



**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 7**

**Marabá
2021**

Microprocessadores e Microcontroladores
Experimento 7

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 um programa e simule um semáforo de 3 tempos, acionado com botão para pedestre. Utilize temporizador.	4
2.2 - Inverter a lógica do LED, fazendo acender quando o botão está solto e apagar quando o botão for pressionado.	12
3 - Conclusão	17
4 - Referências	17

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 um semáforo de 3 tempos, acionado com botão para pedestre. Utilize temporizador.
- 2 - Inverta a lógica do LED, fazendo acender quando o botão está solto e apagar quando o botão for pressionado.

2 - Atividades

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

2.1 - Desenvolva um programa e simule um semáforo de 3 tempos, acionado com botão para pedestre. Utilize temporizador.

Código da atividade 01:

```
; Microcontroladores e Microprocessadores
; Aula 02
; Prof. Elton Alves
; Semaforo

;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 mnemonico para o banco 0 de
memoria
```

```
#define bank1 bsf STATUS, RP0 ; cria um mnemonico para o banco 1 de
memoria

;---Saidas---
#define led1 PORTA, RA0 ; led1 ligado em RA0
#define led2 PORTA, RA1 ;led2 ligado em RA1
#define led3 PORTA, RA2 ;led2 ligado em RA2

;---Vetor de RESET---
org H'000' ; origem no endereco 000h de memoria
goto inicio ; desvia do vetor de interrupcao

;---Vetor de Interrup??o---
org H'0004' ; todas as interrupcoes apontam para este endereco
retfie ; retorna a interrupcao

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

bank1 ; seleciona o banco 1 de mem?ria
    movlw H'A0' ; w=B'0000 0000
    movwf TRISA ; TRISA=H'A0' (todos bits sao saidas)
    movlw H'FF' ; w=B'1111 1111'
    movwf TRISB ;TRISB=H'FF' todos sao entradas

bank0 ; seleciona o banco 0 de mem?ria (padr?o RESET)
    movlw H'A0' ; w=B'0000 0000'
    movwf PORTA; (Leds iniciam desligados-escrevo no PORTB)

loop ; loop infinito
    call apaga_led3
    call acende_led1
    call delay500ms
    call apaga_led1
    call acende_led2
    call delay500ms
    call apaga_led2
    call acende_led3
    call delay500ms

    goto loop ;volta para o label loop
```

```
;---Desenvolvimento das Sub-rotinas---
acende_led1
    bsf led1
    return
apaga_led1
    bcf led1; apaga led1
    return ;retorna da sub-rotina
acende_led2
    bsf led2
    return
apaga_led2
    bcf led2; apaga led2
    return ;retorna da sub-rotina
acende_led3
    bsf led3
    return
apaga_led3
    bcf led3; apaga led3
    return ;retorna da sub-rotina
delay500ms
    movlw D'200' ; move o valor para W (constante)
    movwf H'20' ; inicializa da variavel tempo0 (posi de memoria
do registrador de uso geral)
aux1
    movlw D'250'
    movwf H'21'
aux2 ; gastar 1 ciclo de mquina (aproximar mais o tempo de 500ms)
    nop
    nop
    nop
    nop
    nop
    nop
    nop
    nop
    decfsz H'20' ; decrementa o tempo1 at que seja igual a
0(decremente em uma unidade e verifica se 0)
    goto aux2 ; vai para label aux2
    ; 250 x 10 ciclos de mquina = 2500
    decfsz H'32'; decrementa o tempo0 at seja igual a 0
    goto aux1; vai para label aux1
    ; 3 ciclos de mquina
    ; 2500x200 = 500000
```

```
return  
end
```

Nessa atividade foi implementado apenas a parte dos leds funcionando como semáforo sem o botão do pedestre.

