

```
1: /*-----
2: Programa : IntTimer.c
3: Data      : 27/03/2017
4: Autor     : Prof. Fernando Simplicio
5: Atualizado em 12/04/2017 - Prof. Vargas
6:
7: Utilizando o Interrupt Assistant
8: -----*/
9: //Protótipo de Função
10: void ConfigMCU();
11: void ConfigTempoTIMER();
12: void ConfigInterrupcao_Global();
13: void ConfigInterrupcao_TIMER0();
14:
15: //Rotina de Interrupção gerada pelo Assistente do mikroC
16: //Menu Tools + Interrupt Assistant
17:
18: void INTERRUPTAO_HIGH() iv 0x0008 ics ICS_AUTO
19: {
20:     if(INTCON.TMR0IF == 1) //Checa quem interrompeu
```

```
21:     {
22:         PORTB.RB2 = ~LATB.RB2; //Altera LED RB2
23:         TMR0H = 0xCF;          //Recarrega o TIMER0
24:         TMR0L = 0x2C;
25:         INTCON.TMR0IF = 0;     //Limpa Flag
26:     }
27: }
28:
29: void main()
30: {
31:     ConfigMCU();
32:     ConfigTempoTIMER();
33:     ConfigInterrupcao_Global();
34:     ConfigInterrupcao_TIMER0();
35:     while(1)
36:     {
37:         PORTB.RB1 = ~LATB.RB1; //Inverte LED em RB1
38:         Delay_ms(1000);        //Atrasa 1 Seg
39:     }
40: }
```

```

41:
42: void ConfigMCU()
43: {
44:     LATD = 0;
45:     LATB = 0xFF;
46:     TRISD = 0b00011100;
47:     TRISB = 0b00000001;
48: }
49:
50: void ConfigTempoTIMER()
51: {
52:     /*-----[Cálculo para 200ms]-----
53:         Tempo de overflow = (65536 - Ref) * (4/Fosc) * Prescaler
54:         200.000 uS = (65536 - Ref) * 0.5 uS * 32
55:         Ref = 65536 - 12500 = 53036 ou 0xCF2C
56:         Carregamos TMR0L com 2C
57:         e           TMR0H com CF
58:         -----*/
59:     T0CON = 0b00000100; //TIMER_OFF, MOD_16BITS, TIMER, PRES_1:32
60:     TMR0H = 0xCF;

```

```
61:     TMR0L = 0x2C;
62:     INTCON.TMR0IF = 0;
63:     T0CON.TMR0ON = 1;    //Liga o TIMER0
64: }
65:
66: void ConfigInterrupcao_Global()
67: {
68:     INTCON.GIEH = 1;    //Habilita interrupção High
69:     INTCON.GIEL = 1;    //Habilita interrupção Low
70:     RCON.IPEN = 1;      //0 -> (default) herança 16F, só um nível
71:                          //1 -> Usa dois níveis de prioridade
72: }
73:
74: void ConfigInterrupcao_TIMER0()
75: {
76:     INTCON.TMR0IF = 0;    //Limpa Flag de TIMER0
77:     INTCON2.TMR0IP = 1;   //Prioridade de TIMER0 é High
78:     INTCON.TMR0IE = 1;   //Habilita a interrupção do TIMER0
79: }
```