



UNIVERSIDADE ESTADUAL DE CAMPINAS

Faculdade de Tecnologia

Relatório Microcontroladores II

FRDM-KL25Z - Micro servo motor

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

Prof. Dr. Talia Simões

Limeira/SP

Novembro

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.

b. Materiais

- Micro servo motor
- *Jumpers*
- Computador provido de Ubuntu
- Placa FRDM-KL25Z e cabo USB
- Compilador Mbed armazenado na nuvem

c. Objetivos

Evidenciar o funcionamento de um micro servo motor 5G-90 e a placa Freedom KL25Z através do software para criação e simulação de programas escritos no ambiente Mbed. Os códigos deste relatório são direcionados ao funcionamento do micro servo motor.

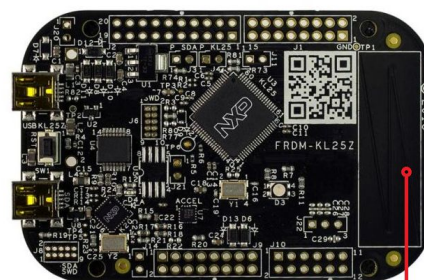
d. Desenvolvimento

Especificações do micro servo motor 5G -90, passadas em aula:

- Tensão de operação: 3 a 7,2V
- Ângulo de rotação: 180°
- Duty cycle: 1 a 2ms
- Período de PWM: 20ms (50Hz)
- Sinal: 4,8V
- Velocidade: 0,12s/60° (4,8V) sem carga
- Torque 1,2kg.cm(4,8V) e 1,6kg.cm(6,0V)



Figura 1 - Micro servo motor com haste.



Sensor capacitivo

Figura 2 -Indicação do TSI na FRDM-KL25Z.

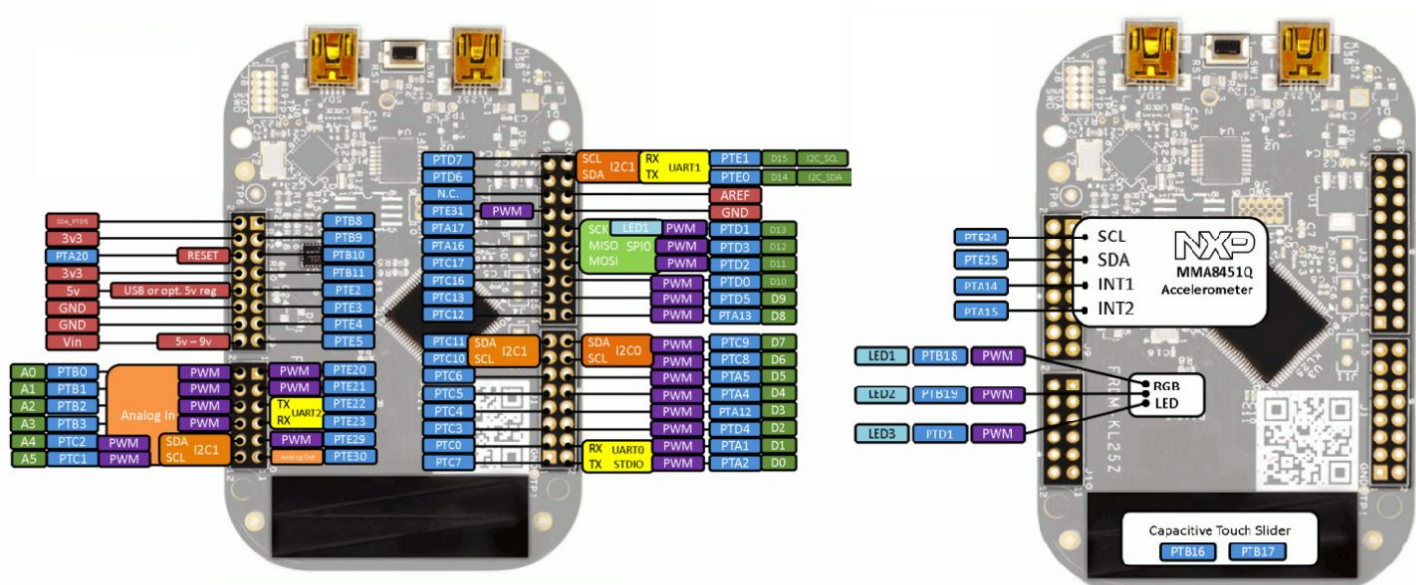


Figura 3 - Pinagem da placa FRDM-KL25Z.[3]

Ligações físicas, por meio de *jumpers*, feitas entre a placa e o servo motor:

<i>Jumpers</i> (cor)	Porta	Correspondência
Laranja	PTD0	Pwm
Vermelho	5V	Vcc
Marrom	GND	Terra

Tabela 1 - Conexões propostas em sala de aula.

