```
#use "sencdr2.icb"

/*variaveis globais*/
float blocoAzul=67.0;
float blocoVerde=234.0;
float blocoVermelho=158.0;
float blocoAmarelo=13.0;

float desvioAzul=0.0;
float desvioVerde=0.0;
float desvioVermelho=0.0;
float desvioAmarelo=0.0;

/*sensores fixos*/
int sensorLeftPort=2;
int sensorRightPort=3;
int sensorBWLeft=4;
int sensorBWRight=5;
int sensorWallLeft=7;
int sensorWallRight=8;

/*sensores moveis*/
int sensorColorPort=2;
int sensorDistancePort=3;

int limiar=170;

int PolLeft=0;
int PolRight=0;

/*motor 0 esquerda e 1 direita*/

/*  leds

  amarelo    azul
  vermelho   verde*/


/*inicia o programa*/
void main(){
    int main_id;
    main_id=start_process(menu());
    start_process(finaliza_stop(main_id));
}

/*funcao responsavel por fazer o 'stop' parar o programa*/
void finaliza_stop(int main_id){
    while(!stop_button());
    while(stop_button());
    kill_process(main_id);
    alloff();
    printf("Execucao abortada.\n");
    beep();
    sleep(1.0);
    main_id=start_process(main());
    start_process(finaliza_stop(main_id));
}
```

```
/*calibra o valor das cores*/
void calibrar(){

    int bloco,i_motor;
    int soma,i;
    float media,delay;

    while(!start_button()){
        delay=(float)(knob()/10);
        printf("Delay = %f \n",delay);
        sleep(0.25);
    }

    /*escolher o bloco que sera calibrado*/
    while(start_button());
    while(!start_button()){
        sleep(0.25);
        if(knob()<60){
            printf("Bloco = Vermelho\n");
            bloco=1;
        }
        else if(knob()<120){
            printf("Bloco = Verde\n");
            bloco=2;
        }
        else if(knob()<180){
            printf("Bloco = Azul\n");
            bloco=3;
        }
        else{
            printf("Bloco = Amarelo\n");
            bloco=4;
        }
    }


    /*delay para a preparacao*/
    sleep((float)delay);

    /*
    if(i_motor==1){
        motor(0,0);
        motor(1,0);
    }*/


    /* ligando led de acordo com o bloco*/
    poke(0x1009,0x3c);
    poke(0x1008,4<<bloco);


    /* coletando dados do sensor */
    soma=0;
    for(i=0;i<50;i++){
        soma+=analog(sensorColorPort);
        sleep(0.2);
    }
    alloff();
```

```
    /* fazendo media de 50 valores */
    media=(float)(soma/50);
    printf("Media = %f\n",media);

    /* setando a media na variavel global */
    switch(bloco){
        case 1:
          blocoVermelho=media;
        break;

        case 2:
          blocoVerde=media;
        break;

        case 3:
          blocoAzul=media;
        break;

        case 4:
          blocoAmarelo=media;
        break;
    }

    /* desligando led */
    poke(0x1008,0);

}

/* funcao para o robo girar */
void gira(int pow_motor1, int pow_motor2, float time){
    motor(0,pow_motor1);
    motor(1,pow_motor2);
    sleep(time);
}

/* funcao para o robo andar*/
void anda(int pow_motor1, int pow_motor2, float time){
    motor(0,pow_motor1);
    motor(1,pow_motor2);
    sleep(time);
}

/* calibrar polarizacao */
void calibrarPolarizacao(){

    int aux1,aux2,cont;

    printf("Posicione o robo.\n");
    while(!start_button());

    aux1=0;
    aux2=0;
    for(cont=0;cont<20;cont++){
        aux1+=analog(sensorLeftPort);
        aux2+=analog(sensorRightPort);
    }

    PolLeft=aux1/20;
```

```
    PolRight=aux2/20;
}



/*funcao que segue a luz*/
void seguirLuz(){
    int sensorLeft,sensorRight;
    int sensorDistance;

    calibrarPolarizacao();

    sensorDistance=digital(sensorDistancePort);
    printf("Seguindo Luz\n");
    sleep(0.5);

    /* enquanto nao bater */
    while(1){
        sensorDistance=digital(sensorDistancePort);
