



**Faculdade de Computação e Engenharia Elétrica**  
**Microprocessadores e Microcontroladores – Prof. Dr. Elton Alves**  
**Experimento 7 – Projeto Assembly no PIC16F628A**

- **Objetivo:**
  - Desenvolver códigos em Assembly com dados de entrada, sub-rotina e temporização.
- **Rodar o código 3 no Mplab para Acender um LED com botão.**

```
; Microcontroladores e Microprocessadores
; Aula 02
; Prof. Elton Alves
; programa botão com led em Assembly
; botão ligado em pull-up:
; 0 - botão acionado
; 1 - botão desligado
; Aciona LED ligado em RB7, a partir de um botão ligado em RB0

;list p=16f628A ; microcontrolador utilizado
;---Arquivos incluídos no projeto---
#include <P16f628a.inc> ; inclui o arquivo do 16f628a (registradores)

;---FUSE bits---
;Cristal oscilador externo 4MHZ
;Sem watchdog time
;Com power up time
    __config _XT_OSC & _WDT_OFF & _PWRTE_ON & _CP_OFF

;---Paginação de Memória
#define bank0    bcf STATUS, RP0 ;cria um mnemônico para o banco 0 de
memória
#define bank1    bsf STATUS, RP0 ; cria um mnemônico para o banco 1 de
memória

;---Entradas---
#define botao1 PORTB,RB0 ; botão 1 ligado em RB0

;---Saídas---
#define led1 PORTB,RB7

;---Vetor de RESET---
    org H'000' ; origem no endereço 000h de memória
    goto inicio ; desvia do vetor de interrupção

;---Vetor de Interrupção----
    org H'0004' ; todas as interrupções apontam para este endereço
    retfie ; retorna a interrupção
```

;---Programa Principal----

inicio

```
bank1 ; seleciona o banco 1 de memória
movlw H'FF' ; w=B'11111111
movwf TRISA ; TRISA=H'FF' (todos bits são entradas)
movlw H'7F' ; w=B'01111111'
movwf TRISB ;TRISB=H'7F' (apenas RB7 como saída)
```

```
bank0 ; secciona o banco 0 de memória (padrão RESET)
movlw H'FF' ; w=B'11111111'
movwf PORTB; (RB7 configurado como saída - HIGH)
```

; goto \$ ; segura o código - \$ indica a posicional atual do código (loop infinito)

