

Questões – Aula 03

1. Dada uma variável 'a' do tipo 'char' (um byte), escreva os trechos de código em C para:

(a) Somente setar o bit menos significativo de 'a'.

```
a |= BIT0;
```

(b) Somente setar dois bits de 'a': o menos significativo e o segundo menos significativo.

```
a |= BIT0 | BIT1;
```

(c) Somente zerar o terceiro bit menos significativo de 'a'.

```
a &= ~(BIT2);
```

(d) Somente zerar o terceiro e o quarto bits menos significativo de 'a'.

```
a &= ~(BIT2 + BIT3);
```

(e) Somente inverter o bit mais significativo de 'a'.

```
a ^= BIT7;
```

(f) Inverter o nibble mais significativo de 'a', e setar o nibble menos significativo de 'a'.

2. Considerando a placa Launchpad do MSP430, escreva o código em C para piscar os dois LEDs ininterruptamente.

```
#include <msp430g2553.h>
void main (void)
{
    WDTCTL = WDTPW | WDTHOLD;
    P1DIR = 0x41;
    P1OUT = 0x41;
    for(;;)
    {
    }
}
```

3. Considerando a placa Launchpad do MSP430, escreva o código em C para piscar duas vezes os dois LEDs sempre que o usuário pressionar o botão.

```
#include <msp430g2553.h>

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

```

P1OUT = 0;
P1DIR = BIT0 + BIT6;
for(;;)
{
    if(P1IN & BIT2 == 0)
    {
        P1OUT |= BIT0 + BIT6;
        P1OUT &= ~(BIT0 + BIT6);
        P1OUT |= BIT0 + BIT6;
        P1OUT &= ~(BIT0 + BIT6);
    }
    else
        P1OUT = 0;
}
}

```

4. Considerando a placa Launchpad do MSP430, faça uma função em C que pisca os dois LEDs uma vez.

```

#include <msp430g2553.h>

void pisca_led ()
{
    P1DIR = BIT0 + BIT6;
    P1OUT |= BIT0 + BIT6;

}

```

5. Reescreva o código da questão 2 usando a função da questão 4.

```

#include <msp430g2553.h>

void pisca_led()
{
    P1DIR = BIT0 + BIT6;
    P1OUT |= BIT0 + BIT6;

}

void main (void)
{
    WDTCTL = WDTPW | WDTHOLD;
    pisca_led();
    for(;;)
    {
    }
}

```

6. Reescreva o código da questão 3 usando a função da questão 4.

```

#include <msp430g2553.h>

void pisca_led()
{
    P1DIR = BIT0 + BIT6;
    P1OUT |= BIT0 + BIT6;

}

void main (void)
{
    WDTCTL = WDTPW | WDTHOLD;
    P1OUT = 0;
    P1DIR = BIT0+BIT6;
    for(;;)
    {
        if(P1IN & BIT2 == 0)
        {
            pisca_led();
            P1OUT &= ~(BIT0 + BIT6);
            pisca_led();
            P1OUT &= ~(BIT0 + BIT6);
        }
        else
            P1OUT = 0;
    }
}

```