

ANO
2023



UNINTER

CADERNO DE RESPOSTAS DA
ATIVIDADE PRÁTICA DE:

MICROPROCESSADORES E
MICROCONTROLADORES

ALUNO: **GILBERTO OLIVEIRA JUNIOR**
RU 3326662

Prof. Winston Sen Lun Fung, Esp.

PRÁTICA 01 - PROGRAMANDO O PIC16F877A

A) PISCAR OS LED D6, LED D9 E LED D11

ENUNCIADO: Neste item, deve-se fazer os leds LED D6, LED D9 e LED D11 piscar em sequência.

I. Apresentação do Código (não esquecer do identificador pessoal):

```

;***** ATIVIDADE PRATICA I *****
;***** PRATICA 1 - PISCA LED D6, LED D9 e LED D11 *****
;***** Descricao:
; Esse codigo demonstra como piscar os LED D6, LED D9 e LED D11 em sequencia
;
; Configuracao do teste:
; MCU:          PIC16F877A
;                 http://www.microchip.com/wwwproducts/en/en010210
;
; Ext. Modulo:   SIMULADOR KIT SENAI PIC 16 (Versao 3.3)
;
; SW:           MPLAB IDE v7.00
;                 http://www.microchip.com/mplab/
;
; Autor:
; ***** RU 3326662 GILBERTO OLIVEIRA JUNIOR
; ***** CURSO: ENGENHARIA DA COMPUTACAO
; ***** ESCOLA SUPERIOR POLITECNICA UNINTER - BRAZIL
; ***** Date : feb 2023
;***** Definição do processador *****

#include p16F877a.inc
_config _HS_OSC & _WDT_OFF & _LVP_OFF & _PWRTE_ON

;***** Memória de programa *****
ORG 0 ;Posiciona o ponteiro para o endereço 0x0 de memória

RESET ;Bloco de RESET do microcontrolador
NOP
GOTC SETUP ;Pula para o marcador SETUP
;***** Interrupção *****
ORG 4

;***** Declaração de variáveis *****
DELAY EQU 0x20
VEZES EQU 0x21

```

```

;***** Inicio do programa *****

;***** Configurações *****

SETUP           ;Bloco de configuração do microcontrolador
    BSF STATUS,RPO ;Aiona o Bank1 de memória
    MOVlw b'00000000' ;Carrega '00000000' no registrador W
    MOVwf TRISA      ;Configura PORTA como saída
    MOVlw b'00000000' ;Carrega '00000000' no registrador W
    MOVwf TRISD      ;Configura o PORTD como saída
    BCF STATUS,RPO   ;Retorna para o Bank0 de memória
    BSF PORTA,5       ;Coloca 5V, nível lógico 1 no pino RA5 - Enable dos LEDs
;***** Programa principal *****

LOOP            ;Bloco do principal do programa
    MOVlw b'00000001' ;Carrega '00000001' no registrador W; acende LED D6
    MOVwf PORTD      ;Envia o conteúdo de W para o registrador PORTD colocando
                      ;nível logico alto (+5V) no pino 19 (RD0)do PORTD
    CALL PERDE_TEMPO ;Chama o procedimento PERDE_TEMPO; acende o LED D6
    MOVlw b'00000000' ;Carrega '00000000' ou 0x00 no registrador W
    MOVwf PORTD      ;Envia o conteúdo de W para o registrador PORTD colocando
                      ;nível logico baixo (0V) em todos os pinos do PORTD (RD7 -> RD0)
    CALL PERDE_TEMPO ;Chama o procedimento PERDE_TEMPO para poder visualizar os
                      ;leds apagados

    MOVlw b'00001000' ;Carrega '00001000' no registrador W; acende LED D9
    MOVwf PORTD      ;Envia o conteúdo de W para o registrador PORTD colocando
                      ;nível logico alto (+5V) no pino 22 (RD3)do PORTD
    CALL PERDE_TEMPO ;Chama o procedimento PERDE_TEMPO para poder visualizar os
                      ;leds acesos

    MOVlw b'00000000' ;Carrega '00000000' ou 0x00 no registrador W
    MOVwf PORTD      ;Envia o conteúdo de W para o registrador PORTD colocando
                      ;nível logico baixo (0V) em todos os pinos do PORTD (RD7 -> RD0)
    CALL PERDE_TEMPO ;Chama o procedimento PERDE_TEMPO para poder visualizar os
                      ;leds apagados

    MOVlw b'00100000' ;Carrega '00100000' no registrador W; acende LED D11
    MOVwf PORTD      ;Envia o conteúdo de W para o registrador PORTD colocando
                      ;nível logico alto (+5V) no pino 28 (RD5)do PORTD
    CALL PERDE_TEMPO ;Chama o procedimento PERDE_TEMPO para poder visualizar os
                      ;leds acesos

```

```

MOVlw b'00000000'      ;Carrega '00000000' ou 0x00 no registrador W
MOVwf PORTD             ;Envia o conteúdo de W para o registrador PORTD colocando
                         ;nível logico baixo (0V) em todos os pinos do PORTD (RD7 -> RD0)
CALL PERDE_TEMPO        ;Chama o procedimento PERDE_TEMPO para poder visualizar os
                         ;leds apagados

GOTC LOOP               ;Manda o programa pular para o inicio do bloco LOOP
                         ;mantendo o programa em execução eterna

PERDE_TEMPO             ;Procedimento que força o PIC a perder tempo(delay)
  MOVlw d'80'             ;Para gravar no PIC utilizar 80, no simulador SENAI use 1.
  MOVwf VEZES

LOOP_VEZES              ;Para gravar no PIC utilizar 255, no simulador SENAI use 1.
  MOVlw d'255'            ;Para gravar no PIC utilizar 255, no simulador SENAI use 1.
  MOVwf DELAY
  CALL DELAY_US
  DECFSZ VEZES, 5
  GOTC LOOP_VEZES
  RETURN

DELAY_US                ;NOP
  NOP
  NOP
  DECFSZ DELAY, 1
  GOTC DELAY_US
  RETURN

END                     ;Marcador de última linha do programa

```

II.

III. Apresentação das Imagens/Fotos (não esquecer do identificador):

a. Imagem do simulador com o LED aceso

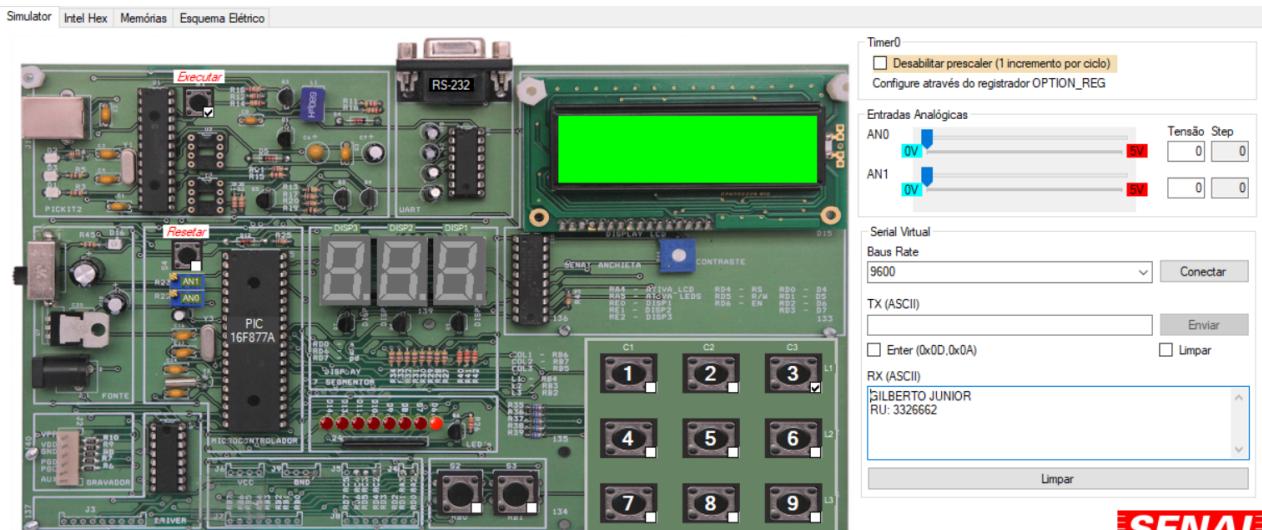


Figura 1 - a: Código rodando - Fazendo LED D6 piscar, em seguida...

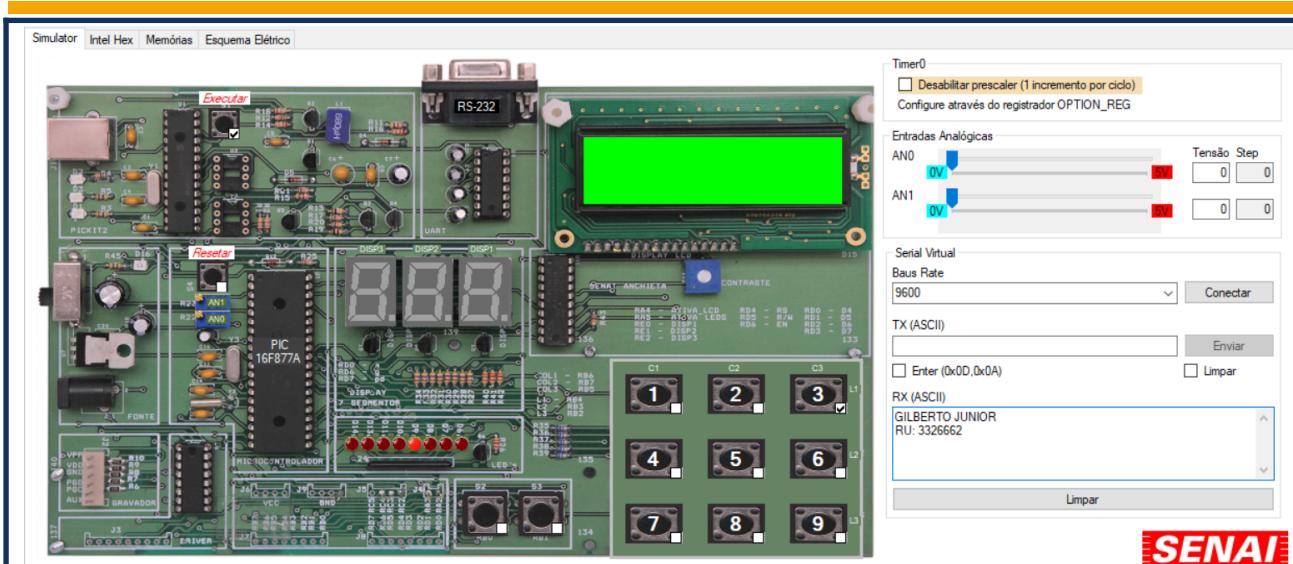


Figura 1 - b: Código rodando - Fazendo LED D9 piscar, para finalmente...

