

```

/*sensores fixos*/
int sensorPol=2;

int sensorBWLeft=4;
int sensorBWRight=5;

int sensorWallLeft=7;
int sensorWallRight=8;

int sensorDistancePort=9;
int sensorDistancePort2=10;

int limiar=180;
int limiarLargada=50;

int Pol=200;
int PolDesvio=5 ;

int pos0,pos1,pos2,pos3;
int DIF=255, POS;
/*motor 0 esquerda e 1 direita*/

/* leds

    amarelo  azul
    vermelho  verde*/

//void calibrarprevia();
//void previa();

/*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));
}

/* 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);
}

```

```

/* 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;
}

```

```

void pegaPol(){
    int aux,i;

    printf("Sensor em frente a luz.Start.\n");
    while(!start_button());
    while(start_button());

    aux=0;
    for(i=0;i<10;i++)
        aux+=analog(sensorPol);

    Pol=aux/10;
}

```

```

/*funcao que segue a linha preta*/
void seguirLinha(){
    int sensorLeft,sensorRight;
    int sensorDistance, sensorDistance2;
    int x,y;

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

    x=0;
    y=0;
    sensorDistance=digital(sensorDistancePort);
    sensorDistance2=digital(sensorDistancePort2);
    printf("Seguindo Linha\n");
    sleep(0.5);
}

```

```
while(sensorDistance!=1 || sensorDistance2!=1){ /*medir distancia para verificar obstaculo*/
```

```
    sensorLeft=analog(sensorBWLeft);
    sensorRight=analog(sensorBWRight);
    sensorDistance=digital(sensorDistancePort);
    sensorDistance2=digital(sensorDistancePort2);
    printf("Lendo Dados\n");
```

```
    motor(0,80+x);
    motor(1,85+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();
    }
```

```
    if(sensorDistance==1){
        motor(0,0);
        sleep(0.2);
    }
```

```
    if(sensorDistance2==1){
        motor(1,0);
        sleep(0.2);
    }
```

```
    }
    printf("Bateu\n");
    alloff();
```

```
}
```

```
/*funcao que segue parede*/
```

```
void wall(){
    int sensorLeft,sensorRight;
    int sensorDistance, sensorDistance2;
    int x,y,tmp;
```

```
    x=0;
    y=0;
    sensorDistance=digital(sensorDistancePort);
```

```

sensorDistance2=digital(sensorDistancePort2);
printf("Seguindo Parede\n");
sleep(0.5);
printf("Lendo Dados\n");

while(sensorDistance!=1 || sensorDistance2!=1){          /*medir distancia para verificar obstaculo*/

    sensorLeft=digital(sensorWallLeft);
    sensorRight=digital(sensorWallRight);
    sensorDistance=digital(sensorDistancePort);
    sensorDistance2=digital(sensorDistancePort2);

    //sleep(1.5);
    motor(0,80+x);
    motor(1,85+y);

    if(sensorLeft==0 && sensorRight==1){ /* direita bateu */
        printf("<<< virando esquerda <<<\n");
        x=0;
        y=15;
    }
    else if(sensorLeft==1 && sensorRight==0){ /* esquerda 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(0,80+x);
            motor(1,85+y);
            sensorLeft=digital(sensorWallLeft);
            sensorRight=digital(sensorWallRight);
        };
    }
}
printf("Bateu\n");
alloff();

}

/* funcao que mostra os valores dos sensores */
void medirsensores(){
    while(1){
        printf("Pol = %d\n",analog(sensorPol));
        sleep(1.0);
        printf("Linha Left = %d\n",analog(sensorBWLeft));
        sleep(1.0);
        printf("Linha Right = %d\n",analog(sensorBWRright));
        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()<80){
            value=0;
            printf("Calibrar.\n");
        }
        else if(knob()<160){
            value=1;
            printf("Previa.\n");
        }
        else if(knob()<210){
            value=3;
            printf("Teste.\n");
        }
        else{
            value=2;
            printf("Medir sensores.\n");
        }
    }

    /*realiza as tarefas*/
    while(start_button());
    switch(value){
        case 0:
            calibrarprevia();
            break;
        case 1:
            previa();
            break;
        case 2:
            medirsensores();
            break;
        case 3:
            teste();
            break;
        default:
            printf("HELP ME! EXCEPTION ERROR! \n");
            break;
    }

}

void calibrarprevia(){

    pegaLimiar();
    pegaPol();

    printf("Start para a largada.\n");
    while(!start_button());
    while(start_button());
    previa();
}