Na figura 01 na linha 01 à 31 temos as configurações iniciais de memória, saídas e vetores de interrupção e reset.

```
1 ; Microcontroladores e Microprocessadores  
2 ; Aula 02  
3 ; Prof. Elton Alves  
4 ; Semáforo  
5  
6 ;list p=16f628a ; microcontrolador utilizado  
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 mnemonico para o banco 0 de memoria  
18 #define bank1 bsf STATUS, RPO ; cria um mnemonico para o banco 1 de memoria  
19  
20 ;---Saídas---  
21 #define led1 PORTA, RA0 ; led1 ligado em RA0  
22 #define led2 PORTA, RA1 ; led2 ligado em RA1  
23 #define led3 PORTA, RA2 ; led2 ligado em RA2  
24  
25 ;---Vetor de RESET---  
26 org H'000' ; origem no endereço 000h de memoria  
27 goto inicio ; desvia do vetor de interrupcao  
28  
29 ;---Vetor de Interrupção---  
30 org H'0004' ; todas as interrupcoes apontam para este endereço  
31 retfie ; retorna a interrupcao
```

Figura 01: Código fonte da atividade 01 - Parte 01

Na figura 02 na linha 34 até 59 temos o início do programa com a seleção dos bancos de memória e início do loop infinito onde serão chamadas as subrotinas para funcionamento do semáforo.

```
31      retfie ; retorna a interrupcao
32
33      ;---Programa Principal---
34      inicio
35          CLRF PORTA ; Limpa PORTA
36          CLRF PORTB ; Limpa PORTB
37
38      bank1 ; seleciona o banco 1 de mem?ria
39          movlw H'AO' ; w=B'0000 0000
40          movwf TRISA ; TRISA=H'AO' (todos bits sao saidas)
41          movlw H'FF' ; w=B'1111 1111'
42          movwf TRISB ;TRISB=H'FF' todos sao entradas
43
44      bank0 ; seleciona o banco 0 de mem?ria (padr?o RESET)
45          movlw H'AO' ; w=B'0000 0000'
46          movwf PORTA; (Leds iniciam desligados-escrevo no PORTB)
47
48      loop ; loop infinito
49          call apaga_led3
50          call acende_led1
51          call delay500ms
52          call apaga_led1
53          call acende_led2
54          call delay500ms
55          call apaga_led2
56          call acende_led3
57          call delay500ms
58
59          goto loop ;volta para o label loop
60
```

Figura 02: Código fonte da atividade 01 - Parte 02

Na figura 03 temos as implementações das subrotinas.

```
61 ;---Desenvolvimento das Sub-rotinas---
62 acende_led1
63     bsf led1
64     return
65 apaga_led1
66     bcf led1; apaga led1
67     return ;retorna da sub-rotina
68 acende_led2
69     bsf led2
70     return
71 apaga_led2
72     bcf led2; apaga led2
73     return ;retorna da sub-rotina
74 acende_led3
75     bsf led3
76     return
77 apaga_led3
78     bcf led3; apaga led3
79     return ;retorna da sub-rotina
80 delay500ms
81     movlw D'200' ; move o valor para W (constante)
82     movwf H'20' ; inicializa o valor da variavel tempo0 (posição de
83 aux1
84     movlw D'250'
85     movwf H'21'
86 aux2 ; gastar 1 ciclo de máquina (aproximar mais o tempo de 500ms)
87     nop
88     nop
89     nop
90     nop
91     nop
```

Figura 03: Loop infinito e sub rotinas do código da atividade 01.

Na figura 03, temos a execução da atividade 01, com o esquema montado com o PIC16F628A.

