

Exemplo Programação de Microcontroladores

TIMER 0/TIMER 1/CONVERSOR AD/LCD/USART

Elabore as rotinas do programa em C que atenda ao seguinte projeto. Através de uma leitura analógica de um potenciômetro ligado a entrada PC0, valor lido (binário entre 0 e 1023) é utilizado para ajuste do PWM da saída PB2, saída na qual esta conectado um LED. Este valor binário é convertido para a escala (0 a 100) e enviado para o LCD e para a porta serial pré-configurada em 9600 bauds. O pino PC1 deve alternar a cada 100ms. Utilize as seguintes funções para configuração.

```
void configura_AD(){// configura
ADMUX = (1<<REFS0) | (1<<MUX2) | (1<<MUX0);
ADCSRA = (1<<ADEN) | (1<<ADPS2)|(1<<ADPS1)|(1<<ADPS0); }

signed int le_temp()
{ set_bit(ADCSRA, ADSC);//inicia a conversão
  while(tst_bit(ADCSRA,ADSC));    //espera a conversao ser finalizada
  return (ADC); }
```

```
ISR(TIMER0_OVF_vect) // interrupção estouro do TC0
{  conta++;
   TCNT0=100; // 156 contagens equivalem a 10ms em 16mhz
   if(conta==100) // 100* 10ms = 1 seg
       { cpl_bit(PORTB,0);  conta=0; } }
```

```
ISR(TIMER1_OVF_vect) // interrupção estouro timer 1
{  TCNT1=49910;
   cpl_bit(PORTB,3); }
```

```
ISR(TIMER1_COMPA_vect) // interrupção comparacao valor A
{  cpl_bit(PORTB,5); }
```

```
void set_pwm_pd6(unsigned char valor) // timer 1
{  TCCR0A |= (1<< COM0A1) | (1 << WGM01) | (1<<WGM00);
   TCCR0B = 0b00000011;
   OCR0A=valor; sei(); }
```

```
void set_pwm_pd5(unsigned char valor) // timer 0
{  TCCR0A |= (1<< COM0B1) | (1 << WGM01) | (1<<WGM00);
   TCCR0B = 0b00000011;
   OCR0B=valor;
   sei(); }
```

```
void set_pwm_pb1(unsigned int valor) // timer 1
{  ICR1 = 1023; // estabelece o valor TOP para o PWM em 1023 = 10bit
   TCCR1A |= (1<<COM1A1);
   TCCR1B = (1<< WGM13)|(1<< CS11); //T=20ms prescaler 64
   TCCR1C = 0;
   OCR1A = valor; //inicializa PWM para saida em tens, o = 0 Vcc
   sei(); }
```

```
void set_pwm_pb2(unsigned int valor) // timer 1
{  ICR1 = 1023; // estabelece o valor TOP para o PWM em 1023 = 10bit
   TCCR1A |= (1<<COM1B1);
   TCCR1B = (1<< WGM13)|(1<< CS11); //T=20ms //T=20ms prescaler 64
   TCCR1C = 0;
   OCR1B = valor; //inicializa PWM para saida em tens, o = 0 Vcc
   sei(); }
```

```
void set_timer1_overflow(unsigned int valor) // estouro timer 1
{  TCCR1A=0b01010000;
   TCCR1B = (1<<CS12) | (1<<CS10); // prescaler 1024
```

```

TIMSK1= 1 << TOIE1;
TCNT1=valor;
OCR1A=52000;
OCR1B=60000;
sei(); }

```

```

void set_timer0_overflow(unsigned int valor) // estouro timer 0
{ TCCR0A = 0b01010000;
  TCCR0B = (1<<CS02) | (1<<CS00); //TC0 com prescaler de 1024,
  TIMSK0 = 1<<TOIE0 ; //habilita a interrupção, o estouro TC0
  TCNT0= valor; // contagem iniciando
  sei(); }

```

+++++ E X E M P L O P R O G R A M A +++++

```

#include "LCD.h"
#include "def_principais.h" //inclusão do arquivo com as principais
definições
#include "USART.h"
#include "LCD.h"

```

```

void main() {
    DDRC= 0b00000000; // entrada

```

```

    unsigned char digitos[tam_vetor];
    unsigned int valorbinario;
    USART_Inic(MYUBRR);
    inic_LCD_4bits();
    configura_AD();

```

```

while (1){

```

```

    valorbinario=

```

```

    // imprime valor binário LCD
    ident_num(valorbinario,digitos); // separa digitos
    cmd_LCD(0xCC,0);
    cmd_LCD(digitos[3],1);
    cmd_LCD(digitos[2],1);
    cmd_LCD(digitos[1],1);
    cmd_LCD(digitos[0],1);
    // imprime valor binário serial
    USART_Transmite(digitos[3]);
    USART_Transmite(digitos[2]);
    USART_Transmite(digitos[1]);
    USART_Transmite(digitos[0]);
    USART_Transmite("\n");

```

```

}

```