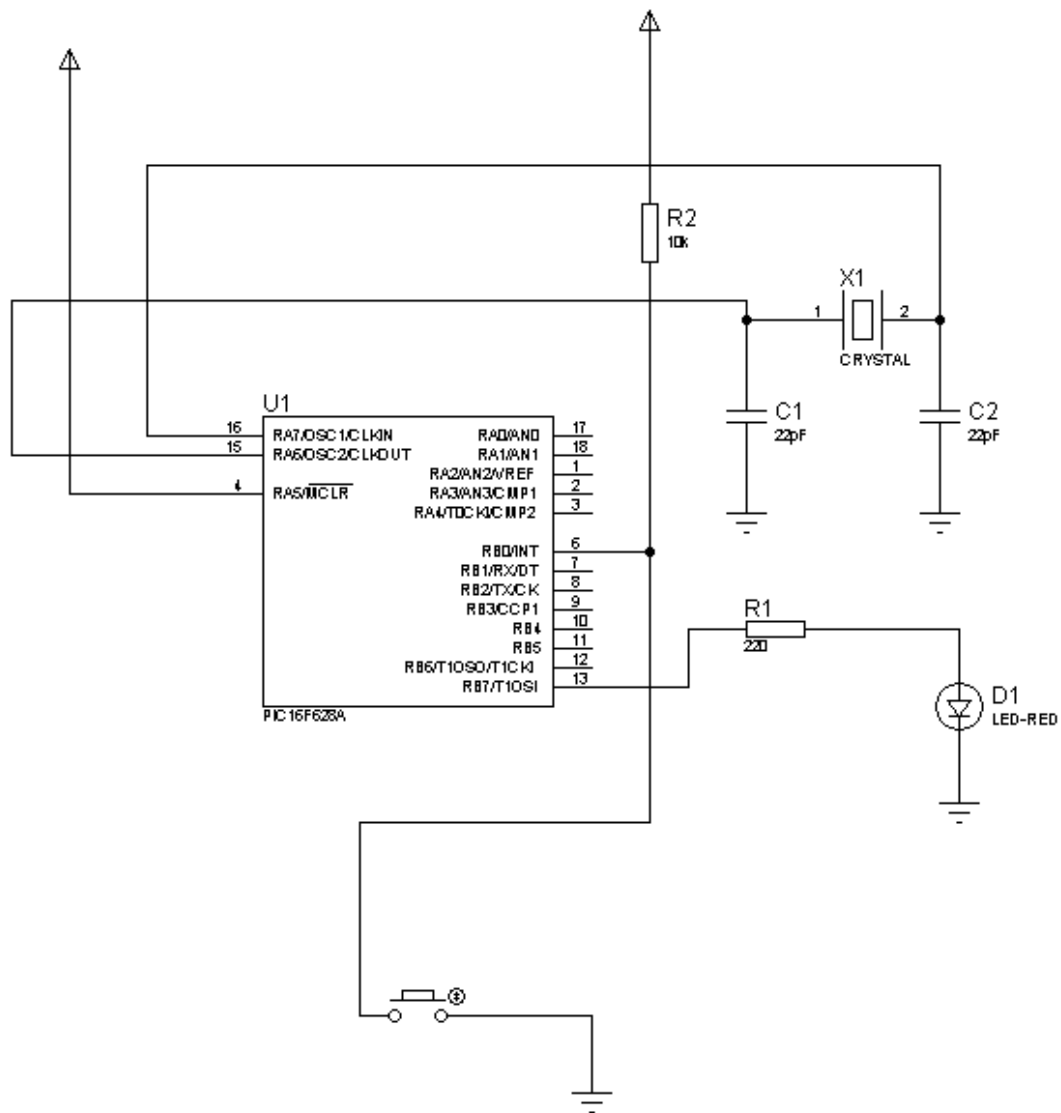
loop

```
btfsc botao1 ; testa bit, se for 0, pula uma linha
goto apaga_led1; Não pressiona, devia para apaga_led1
bsf led1; Sim, liga led1
goto loop; volta label loop
```

```
apaga_led1
bcf led1; apaga led1
goto loop; volta para o label loop
```

end

- Após criar o Projeto do código 3, o mesmo deve ser importado para o Proteus ou outro software de simulação.



- **Rodar o código 4 no Mplab para criar uma sub-rotina.**

; Microcontroladores e Microprocessadores

; Aula 02

; Prof. Elton Alves

; programa botão com led em Assembly (Exemplo de Sub-rotina)

; botão ligado em pull-up:

; 0 - botão acionado

; 1 - botão desligado

; Aciona LED1 ligado em RB1, a partir de um botão1 ligado em RB0

; Aciona LED2 em RB3, a partir de botão2 ligado em RB2

; list p=16f628A ; microcontrolador utilizado

; ---Arquivos incluídos no projeto---

#include <P16f628a.inc> ; inclui o arquivo do 16f628a (registradores)

