

175480_02

Patrick de Carvalho Tavares Rezende Ferreira

EA871

ATIVIDADE 1:

1. a = GPIOA_PDIR;
2. 0x00101030;
3. 0x00101020;
4. 0x00101010;
5. 0x00100030;
6. 0x00100000;
7. Não pressionado: 0x00000010, Pressionado:0x00000000;
8. Não pressionado: 0x00000020, Pressionado:0x00000000;
9. Não pressionado: 0x00001000, Pressionado:0x00000000;
10. 0x00000707-antes, 0x00000105-depois;

ATIVIDADE 2:

1. Adiciona um arquivo de header ao programa, que será utilizado no programa como se todo seu conteúdo houvesse sido de fato digitado naquele ponto do código;
2. O arquivo adicionado entre aspas contém apenas definições de código, por exemplo, o cabeçalho de funções, mas sem conter as implementações em si, mesmo que contenha uma diretiva para incluir um arquivo entre os símbolos <>. Um arquivo incluído dentre estes símbolos contém não só definições, mas também contém as implementações das funções, por exemplo;
3. Separar certos trechos de código em elementos reutilizáveis;
4. Contém apenas uma instrução de inclusão do header MKL25Z4.h, que, por sua vez, contém uma série de definições em seu interior;
5. Ambos representam o mesmo endereço cujo valor é 0x40048038 (Endereço da struct: 0x40047000, endereço relativo do elemento SCGC5: 0x1038);
6. Considerando-se a posição de memória ocupada por cada componente, segundo o manual, a estrutura de dados definida em MKL25Z4.h define os elementos da struct de forma a coincidir com os elementos do hardware. Isto viabiliza a manipulação direta destes elementos de hardware.

ATIVIDADE 3:

```
#include "derivative.h" /* include peripheral declarations */
```

```
/* Inicializa os GPIOs */
```

```
void inicGPIO(void) {
```

```

SIM_SCGC5 = 0x00000A00; // Habilita PORTA e PORTC

PORTC_PCR0 = 0x00000100; // Configura pin MUX control para PORTC aparecer nos
pinos: (001)
PORTC_PCR1 = 0x00000100;
PORTC_PCR2 = 0x00000100;
PORTC_PCR3 = 0x00000100;
PORTC_PCR4 = 0x00000100;
PORTC_PCR5 = 0x00000100;
PORTC_PCR6 = 0x00000100;
PORTC_PCR7 = 0x00000100;
PORTC_PCR10 = 0x00000100;

GPIOC_PDDR = 0x000004FF; // Configura bits 0-7 e 10 do PORTC como saídas
// LEDs apagados (bits 0-7) em zero
// LE (Latch Enable) do registrador 74573 desabilitado (0)

PORTA_PCR4 = 0x00000100; // Configura MUX para PORTA aparecer nos pinos
PORTA_PCR5 = 0x00000100;
PORTA_PCR12 = 0x00000100;

GPIOA_PDDR = 0;
}

void delay(unsigned int tempo) {
    while (tempo) tempo--;
}

int main(void)
{
    inicGPIO();
    int s, b1 = 0, b1old = 0, b2 = 0, b2old = 0, b3 = 0, b3old = 0, toggle1 = 0, toggle2 = 0,
    toggle3 = 0, block = 0;
    unsigned int a, pta4, pta5, pta12;

    GPIOC_PDOR = 0x00000000;

    for(;;){
        s = GPIOC_PDOR & 0x000000FF;

        a = GPIOA_PDIR; // Faz a leitura na PORTA (32 bits)
        pta4 = a & (1<<4);
        pta5 = a & (1<<5);
        pta12 = a & (1<<12);
    }
}

```

```

    if(!pta4){ b1=1;} else { b1=0;} //Aqui apenas lemos quais botões estão pressionados no
momento.
    if(!pta5){ b2=1;} else { b2=0;}
    if(!pta12){ b3=1;} else { b3=0;}

    //Os delays são utilizados apenas para fins de debouncing.
    if(!b1 && b1old){toggle1 = 1;delay(50000);} //Este três if fazem apenas detecção de borda
de descida dos botoes.
    if(!b2 && b2old && !block){toggle2 = 1; delay(50000);}
    if(!b3 && b3old && !block){toggle3 = 1; delay(50000);}

    b1old = b1; b2old = b2; b3old = b3;

    if(toggle1){
        s += ((s & 1) == 0 ? 1:-1); //Se o bit zero valia 1, agora vale zero. Se valia zero, agora vale
1.
        toggle1--; //Reseta o toggle.
    }

    if(toggle2 && !b1){
        s = (s>>1);
        toggle2 = 0;
        block = 1;
    }

    if(toggle3 && !b1){
        s = (s<<1);
        toggle3 = 0;
        block = 1;
    }

    if(!b2 && !b3) block = 0; //Se ambos os botoes houverem sido soltos, desative o bloqueio
imposto para que somente o primeiro deles fosse levado em conta.

    GPIOC_PDOR &= 0xFFFFF00; //Zera os 8 bits iniciais do registrador.
    GPIOC_PDOR |= s & 0x000000FF; // Passa ao registrador somente os bits que nos
interessam.
    GPIOC_PSOR = (1<<10); // pulsa LE (Latch Enable) do 74573, em 1 (set)
    GPIOC_PCOR = (1<<10); // em 0 (clear)

}
}

```