



UNIVERSIDADE ESTADUAL DE CAMPINAS

Faculdade de Tecnologia

Relatório Microcontroladores II

FRDMKL25Z - Acelerômetro

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

Prof. Dr. Talia Simões

Limeira/SP

Setembro

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 acelerômetro.

Os valores dos eixos X, Y e Z são fornecidos pelo acelerômetro da FRDM-KL25Z via protocolo I2C e, conforme haja variação de aceleração, o LED RGB da placa alterna suas cores, ou seja, a percepção de inclinação e inércia se utiliza da aceleração gravitacional para referência.

b. Materiais

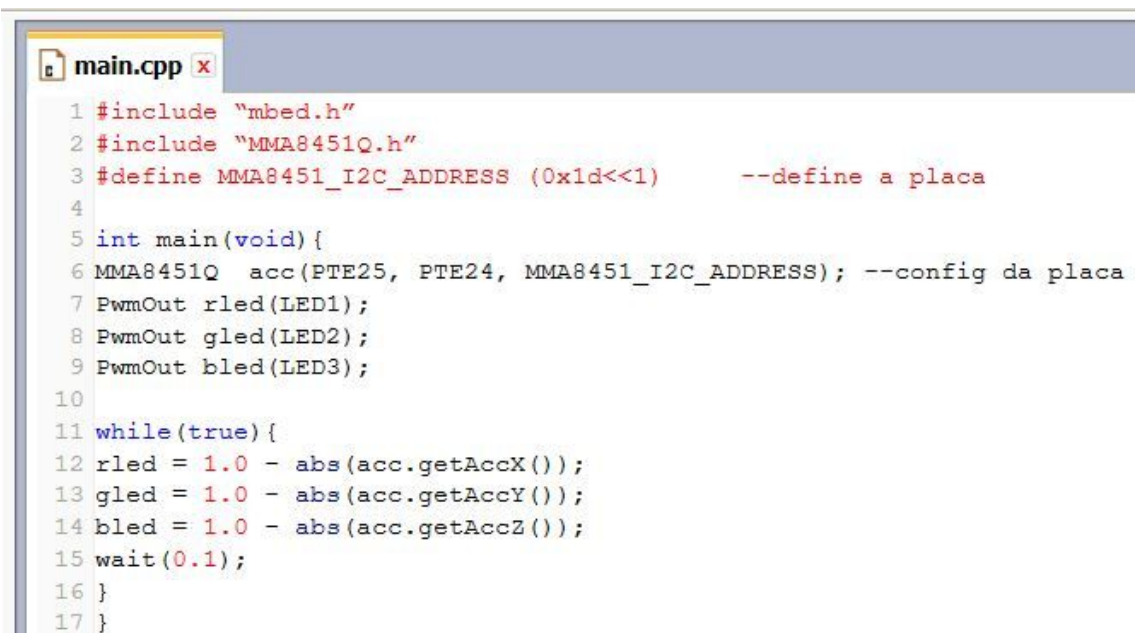
- Computador provido de Windows 7 ou 8
- Placa FRDMKL25Z e cabo USB
- Compilador Mbed armazenado na nuvem

c. Objetivos

Evidenciar o funcionamento do componente acelerômetro da placa Freedom através do software para criação e simulação de programas escritos no ambiente Mbed.

