de setembro de 2019.
- [3] Baú da eletrônica. Display LCD 16 x 2 Azul. Disponível em:<<https://www.baudaeletronica.com.br/display-lcd-16x2-azul.html>>Acesso em 24 de outubro de 2019.
- [4] FRDM-KL25Z. Disponível em: <<https://os.mbed.com/platforms/KL25Z/>>Acesso em 18 de outubro de 2019.
- [5] UNICAMP. Aulas teóricas e práticas de microcontroladores II.