“SULLA, que sentava inerte durante o ditato, agora digita rapidamento por alguns segundos e para, removendo completamente a carta.” R.O.R., Karel Capek.

**Resumo**

Este trabalho relata a recuperação de um equipamento que ficou abandonado no instituto federal da bahia por mais de 10 anos, sem nunca ter sido usado. Um braço robótico RD5NT, da Didactica Itália. A estrutura eletrônica não existia e a estrutura mecânica encontra-se com muitos pontos de oxidação. Ao final do trabalho, o robô foi entregue com nova estrutura de controle de potência, posicionamento dos motores, e uma adaptação técnica para o fechamento do efetuador. As estruturas novas foram montadas com componentes adquiridos pelos alunos. Além disso, o robô passou a ter nova estrutura de programação, mais moderna e flexível. Na introdução será mostrado o robô em questão e discutidos os temas introdutórios da robótica de manipulação. Na etapa de Planejamento serão mostrados os componentes escolhidos e os circuitos montados. Na etapa de Execução será mostrada a programação realizada, os circuitos montados e o estado final do braço robótico. Esse trabalho termina com uma lista de próximos passos para continuar melhorando o equipamento.

**Abstract**

This work reports the recovery of an equipment left in abandonment in Federal Institute of Bahia for more than 10 years, which was never used. A Robotic Arm RD5NT from Didactica Itália. The electronic strutucture does not exist and the mechanical struture is damaged by oxidation. At the end, the arm was returned with new power and positioning structure, and a technical adaptation for the end effector. The new structures were mounted with componentes acquired by the students. Also, the robot has now a new programming structure, modern and more flexibel. Introduction will be shown the robot in question and discussed the introductory topics of robotics manipulation. In the Planning phase will be shown the chosen components and circuits planned. Execution will shown the control programmed, the assembled circuits and the final state of the robotic arm. This work finishes with possible steps to further improve the equipment.

**Introdução**

Foi numa peça de teatro chamada Rossum’s Universal Robot, de 1921, do checo Karel Capek, que o termo Robot foi introduzido ao mundo. A palavra tem origem em Robota, que significa trabalho forçado. A peça relata um pedaço da vida de Rossum, um cientista de responsável pela criação de um composto químico cuja aplicação seria na construção de humanoides, com o objetivo de transferir todo o trabalho pesado para eles. Hoje, os robôs que conhecemos fazem trabalhos diversos, desde a complexa manipulação de objetos pesados até as mais tediosas e enfadonhas tarefas de casa.

O que se percebe é o aumento do seu uso, já que as pessoas têm menos tempo disponível para as atividades enfadonhas e repetitivas, e não querem correr riscos com atividades como a troca de peças pesadas num forno a 1200ºC, comuns em ambientes industriais, que têm ganhado competitividade com a aplicação desta tecnologia. Aliás, é nesses ambientes que as maiores vantagens destes sistemas aparecem: menores custos de trabalho, maior precisão e maior produtividade, flexibilidade moderada porém alcançável, graças às reprogramações.

Uma definição atual traria o robô como sistema ou grupo de sistemas mecatrônicos capazes de realizar trabalhos de maneira autônoma ou pré-programada. Como elemento principal desta definição temos a programabilidade. O robô como conhecemos advém do pós desenvolvimento da teleoperação (aplicada até em exploração espacial) e do CNC (controle numérico computadorizado), e logo após começaram as aplicações em que não seria possível a utilização de um homem, como a exploração submarina de longa duração, a exploração exoplanetária, o desarmamento de dispositivos explosivos e a busca em ambientes radioativos. Assim, reforça-se que as aplicações dos robôs não estão reduzindo postos de trabalho, pela substituição de um humano com a sua introdução, e sim aumentando a nossa longevidade.

O robô que foi restaurado neste trabalho é um RD5NT, da Didactica Itália. O exemplar da instituição encontrava-se sem estrutura de programação, sem estrutura eletrônica e com marcas de oxidação, apenas com o funcionamento dos servomotores das juntas e do motor do efetuador preservados, o que motivou a restauração.