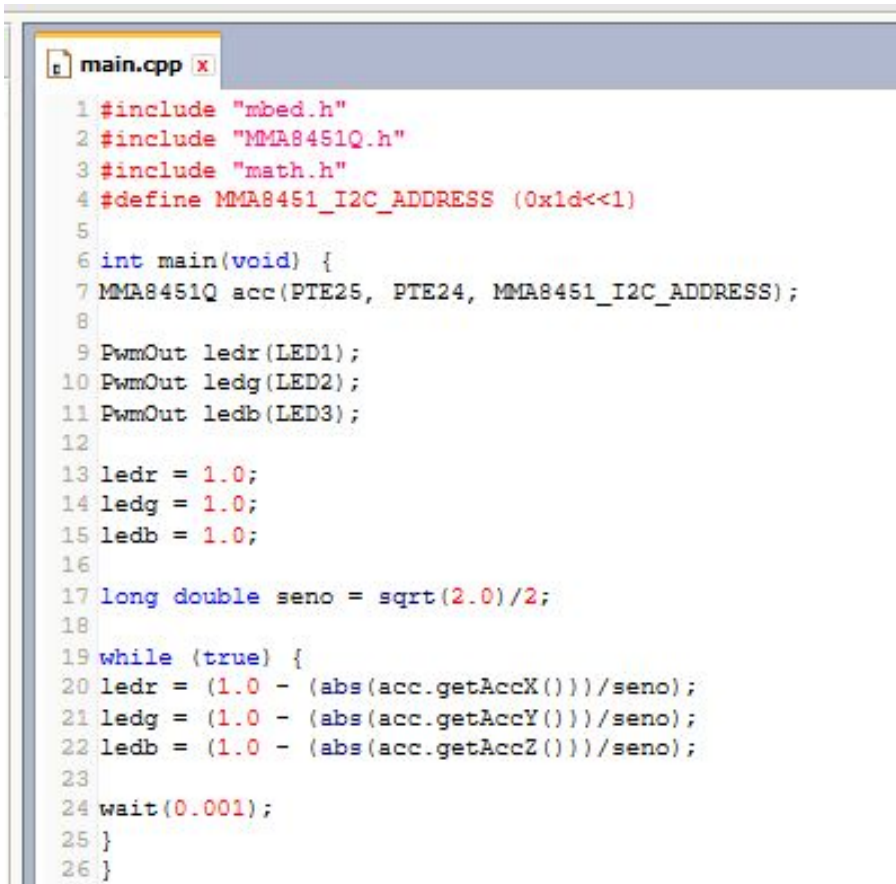
d. Desenvolvimento

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



```
main.cpp x
1 #include "mbed.h"
2 #include "MMA8451Q.h"
3 #define MMA8451_I2C_ADDRESS (0x1d<<1)    --define a placa
4
5 int main(void){
6 MMA8451Q acc(PTE25, PTE24, MMA8451_I2C_ADDRESS); --config da placa
7 PwmOut rled(LED1);
8 PwmOut gled(LED2);
9 PwmOut bled(LED3);
10
11 while(true){
12 rled = 1.0 - abs(acc.getAccX());
13 gled = 1.0 - abs(acc.getAccY());
14 bled = 1.0 - abs(acc.getAccZ());
15 wait(0.1);
16 }
17 }
```

Primeiramente, em todos os códigos, deve-se incluir a biblioteca, no caso a biblioteca oficial mbed, a biblioteca da placa e acelerômetro. Em seguida, no int main(), ocorre a configuração da placa, a inicialização designando o tipo de saída para todos os leds a serem utilizados, neste caso os leds rgb, vermelho, verde e azul, com saída PWM digital. O laço while é composto por condições para o código operar. Neste caso, o led vermelho recebe o valor da diferença entre o inteiro 1 e o valor da aceleração na abscissa X enquanto o led verde recebe o valor da diferença entre o inteiro 1 e o valor da aceleração na abscissa Y e o led azul recebe o valor da diferença entre o inteiro 1 e o valor da aceleração na abscissa Z. Logo, conforme varia-se a posição da placa, o acelerômetro detecta variação na posição tomando a aceleração da gravidade como base e, obedecendo a programação, altera as cores dos leds de acordo com tais variações. Finalmente, ocorre a espera de 0.1 segundos.



```
1 #include "mbed.h"
2 #include "MMA8451Q.h"
3 #include "math.h"
4 #define MMA8451_I2C_ADDRESS (0x1d<<1)
5
6 int main(void) {
7     MMA8451Q acc(PTE25, PTE24, MMA8451_I2C_ADDRESS);
8
9     PwmOut ledr(LED1);
10    PwmOut ledg(LED2);
11    PwmOut ledb(LED3);
12
13    ledr = 1.0;
14    ledg = 1.0;
15    ledb = 1.0;
16
17    long double seno = sqrt(2.0)/2;
18
19    while (true) {
20        ledr = (1.0 - (abs(acc.getAccX())/seno));
21        ledg = (1.0 - (abs(acc.getAccY())/seno));
22        ledb = (1.0 - (abs(acc.getAccZ())/seno));
23
24        wait(0.001);
25    }
26 }
```

Primeiramente, em todos os códigos, deve-se incluir a biblioteca, no caso a biblioteca oficial mbed, a biblioteca da placa e acelerômetro. Em seguida, no int main(), ocorre a configuração da placa, pinagem utilizada, a inicialização designando o tipo de saída para todos os leds a serem utilizados, neste caso os leds rgb, vermelho, verde e azul, com saída PWM digital. Além de definir o estado para cada variável dos pinos, no caso, o valor definido é 1 e por isso, os pinos inicializam desligados. Após, ocorre a definição de uma variável para a constante seno, correspondente ao ângulo de 45 graus.

O laço while é composto por condições para o código operar. Neste caso, o led vermelho recebe o valor da diferença entre o inteiro 1 e o valor da aceleração na abscissa X dividido pelo seno enquanto o led verde recebe o valor da diferença entre o inteiro 1 e o valor da aceleração na abscissa Y dividido pelo seno e o led azul recebe o valor da diferença entre o inteiro 1 e o valor da aceleração na abscissa Z dividido pelo seno. Logo, conforme varia-se a posição da placa, o acelerômetro detecta variação na posição tomando a aceleração da gravidade como base e, obedecendo a programação, altera as cores dos leds de acordo com a angulação de 45 graus entre os eixos. Finalmente, ocorre a espera de 0.001 segundos.

e. Resultados

O trabalho expôs pequenas diferenças entre os códigos, sendo que o primeiro faz a transição de cores nas transições de coordenadas enquanto o segundo mudava as cores de acordo com o ângulo de 45 graus em cada eixo. Durante a elaboração do código foi preciso pensar na relação de proporcionalidade de um triângulo retângulo assim como na projeção do ângulo.

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 sensibilidade do acelerômetro na detecção das coordenadas tridimensionais baseando-se na aceleração da gravidade.

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] Utilizando acelerômetro. Disponível em: <http://www.seucurso.com.br/index.php?option=com_content&view=article&id=281:3-programa-utilizando-acelerometro&catid=11&Itemid=128> Acesso em 27 de setembro de 2019.

[4] Utilizando o acelerômetro e LED RGB da FRDM KL25Z no mbed. Disponível em: <<https://www.embarcados.com.br/frdm-kl25z-no-mbed/>> Acesso em 27 de setembro de 2019.

[5] UNICAMP. Aulas teóricas e práticas de microcontroladores II.