

NAC 02 PCM

Alunos:

Jan Peter Merkel

Rm:81896

Igor Tomazelli Nietmann

Rm:82808

Exercício 1)

Código Fonte:

```
; 1. CABEÇALHO DE IDENTIFICAÇÃO

;FIAP
;PCM
;EXEMPLO EXPERIÊNCIA 2
;Jan Peter Merkel RM:81896
;Igor Tomazelli Nietmann RM:82808


; 2. ARQUIVO DE DEFINIÇÃO DO MICROCONTROLADOR

#include <P18F452.INC>


; 3. DEFINIÇÃO DE VARIÁVEIS


; 3.1 GERAIS

CBLOCK 0x80          ; ENDEREÇO INICIAL DA MEMÓRIA DE DADOS
    FILTRO           ; VARIÁVEL FILTRO PARA O BOTÃO
    FILTRO2
    FILTRO3          ; TIPO INT COM TAMANHO ATÉ 256 DECIMAL
                     ; DE 0 A 255
ENDC                ; FIM DO BLOCO


; 3.2 CONSTANTES

T_FILTRO EQU .230    ; CONSTANTE T_FILTRO COM 230 DECIMAL
```

; 3.3 I/Os

```
#DEFINE BOTAO_RB3 PORTB,3      ; PORTB = BIT 3 (RB3) - MP4
#DEFINE BOTAO_RB4 PORTB,4      ; SITUAÇÕES LÓGICAS: 0- PRESSIONADO - Blue Ray
#DEFINE BOTAO_RB5 PORTB,5      ; PORTB = BIT 3 (RB3) - TV
                                ; 1 = LIBERADO

#DEFINE LED_MP4 PORTD,2        ; PORTD = BIT 0 (RD0)
#DEFINE LED_BLUE_RAY PORTD,3   ; SITUAÇÕES LÓGICAS: 0 = APAGADO
#DEFINE LED_TV PORTD,4
```

; 4. VETOR DE RESET

ORG 0x00

GOTO INICIO

; 5. INÍCIO DO PROGRAMA

; CONFIGURAÇÕES INICIAIS (REGISTRADORES DE FUNÇÕES ESPECIAIS)

INICIO

```
        CLRF PORTD      ; LIMPA PORTD (ESCREVE ZERO EM TODOS OS PINOS DO
PORTD)
```

```
        MOVLW B'00000000' ; MOVE O VALOR BINÁRIO PARA O REGISTRADOR
WORK
```

```
        MOVWF TRISA      ; DEFINE TODOS OS TERMINAIS DO PORTA COMO
SAÍDA
```

```
        MOVLW B'00111000' ; MOVE O VALOR BINÁRIO PARA O REGISTRADOR
WORK
```

```

MOVWF TRISB          ; DEFINE RB3 COMO ENTRADA E DEMAIS COMO
SAÍDA

MOVLW B'00000000'    ; MOVE O VALOR BINÁRIO PARA O REGISTRADOR
WORK

MOVWF TRISC          ; DEFINE TODOS OS TERMINAIS DO PORTC COMO
SAÍDA

MOVLW B'00000000'    ; MOVE O VALOR BINÁRIO PARA O REGISTRADOR WORK

MOVWF TRISD          ; DEFINE TODOS OS TERMINAIS DO PORTD COMO
SAÍDA

MOVLW B'00000000'    ; MOVE O VALOR BINÁRIO PARA O REGISTRADOR
WORK

MOVWF TRISE          ; DEFINE TODOS OS TERMINAIS DO PORTE COMO
SAÍDA

; 6. FUNÇÃO PRINCIPAL

MAIN

;MOVE PRO WORK e do WORK pra variável.

MOVLW T_FILTRO      ;MOVE T_FILTRO(230) PARA FILTRO OU SEJA FILTRO =
T_FILTRO

MOVWF FILTRO        ; INICIA O FILTRO (FILTRO = T_FILTRO)


GOTO VERIFICA_MP4


VERIFICA_MP4

BTFSC BOTAO_RB3      ; O BOTÃO 3 ESTÁ PRESSIONADO?

```

```

    GOTO BOTAO_RB3_APAGADO;GOTO PROC_2          ; NÃO, ENTÃO CONTINUE
Compare o botao rb4

    DECFSZ FILTRO,1      ; SIM, ENTÃO DECREMENTA O FILTRO DO BOTÃO Se for 0 o
resultado vai pro work se 1 fica no próprio registrador

    BSF LED_MP4          ; AVALIA O LED E INVERTE SUA CONDIÇÃO LÓGICA

    BCF LED_BLUE_RAY

    BCF LED_TV

    GOTO MAIN

BOTAO_RB3_APAGADO

    BCF LED_MP4

    MOVLW T_FILTRO      ;MOVE T_FILTRO(230) PARA FILTRO OU SEJA FILTRO =
T_FILTRO

    MOVWF FILTRO2

BTFSB BOTAO_RB4        ; O BOTÃO 3 ESTÁ PRESSIONADO?

    GOTO BOTAO_RB4_APAGADO ; NÃO, ENTÃO CONTINUE Compare o botao rb4

    DECFSZ FILTRO2,1    ; SIM, ENTÃO DECREMENTA O FILTRO DO BOTÃO Se for 0 o
resultado vai pro work se 1 fica no próprio registrador

    BSF LED_BLUE_RAY    ; AVALIA O LED E INVERTE SUA CONDIÇÃO LÓGICA

    BCF LED_TV

    GOTO MAIN

BOTAO_RB4_APAGADO

    BCF LED_BLUE_RAY

    MOVLW T_FILTRO      ;MOVE T_FILTRO(230) PARA FILTRO OU SEJA FILTRO =
T_FILTRO

    MOVWF FILTRO3

BTFSB BOTAO_RB5        ; O BOTÃO 3 ESTÁ PRESSIONADO?

    GOTO BOTAO_RB5_APAGADO ; NÃO, ENTÃO CONTINUE Compare o botao rb4

```

```
    DECFSZ FILTRO3,1    ; SIM, ENTÃO DECREMENTA O FILTRO DO BOTÃO Se for 0 o  
    resultado vai pro work se 1 fica no próprio registrador
```

```
    BSF LED_TV          ; AVALIA O LED E INVERTE SUA CONDIÇÃO LÓGICA
```

```
    GOTO MAIN
```

```
    BOTAO_RB5_APAGADO
```

```
    BCF LED_TV
```

```
    GOTO MAIN
```

```
END
```

Situações Lógicas:

Botao Apertado – 0

Botão Liberado - 1

Link do vídeo:

<https://drive.google.com/file/d/1QislaTJNdv61xY5qbipkkjqv676eaOZG/view?usp=sharing>