; ---FUSE bits---

; Cristal oscilador externo 4MHZ

```

;Sem watchdog time
;Com power up time
__config _XT_OSC & _WDT_OFF & _PWRTE_ON & _CP_OFF

;---Paginação de Memória
#define bank0 bcf STATUS, RP0 ;cria um mnemônico para o banco 0 de memória
#define bank1 bsf STATUS, RP0 ; cria um mnemônico para o banco 1 de memória

;---Entradas---
#define botao1 PORTB,RB0 ; botão 1 ligado em RB0
#define botao2 PORTB,RB2; botão 2 ligado em RB2

;---Saídas---
#define led1 PORTB,RB1 ; led1 ligado em RB1
#define led2 PORTB,RB3 ;led2 ligado em RB3

;---Vetor de RESET---
org H'000' ; origem no endereço 000h de memória
goto inicio ; desvia do vetor de interrupção

;---Vetor de Interrupção----
org H'0004' ; todas as interrupções apontam para este endereço
retfie ; retorna a interrupção

;---Programa Principal----
inicio
    CLRF PORTA ; Limpa PORTA
    CLRF PORTB ; Limpa PORTB

    bank1 ; seleciona o banco 1 de memória
    movlw H'FF' ; w=B'11111111
    movwf TRISA ; TRISA=H'FF' (todos bits são entradas)
    movlw H'F5' ; w=B'1111 0101'
    movwf TRISB ;TRISB=H'F5' (apenas RB1 e RB3 como saída)

    bank0 ; seleciona o banco 0 de memória (padrão RESET)
    movlw H'F5' ; w=B'11110101'
    movwf PORTB; (Leds iniciam desligados-escrevo no PORTB)

    ; goto $ ; segura o código - $ indica a posicional atual do código (loop infinito)

loop ; loop infinito

    call trata_but1 ; chama sub-rotina-trata_but1
    call trata_but2; chama sub-rotina_trata_but2

    goto loop ;volta para o label loop

;---Desenvolvimento das Sub-rotinas----

```

```

trata_but1 ; sub-rotina para tratar botão 1
  btfsc botao1 ; botão foi pressionado?
  goto apaga_led1; Não pressiona, devis para label apaga_led1
  bsf led1; Sim, liga led1
  return ; retorno da subrotina

```

```

apaga_led1
  bcf led1; apaga led1
  return ;retorna da sub-rotina

```

```

trata_but2 ; sub-rotina para tratar botão 2
  btfsc botão2 ; botão foi pressionado?
  goto apaga_led2; Não pressiona, devis para label apaga_led1
  bsf led2; Sim, liga led1
  return ; retorno da subrotina

```

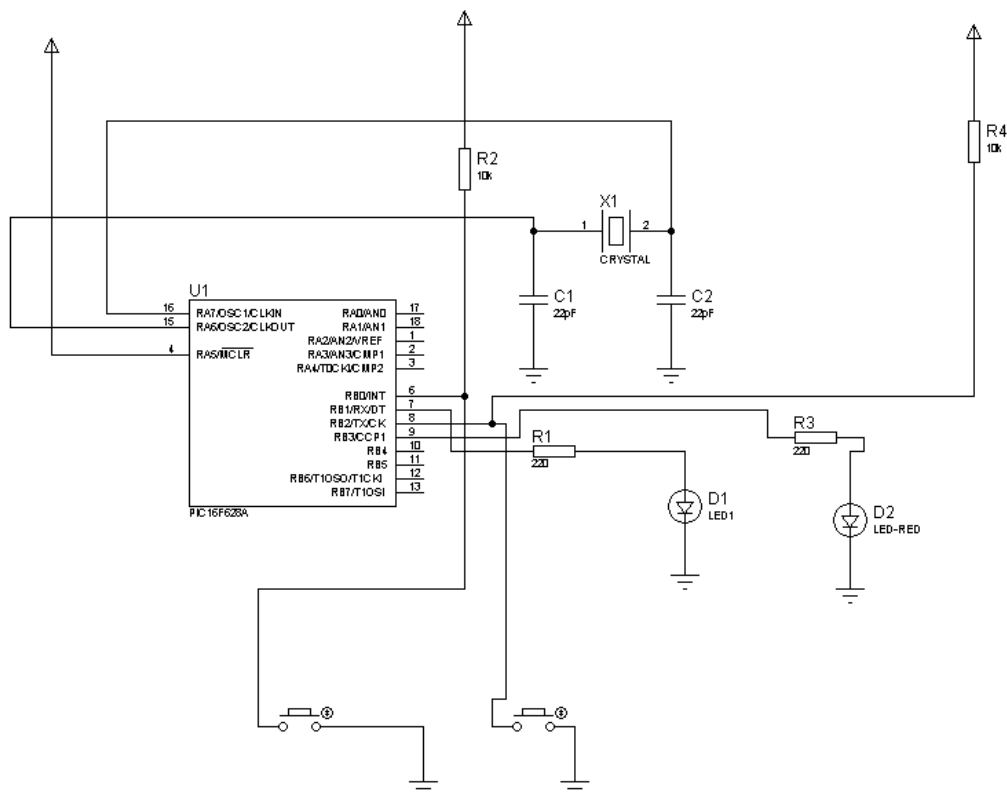
```

apaga_led2
  bcf led2; apaga led1
  return ;retorna da sub-rotina