        sensorLeft=analog(sensorLeftPort);
        sensorRight=analog(sensorRightPort);
        printf("Lendo Dados\n");
        sleep(0.5);

        if((sensorLeft - PolLeft) < (sensorRight - PolRight)){
            beep();
            printf("<<< virando esquerda <<<\n");
            gira(-45,50,1.0);
            anda(45,50,0.3);
        }else if((sensorLeft - PolLeft) > (sensorRight - PolRight)){
            beep();
            printf(">>> virando direita >>>>\n");
            gira(45,-50,1.0);
            anda(45,50,0.3);
        }else{
            beep();
            printf("||| seguindo em frente |||\n");
            anda(45,50,0.5);
        }
    }

    alloff();
    printf("Chegou!\n");
}

/* funcao que pega limiar*/
void pegaLimiar(){
    int aux;

    printf("Sensor sobre branco.Start.\n");
    while(!start_button());
    while(start_button());
    aux=analog(sensorBWLeft)+analog(sensorBWRight);

    printf("Sensor sobre preto.Start.\n");
    while(!start_button());
    while(start_button());
    aux+=analog(sensorBWLeft)+analog(sensorBWRight);
```

```
    limiar=aux/4;
}


/*funcao que segue a linha preta*/
void seguirLinha(){
    int sensorLeft,sensorRight;
    int sensorDistance;
    int x,y;

    pegaLimiar();
    printf("limiar = %d",limiar);
    sleep(1.5);

    x=0;
    y=0;
    sensorDistance=analog(sensorDistancePort);
    printf("Seguindo Linha\n");
    sleep(0.5);

    while(/*sensorDistance!=*/1){                   /*medir distancia para verificar obstaculo*/

        sensorLeft=analog(sensorBWLeft);
        sensorRight=analog(sensorBWRight);
        sensorDistance=analog(sensorDistancePort);
        printf("Lendo Dados\n");
        beep();
        //sleep(0.5);
        motor(0,45+x);
        motor(1,50+y);

        if(sensorLeft<limiar && sensorRight<limiar){  /* dois sensores leram brancos */
            printf("||| seguindo em frente |||\n");
            x=0;
            y=0;
        }
        else if(sensorLeft>limiar && sensorRight<limiar){  /* esquerda leu preto*/
            printf("<<< virando esquerda <<<\n");
            x=0;
            y=15;
        }
        else if(sensorLeft<limiar && sensorRight>limiar){ /* direita leu preto */
            beep();
            printf(">>> virando direita >>>>\n");
            x=13;
            y=0;
        }
        else{                   /* atravessar linha preta */
            x=0;
            y=0;
            beep();
        }
    }
    printf("Bateu\n");
    alloff();

}
```

```
/*funcao que segue parede*/
void wall(){
   int sensorLeft,sensorRight;
   int sensorDistance;
   int x,y,tmp;

   x=0;
   y=0;
   sensorDistance=analog(sensorDistancePort);
   printf("Seguindo Parede\n");
   sleep(0.5);
   printf("Lendo Dados\n");

   while(1){                  /*medir distancia para verificar obstaculo*/

      sensorLeft=digital(sensorWallLeft);
      sensorRight=digital(sensorWallRight);
      sensorDistance=analog(sensorDistancePort);

      //sleep(1.5);
      motor(0,45+x);
      motor(1,50+y);

      if(sensorLeft==0 && sensorRight==1){  /* direita bateu */
         printf("<<< virando esquerda <<<\n");
         x=0;
         y=15;
      }
      else if(sensorLeft==1 && sensorRight==0){ /* esqueda bateu */
         printf(">>> virando direita >>>>\n");
         x=15;
         y=0;
      }
      else{                  /* nao ta encostado */
         tmp=x;
         x=y;
         y=tmp;
         while(sensorLeft==sensorRight){
            printf("--- voltando ---\n");
            motor(0,45+x);
            motor(1,50+y);
            sensorLeft=digital(sensorWallLeft);