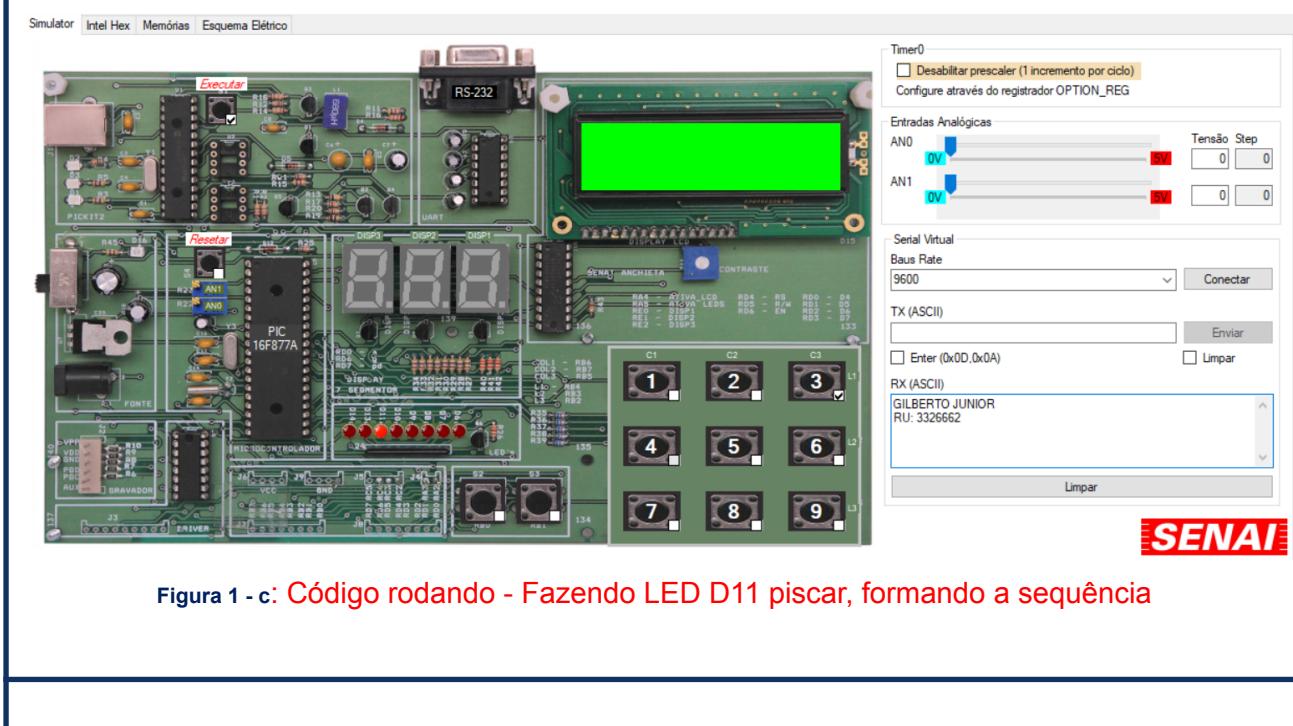


Figura 1 - c: Código rodando - Fazendo LED D11 piscar, formando a sequência

PRÁTICA 01 - PROGRAMANDO O PIC16F877A

B) ACENDER O DISPLAY DE 7 SEGMENTOS

ENUNCIADO: Nesse item deve-se acender o último número do seu RU no display de 7 segmentos.

O display está ligado ao PORT D do PIC.

1. Habilite o display de 7 segmentos desejado.
2. Configure a saída no PORTD para acender o número desejado.

I. Apresentação do Código (não esquecer do identificador pessoal):

```

;***** ATIVIDADE PRATICA II *****
;***** PRATICA 2 - ACENDER O DISPLAY DE 7 SEGMENTOS *****
;***** Descrição: *****
; Esse código demonstra como acender o primeiro dígito do
; Display 7-Segmentos para formar o número 2, correspondente ao
; último dígito do meu RU

;

; Configuração do teste:
; MCU:          PIC16F877A
;              http://www.microchip.com/wwwproducts/en/en010210

; Ext. Modulo:   SIMULADOR KIT SENAI PIC 16 (Versão 3.3)
;

; SW:           MPLAB IDE v7.00
;              http://www.microchip.com/mplab/
; Autor:
; ***** RU 3326662 GILBERTO OLIVEIRA JUNIOR
; ***** CURSO: ENGENHARIA DA COMPUTAÇÃO
; ***** ESCOLA SUPERIOR POLITÉCNICA UNINTER - BRAZIL
; ***** Date : feb 2023
;***** Definição do processador *****

#include p16F877a.inc
_config _HS_OSC & _WDT_OFF & _LVP_OFF & _PWRTE_ON

;***** Memória de programa *****
ORG 0           ;Posiciona o ponteiro para o endereço 0x0 de memória

RESET           ;Bloco de RESET do microcontrolador
NOP
GOTOC SETUP    ;Pula para o marcador SETUP
;***** Interrupção *****
ORG 4

;***** Declaração de variáveis *****
DELAY EQU 0x20
VEZES EQU 0x21

;***** Início do programa *****
|

```

```

;***** Inicio do programa *****

;***** Configurações *****
SETUP           ;Bloco de configuração do microcontrolador
    BSF STATUS,RP0 ;Aiona o Bank1 de memória
    MOVWF b'00000000' ;Carrega '00000000' no registrador W
    MOVWF TRISD ;Configura PORTD como saída - Display de 7 Segmentos
    MOVWF b'00000000' ;Carrega '00000000' no registrador W
    MOVWF TRISE ;Configura o PORTE como saída
    BCF STATUS,RP0 ;Retorna para o Bank0 de memória
    BSF PORTE,0 ;Coloca 5V, nível lógico 1 no pino RE0
                ;Enable do ultimo Display de 7 Segmentos

;***** Programa principal *****

LOOP            ;Bloco do principal do programa
    MOVWF b'01011011' ;Carrega '01011011' no registrador W para formar o numero 2 no 7SEGs
    MOVWF PORTD ;Envia o conteúdo de W para o registrador PORTD colocando
                ;nível logico alto (+5V) os respectivos pinos do PORTD (RD7 -> RD0)
    GOTO LOOP        ;Manda o programa pular para o inicio do bloco LOOP
                ;mantendo o programa em execução eterna
END             ;Marcador de última linha do programa

```

Códigos elaborados fora do MPLAB IDE terão sua nota descontada em 50%

I. Apresentação das Imagens (não esquecer do identificador):

a. Imagem do simulador com o número **aceso** no display de 7 segmentos.

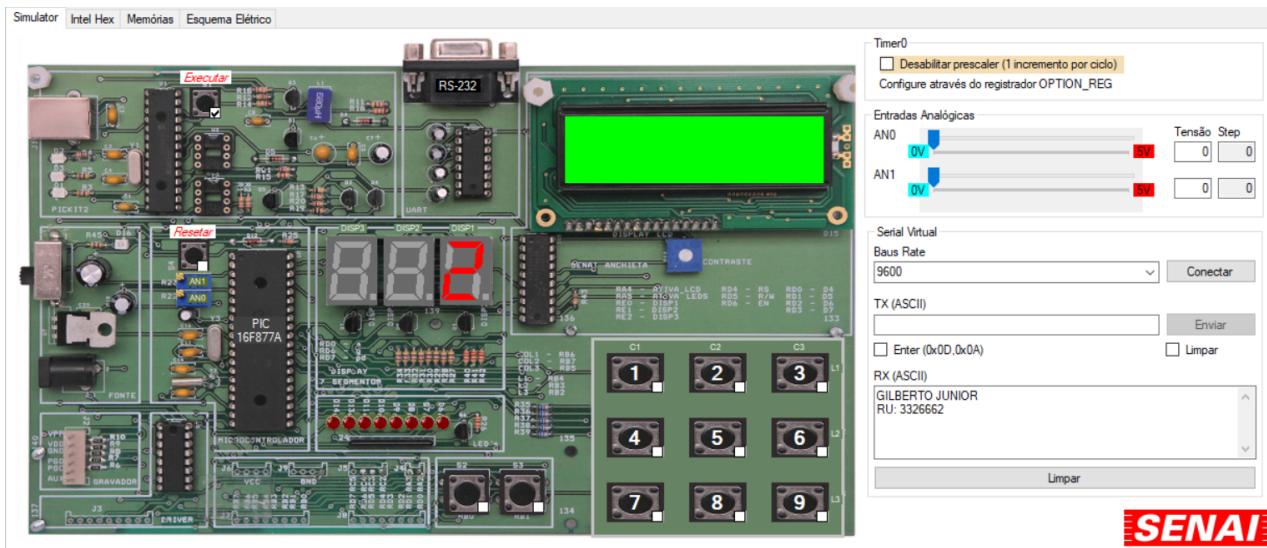
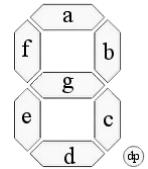


Figura 2: Display 7-Segmentos aceso no número 2, correspondente ao último dígito do meu RU

- Preencha a tabela abaixo verificando qual os segmentos devem acender (1) ou ficar apagado (0).

