



**Universidade Federal do Sul e Sudeste do Pará  
Faculdade de Computação e Engenharia Elétrica  
Curso de Engenharia da Computação  
Iago Costa das Flores**

**Microprocessadores e Microcontroladores  
Experimento 6**

**Marabá  
2021**

**Microprocessadores e Microcontroladores**  
**Experimento 6**

Relatório apresentado no curso de Engenharia da Computação, turma de 2018 como obtenção de nota parcial na disciplina de microprocessadores e microcontroladores, ministrada pelo Professor Dr. Elton Alves.



## **Sumário**

<b>1 - Introdução</b>	<b>4</b>
<b>2 - Atividades</b>	<b>4</b>
2.1 - Desenvolva um programa e simule para piscar dois LEDs.	4
<b>3 - Conclusão</b>	<b>9</b>
<b>4 - Referências</b>	<b>9</b>

# 1 - Introdução

O Trabalho visa apresentar os códigos fontes e resultados de execução das três atividades avaliativas a seguir:

- 1- Desenvolva um programa e simule para piscar dois LEDs.

## 2 - Atividades

As atividades demonstradas a seguir foram feitas com a ajuda do programa emu8086 para escrever e executar os códigos em assembly.

### 2.1 - Desenvolva um programa e simule para piscar dois LEDs.

#### **Código da atividade 01:**

```
; Rodar o código 1 no Mplab para piscar dois leds.
; Microcontroladores e Microprocessadores
; Aula 01
; Prof. Elton Alves
; 13/07/2021

;---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

;---Saídas---
#define led1 PORTB,0 ; 0 -> Apagado e 1-> Aceso
#define led2 PORTA,0
;---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

;---Inicio do Programa----

inicio

CLRF PORTB ; limpa a PORTB

CLRF PORTA

bank1

movlw B'00000000'

movwf TRISB ; Define toda PORTB como saída

movwf TRISA

;movwf b'10000000'

;movwf OPTION\_REG

movlw b'00000000'

movwf INTCON ; todas as interrupções desligadas

bank0

movlw b'00000111'

movwf CMCON ; define o modo de operação do comparador

;bank0 ; seleciona o banco 0 de memória (padrão RESET)

; movlw H'FF' ; w=B'11111111'

; movwf PORTB; (RB7 configurado como saída - HIGH)

loop

bsf led1

bcf led1

bsf led2

bcf led2

goto loop

END

Na figura 01 na linha 22 temos a linha que usamos para configurar a portaA como saída para o led2, estando inicialmente desativada.

