Universidade Federal de Itajubá

Instituto de Engenharia de Sistemas e Tecnologias da Informação - IESTI

2a Prova de ECOP04 – Programação para Sistemas Embarcados 1º sem. 2023

Prof. Décio Rennó de M. Faria

(20 pts) Questão 1: Uma máquina recebe comandos pela porta de comunicação serial, o comando M1 ajusta a velocidade do motor em 10%, M2 em 50% e M3 no máximo (100%). A velocidade é controlada através da saída PWM1 do microcontrolador. Para cada comando recebido o microcontrolador deve enviar uma mensagem "OK" para o computador. Implemente um **programa** para essa máquina.

```
#include "pic18f4520.h"
#include "config.h"
#include "serial.h"
#include "pwm.h"

void main(void){
unsigned char val;
serialInit();
pwmInit();
pwmFrequency(10000);
pwmSet1(0);
for(;;)    {
           val=serialRead();
           if(val=='M'){
                      val=serialRead();
                      while(val==0) val=serialRead();
                      if(val=='1') pwmSet1(10);
                      if(val=='2') pwmSet1(50);
                      if(val=='3') pwmSet1(100);
                      serialSend('O');
                      serialSend('K');
                      }
           }
}
```

(25 pts) Questão 2: Um reservatório possui um sensor ligado a um microcontrolador que indica o nível utilizando 4 fios (bits 0 a 3 da porta B). Quando todos estão em zero o reservatório está vazio, quando o primeiro é ativado o reservatório está em 25%, o segundo em 50%, o terceiro 75% e o quarto 100%. Monte uma **biblioteca** (arquivo nivel.h e nivel.c) que ajusta o display de sete segmentos com o valor do nível do reservatório. Utilize a biblioteca ssd.h para a implementação. O comando ssdUpdate() será executado no programa principal e não pela biblioteca. As funções da biblioteca devem ser:

- void mostradorInit(void); //Inicializa o mostrador
- void mostradorAjustaValor(void); //Verifica o status de cada fio e ajusta display.
- unsigned char mostradorNivel(void); //Fornece o valor do nível atual.

nivel.h

```
#ifndef NIVEL
        #define NIVEL

void mostradorInit(void);
void mostradorAjustaValor(void);
unsigned char mostradorNivel(void);

#endif
```

nivel.c

```
#include “ssd.h”
#include “pic18f4520.h”

void mostradorInit(void){
TRISB = 1;
ssdInit();
ssdDigit(0,0);
ssdDigit(0,1);
ssdDigit(0,2);
ssdDigit(0,3);
}

void mostradorAjustaValor(void){
if(bitTst(PORTB,0) { ssdDigit(0,2); ssdDigit(2,1); ssdDigit(5,0);}
        else { ssdDigit(0,2); ssdDigit(0,1); ssdDigit(0,0);}
if(bitTst(PORTB,1) { ssdDigit(0,2); ssdDigit(5,1); ssdDigit(0,0);}
if(bitTst(PORTB,2) { ssdDigit(0,2); ssdDigit(7,1); ssdDigit(5,0);}
if(bitTst(PORTB,3) { ssdDigit(1,2); ssdDigit(0,1); ssdDigit(0,0);}
}

unsigned char mostradorNivel(void){
if(bitTst(PORTB,0) return 25;
        else return  0;
if(bitTst(PORTB,1) return 50;
if(bitTst(PORTB,2) return 75;
if(bitTst(PORTB,3) return 100;
}
```