Pino PIC	D0	D1	D2	D3	D4	D5	D6
Seg. Display	a	b	c	d	e	f	g
0 / 1	1	1	0	1	1	0	1



PRÁTICA 01 - PROGRAMANDO O PIC16F877A

C) CONTADOR DE 0 A F NO DISPLAY DE 7 SEGMENTOS

ENUNCIADO: Nesse item deve-se fazer um contador de 0 até F utilizando o display de 7 segmentos.

O display está ligado ao PORT D do PIC.

1. Ligue o display conforme orientado na atividade B.
2. Utilize o procedimento PERDE_TEMPO indicada na atividade A.

I. Apresentação do Código (não esquecer do identificador pessoal):

```

;***** ATIVIDADE PRATICA III *****
;***** PRATICA 3 - CONTADOR A ATÉ F NO DISPLAY DE 7 SEGMENTOS *****
;***** Descrição:
; Esse código demonstra como contar de A - F no primeiro dígito do
; Display 7-Segmentos, sequencialmente.
; Configuração do teste:
;     MCU:          PIC16F877A
;                 http://www.microchip.com/wwwproducts/en/en010210

;     Ext. Módulo:  SIMULADOR KIT SENAI PIC 16 (Versão 3.3)
;

;     SW:           MPLAB IDE v7.00
;                 http://www.microchip.com/mplab/
; Autor:
; ***** RU 33266662 GILBERTO OLIVEIRA JUNIOR
; ***** CURSO: ENGENHARIA DA COMPUTAÇÃO
; ***** ESCOLA SUPERIOR POLITÉCNICA UNINTER - BRAZIL
; ***** Data : feb 2023
;***** Definição do processador *****

#include p16F877a.inc
__config _HS_OSC & _WDT_OFF & _LVP_OFF & _PWRTE_ON

;***** Memória de programa *****
ORG 0           ;Posiciona o ponteiro para o endereço 0x0 de memória

RESET           ;Bloco de RESET do microcontrolador
NOP
GOTO  SETUP      ;Pula para o marcador SETUP
;***** Interrupção *****
ORG 4

;***** Declaração de variáveis *****
DELAY EQU 0x20
VEZES EQU 0x21

```

```

;***** Inicio do programa *****

;***** Configurações *****
SETUP           ;Bloco de configuração do microcontrolador
    BSF STATUS,RPO ;Aiona o Bank1 de memória
    MOVLW b'00000000' ;Carrega '00000000' no registrador W
    MOVWF TRISD      ;Configura PORTD como saída - Display de 7 Segmentos
    MOVLW b'00000000' ;Carrega '00000000' no registrador W
    MOVWF TRISE      ;Configura o PORTE como saída
    BCF STATUS,RPO   ;Retorna para o Bank0 de memória
    BSF PORTE,0       ;Coloca 5V, nível lógico 1 no pino RE0
                      ;Enable do ultimo Display de 7 Segmentos

;***** Programa principal *****

LOOP            ;Bloco do principal do programa
    MOVLW b'00111111' ;Carrega '00111111' no registrador W para formar o numero 0 no 7SEGs
    MOVWF PORTD        ;Envia o conteúdo de W para o registrador PORTD colocando
                      ;nível logico alto (+5V) os respectivos pinos do PORTD (RD7 -> RD0)
    CALL PERDE_TEMPO ;Chama o procedimento PERDE_TEMPO para poder visualizar os
                      ;leds apagados
    MOVLW b'00000110' ;Carrega '01011011' no registrador W para formar o numero 1 no 7SEGs
    MOVWF PORTD        ;Envia o conteúdo de W para o registrador PORTD colocando
                      ;nível logico alto (+5V) os respectivos pinos do PORTD (RD7 -> RD0)
    CALL PERDE_TEMPO ;Chama o procedimento PERDE_TEMPO para poder visualizar os
                      ;leds apagados
    MOVLW b'01011011' ;Carrega '01011011' no registrador W para formar o numero 2 no 7SEGs
    MOVWF PORTD        ;Envia o conteúdo de W para o registrador PORTD colocando
                      ;nível logico alto (+5V) os respectivos pinos do PORTD (RD7 -> RD0)
    CALL PERDE_TEMPO ;Chama o procedimento PERDE_TEMPO para poder visualizar os
                      ;leds apagados
    MOVLW b'01001111' ;Carrega '01011011' no registrador W para formar o numero 3 no 7SEGs
    MOVWF PORTD        ;Envia o conteúdo de W para o registrador PORTD colocando
                      ;nível logico alto (+5V) os respectivos pinos do PORTD (RD7 -> RD0)
    CALL PERDE_TEMPO ;Chama o procedimento PERDE_TEMPO para poder visualizar os
                      ;leds apagados
    MOVLW b'01100110' ;Carrega '01100110' no registrador W para formar o numero 4 no 7SEGs
    MOVWF PORTD        ;Envia o conteúdo de W para o registrador PORTD colocando
                      ;nível logico alto (+5V) os respectivos pinos do PORTD (RD7 -> RD0)
    CALL PERDE_TEMPO ;Chama o procedimento PERDE_TEMPO para poder visualizar os
                      ;leds apagados

```

```

MOVlw b'01101101' ;Carrega '01101101' no registrador W para formar o numero 5 no 7SEGs
MOVwf PORTD ;Envia o conteúdo de W para o registrador PORTD colocando
;nível logico alto (+5V) os respectivos pinos do PORTD (RD7 -> RD0)
CALL PERDE_TEMPO ;Chama o procedimento PERDE_TEMPO para poder visualizar os
;leds apagados

MOVlw b'01111101' ;Carrega '01111101' no registrador W para formar o numero 6 no 7SEGs
MOVwf PORTD ;Envia o conteúdo de W para o registrador PORTD colocando
;nível logico alto (+5V) os respectivos pinos do PORTD (RD7 -> RD0)
CALL PERDE_TEMPO ;Chama o procedimento PERDE_TEMPO para poder visualizar os
;leds apagados

MOVlw b'00000111' ;Carrega '00000111' no registrador W para formar o numero 7 no 7SEGs
MOVwf PORTD ;Envia o conteúdo de W para o registrador PORTD colocando
;nível logico alto (+5V) os respectivos pinos do PORTD (RD7 -> RD0)
CALL PERDE_TEMPO ;Chama o procedimento PERDE_TEMPO para poder visualizar os
;leds apagados

MOVlw b'01111111' ;Carrega '01111111' no registrador W para formar o numero 8 no 7SEGs
MOVwf PORTD ;Envia o conteúdo de W para o registrador PORTD colocando
;nível logico alto (+5V) os respectivos pinos do PORTD (RD7 -> RD0)
CALL PERDE_TEMPO ;Chama o procedimento PERDE_TEMPO para poder visualizar os
;leds apagados

MOVlw b'01101111' ;Carrega '01101111' no registrador W para formar o numero 9 no 7SEGs
MOVwf PORTD ;Envia o conteúdo de W para o registrador PORTD colocando
;nível logico alto (+5V) os respectivos pinos do PORTD (RD7 -> RD0)
CALL PERDE_TEMPO ;Chama o procedimento PERDE_TEMPO para poder visualizar os
;leds apagados

MOVlw b'01110111' ;Carrega '01110111' no registrador W para formar o numero A no 7SEGs
MOVwf PORTD ;Envia o conteúdo de W para o registrador PORTD colocando
;nível logico alto (+5V) os respectivos pinos do PORTD (RD7 -> RD0)
CALL PERDE_TEMPO ;Chama o procedimento PERDE_TEMPO para poder visualizar os
;leds apagados

MOVlw b'01111100' ;Carrega '01111100' no registrador W para formar o numero B no 7SEGs
MOVwf PORTD ;Envia o conteúdo de W para o registrador PORTD colocando
;nível logico alto (+5V) os respectivos pinos do PORTD (RD7 -> RD0)
CALL PERDE_TEMPO ;Chama o procedimento PERDE_TEMPO para poder visualizar os
;leds apagados

MOVlw b'00111001' ;Carrega '00111001' no registrador W para formar o numero C no 7SEGs
MOVwf PORTD ;Envia o conteúdo de W para o registrador PORTD colocando
;nível logico alto (+5V) os respectivos pinos do PORTD (RD7 -> RD0)
CALL PERDE_TEMPO ;Chama o procedimento PERDE_TEMPO para poder visualizar os
;.
.
```

```

MOVlw b'01011110' ;Carrega '01011110' no registrador W para formar o numero D no 7SEGs
MOVwf PORTD ;Envia o conteúdo de W para o registrador PORTD colocando
;nível logico alto (+5V) os respectivos pinos do PORTD (RD7 -> RD0)
CALL PERDE_TEMPO ;Chama o procedimento PERDE_TEMPO para poder visualizar os
;leds apagados
MOVlw b'01111001' ;Carrega '01111001' no registrador W para formar o numero E no 7SEGs
MOVwf PORTD ;Envia o conteúdo de W para o registrador PORTD colocando
;nível logico alto (+5V) os respectivos pinos do PORTD (RD7 -> RD0)
CALL PERDE_TEMPO ;Chama o procedimento PERDE_TEMPO para poder visualizar os
;leds apagados
MOVlw b'01110001' ;Carrega '01110001' no registrador W para formar o numero F no 7SEGs
MOVwf PORTD ;Envia o conteúdo de W para o registrador PORTD colocando
;nível logico alto (+5V) os respectivos pinos do PORTD (RD7 -> RD0)
CALL PERDE_TEMPO ;Chama o procedimento PERDE_TEMPO para poder visualizar os
;leds apagados
GOTC LOOP ;Manda o programa pular para o inicio do bloco LOOP
;mantendo o programa em execução eterna

;***** Rotina Delay *****
PERDE_TEMPO ;Procedimento que força o PIC a perder tempo(delay)
    MOVlw d'10' ;Para gravar no PIC utilizar 80, no simulador SENAI use 1.
    MOVwf VEZES

LOOP_VEZES
    MOVlw d'10' ;Para gravar no PIC utilizar 255, no simulador SENAI use 1.
    MOVwf DELAY
    CALL DELAY_US
    DECFSZ VEZES,5
    GOTC LOOP_VEZES
    RETURN

DELAY_US
    NOP
    NOP
    DECFSZ DELAY,1
    GOTC DELAY_US
    RETURN

END ;Marcador de última linha do programa

```

Códigos elaborados fora do MPLAB IDE terão sua nota descontada em 50%

II. Apresentação das Imagens (não esquecer do identificador):

- Imagen do simulador com o número 0 **aceso**.