```
1  ; Rodar o código 1 no Mplab para Acender um LED.
2  ; Microcontroladores e Microprocessadores
3  ; Aula 01
4  ; Prof. Elton Alves
5  ; programa para acender um Led em Assembly
6
7  ;---Arquivos incluídos no projeto---
8  #include <P16f628a.inc> ; inclui o arquivo do 16f628a (registradores)
9
10 ;---FUSE bits---
11 ;Cristal oscilador externo 4MHZ
12 ;Sem watchdog time
13 ;Com power up time
14 __config _XT_OSC & _WDT_OFF & _PWRTE_ON & _CP_OFF
15
16 ;---Paginação de Memória
17 #define bank0 bcf STATUS, RPO ; cria um mnemônico para o banco 0 de memória
18 #define bank1 bsf STATUS, RPO ; cria um mnemônico para o banco 1 de memória
19
20 ;---Saídas---
21 #define led1 PORTB,0 ; 0 -> Apagado e 1-> Aceso
22 #define led2 PORTA,0
23 ;---Vetor de RESET---
24 org H'000' ; origem no endereço 000h de memória
25 goto inicio ; desvia do vetor de interrupção
26
27 ;---Vetor de Interrupção---
28 org H'0004' ; todas as interrupções apontam para este endereço
29 retfie ; retorna a interrupção
30
```

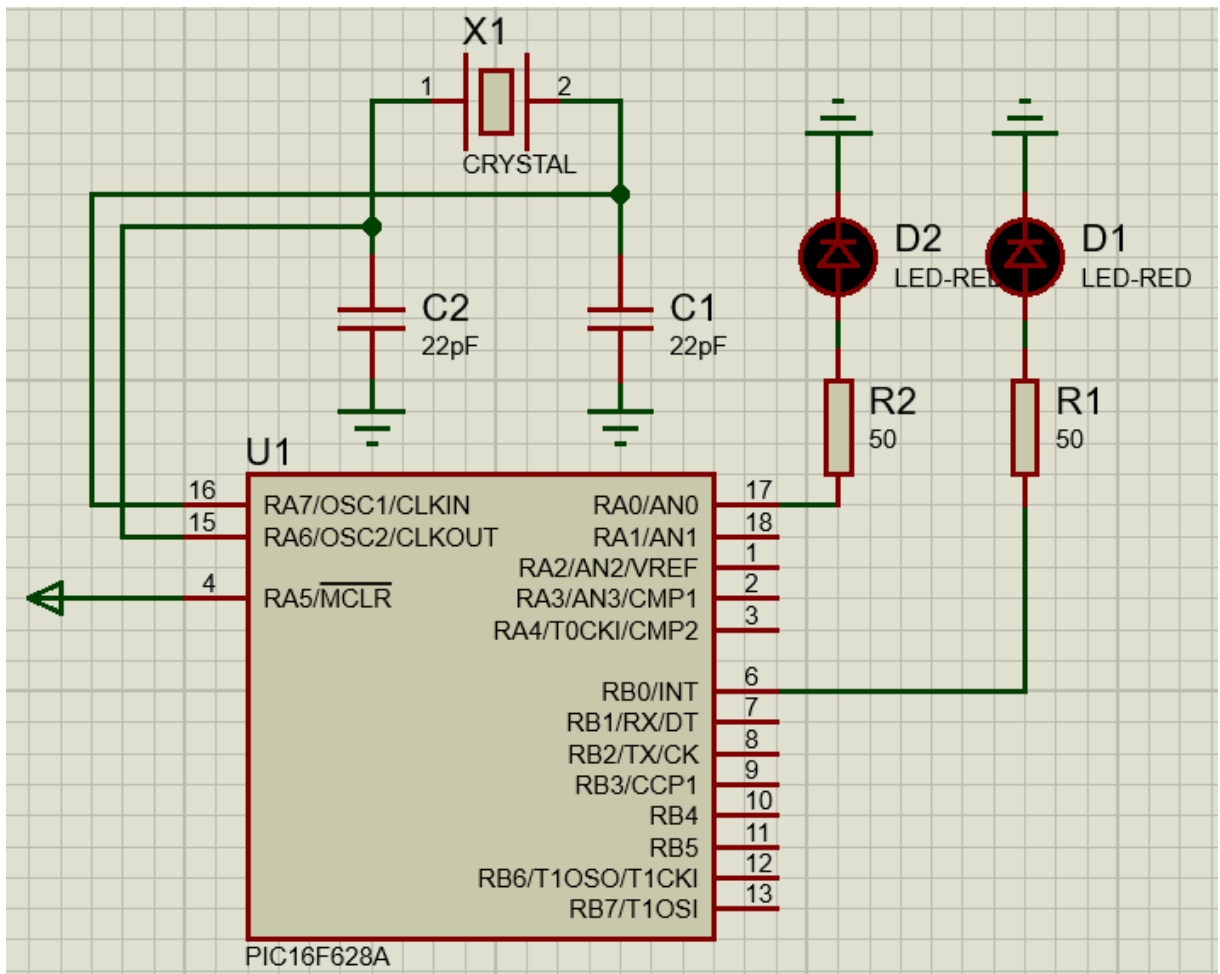
**Figura 01:** Código fonte da atividade 01 - Parte 01

Na figura 02 na linha 51 a 54 temos as linhas que ativam e desativam as saídas das portas onde estão conectados os leds. Na linha 38 temos a definição da Porta A como saída.

