Aluna: Tatielen Rodrigues Dutra Pereira
Matricula: 12/0136074
Data: 21/11/2017

1. **Defina a função void Atraso(volatile unsigned int x); que fornece um atraso de x milissegundos. Utilize o Timer_A para a contagem de tempo, e assuma que o SMCLK já foi configurado para funcionar a 1 MHz. Esta função poderá ser utilizada diretamente nas outras questões desta prova.**

```
void Atraso(volatile unsigned int x)
{
   TACCR0 = 1000-1;
   TACTL |= TACLR;
   TACTL = TASSEL_2 + ID_0 + MC_1;
   while(x>0)
   {
    x--;
    while((TACTL&TAIFG)==0);
          TACTL &= ~TAIFG;
   }
   TACTL = MC_0;

}

int main(void) {
   WDTCTL = WDTPW | WDTHOLD;
   BCSCTL1 = CALBC1_1MHZ;
   DCOCTL = CALDCO_1MHZ;
       return 0;
}
```

2. **Pisque os LEDs da Launchpad numa frequência de 100 Hz.**

```
#include <msp430g2553.h>
#include <intrinsics.h>

#define LED1 BIT0
#define LED2 BIT6

int main(void)
{
   WDTCTL = WDTPW + WDTHOLD;   // Stop WDT

   BCSCTL1 = CALBC1_1MHZ;      //MCLK e SMCLK @ 1MHz
   DCOCTL = CALDCO_1MHZ;       //MCLK e SMCLK @ 1MHz
```

```
  P1OUT &= ~(LED1+LED2);
  P1DIR |= LED1 + LED2;
  TA0CCR0 = 5000-1;
  TA0CTL = TASSEL_2 + ID_0 + MC_1 + TAIE;
  _BIS_SR(LPM0_bits+GIE);
  return 0;
}

#pragma vector = TIMER0_A1_VECTOR
__interrupt void TIMER0_TA0_ISR(void)
{
  P1OUT ^= LED1 + LED2;
  TA0CTL &= ~TAIFG;
}
```

3. Pisque os LEDs da Launchpad numa frequência de 20 Hz.

```
#include <msp430g2553.h>
#include <intrinsics.h>

#define LED1 BIT0
#define LED2 BIT6

int main(void)
{
  WDTCTL = WDTPW + WDTHOLD;   // Stop WDT

  BCSCTL1 = CALBC1_1MHZ;      //MCLK e SMCLK @ 1MHz
  DCOCTL = CALDCO_1MHZ;       //MCLK e SMCLK @ 1MHz
  P1OUT &= ~(LED1+LED2);
  P1DIR |= LED1 + LED2;
  TA0CCR0 = 25000-1;
  TA0CTL = TASSEL_2 + ID_0 + MC_1 + TAIE;
  _BIS_SR(LPM0_bits+GIE);
  return 0;
}

#pragma vector = TIMER0_A1_VECTOR
__interrupt void TIMER0_TA0_ISR(void)
{
  P1OUT ^= LED1 + LED2;
  TA0CTL &= ~TAIFG;
}
```

4. Pisque os LEDs da Launchpad numa frequência de 1 Hz.

```
#include <msp430g2553.h>
#include <intrinsics.h>

#define LED1 BIT0
#define LED2 BIT6

int main(void)
{
   WDTCTL = WDTPW + WDTHOLD;   // Stop WDT

   BCSCTL1 = CALBC1_1MHZ;      //MCLK e SMCLK @ 1MHz
   DCOCTL = CALDCO_1MHZ;       //MCLK e SMCLK @ 1MHz
   P1OUT &= ~(LED1+LED2);
   P1DIR |= LED1 + LED2;
   TA0CCR0 = 62500-1; //10000-1;
   TA0CTL = TASSEL_2 + ID_3 + MC_1 + TAIE;
   _BIS_SR(LPM0_bits+GIE);
   return 0;
}

#pragma vector = TIMER0_A1_VECTOR
__interrupt void TIMER0_TA0_ISR(void)
{
   P1OUT ^= LED1 + LED2;
   TA0CTL &= ~TAIFG;
}
```