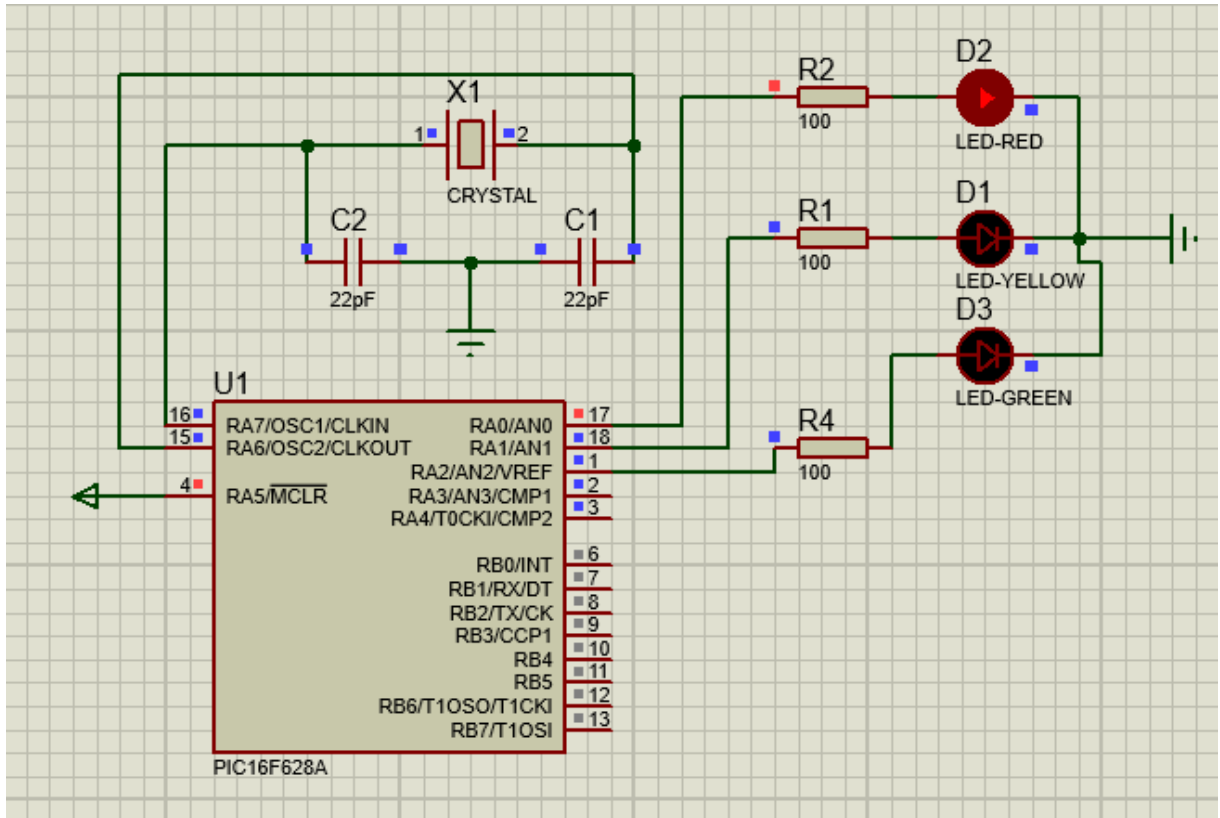


Figura 04: Execução da atividade 01 - Parte 01

Na figura 04, temos a execução da atividade 01, com o esquema montado com o PIC16F628A e os três leds conectados nas saídas, representando o semáforo.