```
30
31 ;---Inicio do Programa-----
32 inicio
33 CLRF PORTB ; limpa a PORTB
34 CLRF PORTA
35 bank1
36     movlw B'00000000'
37     movwf TRISB ; Define toda PORTB como saída
38     movwf TRISA
39     ;movwf b'10000000'
40     ;movwf OPTION_REG
41
42     movlw b'00000000'
43     movwf INTCON ; todas as interrupções desligadas
44 bank0
45     movlw b'00000111'
46     movwf CMCON ; define o modo de operação do comparador
47 ;bank0 ; sequeciona o banco 0 de memória (padrão RESET)
48     ; movlw H'FF' ; w=B'11111111'
49     ; movwf PORTB; (RB7 configurado como saída - HIGH)
50 loop
51 bsf led1
52 bcf led1
53 bsf led2
54 bcf led2
55 goto loop
56 END
```

Figura 02: Código fonte da atividade 01 - Parte 02

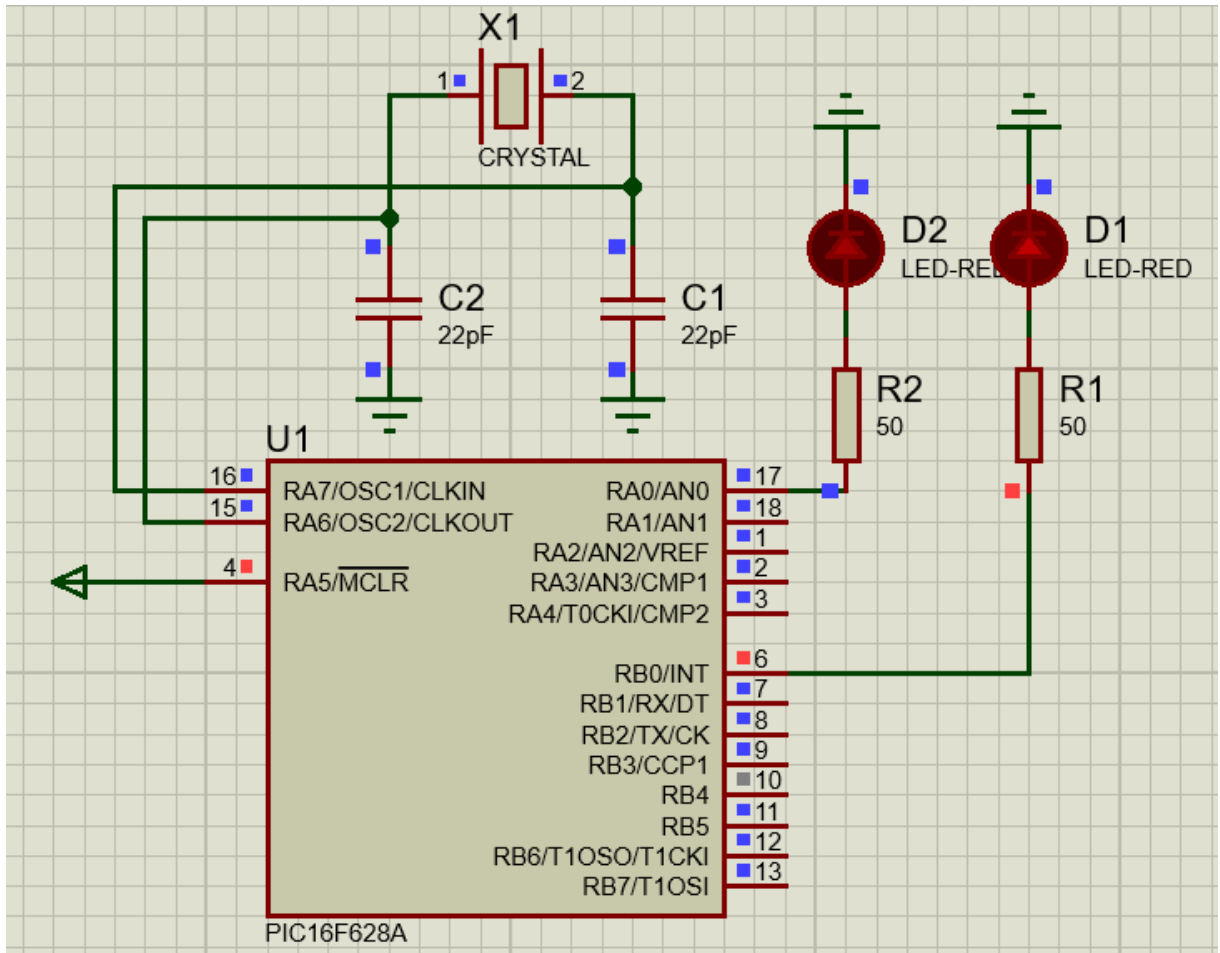
Na figura 03, temos a execução da atividade 01, com o esquema montado com o PIC16F628A e os dois leds conectados nas saídas.



**Figura 03:** Execução da atividade 01 - Parte 01



Na figura 04, temos a execução da atividade 01, com o esquema montado com o PIC16F628A e os dois leds conectados nas saídas. Onde mostra um dos leds ativando com a simulação ativa.



**Figura 04:** Execução da atividade 01 - Parte 02

### 3 - Conclusão

Foram demonstradas as impressões do código e execução dos mesmos através das imagens apresentadas com suas devidas explicações. Os códigos fontes estão comentados e a atividade foi feita conforme o solicitado no comando do trabalho.

### 4 - Referências

MANZANO, J. A. **Programação assembly: padrão IBM - PC 8086/8088**. 6ªed. Ed.Erica, 2012.