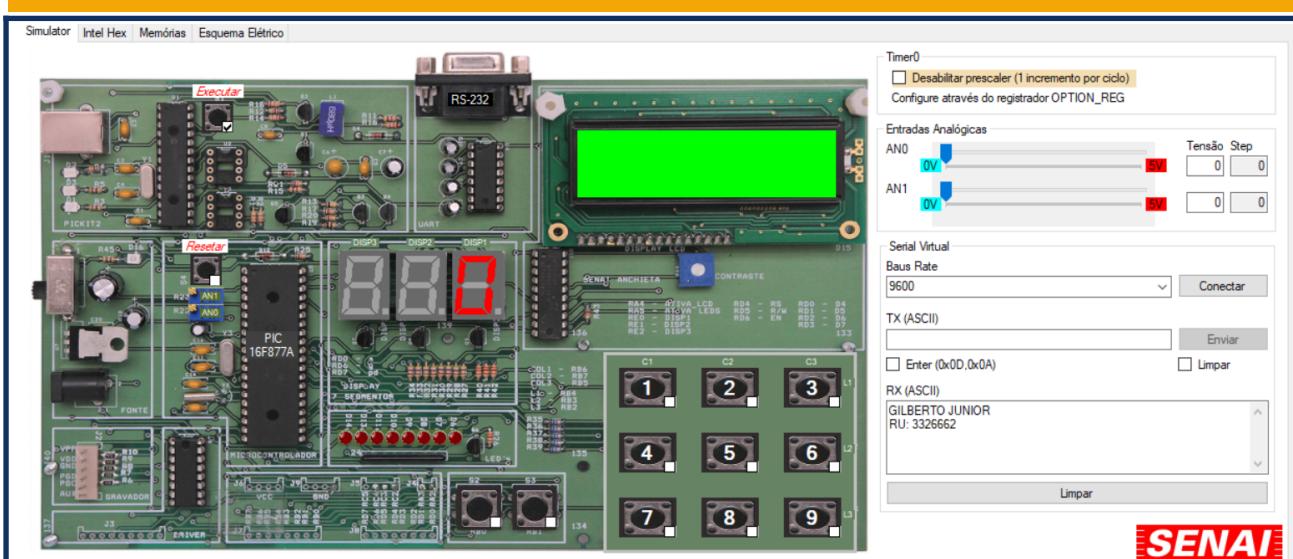


Figura 3: 0 hexadecimal

b. Imagem do simulador com o número 1 aceso.

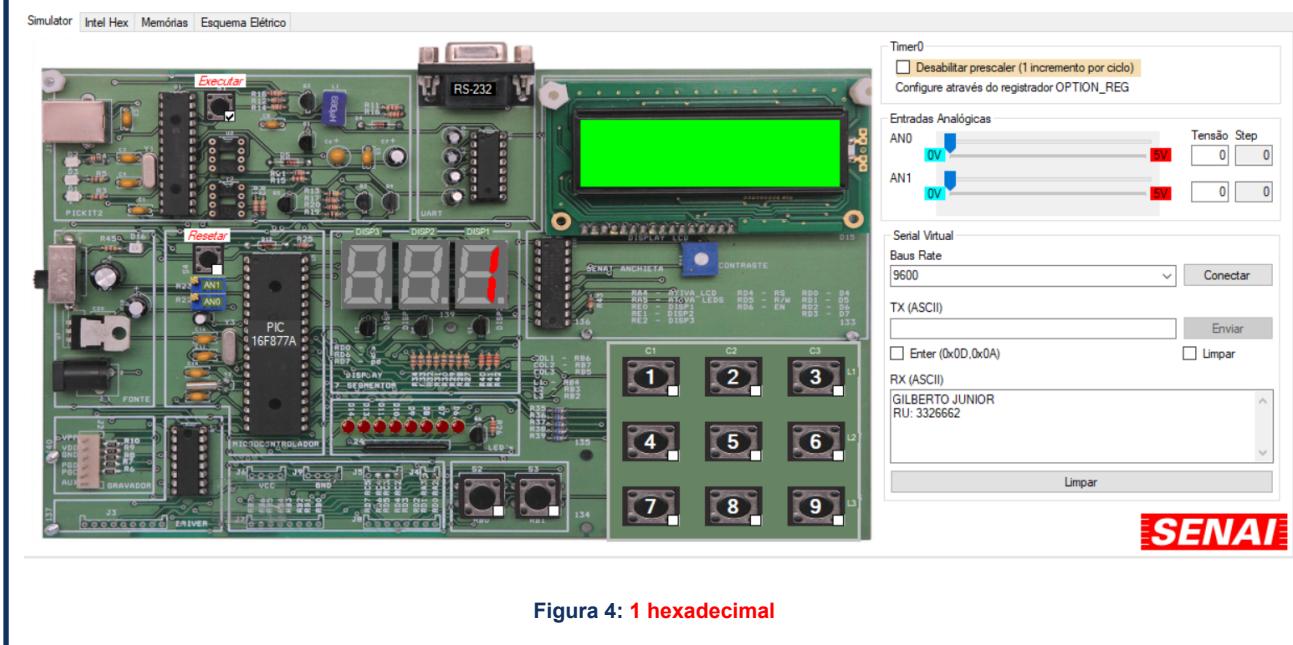


Figura 4: 1 hexadecimal

c. Imagem do simulador com o número 2 aceso.

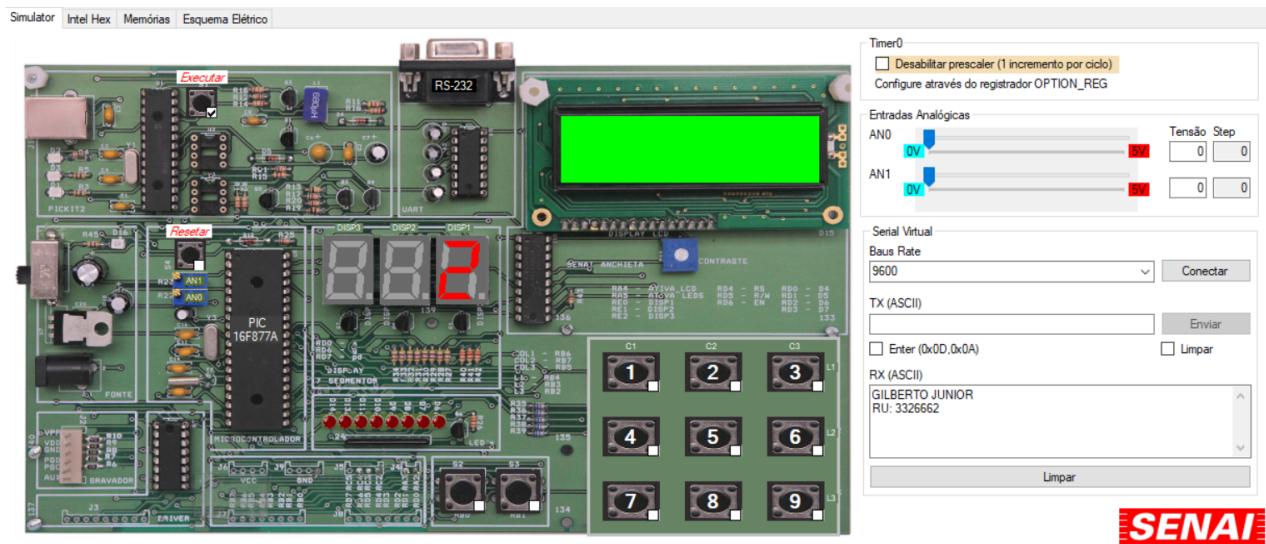


Figura 5: 2 hexadecimal

d. Imagem do simulador com o número 3 aceso.

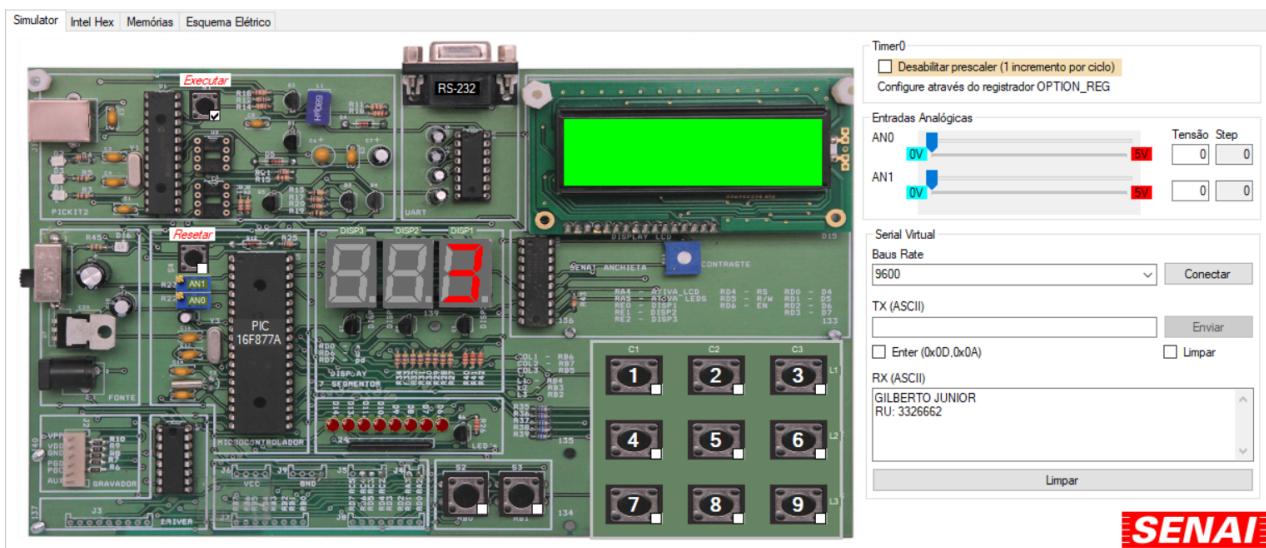


Figura 6: 3 hexadecimal

e. Imagem do simulador com o número 4 aceso.

