



Universidade Estadual de Campinas  
Faculdade de Engenharia Elétrica  
e de Computação



---

# EA871 - Laboratório de Programação Básica de Sistemas Digitais

Roteiro 4 - Primeiro Programa em C para Microcontroladores

---

Aluno: Fernando Teodoro de Cillo

RA: 197029

Campinas  
Abril de 2022

## Introdução:

O intuito deste experimento é entender o princípio básico de operação do microcontrolador KL2x, desenvolver um pseudo-código e, a partir disso, um código em linguagem C. Conhecimentos necessários para esse experimento são o mascaramento de *bits* e o uso do qualificador *volatile* para evitar otimizações indevidas do compilador.

## Experimento:

### 4

subrotina delay

Entrada: Inteiro

Saída: Nenhuma

Enquanto Inteiro for diferente de 0 decrementa 1

Fim Enquanto

FIM

programa main

INÍCIO

Entrada: Nenhuma

Saída: Sinais nos pinos PORTB 18, PORTB 19 e PORTD 1

Inicialização:

Habilitar os relógios (PORTB e PORTD)

Habilitar o pino 18 da PORTB (LED vermelho), 18 da PORTB (LED azul) e 1 da PORTD (LED verde)

Habilitar GPIO

Função **vermelho** (ativa alta):

LED vermelho aceso

LED verde apagado

LED azul apagado

Função **verde** (ativa alta):

LED vermelho apagado

LED verde aceso

LED azul apagado

Função **azul** (ativa alta):

LED vermelho apagado

LED verde apagado

LED azul aceso

LOOP:

Função **vermelho**

Função verde

Função azul

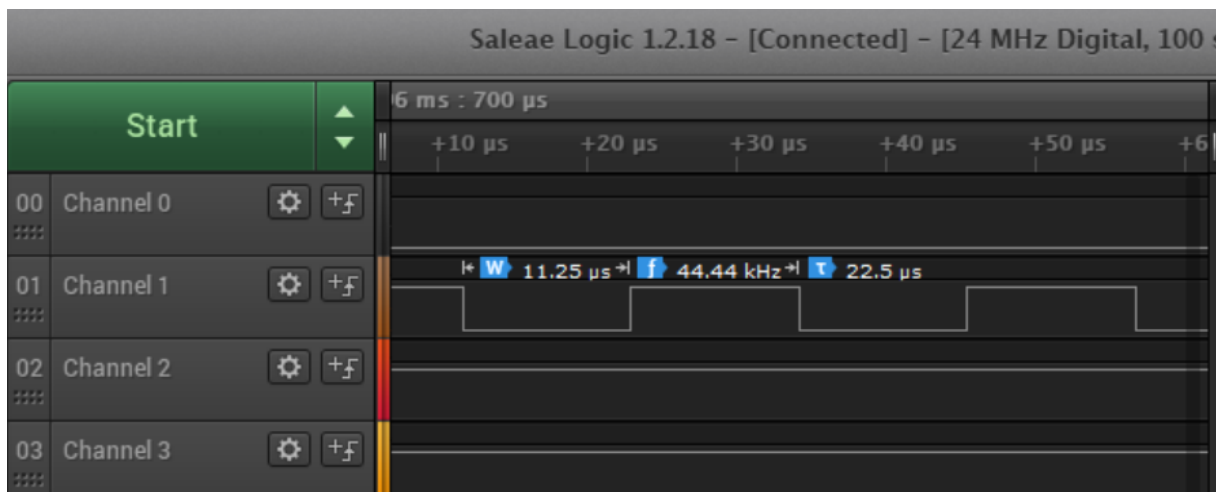
Função vermelho + Função verde (LED amarelo)

Função vermelho + Função verde + Função azul (LED branco)