5. Pisque os LEDs da Launchpad numa frequência de 0,5 Hz.

```
#include <msp430g2553.h>
#include <intrinsics.h>

#define LED1 BIT0
#define LED2 BIT6

int main(void)
{
   WDTCTL = WDTPW + WDTHOLD;   // Stop WDT

   BCSCTL1 = CALBC1_1MHZ;      //MCLK e SMCLK @ 1MHz
   DCOCTL = CALDCO_1MHZ;       //MCLK e SMCLK @ 1MHz
   P1OUT &= ~(LED1+LED2);
   P1DIR |= LED1 + LED2;
   TA0CCR0 = 62500-1; //10000-1;
```

```
   TA0CTL = TASSEL_2 + ID_3 + MC_3 + TAIE;
   _BIS_SR(LPM0_bits+GIE);
   return 0;
}

#pragma vector = TIMER0_A1_VECTOR
__interrupt void TIMER0_TA0_ISR(void)
{
   P1OUT ^= LED1 + LED2;
   TA0CTL &= ~TAIFG;
}
```

6. **Repita as questões 2 a 5 usando a interrupção do Timer A para acender ou apagar os LEDs.**

7. **Defina a função void paralelo_para_serial(void); que lê o byte de entrada via porta P1 e transmite os bits serialmente via pino P2.0. Comece com um bit em nivel alto, depois os bits na ordem P1.0 - P1.1 - … - P1.7 e termine com um bit em nível baixo. Considere um período de 1 ms entre os bits.**

```
#include <msp430.h>

/*
 * main.c
 */

void Atraso(volatile unsigned int x)
{
   TACCR0 = 1000-1;
   TACTL |= TACLR;
   TACTL = TASSEL_2 + ID_0 + MC_1;
   while(x>0)
   {
     x--;
     while((TACTL&TAIFG)==0);
           TACTL &= ~TAIFG;
   }
   TACTL = MC_0;

}

void paralelo_para_serial(void)
{

   int i,x;
   x = P1IN;
   P2OUT |= BIT0;
```

```
   Atraso(1);
   for(i=0;i<8;i++)
   {
      P2OUT |= (x&BIT0);
      Atraso(1);
      x = (x >> 1);
   }
   P2OUT &= ~(BIT0);
}

int main(void) {
   WDTCTL = WDTPW | WDTHOLD;
   BCSCTL1 = CALBC1_1MHZ;
   DCOCTL = CALDCO_1MHZ;
   P1DIR = 0;
   P2OUT = 0;
   P2DIR |= BIT0;
   for(;;)
   {
      paralelo_para_serial();
   }
   return 0;
}
```

8. **Faça o programa completo que lê um byte de entrada serialmente via pino P2.0 e transmite este byte via porta P1. O sinal serial começa com um bit em nivel alto, depois os bits na ordem 0-7 e termina com um bit em nível baixo. Os pinos P1.0-P1.7 deverão corresponder aos bits 0-7, respectivamente. Considere um período de 1 ms entre os bits.**

9. **Defina a função void ConfigPWM(volatile unsigned int freqs, volatile unsigned char ciclo_de_trabalho); para configurar e ligar o Timer_A em modo de comparação. Considere que o pino P1.6 já foi anteriormente configurado como saída do canal 1 de comparação do Timer_A, que somente os valores {100, 200, 300, …, 1000} Hz são válidos para a frequência, que somente os valores {0, 25, 50, 75, 100} % são válidos para o ciclo de trabalho, e que o sinal de clock SMCLK do MSP430 está operando a 1 MHz.**

```
ConfigPWM(volatile unsigned int freqs, volatile unsigned char ciclo_de_trabalho)
{
      TACCR0 = (1000000/freq) - 1;
      TACCR1 = (ciclo_de_trabalho*(TACCR0+1))/100 - 1;
      TACCTL1 = OUTMOD_7;
      TACTL = TASSEL_2 + ID_0 + MC_1;
}
```