



**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 8**

**Marabá
2021**

Microprocessadores e Microcontroladores
Experimento 8

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

| | |
|--|----------|
| 1 - Introdução | 4 |
| 2 - Atividades | 4 |
| 2.1 - Desenvolva uma simulação que permita um LED ficar aceso enquanto o TIMER0 ficar rodando. | 4 |
| 3 - Conclusão | 8 |
| 4 - Referências | 8 |

1 - Introdução

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

- 1 - Desenvolva uma simulação que permita um LED ficar aceso enquanto o TIMER0 ficar rodando.

2 - Atividades

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

2.1 - Desenvolva uma simulação que permita um LED ficar aceso enquanto o TIMER0 ficar rodando.

Código da atividade 01:

```
; Temporização com TIMER0 e led
; Interrupção com TIMER0 e led
; Clock: 4MHz e Ciclo de Máquina = 1us
;---Listagem do Processador Utilizado---list p=16F628A
#include <P16f628a.inc>
;---Arquivos Inclusos no Projeto ---

; --- FUSE Bits ---
; - Cristal de 4MHz
; - Desabilitamos o Watch Dog Timer
; - Habilitamos o Power Up Timer
; - Brown Out desabilitado
; - Sem programação em baixa tensão, sem proteção de código, sem
proteção da memória EEPROM
__config _XT_OSC & _WDT_OFF & _PWRTE_ON & _BOREN_OFF & _LVP_OFF &
_CP_OFF & _CPD_OFF & _MCLRE_ON

;---Paginação de Memória---
#define bank0 bcf STATUS, RP0
#define bank1 bsf STATUS, RP0

;---Saidas---
#define led1 PORTA,0 ;led1 ligado em RA0
```

```
;---Registadores de uso geral---
cblock H'20'
W_TEMP ; registrador para armazenar o conteúdo temporário de w
STATUS_TEMP ; registrador para armazenar o conteúdo temporário de STATUS
endc

;---Vetor de RESET
org H'0000'
goto inicio

;---Vetor de Interrupção---
org H'0004' ; as interrupções apontam para esse endereço de memória

;---Salva Contexto---
MOVWF W_TEMP ; copia o conteúdo de w para W_TEMP
SWAPF STATUS,W ; move o conteúdo de STATUS com os nibles invertidos para w
bank0 ; seleciona o banco zero de memória
MOVWF STATUS_TEMP ; copia o conteúdo de STATUS com os nibles invertidos para STATUS_TEMP
;--- Final de Salvamento de Contexto ---

;--- Trata ISR ---
btfss INTCON, T0IF ; ocorreu um overflow no TIMER0
goto exit_ISR ; NÃO, desvia para saída da interrupção
movlw D'10' ; move a literal 10d para w
bcf INTCON,T0IF ; SIM limpa a flag
comf PORTA ; complementando todo registrador PORTA

;----Recupera contexto (saída da interrupção)--
exit_ISR

SWAPF STATUS_TEMP,W ; copia em w o conteúdo de STATUS_TEMP com nibles invertidos.
MOVWF STATUS ; recuperando o conteúdo de STATUS
SWAPF W_TEMP,F ; W_TEMP = W_TEMP com nibles invertidos
SWAPF W_TEMP,W ; recupera conteúdo de w
retfie ; retorna da interrupção

inicio
    bank1 ; seleciona o banco 0 de memória
```

```
movlw H'80' ;w=80h (inicia o registrador em 0)
movwf OPTION_REG ; configurar o OPTION_REG (Interrupções)
;PULL ups internos desabilitados
;Timer0 incrementa com ciclo de máquina
;Prescaler 1:2 associado ao Timer0
movlw H'F3' ; w = F3h
movwf TRISA ; configura a saída dos Leds
bank0
; seleciona banco 0 de memória
movlw H'07' ; w=7h
movwf CMCON; CMCON = 7h desabilita os comparadores
movlw H'A0'; w=A0h (inicia o registrador em 0)
movwf INTCON ; configurar INTCON (Configura interrupções)
; habilita a interrupção global
; habilita a interrupção do TIMER0

movlw D'10'; move a literal 10 para w
movwf TMR0 ; inicializando timer0 com 10d (256-10 = 246)
bcf led1 ; Sim, desliga led1
bsf led1 ; acende led1
goto $ ; aguarda a interrupção

end
```