```

end

- Após criar o Projeto do código 4, o mesmo deve ser importado para o Proteus ou outro software de simulação.



- **Rodar o código 5 no Mplab para cálculo de Delay e Ciclo de Máquina (temporização)**

; Microcontroladores e Microprocessadores

; Aula 02 - Temporização (Delay)

; Prof. Elton Alves

; Aciona LED1 ligado em RB1 e apaga LED2 ligado em RB3

;aguarda 500 milisegundos

;Aciona LED2 ligado em RB3 e apaga LED1 em RB1

;aguarda 500 milisegundos

;0 - botão acionado

;1 - botão desligado

;calculo de ciclos de Máquina

;ciclo de máquina =  $1/(\text{Freq. Cristal}/4) = 1\mu\text{s}$

;list p=16f628A ; microcontrolador utilizado

;---Arquivos incluídos no projeto---

#include <P16f628a.inc> ; inclui o arquivo do 16f628a (registradores)

;---FUSE bits---

;Cristal oscilador externo 4MHZ

;Sem watchdog time

;Com power up time

\_\_config \_XT\_OSC & \_WDT\_OFF & \_PWRTE\_ON & \_CP\_OFF

;---Paginação de Memória

#define bank0 bcf STATUS, RP0 ; cria um mnemônico para o banco 0 de memória

#define bank1 bsf STATUS, RP0 ; cria um mnemônico para o banco 1 de memória

;---Entradas---

#define botao1 PORTB, RB0 ; botão 1 ligado em RB0

#define botão2 PORTB, RB2; botão 2 ligado em RB2

;---Saídas---

#define led1 PORTB, RB1 ; led1 ligado em RB1

#define led2 PORTB, RB3 ; led2 ligado em RB3

;---Vetor de RESET---

org H'000' ; origem no endereço 000h de memória

goto inicio ; desvia do vetor de interrupção

;---Vetor de Interrupção----

org H'0004' ; todas as interrupções apontam para este endereço

retfie ; retorna a interrupção

;---Programa Principal----

inicio

```
CLRF PORTA ; Limpa PORTA
CLRF PORTB ; Limpa PORTB
```

```
bank1 ; seleciona o banco 1 de memória
movlw H'FF' ; w=B'11111111
movwf TRISA ; TRISA=H'FF' (todos bits são entradas)
movlw H'F5' ; w=B'11110101'
movwf TRISB ;TRISB=H'F5' (apenas RB1 e RB3 como saída)
```

```
bank0 ; seleciona o banco 0 de memória (padrão RESET)
movlw H'F5' ; w=B'11110101'
movwf PORTB; (Leds iniciam desligados)
```

```
; goto $ ; segura o código - $ indica a posicional atual do código (loop infinito)
```

```
loop ; loop infinito
```

```
bsf led1 ; liga LED1
bcf led2 ; desliga LED2
call delay500ms ;chama sub_rotina
bcf led1 ; desliga LED1
bsf led2 ; liga LED1
call delay500ms ;chama sub-rotina
```

```
goto loop ;volta para o label loop
```

```
;---Desenvolvimento das Sub-rotinas---
```

```
delay500ms
    movlw D'200' ; move o valor para W (constante)
    movwf H'20' ; inicialização da variavel tempo0 (posição de memória do
    registrador de uso geral)
```

```
aux1
    movlw D'250'
    movwf H'21'
```

```
aux2 ; gastar 1 ciclo de máquina (aproximar mais o tempo de 500ms)
    nop
    nop
    nop
    nop
    nop
    nop
    nop
```

```
    decfsz H'20' ; decrementa o tempo1 até que seja igual a 0(decremente em uma
    unidade e verifica se é 0)
    goto aux2 ; vai para label aux2
```

