



UNIVERSIDADE ESTADUAL DE CAMPINAS

Faculdade de Tecnologia

Relatório Microcontroladores II

**FRDMKL25Z**

Raphaela Carvalho Cruz

Prof. Dr. Talia Simões

Limeira/SP

Setembro

2019

## **a. Introdução**

A placa de desenvolvimento Freedom FRDM-KL25Z NXP tem baixo custo e 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. A FRDM-KL25Z tem o apoio de uma enorme gama de desenvolvedores pelo mundo, facilitando seu acesso aos usuários e leigos.

## **b. Materiais**

- Computador provido de Windows 7 ou 8
- Placa FRDMKL25Z
- Compilador Mbed armazenado na nuvem

## **c. Objetivos**

Introduzir a utilização do software e placa, criar e simular programas escritos no ambiente Mbed.

## **d. Desenvolvimento**

**d.1)** Passo a passo referente à instalação e utilização da placa para criação dos códigos:

1)Configurações iniciais

2)Em [www.mbed.com](http://www.mbed.com), acessar:

Hardware>Boards>FRDM KL25Z>Step by step firmware upgrade instructions

3)Baixar o firmware:

- 20140530\_k20...

4)Conexão da placa FRDMKL25Z: conexão USB oposta ao QRCODE

5)Instalando o firmware

- Segurar reset e conectar a placa, pop-up BOOTLOADER aparece

6)Salvar o firmware em BOOTLOADER e resetar a placa para surgir o pop-up MBED

7) Ir em compiler do Mbed e fazer login. Para adicionar a placa:

- Selecionar No device selected
- Clicar em Add platform
- Procurar por KL25Z
- Clicar em Add
- Abrir novamente o compilador

## 8)Dentro do compiler

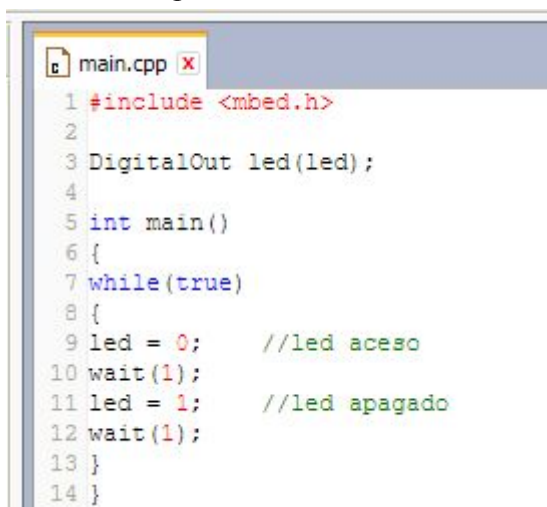
- Clicar em new, escolher FRDM KL25Z e “empty program”
- Nomear e clicar em ok
- Em program workspace, clicar com o botão direito sobre o programa,
- Selecionar Import Library -> From Import Wizard
- Em Import library from mbed.org, escolher a opção “mbed” e clicar no botão import

## 9)Criando um programa

- Clicar com o botão direito no projeto(nome) e escolher new file
- Nomear o arquivo: main.cpp
- Digitar o primeiro programa

## d.2) Programação:

### Pisca-led simples



```
1 #include <mbed.h>
2
3 DigitalOut led(led);
4
5 int main()
6 {
7     while(true)
8     {
9         led = 0;    //led aceso
10        wait(1);
11        led = 1;    //led apagado
12        wait(1);
13    }
14 }
```

Primeiramente, em todos os códigos, deve-se incluir a biblioteca, no caso a biblioteca oficial mbed. Em seguida, ocorre a inicialização designando o tipo de saída para todos os leds a serem utilizados, neste caso apenas um led com saída digital. O int main() é provido das condições para o código operar. Neste caso, o led começa aceso, assumindo 0 pois a lógica é invertida, e apaga posteriormente ao intervalo de 1 segundo, apagando e após obedecer o intervalo de 1 segundo, acende novamente, cumprindo um looping infinito.

```
main.cpp x
1 #include "mbed.h"
2
3 DigitalOut led1(led1); //vermelho
4 DigitalOut led2(led2); //verde
5 DigitalOut led3(led3); //azul
6
7 int main()
8 {
9 while(true)
10 {
11 led1 = 0; //led aceso
12 wait(5);
13 led1 = 1; //led apagado
14 led3 = 0; //led aceso
15 wait(5);
16 led3 = 1; //led apagado
17 led2 = 0; //led aceso
18 wait(5);
19 led2 = 1; //led apagado
20 }
21 }
```

Primeiramente, em todos os códigos, deve-se incluir a biblioteca, no caso a biblioteca oficial mbed. Em seguida, ocorre a inicialização designando o tipo de saída para todos os leds a serem utilizados, no caso 3 leds com saída digital. O int main() é provido das condições para o código operar. Neste caso, o led vermelho começa aceso, assumindo 0 pois a lógica é invertida, e apaga posteriormente ao intervalo de 5 segundos, o led azul acende e após o intervalo de 5 segundos, apaga e finalmente o led verde acende, obedece o tempo de 5 segundos e apaga. O looping recomeça infinitamente.

```
main.cpp x main.cpp x
1 #include <mbed.h>
2
3 DigitalOut led1(LED1);
4 DigitalOut led2(LED2);
5
6 int main()
7 {
8 while(true)
9 {
10 led1 = 1;
11 led2 = 1;
12 led1 = 0; //led vermelho aceso
13 wait(15);
14 led1 = 1; //led vermelho apagado p não misturar cor
15 led2 = 0; //led verde aceso e não apaga pq tem q misturar a cor
16 wait(10);
17 led1 = 0; //led vermelho aceso
18 wait(3);
19 }
20 }
```

Primeiramente, em todos os códigos, deve-se incluir a biblioteca, no caso a biblioteca oficial mbed. Em seguida, ocorre a inicialização designando o tipo de saída para todos os leds a serem utilizados, no caso 2 leds com saída digital. O int main() é provido das condições para o código operar.

Neste caso, o led vermelho e verde começam apagado, assumindo 1 pois a lógica é invertida. O led vermelho acende e apaga posteriormente ao intervalo de 15 segundos, referente ao tempo de espera estipulado em um farol vermelho. O led verde acende imediatamente e permanece aceso por 10 segundos. O led vermelho acende, combinando desta maneira sua cor com o led verde, formando o amarelo, após 3 segundos, tempo estimado de espera no farol amarelo, os leds vermelho e verde apagam novamente. O looping recomeça e opera infinitamente.

## **e. Resultados**

As simulações na placa evidenciam a possibilidade de fazer projetos diferentes utilizando as mesmas ferramentas, físicas e em nuvem, de forma fácil e acessível a fim de testar funcionamento de lógicas e projetos para o dia a dia, tal como o último que simula o funcionamento de um semáforo por meio da programação do acendimento de leds e suas combinações operando em intervalos.

## **f. Conclusão**

O trabalho expôs pequenas diferenças entre os códigos, desde variação de quantidade de leds e intervalos de tempo de forma a mostrar aplicações diferentes. O primeiro código é o funcionamento de um pisca-led simples para introdução do uso da placa, o segundo é um pisca-led com duas cores que poderia representar um pisca-pisca de natal por exemplo, largamente utilizado pelas pessoas. Enquanto que o último código, evolui um pouco e além de intervalo de tempo distintos, há a mistura de duas cores no rgb, vermelho e verde, a fim de formar o amarelo do sinal de um semáforo.

## **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] UNICAMP. Aulas teóricas e práticas de microcontroladores II