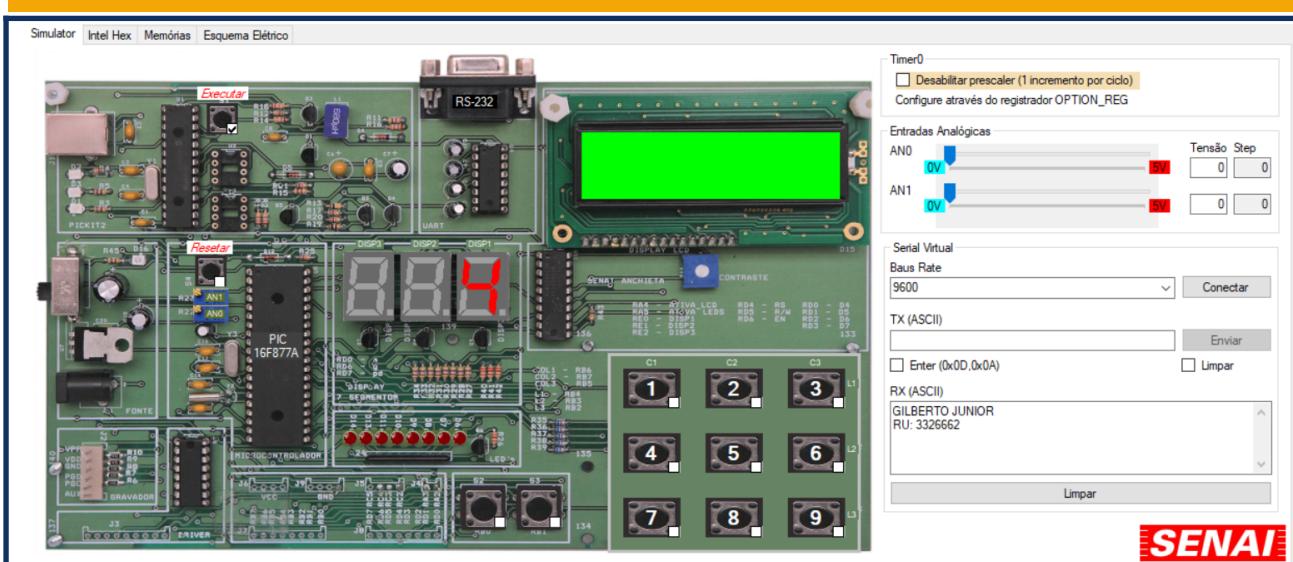


Figura 7: 4 hexadecimais

f. Imagem do simulador com o número 5 aceso.

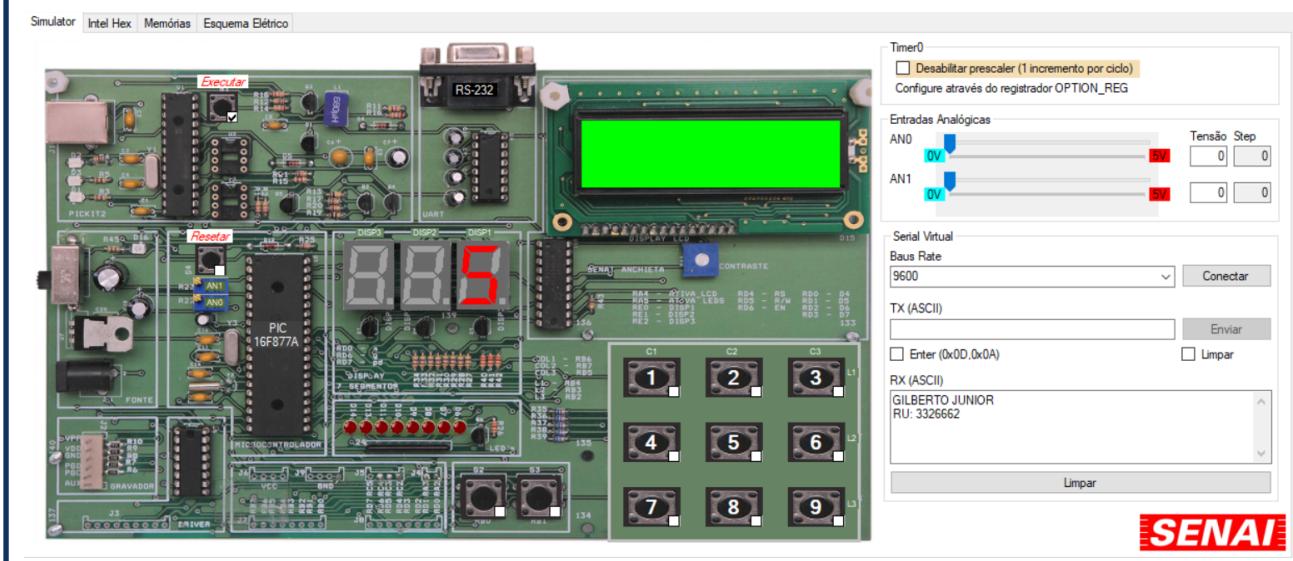


Figura 8: 5 hexadecimais

g. Imagem do simulador com o número 6 aceso.

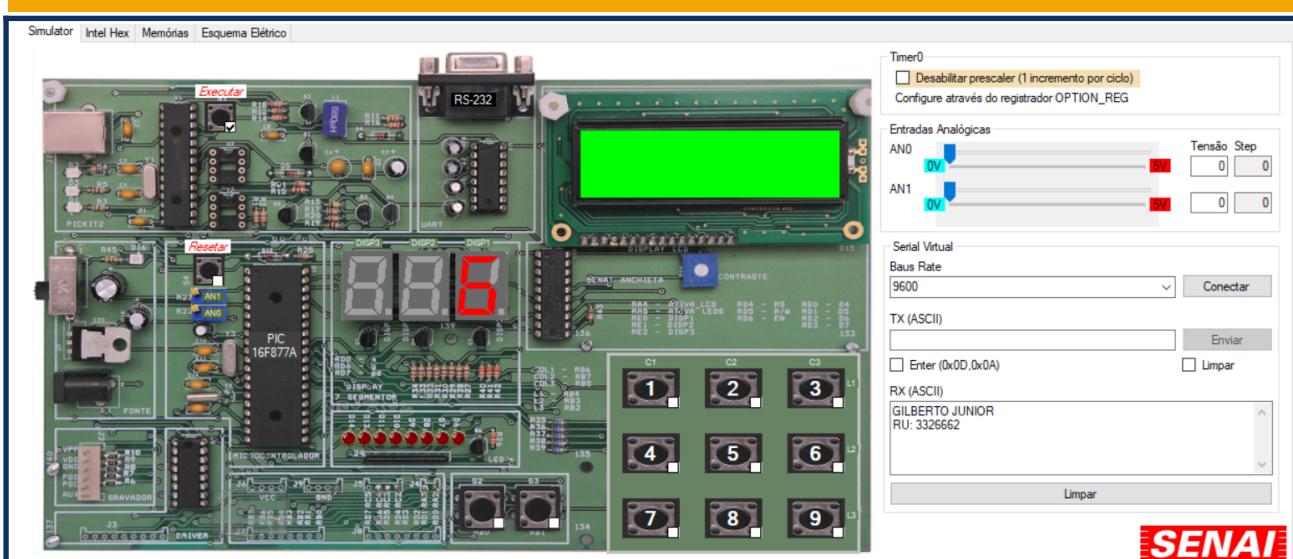


Figura 9: 6 hexadecimal

h. Imagem do simulador com o número 7 aceso.

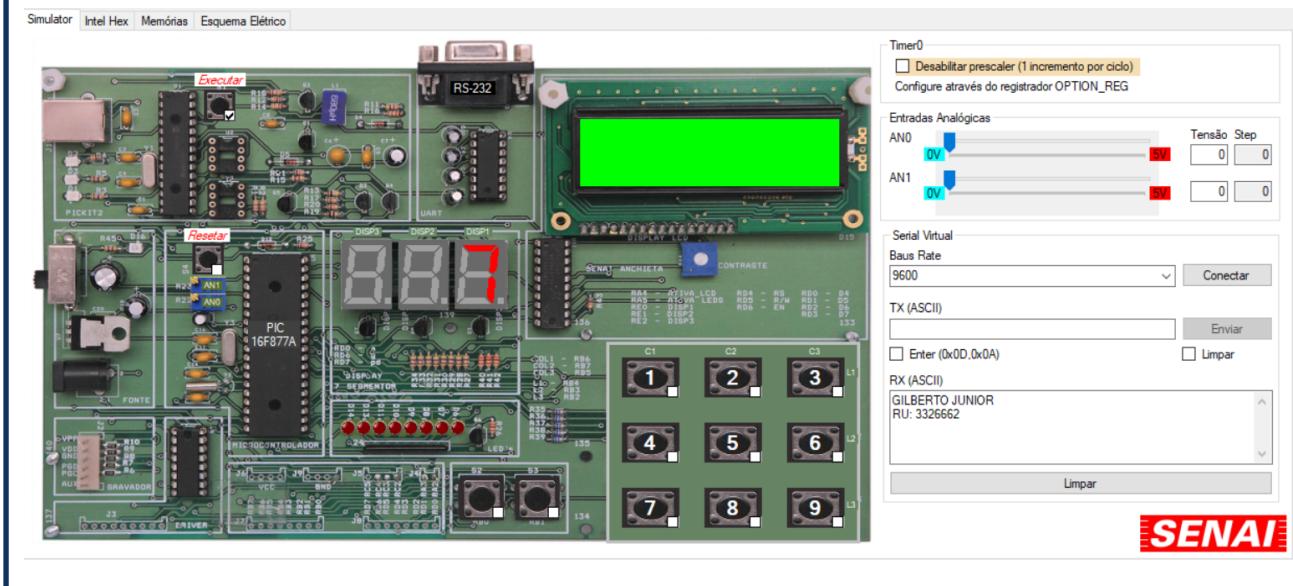


Figura 10: 7 hexadecimal

i. Imagem do simulador com o número 8 aceso.