; 250 x 10 ciclos de máquina = 2500

decfsz H'32'; decreenta o tempo0 até seja igual a 0  
goto aux1; vai para label aux1

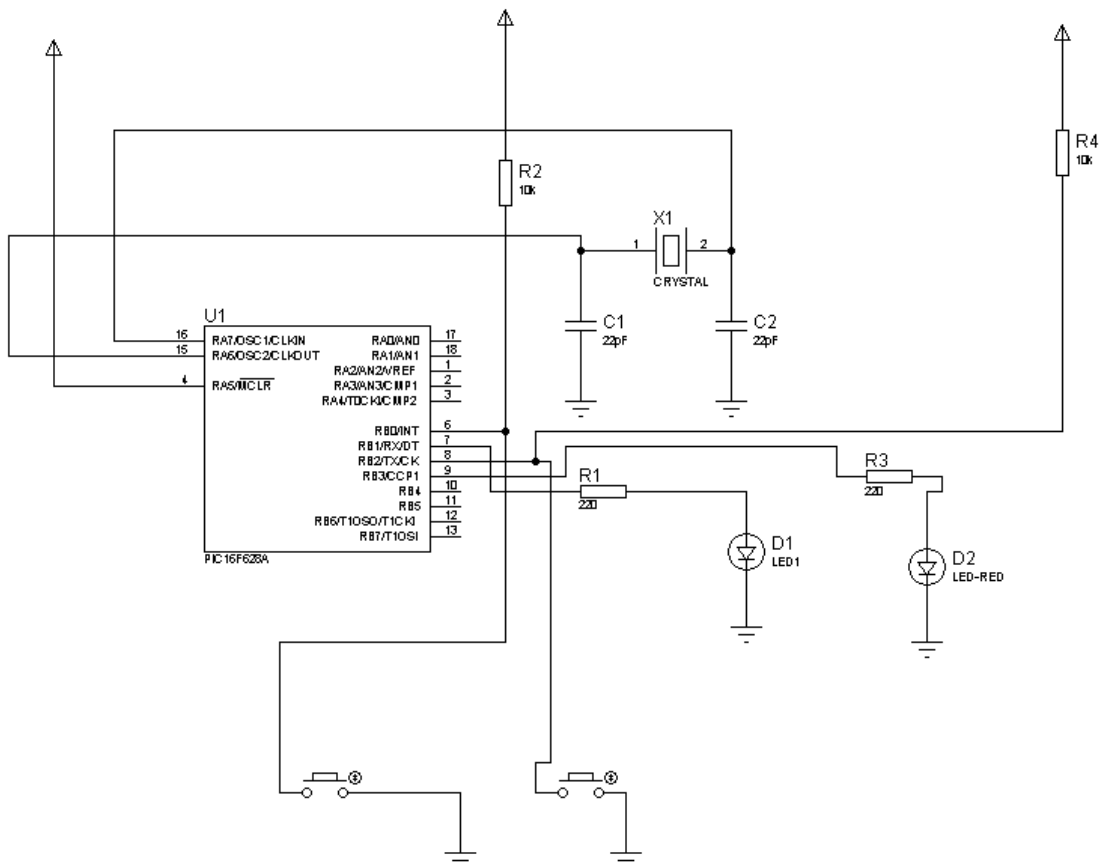
; 3 ciclos de máquina

; 2500x200 = 500000

return

end

- Após criar o Projeto do código 5, o mesmo deve ser importado para o Proteus ou outro software de simulação.



#### ATIVIDADE AVALIATIVA:

- Desenvolva um programa e simule um semáforo de 3 tempos, acionado com botão para pedestre. Utilize temporização.
- Inverta a lógica do LED, fazendo acender quando botão está solto e apagar quando botão for pressionado.
- Data da entrega: 21/07/2021
- O relatório deve ser enviado, juntando com .HEX e o arquivo de simulação.



