

```

1  #include <89c51rd2.h>
2  #include <stdio.h>
3
4  //***** Definicao dos sinais de hardware *****
5  // Enable da placa de sensores      P1.2      Out      Activo a 0
6  // Clock da placa de sensores       P1.0      In       Activo a 0
7  // Dados da placa de sensores       P1.1      In       (S6 .. S0)
8  ///////////////////////////////////////////////////////////////////
9  // Ligar sonar                       P1.3      Out      Activo a 1
10 // Eco do sonar                       P3.3      In       Activo a 1
11 ///////////////////////////////////////////////////////////////////
12 // Direccao do motor 1 (esquerdo)   P3.4      Out      (1-frente 0-tras)
13 // Direccao do motor 2 (direito)    P3.5      Out      (1-frente 0-tras)
14 // PWM motor 1 (esquerdo)          P1.4      Out
15 // PWM motor 2 (direito)           P1.5      Out
16 //***** Placa com IVs *****
17 //
18 //      TRAS
19 //      X X X D4 D3 D2 X X          Leitura da Placa de IV
20 //      FRENTE
21 //
22 //      TRAS
23 //      0 0 0 (0x00)  sobre a linha          estado=0
24 //      0 0 1 (0x04)  fugiu para a esquerda  estado=1
25 //      0 1 1 (0x0C)  fugiu para a esquerda  estado=2
26 //      1 1 1 (0x1C)  fora da linha          estado=3
27 //      1 0 0 (0x10)  fugiu para a direita   estado=4
28 //      1 1 0 (0x18)  fugiu para a direita   estado=5
29 //      FRENTE
30 //*****
31
32 sbit enable_sensores = P1^2;          //Placa de sensores de fita
33 sbit clock_sensores = P1^0;
34 sbit data_sensores = P1^1;
35
36 sbit dir0 = P3^4;                    //Motores
37 sbit dir1 = P3^5;
38
39 bit flag1 = 1;
40 bit flag10ms = 0;
41
42 unsigned char Leitura_IV;
43 unsigned char cont1;
44 unsigned char cont10ms = 10;
45 unsigned char estado = 0;
46
47 const Vm = 80;
48 const k = 20;
49 const k1 = 30;
50 const k2 = 50;
51
52 #define DIR(a,b) dir0=a; dir1=b      // DIR(esquerdo,direito)
53 #define SET_PWM(l,r) CCAP1H=(255-r); CCAP2H=(255-l) // SET_PWM(esquerdo,direito)
54
55
56 //***** Interrupt do T/CO *****
57 void int_tc0 (void) interrupt 1 using 2
58 {

```

```

59     TH0 = 0xf9; // T/C1 para gerar overflow em 1mS
60     TL0 = 0x7E;
61     cont10ms--; // Contador para 10mS
62     if(cont10ms == 0)
63     {
64         cont10ms = 10; // Mudar este valor para alterar a frequência de controlo
65         flag10ms = 1;
66     }
67     cont1--; // Contador de mS
68     if(cont1 == 0) flag1=1;
69 }
70
71 //***** Inicializacao do uC *****
72 void inic(void)
73 {
74     CKCON = 0x2f; // Clock dobrado, perifericos com 12 ciclos
75
76     RCAP2H = 0xff; // T/C2 para gerar baud rate de 9600 bps
77     RCAP2L = 0xbf;
78     T2CON = 0x34;
79     SCON = 0x52; // UART em modo 1
80
81     CMOD = 0; // PCA configuration
82     CCON = 0x40;
83     CCAPM1 = 0x42; // Configuração dos módulos para PWM
84     CCAPM2 = 0x42;
85
86     TMOD = 0x91; // T/Cs em modo de 16 bits, T/C1 GATE on (para sonar)
87     TH0 = 0xf9; // T/CO para gerar overflow em 1mS
88     TL0 = 0x7E;
89
90     IPL = 0x02; // Int do T/CO com maior prioridade
91     IE = 0x82; // Enable da int do T/CO
92
93     TR0 = 1; // T/CO on
94 }
95
96 //***** Leitura dos detectores de fita *****
97 unsigned char ler_IV(void)
98 {
99     unsigned char i, leitura;
100
101     leitura=0;
102     enable_sensores=0;
103     for (i=1; i<8; i++)
104     {
105         leitura=leitura<<1;
106         while(clock_sensores==1);
107         if (data_sensores==1) leitura++;
108         while(clock_sensores==0);
109     }
110     enable_sensores=1;
111     return leitura;
112 }
113
114 //***** PROGRAMA PRINCIPAL *****
115 void main(void)
116 {

```

```

117     inic();
118     DIR(1,1);
119     SET_PWM(Vm,Vm);
120     while(1)
121     {
122         while(flag10ms == 0);           // Espera inicio de ciclo de controlo
123         flag10ms = 0;
124         Leitura_IV = ler_IV();
125         Leitura_IV = Leitura_IV & 0x1C;
126
127         switch (Leitura_IV)
128         {
129             case 0x0:                   // Centrado na linha (estado 0)
130                 break;
131
132             case 0x10:                  // Sai pela direita (estado 4)
133                 SET_PWM(Vm-k,Vm+k);
134                 break;
135
136             case 0x18:                  // Sai pela direita + (estado 5)
137                 SET_PWM(Vm-k1,Vm+k1);
138                 break;
139
140             case 0x04:                  // Sai pela esquerda (estado 1)
141                 SET_PWM(Vm+k,Vm-k);
142                 break;
143
144             case 0x0C:                  // Sai pela esquerda + (estado 2)
145                 SET_PWM(Vm+k1,Vm-k1);
146                 break;
147
148             case 0x1C:                  // Fora da linha (estado 3)
149                 SET_PWM(0,0);
150         }
151     }
152 }
153 }
154

```