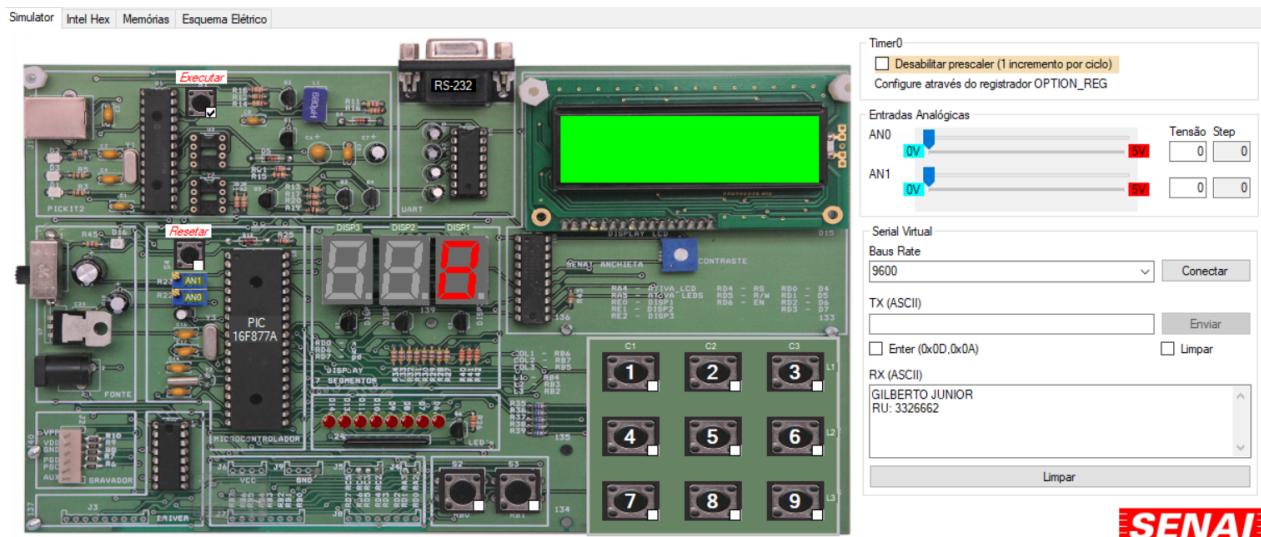


Figura 11:8 hexadecimal

j. Imagem do simulador com o número 9 aceso.

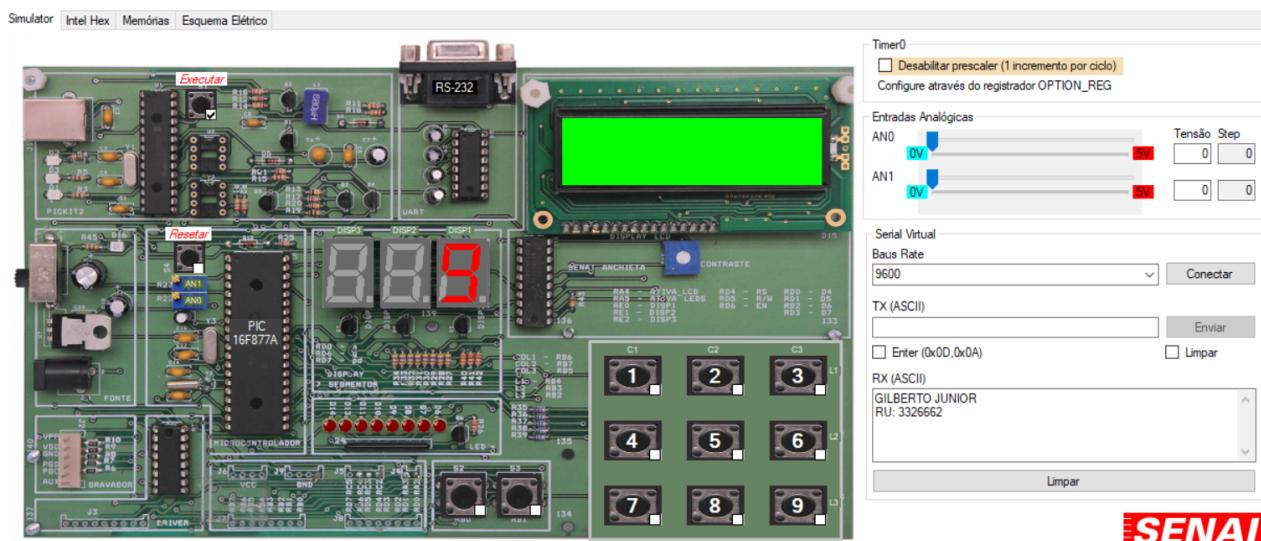


Figura 12: 9 hexadecimal

k. Imagem do simulador com o número A aceso.

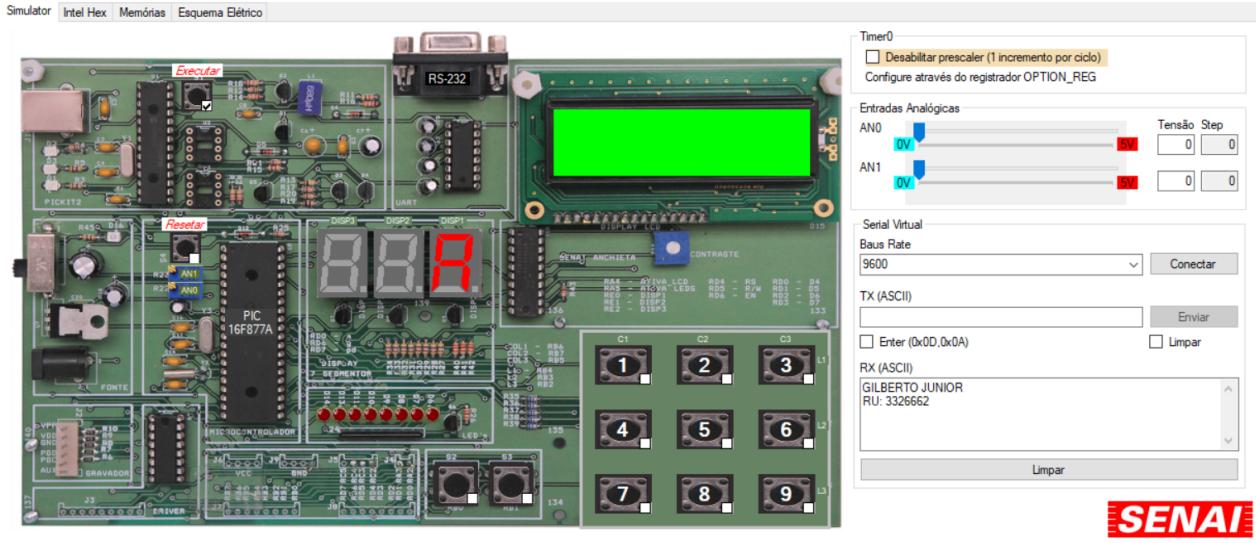


Figura 13: A hexadecimal

I. Imagem do simulador com o número b **aceso**.

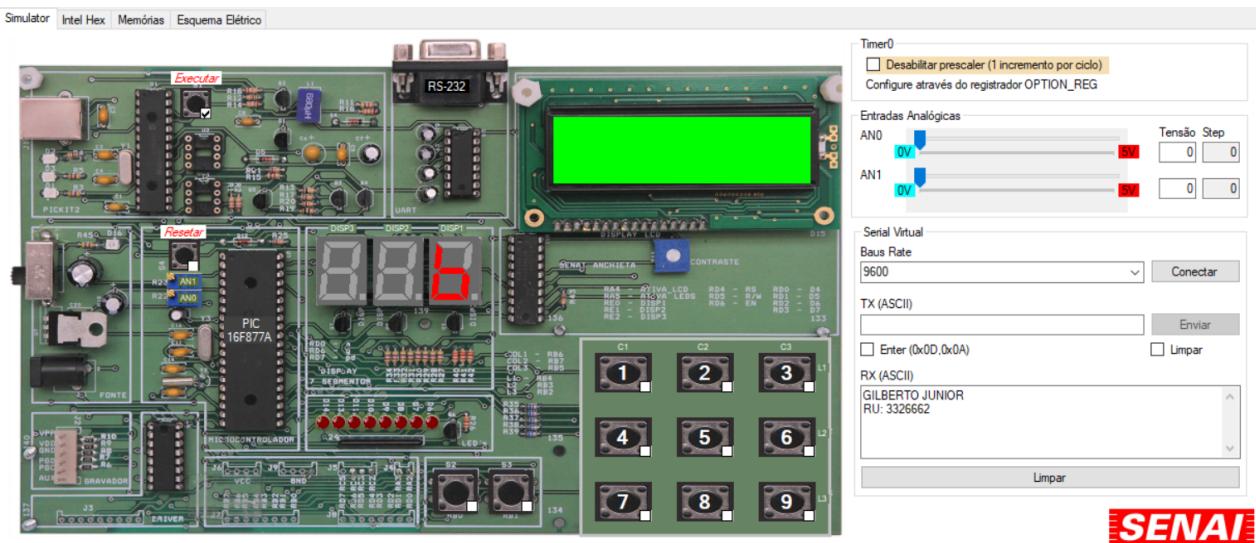


Figura 14: B hexadecimal

m. Imagem do simulador com o número C **aceso**.

