

## 5º Laboratório de Programação para Sistemas Embarcados (ECOP14)

Prof. Otávio Gomes e Prof. Rodrigo Almeida

### Leia com atenção - Informações iniciais:

1. No início de cada tópico/assunto é apresentado um **exercício de revisão** em que basta copiar o código na ferramenta, realizar a compilação e a execução e, então, interpretar o resultado. Este tipo de exercício tem como objetivo auxiliar o aluno a relembrar alguns conceitos e a validar as ferramentas que estão sendo utilizadas. Este código sempre estará correto e funcionando.
2. Os exercícios estão apresentados em **ordem crescente de dificuldade**.
3. Para o **registro de frequência**, o aluno deverá enviar o código relativo ao exercício mais difícil desta lista que conseguir resolver. Por exemplo:
  - a. Se em uma lista contendo 6 exercícios o aluno A conseguiu resolver até o exercício 4, é este que ele deve enviar para registro de frequência.
  - b. Se o aluno B conseguiu resolver toda a lista de exercícios, deve enviar o último exercício da lista.
4. Os exercícios abordam todos os conceitos relacionados ao conteúdo da aula em questão. Deste modo, caso o aluno não consiga resolver alguns dos exercícios, recomenda-se que o mesmo participe dos **plantões de dúvidas** e que busque aprender os conceitos envolvidos na atividade.
5. A **próxima atividade** de laboratório admitirá que os conceitos aqui apresentados já foram plenamente compreendidos.
6. Maiores informações a respeito do controle de frequência e notas estão disponíveis no SIGAA.

### 1) Criar um novo projeto. Criar o arquivo *main.c* e colocar o seguinte código:

```
#include <pic18f4520.h>
#include "config.h"

void main(void){
    unsigned char i, j;
    TRISD = 0x00;
    PORTD = 0x00;
    for(i = 0; i < 255; i++) {
        for(j = 0; j < 255; j++){
            for(j = 0; j < 255; j++){
            }
            PORTD = 0xFF;
            for(;;);
        }
    }
}
```

Link para vídeo tutorial:

<https://youtu.be/ukZE48C45A4>



- a) Compile o programa e grave no microcontrolador (*PICSimLab*). O que acontece com os led's?
- b) Mude o tipo das variáveis "i" e "j" para "int". Repita o procedimento. Mude novamente o tipo das variáveis para "float". O que acontece?

2) Crie dois arquivos: *delay.c* e *delay.h*. Dentro do arquivo *delay.c* crie a função "tempo", que recebe um **unsigned char** e não retorna nada. Esta função deverá gerar um *delay* de X milissegundos, onde X é o parâmetro recebido pela função. Lembre-se de fazer o arquivo de header corretamente, utilizando as diretivas **#ifndef** e **#endif**. Fique atento ao tipo e à capacidade de armazenamento do parâmetro da função. Utilize a função no arquivo *main.c* para gerar um atraso de 1 segundo.

```
//este código gera um delay de aprox 1ms
volatile unsigned char j, k;

for(j = 0; j < 41; j++) {
    for(k = 0; k < 3; k++);
}
```

3) Crie um semáforo utilizando a função anterior. O sinal verde deve ficar ligado 4 segundos, o amarelo meio segundo e o vermelho 5 segundos. Modifique a biblioteca *delay* se for necessário.

```
PORTD = 0b00000001; //acende vermelho
PORTD = 0b00000010; //acende verde
PORTD = 0b00000011; //acende amarelo
```