            sensorRight=digital(sensorWallRight);
         };
         //beep();
      }
   }
   printf("Bateu\n");
   alloff();

}

/* funcao que retorna cor*/
int retornaCor(){
   int aux,max, i_max;
   poke(0x1009,0x3c);   /*map spin digital output*/
```

```
max=999;
poke(0x1008,4<<0);    /*vermelho*/
sleep(1.5);
aux=analog(sensorColorPort);
printf("vermelho - %d\n",aux);
if(aux<max){
   max=aux;
   i_max=1;
}
//poke(0x1008,0);

poke(0x1008,4<<1);    /*verde*/
sleep(1.5);
aux=analog(sensorColorPort);
printf("verde - %d\n",aux);
if(aux<max){
   max=aux;
   i_max=2;
}
//poke(0x1008,0);

poke(0x1008,4<<2);    /*azul*/
sleep(1.5);
aux=analog(sensorColorPort);
printf("azul - %d\n",aux);
if(aux<max){
   max=aux;
   i_max=3;
}
//poke(0x1008,0);

poke(0x1008,4<<3);    /*amarelo*/
sleep(1.5);
aux=analog(sensorColorPort);
/*if(aux<max){
   max=aux;
   i_max=4;
}*/

poke(0x1008,0);
printf("%d\n",i_max);
sleep(1.0);

switch(i_max){
   case 1:
     printf("Vermelho\n");
   break;
   case 2:
     printf("Verde\n");
   break;
   case 3:
     printf("Azul\n");
   break;
   case 4:
     printf("Amarelo\n");
   break;
}
```

```
    return i_max;

}

void shaft(){

}

/*funcao para a tarefa final*/
void MoonWalk(){
    int sensorColor=0;

    while(sensorColor != 1){
        /*if(sensorColor == 2){ bloco verde segue em frente      talvez colocar medidor de tempo interativo
            anda(45,50,0.3);
        }*/
        if(sensorColor == 4){ /*bloco amarelo vira para a direita*/
            gira(45,-50,0.2);
        }
        else if(sensorColor == 3){ /*bloco azul vira para a esquerda*/
            gira(-45,50,0.2);
        }
        else{
            seguirLinha();
            sensorColor=retornaCor();
        }
    }

}

/* funcao que mostra os valores dos sensores */
void medirsensores(){
    while(1){
        printf("Pol Left = %d\n",analog(sensorLeftPort));
        sleep(1.0);
        printf("Pol Right = %d\n",analog(sensorRightPort));
        sleep(1.0);
        printf("Linha Left = %d\n",analog(sensorBWLeft));
        sleep(1.0);
        printf("Linha Right = %d\n",analog(sensorBWRight));
        sleep(1.0);
        printf("Color = %d\n",analog(sensorColorPort));
        sleep(1.0);
        printf("Distance = %d\n",digital(sensorDistancePort));
        sleep(1.0);
    }
}



/*menu de opcoes*/
void menu(){
    int value;
    printf("Escolha uma opcao.\n");
    sleep(2.0);
    while(!start_button()){
        sleep(0.3);
        if(knob()<45){
```

```
        value=0;
        printf("Calibrar.\n");
    }
    else if(knob()<40){
        value=1;
        printf("Seguir Luz.\n");
    }
    else if(knob()<80){
        value=2;
        printf("Seguir Linha.\n");
    }
    else if(knob()<120){
        value=3;
        printf("MoonWalk.\n");
    }
    else if(knob()<160){
        value=4;
        printf("RetornaCor.\n");
    }
    else if(knob()<200){
        value=6;
        printf("WallFollowing.\n");
    }
    else if(knob()<240){
        value=7;
        printf("Seguir Luz.\n");
    }
    else{
        value=5;
        printf("Medir sensores.\n");
    }
}

/*realiza as tarefas*/
while(start_button());
switch(value){
    case 0:
      calibrar();
    break;
    case 1:
      seguirLuz();
    break;
    case 2:
      seguirLinha();
    break;
    case 3:
      MoonWalk();
    break;
    case 6:
      wall();
    break;
    case 7:
      seguirLuz();
    break;
    case 4:
      retornaCor();
    break;
    case 5:
```

```
        medirsensores();
      break;
      default:
        printf("HELP ME! EXCEPTION ERROR! \n");
      break;
    }

}
```