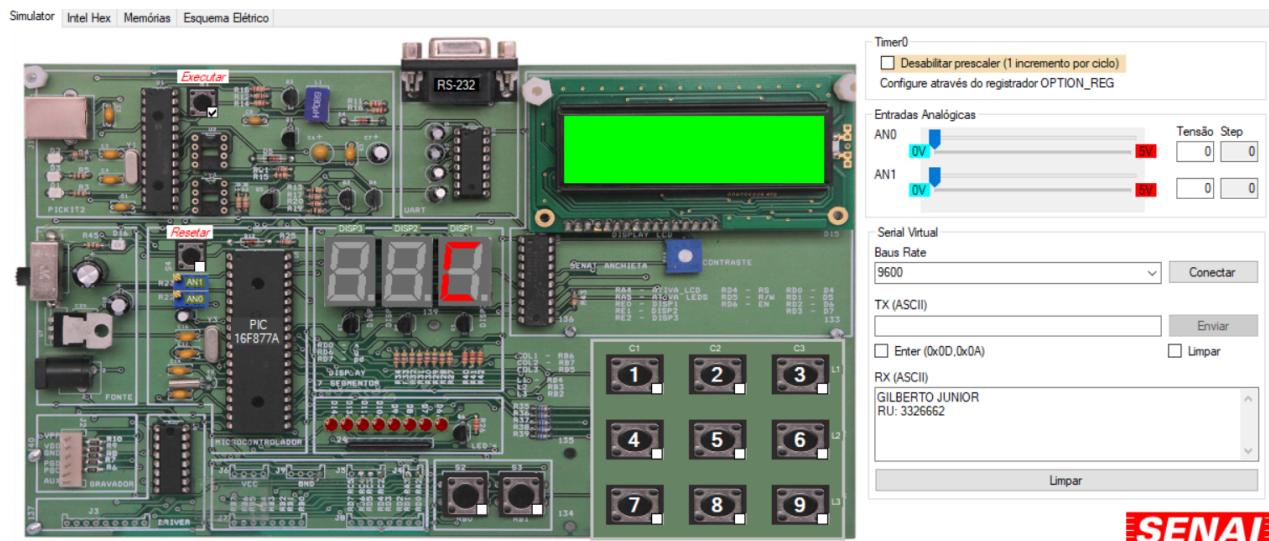


Figura 15: C hexadecimal

n. Imagem do simulador com o número d aceso.

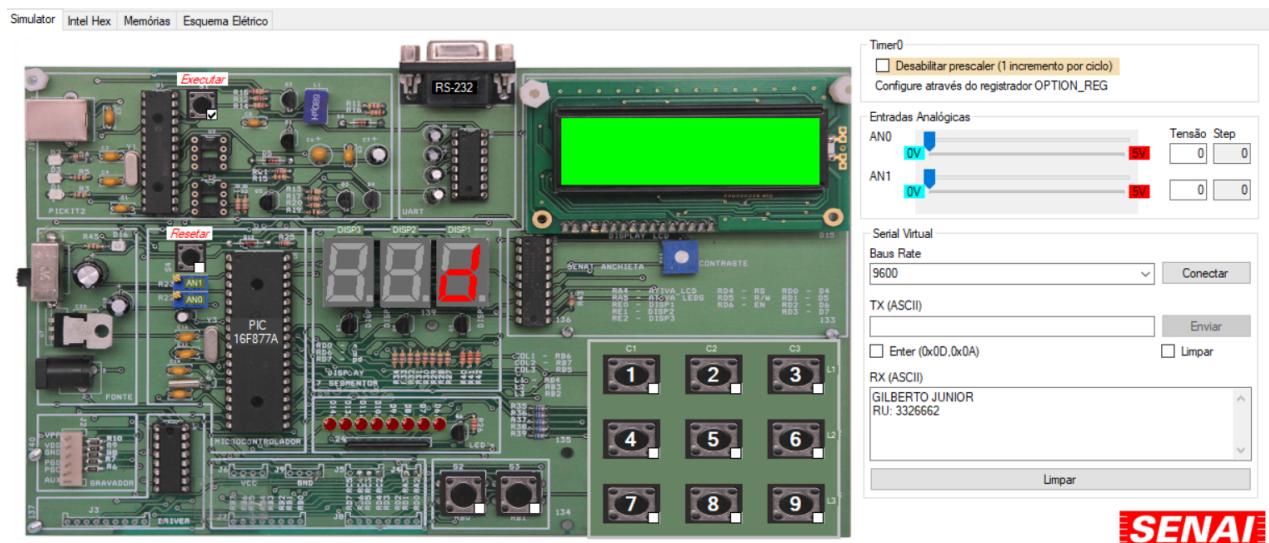


Figura 16: D hexadecimal

o. Imagem do simulador com o número E aceso.

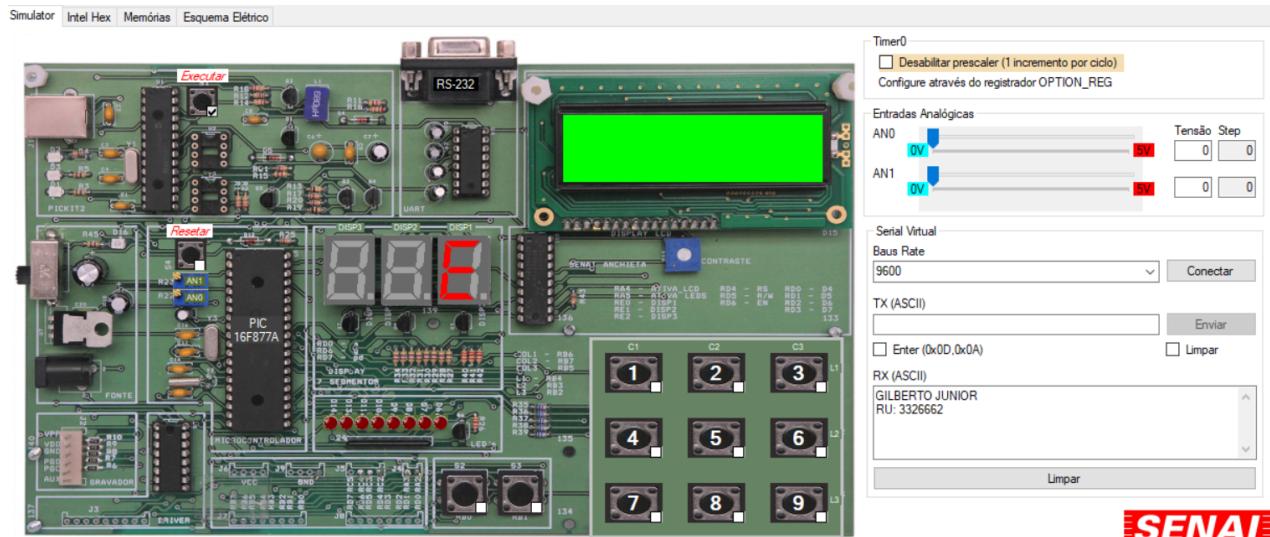


Figura 17: E hexadecimal

p. Imagem do simulador com o número F aceso.

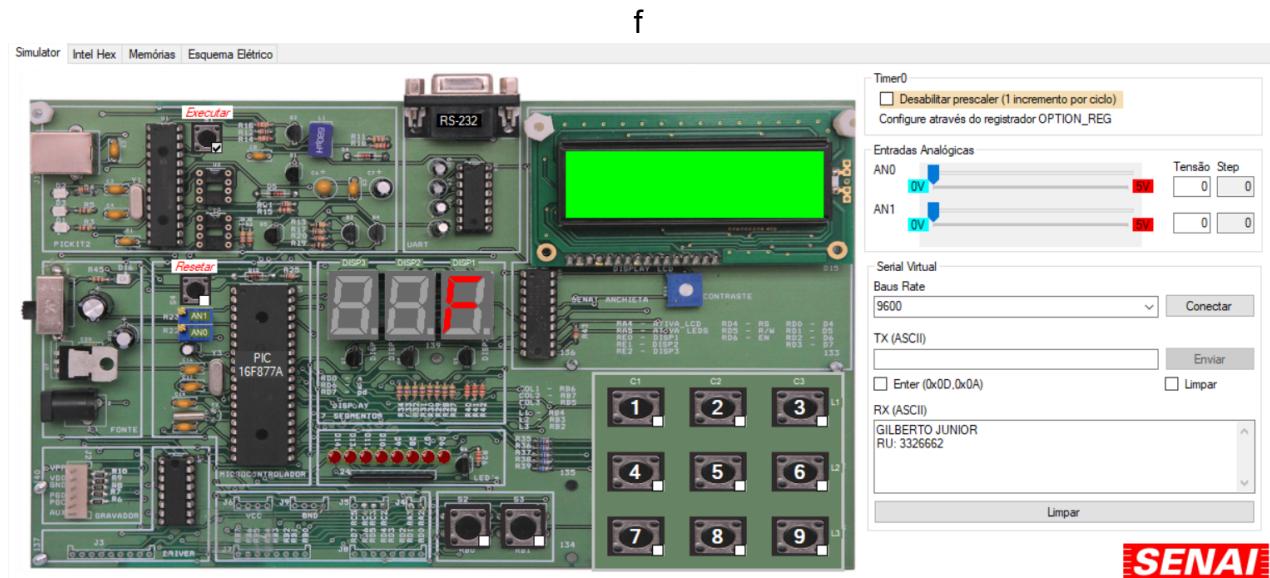
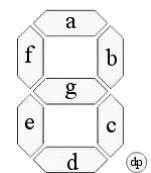


Figura 18: F hexadecimal

- III. Preencha a tabela abaixo verificando qual os segmentos devem acender (1) ou ficar apagado (0) para cada combinação. (**COLOCAR RESPOSTA AQUI**)

Pino PIC	D6	D5	D4	D3	D2	D1	D0
Seg. Display	g	f	e	d	c	b	a
0	0	1	1	1	1	1	1
1	0	0	0	0	1	1	0
2	1	0	1	1	0	1	1
3	1	0	0	1	1	1	1
4	1	1	0	0	1	1	0
5	1	1	0	1	1	0	1
6	1	1	1	1	1	0	1
7	0	0	0	0	1	1	1
8	1	1	1	1	1	1	1
9	1	1	0	1	1	1	1
A	1	1	1	0	1	1	1
B	1	1	1	1	1	0	0
C	0	1	1	1	0	0	1
D	1	0	1	1	1	1	0
E	1	1	1	1	0	0	1
F	1	1	1	0	0	0	1



PRÁTICA 01 - PROGRAMANDO O PIC16F877A

D) LENDO UM BOTÃO E ACENDENDO UM LED

ENUNCIADO: Desenvolva um programa que ao pressionar o botão RB0 os LEDs D6, D8, D10 e D13 devem acender e os LEDs D7, D9, D11 e D14 devem ser apagados. Se o botão não estiver pressionado os LEDs D6, D8, D10 e D13 devem permanecer apagados e os LEDs D7, D9, D11 e D14 devem ser acesos.