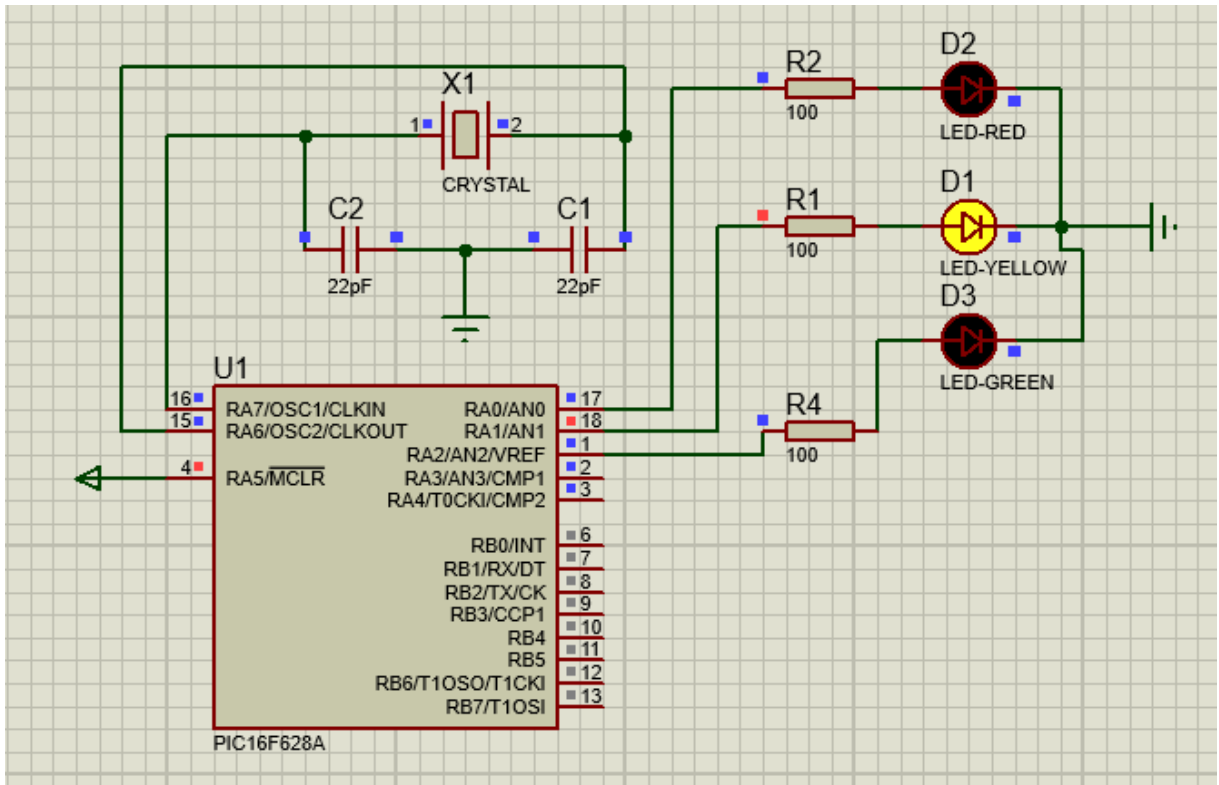


Figura 05: Execução da atividade 01 - Parte 02

2.2 - Inverter a lógica do LED, fazendo acender quando o botão está solto e apagar quando o botão for pressionado.

Código da atividade 02:

```
; Microcontroladores e Microprocessadores
; Aula 02
; Prof. Elton Alves
; Led invertido
; Iago Costa

;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 ; botao 1 ligado em RB0
#define botao2 PORTB, RB1 ; botao 2 ligado em RB1
#define botao3 PORTB, RB2 ; botao 3 ligado em RB2

;---Saídas---
#define led1 PORTA, RA0 ;led1 ligado em RA0
#define led2 PORTA, RA1 ;led2 ligado em RA1
#define led3 PORTA, RA2 ;led2 ligado em RA2

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

```
;---Vetor de Interrupcao---
org H'0004' ; todas as interrupcoes apontam para este endereco
retfie ; retorna a interrupcao

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

bank1 ; seleciona o banco 1 de mem?ria
    movlw H'A0' ; w=B'0000 0000
    movwf TRISA ; TRISA=H'A0' (todos bits sao saidas)
    movlw H'FF' ; w=B'1111 1111'
    movwf TRISB ;TRISB=H'FF' todos sao entradas

bank0 ; seleciona o banco 0 de memoria (padrao RESET)
    movlw H'A0' ; w=B'0000 0000'
    movwf PORTA; (Leds iniciam desligados-escrevo no PORTB)

