

```

1  #include<p18f452.h>
2  #include<adc.h>
3  #include<timers.h>
4  #include<pwm.h>
5
6  #pragma config WDT = OFF, LVP = OFF, OSC = HSPLL, PWRT = ON, BOR = ON, BORV = 42
7
8  // Sintonia do controlador
9  const double a1 = 0.67032;
10 const double b1 = 0.015516144183773*4;      //b1 = 3.1746031*5/1023;
11 const double b2 = -0.014710475073314*4;     //b2 = -3.0097632*5/1023;
12
13 const int x_bar = 500;      //x_bar = Vsensor*1023/5;
14 const float u_bar = 9.5;//8.9; // Volts
15 const unsigned char VCC = 15;
16 float lim_max;
17
18 unsigned int uPWM;
19 volatile double u = 0, u_ant = 0;
20 volatile int e, e_ant = 0;
21
22 //Rotina de interrupção
23 void ISR_High_Priority(void);      //Prototipagem da função
24
25 #pragma code vec_int_high_priority = 0x08 //Alocação na mem. de programa
26 void vec_int_high_priority(void)
27 {
28     _asm GOTO ISR_High_Priority _endasm
29 }
30 #pragma code
31 #pragma interrupt ISR_High_Priority
32 void ISR_High_Priority(void)
33 {
34     if(INTCONbits.TMR0IF)
35     {
36         ConvertADC(); //Inicia conversão
37         WriteTimer0(61536); //Reinicializa TMR0
38         INTCONbits.TMR0IF = 0; //Limpa flag de interrupção
39         uPWM = (u + u_bar)*800/VCC;
40         SetDCPWM1(uPWM); //Atualização do PWM
41
42         while(BusyADC()); //Aguarda conversor A/D
43         e = ReadADC() - x_bar; //Cálculo do erro
44
45         u = a1*u_ant + b1*e + b2*e_ant; //Eq. de diferenças
46
47         //Teste de Saturação
48         if(u > lim_max) u = lim_max;
49         else if(u < -u_bar) u = -u_bar;
50
51         e_ant = e;
52         u_ant = u;
53
54         if(INTCONbits.TMR0IF) //Intervalo de amostragem é
55         { // suficiente? Se não for:
56             PORTD = 0b01111111;
57             while(1); //HALT
58         }
59     }
60 }
61
62 void main(void)
63 {
64     PORTB = 0b00010000;
65     PORTD = 0x00;
66
67     TRISA = 0xFF;
68     TRISB = 0x0F;
69     TRISC = 0b11111011;
70     TRISD = 0x00;
71
72     //Timer0
73     OpenTimer0(TIMER_INT_ON & T0_16BIT & T0_SOURCE_INT & T0_PS_1_1);
74
75     //Timer2 (p/ PWM)
76     OpenTimer2(TIMER_INT_OFF & T2_PS_1_4);

```

```

77
78 OpenPWM1(199);           //Carrega PR2 Tpmw = 200us
79 SetDCPWM1(u_bar*800/VCC); //Duty-cycle inicial
80
81 //Configura conversor A/D interno
82 OpenADC( ADC_FOSC_32 & ADC_RIGHT_JUST &
83         ADC_5ANA_0REF, ADC_CH2 & ADC_INT_OFF );
84
85 INTCONbits.GIE = 1; //Seta flag geral de interrupções
86 RCONbits.IPEN = 1; //Habilita prioridade de interrupções
87 INTCON2bits.TMR0IP = 1; //Int. de timer0 como de alta prioridade
88
89 //Inicialização de variáveis
90 lim_max = VCC - u_bar;
91
92 WriteTimer0(61536);
93 INTCONbits.TMR0IF = 0;
94 INTCONbits.TMR0IE = 1;
95
96 while(1){} //Loop principal
97 }

```