Códigos para simulação de controle do micro servo motor.

Primeiramente, em todos os códigos, deve-se incluir a biblioteca oficial mbed, além de outras utilizadas, neste caso a do sensor TSI. Em seguida, declara-se o sensor capacitivo e define-se a entrada do tipo saída pwm conectada à porta PTD0. Posteriormente, no int main(), ocorre a configuração da lógica do programa a operar. A variável “valor_tsi” é definida como *float* e o período de pwm em como sendo 20 milissegundos, correspondente à largura do pulso/período multiplicado por 100, significando a porcentagem do fator de ciclo do pwm, iniciado em 0°.

```

main.cpp
1 #include <mbed.h>
2 #include <TSISensor.h>
3
4 TSISensor tsi;
5 PwmOut pwm (PTD0);
6
7 int main()
8 {
9     float valor_tsi;
10    pwm.period_ms(20);
11    while (1)
12    {
13        valor_tsi = tsi.readPercentage();
14        pwm.pulsewidth_us(50+350*(valor_tsi));
15        wait(0.01);
16    }

```

Enquanto for verdade, laço *while(1)*, o valor da leitura de porcentagem do TSI é armazenado em “valor_tsi”. Após o recebimento da leitura, a variável do valor do sensor capacitivo é multiplicada por 350 e somado a 50 resultando no valor de pulso do pwm, sinal emitido para variação angular da haste do micro servo motor. Finalmente, ocorre o tempo de espera de 0.01 segundos e as estruturas *while* e *int main()* são finalizadas nesta ordem. Sucintamente, o código opera da seguinte forma: o operador controla a rotação da haste do servo motor por meio do toque no sensor TSI.

```
main.cpp x
1 #include <mbed.h>
2
3 PwmOut pwm(PTD0);
4
5 int main()
6 {
7     pwm.period_ms(20);
8     while(1)
9     {
10         pwm.pulsewidth_us(500);
11         wait(0.5);
12         pwm.pulsewidth_us(2250);
13         wait(0.5);
14         pwm.pulsewidth_us(4000);
15         wait(0.5);
16     }
17 }
```

Primeiramente, em todos os códigos, deve-se incluir a biblioteca oficial mbed, além de outras utilizadas, neste caso a única utilizada. Em seguida, define-se a entrada do tipo saída pwm conectada à porta PTD0. Posteriormente, no int main(), ocorre a configuração da lógica do programa a operar. O período de pwm como sendo 20 milissegundos, correspondente à largura do pulso/período multiplicado por 100, significando a porcentagem do fator de ciclo do pwm, iniciado em 0° e enquanto for verdade, laço *while(1)*, a haste do servo motor varia sua angulação entre 0° passando por 90° até chegar a 180° em um período de 0,5 segundos. Finalmente, as estruturas *while* e *int main()* são finalizadas nesta ordem. Sucintamente, o código opera

da seguinte forma: a haste do servo motor varia entre 0° e 180° automaticamente.

e. Resultados

O teste do primeiro código, se trata do funcionamento do micro servo motor de acordo com o estímulo do toque humano no sensor capacitivo contido na placa Freedom KL25Z. O código do desafio, consiste na automação do funcionamento do servo motor, em que a haste rotaciona independente do sensor capacitivo. A rotação da haste ocorre de forma tão rápida, ou seja, em tão curto intervalo de tempo que, a visão humana é imperceptível a comutação entre os ângulos 0° a 90° e 90° a 180°. Finalmente, pode-se inferir o que o primeiro código contempla um projeto de cunho mais manual e sensível enquanto que o segundo se aproxima de um funcionamento mais independente, e bastante similar ao de um pára-brisas de carro, como comentado em aula.

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 variação de controle do micro servo motor.

g.Referências

- [1] Documentação Mbed OS5. Disponível em:<<https://os.mbed.com/docs/mbed-os/v5.9/introduction/index.html>> Acesso em 15 de outubro de 2019.
- [2] Apresentando a Freedom KL25Z. Disponível em:<<https://www.filipeflop.com/blog/apresentando-frdm-kl25z/>> Acesso em 15 de outubro de 2019.
- [3] FRDM-KL25Z. Disponível em: <<https://os.mbed.com/platforms/KL25Z/>>Acesso em 18 de outubro de 2019.
- [4] UNICAMP. Aulas teóricas e práticas de microcontroladores II.