    ; goto $ ; segura o c?digo - $ indica a posicial atual do c?digo
(loop infinito)

loop ; loop infinito

    call trata_but1 ; chama sub-rotina-trata_but1
    call trata_but2 ; chama sub-rotina_trata_but2
    call trata_but3 ; chama sub-rotina_trata_but3

    goto loop ;volta para o label loop

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

trata_but1 ; sub-rotina para tratar bot?o 1
    btfsc botao1 ; botao foi pressionado?
    goto liga_led1; Nao pressiona, desvia para label apaga_led1
    bcf led1; Sim, desliga led1
    return ; retorno da subrotina

liga_led1
    bsf led1; apaga led1
    return ;retorna da sub-rotina
```

```
trata_but2 ; sub-rotina para tratar bot?o 2
    btfsc botao2 ; botao foi pressionado?
    goto liga_led2; Nao pressiona, desvia para label apaga_led2
    bcf led2; Sim, desliga led2
    return ; retorno da subrotina

liga_led2
    bsf led2; apaga led2
    return ;retorna da sub-rotina

trata_but3 ; sub-rotina para tratar botao 3
    btfsc botao3 ; botao foi pressionado?
    goto liga_led3; Nao pressiona, desvia para label apaga_led2
    bcf led3; Sim, desliga led3
    return ; retorno da subrotina

liga_led3
    bsf led3; apaga led3
    return ;retorna da sub-rotina

end
```

Na figura 06 temos a configuração inicial do código da atividade 02.

```
8      #include <P16f628a.inc> ; inclui o arquivo do 16f628a (registradores)
9      ;---FUSE bits---
10     __config _XT_OSC & _WDT_OFF & _PWRTE_ON & _CP_OFF
11     ;---Pagina??o de Mem?ria
12     #define bank0 bcf STATUS, RPO ;cria um mnemonico para o banco 0 de memoria
13     #define bank1 bsf STATUS, RPO ; cria um mnemonico para o banco 1 de memoria
14     ;---Entradas---
15     #define botao1 PORTB, RB0 ; botao 1 ligado em RB0
16     #define botao2 PORTB, RB1 ; botao 2 ligado em RB1
17     #define botao3 PORTB, RB2 ; botao 3 ligado em RB2
18     ;---Sa?das---
19     #define led1 PORTA, RA0 ;led1 ligado em RA0
20     #define led2 PORTA, RA1 ;led2 ligado em RA1
21     #define led3 PORTA, RA2 ;led2 ligado em RA2
22     ;---Vetor de RESET---
23     org H'000' ; origem no endereco 000h de memoria
24     goto inicio ; desvia do vetor de interrupcao
25     ;---Vetor de Interrupcao---
26     org H'0004' ; todas as interrupcoes apontam para este endereco
27     retfie ; retorna a interrupcao
28     ;---Programa Principal---
29     inicio
30         CLRF PORTA ; Limpa PORTA
31         CLRF PORTB ; Limpa PORTB
32     bank1 ; seleciona o banco 1 de mem?ria
33         movlw H'A0' ; w=B'0000 0000
34         movwf TRISA ; TRISA=H'A0' (todos bits sao saidas)
35         movlw H'FF' ; w=B'1111 1111'
36         movwf TRISB ;TRISB=H'FF' todos sao entradas
37     bank0 ; seleciona o banco 0 de memoria (padrao RESET)
38         movlw H'A0' ; w=B'0000 0000'
39         movwf PORTA; (Leds iniciam desligados-escrevo no PORTB)
```

Figura 06: Configuração do código do led invertido.

Na figura 07 temos o loop infinito com a chamada das sub rotinas que fazem a lógica do led invertido funcionar.

