

11º Laboratório de Programação para Sistemas Embarcados

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. A **entrega** desta atividade para o controle de frequência será realizada pelo SIGAA.

1) Crie um programa *main.c* com o código abaixo. O programa escreve a data/hora a partir do RTC, imprimindo-a no LCD.

```
#include <pic18f4520.h>
#include "config.h"
#include "delay.h"
#include "lcd.h"
#include "ds1307.h"
#include "i2c.h"

#define L_ON    0x0F
#define L_OFF   0x08
#define L_CLR   0x01
#define L_L1    0x80
#define L_L2    0xC0

void main(void) {
    int seg=0;
    TRISA = 0xC3;
    TRISB = 0x03;
    TRISC = 0x01;
    TRISD = 0x00;
    TRISE = 0x00;
    ADCON1 = 0x06;
```

```
lcdInit();
lcdCommand (L_CLR);
lcdCommand (L_L1);
lcdString ("Teste RTC");
atraso(500);

lcdCommand(L_L1);
lcdString("Data: ");
lcdChar(((getDays() / 10) % 10) + 48);
lcdChar((getDays() % 10) + 48);
lcdChar('/');
lcdChar(((getMonths() / 10) % 10) + 48);
lcdChar((getMonths() % 10) + 48);
lcdChar('/');
lcdChar(((getYears() / 10) % 10) + 48);
lcdChar((getYears() % 10) + 48);

while (seg < 1000) {
    lcdCommand(L_L2);
    lcdString("Hora: ");
    lcdChar(((getHours() / 10) % 10) + 48);
    lcdChar((getHours() % 10) + 48);
    lcdChar('-');
    lcdChar(((getMinutes() / 10) % 10) + 48);
    lcdChar((getMinutes() % 10) + 48);
    lcdChar('-');
    lcdChar(((getSeconds() / 10) % 10) + 48);
    lcdChar((getSeconds() % 10) + 48);

    seg++;
}

lcdCommand(L_CLR);
lcdCommand(L_L1);
lcdString ("      Fim");
lcdCommand(L_L2);
lcdString(" Pressione RST");

while(1);
}
```

2) No programa de exemplo a seguir o PICSimLab recebe informações através da serial.

```
#include <pic18f4520.h>
#include "config.h"
#include "atraso.h"
#include "lcd.h"
#include "serial.h"

void main(void) {
    unsigned char i;
    unsigned char tmp;

    TRISA = 0xC3;
    TRISB = 0x03;
    TRISC = 0x01;
    TRISD = 0x00;
    TRISE = 0x00;
    ADCON1 = 0x06;

    lcdInit();
    serial_init();

    TRISCbits.TRISC7 = 1; //RX
    TRISCbits.TRISC6 = 0; //TX
    PORTB = 0;

    serial_tx_str("\r\n Picsimlab\r\n Teste Serial
TX\r\n");

    lcdCommand(L_CLR);
    lcdCommand(L_L1);
    lcdString("Teste Serial RX");

    serial_tx_str(" Digite:\r\n");

    for (i = 0; i < 32; i++) {
        if (!(i % 16)) {
            lcdCommand(L_L2);
            serial_tx_str("\r\n");
        }
        tmp = serial_rx(2000);
        lcdChar(tmp);
        serial_tx(tmp);
    }
    atraso_ms(100);
    serial_tx_str("\r\n FIM!\r\n");
    while (1);
}
```

Link para vídeo tutorial:

<https://youtu.be/UWYlvr-J99M>



Informações para
comunicação serial
utilizando o PICSimLab:
https://lcamboa.github.io/picsimlab_docs/stable/SerialCommunicationlabelseriali.html

Link para os arquivos do
Projeto:

<https://bit.ly/3cejQDp>



*** Atenção:** Para este exercício foi utilizada a biblioteca serial PICSimLab/PICGenius/PIC18F4520.

3) Baseando-se nos programas anteriores, crie uma nova rotina *main* que altere o valor da data e hora armazenada no RTC dependendo do comando recebido pela serial. Não se esqueça de imprimir a data e o horário do RTC no LCD para confirmar a mudança!

Os comandos que irão ser recebidos têm a forma:

Alterar a data do RTC	- Dddmmyy
Alterar a hora do RTC	- Hhhmmss

Observação 1: Não é necessário prever comandos incorretos do usuário (Ex, verificar se os dígitos recebidos via serial são de fato números).

Observação 2: No teste, envie os caracteres individualmente ou de dois em dois para que o microcontrolador consiga tratar a serial.