

Disciplina: Eletrônica Embarcada **Código:** 120871 **Turma:** A
Professor: Diogo Caetano Garcia
Aluno/Matrícula: Fábio Barbosa Pinto – 11/0116356

Questionário: 09_Dig_In

1. Escreva uma função em C que faz o debounce de botões ligados à porta P1.

```
#include
<msp430.h>
#define LED1 BIT0
#define LED2 BIT6
#define LEDS (LED1|LED2)
#define tempo 0x7530

int debounce ()
{
    volatile int Count;
    int comp;

    for (Count =0; Count<1000; Count++);
    {
        if(Count ==0)
            comp = P1IN;
        else if(comp != P1IN)
            {comp = P1IN;
            Count = 0;}

    }
    return P1IN;
}

int main(void)
{
    WDTCTL = WDTPW + WDTHOLD;
    P1DIR = 0xFF;
    P1OUT = 0x00;

    while (1)
    {}
}
```

2. Escreva um código em C que lê 9 botões multiplexados por 6 pinos, e pisca os LEDs da placa Launchpad de acordo com os botões. Por exemplo, se o primeiro botão é pressionado, os LEDs piscam uma vez; se o segundo botão é pressionado, os LEDs piscam duas vezes; e assim por diante. Se mais de um botão pressionado, os LEDs não piscam.

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

```
//main.c
```

```
#define Y_teclado (BIT0 + BIT1 + BIT2)
```

```
#define X_teclado (BIT3 + BIT4 + BIT5)
```

```
#define LED1 BIT6
```

```
#define LED2 BIT7
```

```
int leitor_de_botao(void)
```

```
{
```

```
    int linha,coluna,botao=0,i,j;
```

```
    /* LINHAS
```

```
    * Y1 --> P1.0
```

```
    * Y2 --> P1.1
```

```
    * Y3 --> P1.2
```

```
    *
```

```
    * COLUNAS
```

```
    * X1 --> P1.3
```

```
    * X2 --> P1.4
```

```
    * X3 --> P1.5
```

```
    */
```

```
    /*LEITURA DA LINHA Y*/
```

```
P1DIR &= ~(Y_teclado); //Definindo as linhas Y como entrada
```

```
P1OUT = 0; //Colocando as saidas em 0
```

```
P1DIR |= X_teclado; //Definindo as colunas X como saida
```

```
P1REN |= Y_teclado; //Definindo os resistores de pull-up/pull-down
```

```
P1OUT |= Y_teclado; //Definindo como pull-up
```

```
/*Um botao será pressionado se na entrada aparecer '0 logico'*/
```

```
if(((P1IN&BIT0)==0)&&(((P1IN&BIT1)==BIT1))&&(((P1IN&BIT2)==BIT2)))
{
    /* Deve-se garantir que somente 1 botão será pressionado.
    * Portanto, se por exemplo, a linha 0 for acionada,
    * a linha 1 não pode estar acionada e nem a linha 2
    * por isso esse tanto de ANDs ai
    * */
    linha = 1;
}
else if(((P1IN&BIT0)==BIT0)&&(((P1IN&BIT1)==0))&&(((P1IN&BIT2)==BIT2)))
{
    linha = 2;
}
//else if(((P1IN&BIT0)==1)&&(((P1IN&BIT1)==1))&&(((P1IN&BIT2)==0)))
else if(((P1IN&BIT0)==BIT0)&&(((P1IN&BIT1)==BIT1))&&(((P1IN&BIT2)==0)))
{
    linha = 3;
}
else
{
    linha = 0; /*Retorna 0 quando nao tem botao pressionado,
    ou se foi pressionado dois botoes de uma vez*/
}
/*LEITURA DA COLUNA X*/
```

```
P1DIR &= ~(X_teclado); //Definindo as colunas X como entrada
```

```
P1OUT = 0; //Colocando as saidas em 0
```

```
P1DIR |= Y_teclado; //Definindo as linhas Y como saida
```

```
P1REN &= ~(Y_teclado); //Nao é necessário mais os resistores para as linhas Y
```

```
P1REN |= X_teclado; //Definindo os resistores de pull-up/pull-down para as colunas X
```

```
P1OUT |= X_teclado; //Definindo como modo pull-up
```

```
if(((P1IN&BIT3)==0)&&(((P1IN&BIT4)==BIT4))&&(((P1IN&BIT5)==BIT5)))
{
    coluna = 1;
}
else if(((P1IN&BIT3)==BIT3)&&(((P1IN&BIT4)==0))&&(((P1IN&BIT5)==BIT5)))
{
    coluna = 2;
}
else if(((P1IN&BIT3)==BIT3)&&(((P1IN&BIT4)==BIT4))&&(((P1IN&BIT5)==0)))
{
    linha = 3;
}
else
{
    coluna = 0;
}

if(linha==0 || coluna==0)
{
    return 0; /*Nesse caso, nenhum botao foi pressionado, ou dois foram pressionados de uma
vez*/
}
else
{
    /*Verificando qual botao foi pressionado*/
    for(i=1;i<4;i++)
    {
        /*i - refere-se a linha
        *j - refere-se a coluna
        */
        for(j=1;j<4;j++)
        {
            botao++;
        }
    }
}
```

```
        if(((linha==i) && (coluna==j)))
        {
            break;
        }
    }
}
return botao;
}
}

void pisca_led(int quantidade_piscadas)
{
    int i;
    P1OUT &= ~(LED1 + LED2); /*Inicializar a saídas P1.6 e P1.7 em 0*/
    P1DIR |= LED1 + LED2;
    for(i=0;i<=quantidade_piscadas;i++)
    {
        P1OUT ^= LED1 + LED2;
        P1OUT ^= LED1 + LED2;
    }
}

int main(void) {
    WDTCTL = WDTPW | WDTHOLD; // Stop watchdog timer
    for(;;)
    {
        pisca_led(leitor_de_botoes());
    }
    return 0;
}
```