Na figura 01 é possível ver os comandos de loop infinito e ligar e desligar o led 1.

```
inicio
    bank1 ; seleciona o banco 0 de memória
        movlw H'80' ; w=80h (inicia o registrador em 0)
        movwf OPTION_REG ; configurar o OPTION_REG (Interrupções)
        ;PULL ups internos desabilitados
        ;Timer0 incrementa com ciclo de máquina
        ;Prescaler 1:2 associado ao Timer0
        movlw H'F3' ; w = F3h
        movwf TRISA ; configura a saída dos LEDS
    bank0
        ; seleciona banco 0 de memória
        movlw H'07' ; w=7h
        movwf CMCON; CMCON = 7h desabilita os comparadores
        movlw H'A0'; w=A0h (inicia o registrador em 0)
        movwf INTCON ; configurar INTCON (Configura interrupções)
        ; habilita a interrupção global
        ; habilita a interrupção do TIMERO

        movlw D'10'; move a literal 10 para w
        movwf TMRO ; inicializando timer0 com 10d (256-10 = 246)
        bcf led1 ; Sim, desliga led1
        bsf led1 ; acende led1
        goto $ ; aguarda a interrupção

end
```

Figura 01: Código fonte da atividade 01

Na figura 02, temos a execução da atividade 01, com o esquema montado com o PIC16F628A.

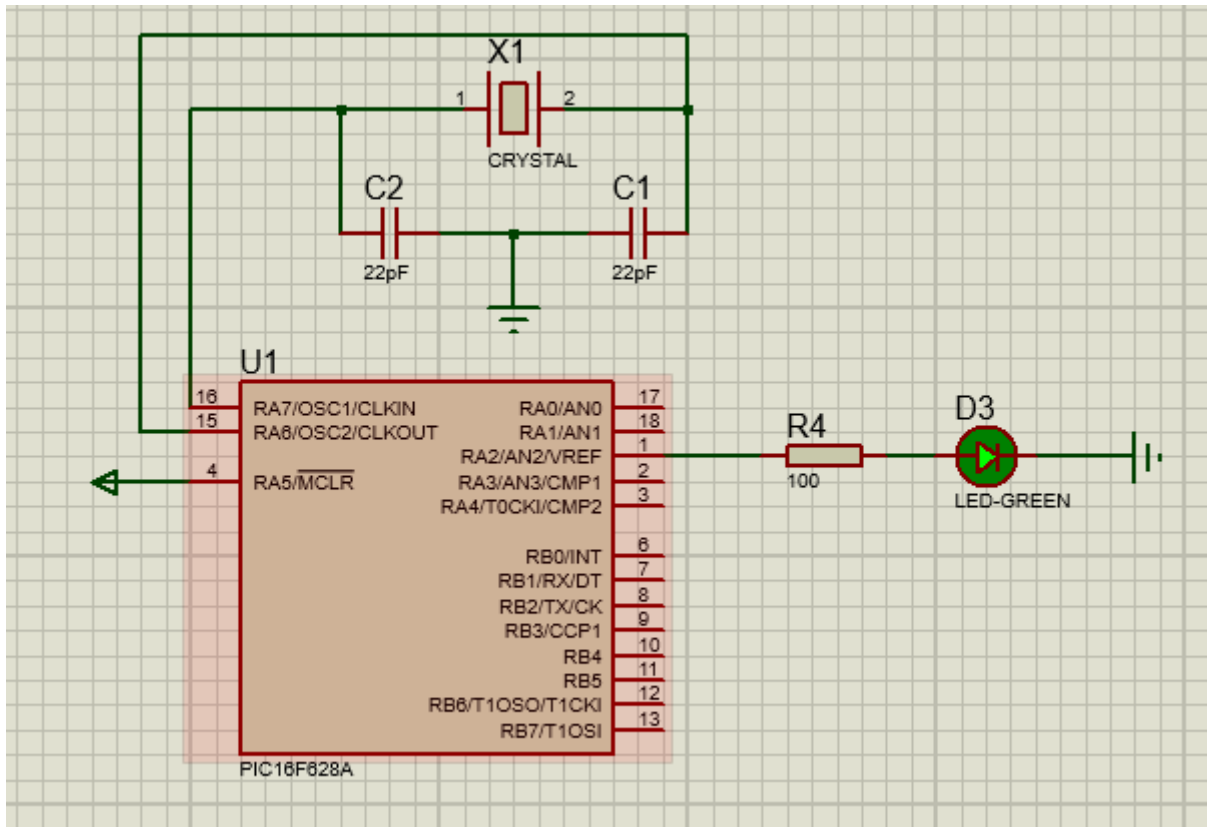


Figura 02: Execução da atividade 01

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. Apenas a atividade 01 ficou sem a implementação do botão do pedestre.

4 - Referências

José, D. S. **Desbravando o PIC - Ampliado e atualizado para PIC16F628A**. 7ª ed. Ed.Érica, 2003.