

Pré-Laboratório 04 - PWM

Questão 1 :

```
1 #INCLUDE <P16F628A.INC>
2
3 ORG      0x00
4     GOTO      INI
5
6 ORG      0x04
7     RETFIE
8
9 INI :
10    BANKSEL   TRISB
11    BCF       TRISB,3
12
13    BANKSEL   CCP1CON
14    MOVLW    B'00101100'
15    MOVWF    CCP1CON
16
17    BANKSEL   PR2
18    MOVLW    D'30'
19    MOVWF    PR2
20
21    BANKSEL   T2CON
22    MOVLW    B'00001001'
23    MOVWF    T2CON
24
25    BANKSEL   CCPR1L
26    MOVLW    B'00001001'
27    MOVWF    CCPR1L
28
29 MAIN :
30    GOTO     MAIN
31
32 END
```

Os comandos “BANKSEL TRISB” e “BCF TRISB,3” são responsáveis por, respectivamente, procurar o banco onde se encontra o registrador TRISB e configurar o pino RB3 do microcontrolador como saída. Já os trechos referentes ao registrador CCP1CON são responsáveis por configurar a função PWM (passando 11xx para os bits menos significativos do registrador) assim como configurar um valor específico calculado para que possamos ter o valor correto de DUTY CYCLE.

A forma como foi obtido o valor específico dos bits 5 e 4 do registrador CCP1CON, o valor de PR2 e o de CCP1RL serão comentadas logo abaixo:

$$Periodo_{PWM} = (PR2 + 1) * 4 * \frac{1}{F_{osc}} * TMR2_{prescaler} \quad (1)$$

$$Periodo_{PWM} = \frac{1}{20 * 10^3} = 0.00005s$$

Utilizando $TMR2_{prescaler} = 4$ e $F_{osc} = 10MHz$, temos :

$$PR2 = 30 \quad (2)$$

Para um DUTY CYCLE de 30%, ou seja, 0.000015 segundos em tempo alto, temos a seguinte equação :

$$DUTY_{CICLE} = (CCP1RL : CCP1CON < 5 : 4 >) * \frac{1}{F_{osc}} * TMR2_{prescaler}$$

$$(CCP1RL : CCP1CON < 5 : 4 >) = 38 \quad (3)$$

Como 38 está em decimal, precisamos convertê-lo para binário, ou seja, 0000100110. Assim precisamos passar os últimos dois bits desse valor para os bits 5 e 4 do registrador CCP1CON e o resto do valor para o registrador CCP1RL. Agora que obtemos todos os valores necessários e escolhemos o valor de prescaler do TMR2, basta passar esses valores para os registradores específicos. Isso é feito nas linhas 13 até a linha 27 do código. Logo após as configurações necessárias temos um loop “MAIN GOTO MAIN” para que o PWM siga funcionando em loop.

Questão 2:

```

1  #INCLUDE <P16F628A.INC>
2
3  AUX1      EQU      0x20
4  AUX2      EQU      0x21
5  AUX3      EQU      0x22
6
7  ORG      0x00
8      GOTO      INI
9
10 ORG      0x04
11      RETFIE
12
13 INI :
14      BANKSEL  CMCON
15      MOVLW    0x07
16      MOVWF    CMCON
17
18      BANKSEL  TRISA
19      CLRF     TRISA
20
21 MAIN :
22      BANKSEL  PORTA
23      MOVLW    0x09
24      MOVWF    PORTA

```

```

25      CALL      DELAY_05S
26      MOVLW     0x03
27      MOVWF     PORTA
28      CALL      DELAY_05S
29      MOVLW     0x06
30      MOVWF     PORTA
31      CALL      DELAY_05S
32      MOVLW     0x0C
33      MOVWF     PORTA
34      CALL      DELAY_05S
35      GOTO      MAIN
36
37 DELAY_05S :
38      MOVLW     .250
39      MOVWF     AUX1
40      MOVWF     AUX2
41      MOVLW     .5
42      MOVWF     AUX3
43 PT1 :
44      NOP
45      DECFSZ     AUX1
46      GOTO      PT1
47 PT2 :
48      MOVLW     .250
49      MOVWF     AUX1
50      DECFSZ     AUX2
51      GOTO      PT1
52 PT3 :
53      MOVLW     .250
54      MOVWF     AUX2
55      DECFSZ     AUX3
56      GOTO      PT1
57      RETURN
58      END

```

Dentro do trecho de código demarcado por “INI” é repassado o valor 0x07 em hexadecimal para o registrador CMCON, dessa forma podemos desativar a função “comparador” que vem por default associada aos primeiros pinos do registrador PORTA, tal passo é necessário pois os pinos associados a essa função serão utilizados como saída para comandar o motor de passo.

Logo após, passamos o valor 0x00 para o registrador TRISA para que seja possível utilizar todos os pinos de PORTA como saída.

O código contido dentro do trecho demarcado pela label “MAIN” é bem simples, o mesmo consiste de repassar os valores 0x09, 0x03, 0x06 e 0x0C, que são os valores descritos no pré-laboratório porém em hexadecimal, para o registrador PORTA, mas a cada troca de valor é chamada a função *DELAY_05S* que nada mais é que um delay de 0.5s. Dessa forma, existe um intervalo de aproximadamente 0.5 segundos entre a transmissão de cada valor.