```

```

void previa(){

    int largada, previa;

    printf("LARGADA..\n");

    largada=(analog(sensorBWLeft)+analog(sensorBWRight))/2;
    while(largada > limiarLargada){
        largada=(analog(sensorBWLeft)+analog(sensorBWRight))/2;
    }

    printf("Iniciado.\n");
    beep();
    previa=start_process(corrida());
    start_process(timer(previa));
}

```

```

void timer(int previa){

    int thiefID;

    sleep(50.0);
    kill_process(previa);
    alloff();
    printf("Sprint final.\n");

    thiefID=start_process(thief());
    sleep(10.0);
    kill_process(thiefID);
    alloff();
    printf("GAME OVER.\n");
}

```

```

void corrida(){

    int livePol,parede,parede2,aux;
    aux=0;

    livePol=analog(sensorPol);
    if(livePol >= (Pol-PolDesvio)){
        motor(0,-60);
        motor(1,65);
        sleep(0.3);
    }else {

    while(livePol < (Pol-PolDesvio)){ //gira buscando luz polarizada
        printf("live = %d   outro = %d\n",livePol,(Pol-PolDesvio));
        sleep(0.2);
        motor(0,-60);
        motor(1,65);
        livePol=analog(sensorPol);
    }
    }

    sleep(0.3);
    parede=digital(sensorDistancePort);
    parede2=digital(sensorDistancePort2);

    motor(0,75);
}

```

```

motor(1,85); // vai em direcao a parede

while(parede==0 || parede2==0){ //enquanto nao bater as duas patas

    if(parede==1 && parede2==0){
        motor(0,-60);
        motor(1,65);
        sleep(0.1);
        motor(0,75);
        motor(1,85);
        aux++;
    }
    if(parede2==1 && parede==0){
        motor(0,60);
        motor(1,-65);
        sleep(0.1);
        motor(0,75);
        motor(1,85);
        aux++;
    }

    parede=digital(sensorDistancePort);
    parede2=digital(sensorDistancePort2);
    if(aux>5)
        {parede==1;parede2=1;}
}

alloff();
printf("Achou parede.\n");

sleep(0.5);
motor(0,-80);
motor(1,-85);
sleep(0.2);

alloff();
sleep(0.5);
motor(0,-60);
motor(1,65);
sleep(0.48);
alloff();

printf("Girou para a borda\n");
alloff();
sleep(2.0);

motor(0,75);
motor(1,85); // vai em direcao a parede

/*while(parede==0 || parede2==0){ //enquanto nao bater as duas patas
    if(parede==1)
        motor(0,0);
    if(parede2==1)
        motor(1,0);

    parede=digital(sensorDistancePort);
    parede2=digital(sensorDistancePort2);
}

```

```

}

alloff();
printf("Achou parede.\n");

anda(-80,-85,0.2); //da uma pequena re

gira(-60,65,0.3); // gira para a esquerda

*/motor(3,100);
seguirLinha();

anda(-60,-65,0.2);

gira(-60,65,0.4); /*baliza*/

printf("Baliza\n");
alloff(); /* espera faltar 10 sec*/
}

void teste(){
    motor(3,100);
}

void thief(){

    int parede,parede2;

    printf("Tentando coletar blocos alheios. \n");
    parede=digital(sensorDistancePort);
    parede2=digital(sensorDistancePort2);

    motor(0,75);
    motor(1,85);
    motor(3,100);

    while(parede != 1 || parede2 != 1){
        parede=digital(sensorDistancePort);
        parede2=digital(sensorDistancePort2);

        if(parede==1)
            motor(0,0);
        if(parede2==1)
            motor(1,0);

    }

    motor(0,0);
    motor(1,0);
}

```