

Aluno: Mateus Alves da Rocha **Matrícula:** 11/0132661

Data: 26/04/2017

Para todas as questões, utilize os LEDs e/ou os botões da placa Launchpad do MSP430.

1. Escreva um código em C que pisca os LEDs ininterruptamente.

```
#include <msp430g2553.h>

#define LED1 BIT0
#define LED2 BIT6
#define LEDS (LED1|LED2)
#define BTN BIT3

void atraso(volatile unsigned int i)
{
    while((i--)>0);
}

int main(void)
{
    WDTCTL = WDTPW | WDTHOLD;
    P1OUT |= LEDS;
    P1DIR |= LEDS;
    while(1)
    {
        atraso(0xffff);
        P1OUT ^= LEDS;
    }
    return 0;
}
```

2. Escreva um código em C que pisca os LEDs ininterruptamente. No ciclo que pisca os LEDs, o tempo que os LEDs ficam ligados deve ser duas vezes maior do que o tempo que eles ficam desligados.

```
#include <msp430g2553.h>

#define LED1 BIT0
#define LED2 BIT6
#define LEDS (LED1|LED2)
#define BTN BIT3

void atraso(volatile unsigned int i)
{
    while((i--)>0);
}

int main(void)
{
    WDTCTL = WDTPW | WDTHOLD;
    P1OUT |= LEDS;
    P1DIR |= LEDS;

    while(1)
    {
        if (P1OUT != LEDS)
        {
            P1OUT ^= LEDS;

            atraso(0xffff);

            atraso(0xffff);

        }

        atraso(0xffff);
        P1OUT ^= LEDS;
    }
}
```

```
    }  
}
```

3. Escreva um código em C que acende os LEDs quando o botão é pressionado.

```
#include <msp430.h>
```

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

```
#define LED1 BIT0
```

```
#define LED2 BIT6
```

```
#define LEDS (LED1|LED2)
```

```
#define BTN BIT3
```

```
int main(void)
```

```
{
```

```
    WDTCTL = WDTPW | WDTHOLD;
```

```
    P1OUT |= LEDS;
```

```
    P1DIR |= LEDS;
```

```
    P1DIR &= ~BTN; //implementa o btn como entrada
```

```
    P1REN |= BTN; //habilita resistor no BTN
```

```
    P1OUT |= BTN; //Leva o pino do BTN para alto
```

```
    while(1)
```

```
    {
```

```
        if((P1IN&BTN)==0)
```

```

        P1OUT |= LEDS;
    else
        P1OUT &= ~LEDS;
    }
    return 0;
}

```

4. Escreva um código em C que pisca os LEDs ininterruptamente somente se o botão for pressionado.

```

#include <msp430.h>
#include <msp430g2553.h>

#define LED1 BIT0
#define LED2 BIT6
#define LEDS (LED1|LED2)
#define BTN BIT3

void atraso(volatile unsigned int i)
{
    while((i--)>0);
}

int main(void)
{
    WDTCTL = WDTPW | WDTHOLD;

    P1OUT |= LEDS;

```

```
P1DIR |= LEDS;
```

```
P1DIR &= ~BTN; //implementa o btn como entrada
```

```
P1REN |= BTN; //habilita resistor no BTN
```

```
P1OUT |= BTN; //Leva o pino do BTN para alto
```

```
P1OUT &= ~LEDS;
```

```
while(1)
```

```
{
```

```
    if((P1IN&BTN)==0)
```

```
    {
```

```
        atraso(0xffff);
```

```
        P1OUT ^= LEDS;
```

```
    }
```

```
}
```

```
    return 0;
```

```
}
```

5. Escreva um código em C que acende os LEDs quando o botão é pressionado. Deixe o MSP430 em modo de baixo consumo, e habilite a interrupção do botão.

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

```
#include <legacymsp430.h> // Para rodar interrupcoes
```

```
#define LED1 BIT0
```

```
#define LED2 BIT6
```

```
#define LEDS (LED1|LED2)
```

```
#define BTN BIT3
```

```
void atraso(volatile unsigned int i)
```

```
{  
    while((i--)>0);  
}
```

```
int main(void)
```

```
{  
    WDTCTL = WDTPW | WDTHOLD;  
    P1OUT |= LEDS;  
    P1DIR |= LEDS;  
    P1DIR &= ~BTN;  
    P1REN |= BTN;  
    P1OUT |= BTN;  
    P1IES |= BTN;  
    P1IE |= BTN;  
    _BIS_SR(GIE);  
    while(1)  
    {  
        atraso(0xFFFF);  
        P1OUT ^= LEDS;  
    }  
}
```

```
        return 0;
    }

interrupt(PORT1_VECTOR) Interrupcao_P1(void)
{
    while((P1IN&BTN)==0);
    P1IFG &= ~BTN;
}
```