Os botões estão ligados ao PORT B do PIC, nos pinos RB0 e RB1.

- Configure no TRISB o pino RB0 como entrada.
- Atenção: O botão quando fechado conecta o pino ao GND

I. Apresentação do Código (não esquecer do identificador pessoal):

```
***** ATIVIDADE PRATICA IV *****
;***** PRATICA 4 - ACENDE LEDs D6, D8, D10 e D13 QUANDO PRESSIONA BOTAO RBO
;          ACENDE LEDs D7, D9, D11 e D14 QUANDO LIBERA BOTAO RBO
***** Descricao:
;
; Esse codigo demonstra como acender os LED acima descritos ao
; se pressionar ou nao o botao de interrupcao RBO.
;
; Configuracao do teste:
;
; MCU:          PIC16F877A
;             http://www.microchip.com/wwwproducts/en/en010210
;
; Ext. Modulo:  SIMULADOR KIT SENAI PIC 16 (Versão 3.3)
;             https://github.com/LePoloni/Simulador-KIT-SENAI-PIC16
;
; SW:           MPLAB IDE v7.00
;             http://www.microchip.com/mplab/
;
; Autor:
;
; ***** RU 3326662 GILBERTO OLIVEIRA JUNIOR
; ***** CURSO: ENGENHARIA DA COMPUTACAO
; ***** ESCOLA SUPERIOR POLITECNICA UNINTER - BRAZIL
; ***** Date : feb 2023
;***** Definição do processador *****
;
;#include p16F877a.inc
;_config _HS_OSC & _WDT_OFF & _LVP_OFF & _PWRTE_ON
;
;***** Memória de programa *****
ORG 0 ;Posiciona o ponteiro para o endereço 0x0 de memória
RESET ;Bloco de RESET do microcontrolador
NOP
GOTO SETUP ;Pula para o marcador SETUP
;***** Interrupção *****
ORG 4
;
;***** Declaração de variáveis *****
DELAY EQU 0x20
VEZES EQU 0x21
```

```

;***** Configurações *****
SETUP           ;Bloco de configuração do microcontrolador
    BSF    STATUS,RPO ;Aiona o Bank1 de memória
    MOVlw b'00000000' ;Carrega '00000000' no registrador W
    MOVwf TRISA      ;Configura PORTA como saída
    MOVlw b'00000000' ;Carrega '00000000' no registrador W
    MOVwf TRISD      ;Configura o PORTD como saída
    BCF    STATUS,RPO ;Retorna para o Bank0 de memória
    BSF    PORTA,5    ;Coloca 5V, nível lógico 1 no pino RA5; Enable dos LEDs
    BCF    PORTE,0    ;Reseta o RB0, colocando-o como entrada do BOTAO 0

;***** Programa principal *****
LOOP            ;Bloco do principal do programa
    CLRF   PORTB      ;Limpa o PORTB
    BTFSC  PORTE,0    ;Testa se o botao RB0 esta pressionado (GND)
    MOVlw b'10101010' ;Sim, botao pressionado; acende os LEDS D7,D9,D11 e D14
    BTFSS  PORTE,0    ;Testa se o botao RB0 nao esta pressionado (5V)
    MOVlw b'01010101' ;Nao, botao nao esta pressionado;
                      ;acende os LEDS D6, D8, D10 e D13
    MOVwf  PORTD      ;Envia o conteúdo de W para o registrador PORTD colocando
                      ;nível logico alto (+5V) pinos correspondentes do PORTD
    GOTC  LOOP        ;Manda o programa pular para o inicio do bloco LOOP
                      ;mantendo o programa em execução eterna

;***** Rotinas de Delay *****
PERDE_TEMPO
    MOVlw d'1'          ;Procedimento que força o PIC a perder tempo(delay)
    MOVwf VEZES         ;Para gravar no PIC utilizar 80(simulador SENAI use 1)

LOOP_VEZES
    MOVlw d'1'          ;Para gravar no PIC utilizar 255,(simulador SENAI use 1)
    MOVwf DELAY
    CALL  DELAY_US
    DECFSZ VEZES,5
    GOTC LOOP_VEZES
    RETURN

DELAY_US
    NOP
    NOP
    DECFSZ DELAY,1
    GOTC DELAY_US
    RETURN

END             ;Marcador de última linha do programa

```

Códigos elaborados fora do MPLAB IDE terão sua nota descontada em 50%

II. Apresentação das Imagens (não esquecer do identificador):

a. Imagem do simulador com o botão NÃO pressionado.

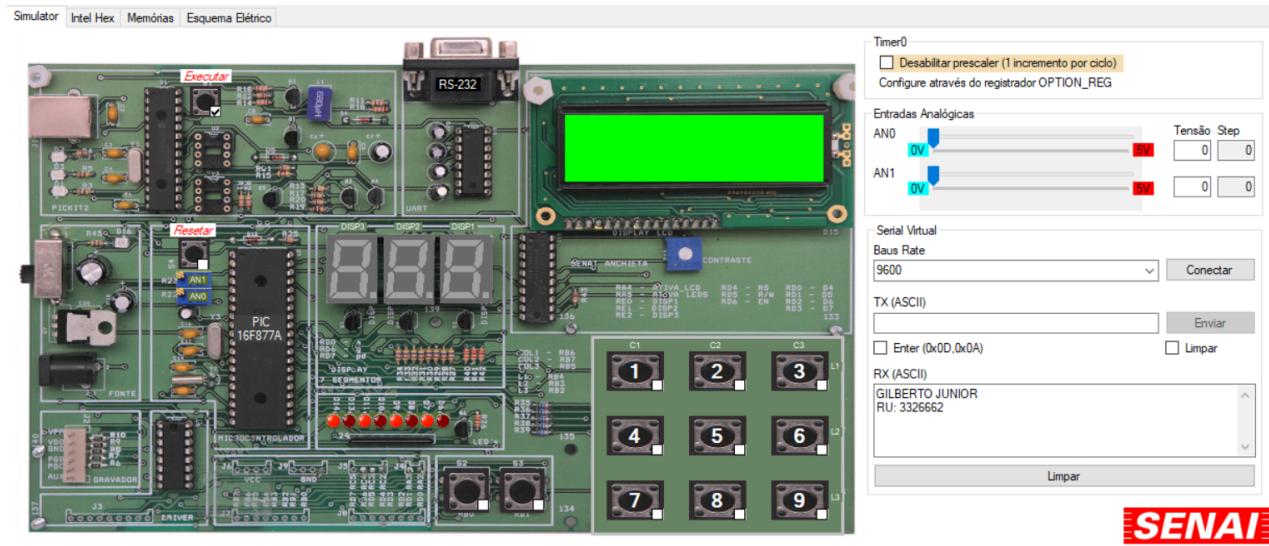


Figura 19: ACENDE LEDs D7, D9, D11 e D14 QUANDO LIBERA BOTÃO RB0

b. Imagem do simulador com o botão pressionado.

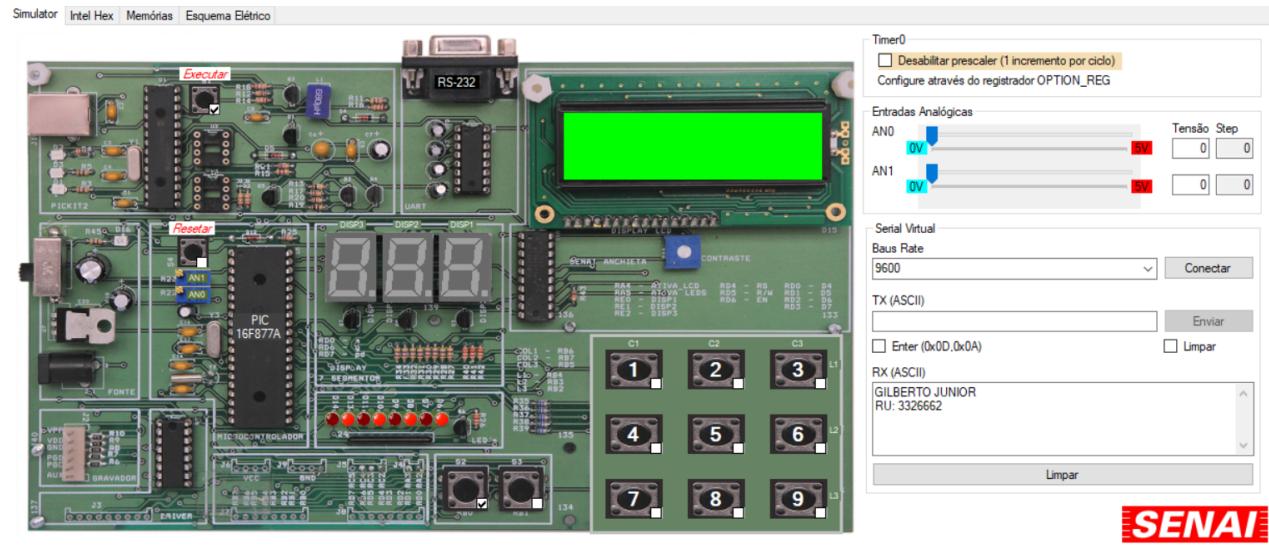


Figura 20: ACENDE LEDs D6, D8, D10 e D13 QUANDO PRESSIONA BOTÃO RB0