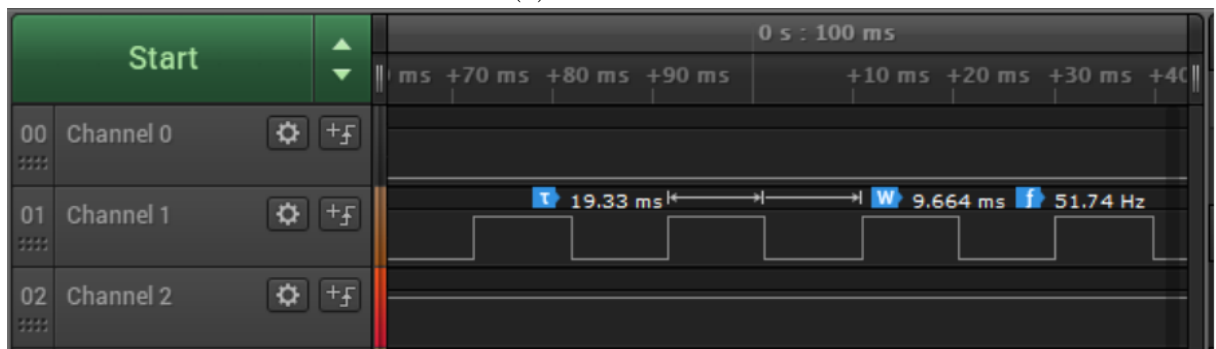
FIM

## 6

Um teste do projeto pode ser visto em [teste do LED RGB](#), provando que a sequência dos LEDs está correta, e a largura de pulso da função *delay\_10us* é mostrada na figura 1. Foram utilizadas 25 repetições e COUNT=1 e COUNT=100, respectivamente, para obter intervalos de tempo próximos de  $10\mu s$  e  $10ms$ . A figura 2 mostra, usando o analisador lógico, que com 52000 iterações conseguimos um intervalo de  $0.5026s$  para o LED mudar de cor, utilizando o código da figura 4.



(a) COUNT=1



(b) COUNT=1000

Figura 1: Largura de pulso da função *delay\_10us* variando o valor de COUNT

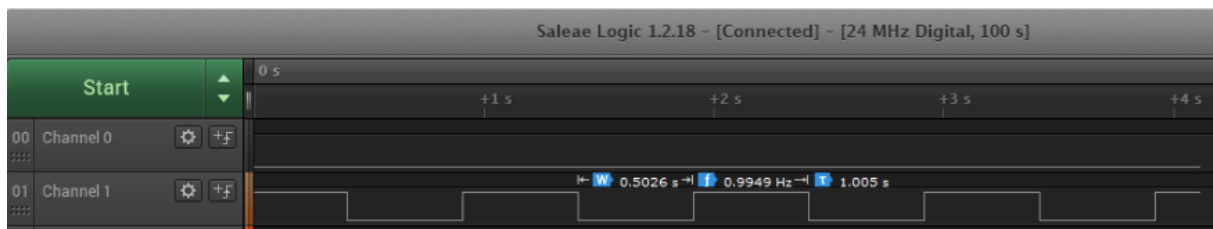


Figura 2: Analisador lógico com o intervalo de 0.5026s

## 7

Comparando a figura 3 com o tamanho do arquivo *.elf* analisado no roteiro 2, nota-se que apenas o arquivo *.txt* teve seu tamanho aumentado. Isso ocorre porque há muito mais linhas de código em linguagem C e Assembly (porque o IDE CodeWarrior permite fazer essa inserção) e, conseqüentemente, o código em linguagem de máquina é muito maior. Os arquivos *.data* e *.bss* armazenam as variáveis inicializadas e não-inicializadas, respectivamente.

```

Problems Console Memory
CDT Build Console [ledRGBYW_FRDMKL25]
'Executing target #7 ledRGBYW_FRDMKL25.siz'
'Invoking: ARM Ltd Windows GNU Print Size'
"C:/Freescall/CW_MCU_v10.6/Cross_Tools/arm-none-eabi-gcc-4_7_3,
  text    data    bss      dec      hex filename
 1364     24    2076    3464    d88 ledRGBYW_FRDMKL25.elf
'Finished building: ledRGBYW_FRDMKL25.siz'

```

Figura 3: Tamanho do arquivo dado pela função *Print size*

```

// Laço que alterna as cores do LED
for(;;) {
    // Acende o LED na cor vermelho
    azul(0);
    verde(0);
    vermelho(1);
    delay_10us(52000);

    // Acende o LED na cor verde
    vermelho(0);
    azul(0);
    verde(1);
    delay_10us(52000);

    // Acende o LED na cor azul
    vermelho(0);
    verde(0);
    azul(1);
    delay_10us(52000);

    // Acende o LED na cor amarela
    azul(0);
    verde(1);
    vermelho(1);
    delay_10us(52000);

    // Acende o LED na cor branca
    vermelho(1);
    azul(1);
    verde(1);
    delay_10us(52000);
}

```

Figura 4: Código em C do laço que alterna as cores do LED