(25 pts) Questão 3: Uma empresa possui uma máquina que testa 4 pontos de tensão em uma placa na linha de produção (Pontos: A,B,C e D correspondentes aos bits 4,5,6 e 7 da porta A). O teste é iniciado quando os pontos A e B são ativados (ligados). Quando isso ocorre e a placa não possui falhas o ponto C deve apresentar nível lógico 0 e o ponto D nível lógico 1. Havendo falha a mensagem "FALHA" deve ser mostrada em um LCD, caso contrário a placa é liberada (bit 0 porta A). Um sensor sinaliza que a placa foi liberada (bit 1 porta A). Faça um **programa** para teste das placas.

```
#include "pic18f4520.h"
#include "config.h"
#include "lcd.h"

void main(void){
unsigned char x;
lcdInit();
bitSet(TRISA,4);
bitSet(TRISA,5);
bitSet(TRISA,6);
bitSet(TRISA,7);
bitSet(TRISA,1); //sensor de placa liberado
bitClr(TRISA,0); // placa será liberada
bitClr(PORTA,0);
for(;;){
        if(bitTst(PORTA,4)&&bitTst(PORTA,5)){
                if( !bitTst(PORTA,6) && bitTst(PORTA,7)) bitSet(PORTA,0)
                    else {
                        lcdCommand(0x80);
                        lcdData('F');
                        lcdData('A');
                        lcdData('L');
                        lcdData('H');
                        lcdData('A');
                        }
        }
        while(!bitTst(PORTA,1); //espera placa ser liberada
        bitClr(PORTA,0);
        lcdCommand(0x80);
        for(x=0;x<16;x++) lcdData(' ');  // limpa display
}
```

(30 pts) Questão 4: Uma geladeira moderna possui um sensor de temperatura com resposta linear ligado a um conversor ADC de um microcontrolador, quando a temperatura está acima de 6 graus o motor deve ligar (bit 0 porta B), quando está abaixo de 2 graus o motor deve desligar. O conversor ADC fornece valores de 0 a 1023 que corresponde a valores de temperatura de 0 a 20 graus. Nesta mesma geladeira se a porta ficar aberta (bit 1 porta B) por mais de 30 segundos um alarme deve ser ligado (bit 2 porta B). Implemente um **programa** que utiliza uma interrupção e a biblioteca adc.h para controlar a geladeira.

```
char flag;
//Esta interrupção é chamada a cada 1ms.
void Int(void) __interrupt(){
    if (BitTst(INTCON, 2)) {      //Verifica se ocorreu overflow de TMR0
        timerReset(1000);         //Ajusta o contador do TMR0
        BitClr(INTCON, 2);        //Desliga o Flag de everflow do TMR0
        flag = 1;
    }
}

Obs: Uma função InitInterrupt() ajusta todos os parâmetros da interrupção.
```

```
#include “pic18f4520.h”
#include “config.h”
#include “adc.h”

void main(void){
unsigned int tempo=0;
unsigned int temperatura;
adcInit();
InitInterrupt();
bitClr(TRISB,0); //Motor
bitSet(TRISB,1); //Porta
bitClr(TRISB,2); //Alarme
bitClr(PORTB,0); //Motor desligado
bitClr(PORTB,2); //Alarme desligado
for(;;){
        if(bitTst(PORTB,1) tempo++;  //porta aberta
                else tempo=0;
        temperatura=(adcRead()*20/1023);
        if(temperatura>6) bitSet(PORTB,0); //liga motor
        if(temperatura<2) bitClr(PORTB,0); //desliga motor
        if(tempo>30000) bitSet(PORTB,2); //alarme
                else bitClr(PORTB,2);
        while(!flag);  // Espera passar 1ms
        flag=0;
}
```

```
//adc.h
void adcInit(void);
int adcRead(void);

//config.h
Necessário para configurar o hardware
Possui comandos: #pragma …

//pic18f4520.h
Define TRISA, PORTA,... bem como:
BitSet(arg,bit)  BitClr(arg,bit)
BitFlp(arg,bit)  BitTst(arg,bit)

//Observação: TRIS = 0 → Saída
//ssd.h
void ssdInit(void);
void ssdUpdate(void);
void ssdDigit(int val, int pos);
```

```
//serial.h
void serialInit(void);
unsigned char serialRead(void);
void serialSend(unsigned char c);

//pwm.h
void pwmInit(void);
void pwmFrequecy(unsigned int freq);
void pwmSet1(unsigned char porcento);

//lcd.h
void lcdInit(void);
void lcdCommand(unsigned char c);
void lcdData(unsigned char c);
```