```
40      loop ; loop infinito
41      call trata_but1 ; chama sub-rotina-trata_but1
42      call trata_but2 ; chama sub-rotina_trata_but2
43      call trata_but3 ; chama sub-rotina_trata_but3
44      goto loop ;volta para o label loop
45      ;---Desenvolvimento das Sub-rotinas----
46      trata_but1 ; sub-rotina para tratar bot?o 1
47      btfsc botao1 ; botao foi pressionado?
48      goto liga_led1; Nao pressiona, desvia para label apaga_led1
49      bcf led1; Sim, desliga led1
50      return ; retorno da subrotina
51      liga_led1
52      bsf led1; apaga led1
53      return ;retorna da sub-rotina
54      trata_but2 ; sub-rotina para tratar bot?o 2
55      btfsc botao2 ; botao foi pressionado?
56      goto liga_led2; Nao pressiona, desvia para label apaga_led2
57      bcf led2; Sim, desliga led2
58      return ; retorno da subrotina
59      liga_led2
60      bsf led2; apaga led2
61      return ;retorna da sub-rotina
62      trata_but3 ; sub-rotina para tratar botao 3
63      btfsc botao3 ; botao foi pressionado?
64      goto liga_led3; Nao pressiona, desvia para label apaga_led2
65      bcf led3; Sim, desliga led3
66      return ; retorno da subrotina
67      liga_led3
68      bsf led3; apaga led3
69      return ;retorna da sub-rotina
70
71      end
```

Figura 07: Configuração do loop e sub rotinas de execução.

Na figura 08 temos a demonstração do atividade 02 onde é solicitado implementar o led invertido.

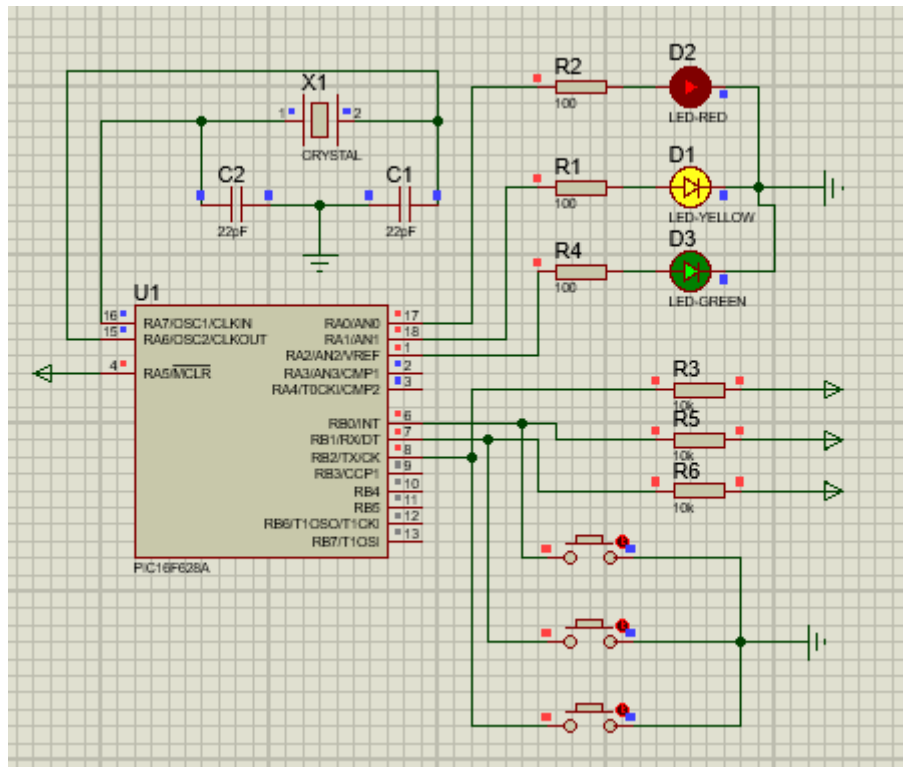


Figura 08: Demonstração da simulação do sistema do led invertido.

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**. 7ªed. Ed.Erica, 2003.