Trata-se de um braço mecatrônico com 4 servomecanismos de rotação (base, ombro, cotovelo e pulso), 5 elos mecânicos e uma garra mecatrônica. Os elos são as estruturas mecânicas que se ligam entre as juntas, com função de principal de suportar a carga mecânica suportada pelo robô. As juntas que são as estruturas responsáveis pela movimentação e posicionamento e ligam dois elos ou um elo e um efetuador e podem ser do tipo rotacional ou translacional (são os servomecanismos de rotação). Já o efetuador é o dispositivo responsável por realizar a ação proposta, seja ela a soldagem de dois componentes, o suporte de uma peça, a execução de um furo ou o disparo de uma arma. Eles podem ser pneumáticos, mecatrônicos, hidráulicos ou eletromagnéticos.

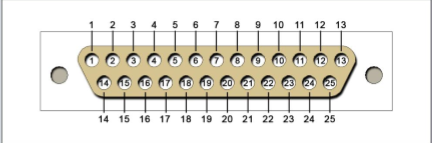
Além dos componentes citados, um robô precisa para o seu funcionamento de uma unidade de potência para o acionamento correto dos servomecanismos, e uma unidade de controle que proporcionará trajetórias condizentes com a ação a ser efetuada por ele. Para tarefas como complexas como a usinagem de peças para a indústria aeronáutica, por exemplo, o efetuador deve seguir uma trajetória planejada com altíssima precisão. No entanto, para alcançar a melhor estratégia de controle é necessário um modelo matemático do robô. Tal modelagem não foi alcançada, por restrições de tempo do projeto. Além da unidade de controle, é necessária também uma unidade de processamento, responsável por informar a unidade de controle qual posição deve ser alcançada pelo sistema, muitas vezes com restrições de tempo e velocidades bem severas. Esta unidade lê as posições (indicadas provavelmente, por um homem), planeja a movimentação, e se comunica com o espaço exterior, quando necessário.

Para as unidades de controle e processamento, foi adquirido um sistema Arduíno e para as unidades de potência foram adquiridos circuitos do tipo Ponte H (L298N). As características básicas dos componentes adquiridos serão detalhadas na parte de planejamento.

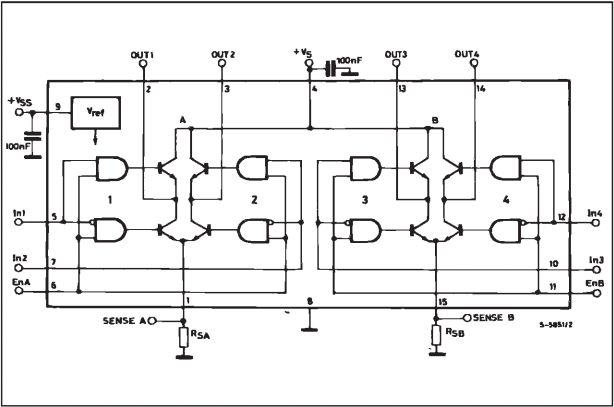
Os robôs manipuladores podem ser classiﬁcados por vários critérios, tais como sua geometria, ou estrutura cinemática, o tipo de aplicação para a qual foram projetados, a maneira pela qual são controlados, etc. O robô em questão possui as juntas de rotação numa configuração esférica, seguida por mais dois graus de liberdade de rotação e o efetuador final. A esse tipo é dado o nome RRR (como 3 rotações em sequência). Tem arranjo cinemático antropomórfico do tipo braço.

**Planejamento**

Levantado o estado atual do robô, no qual verificamos a integridade das conexões entre a porta de saída e os motores, potenciômetros de posição e chave de fim de curso, iniciamos o detalhamento do que seria necessário construir ou comprar para colocar o equipamento em funcionamento. A porta de saída tem vinte e cinco pinos dos quais oito não apresentavam conexão com o robô. As conexões identificadas foram: motor\_1, pinos 1 e 14; motor\_2, pinos 2 e 15; motor\_3, pinos 4 e 17; motor\_4, pinos 5 e 18; motor\_garra, pinos 6 e 19; Pot\_1, pino 20; Pot\_2, pino 22; Pot\_3, pino 23; Pot\_4, pino 24. Os Pinos Vcc e Terra são 13 e 25 respectivamente. O pino 7 está aterrado à carcaça. Os pinos não mencionados estão sem conexão. A figura abaixo mostra a configuração da porta do robô.

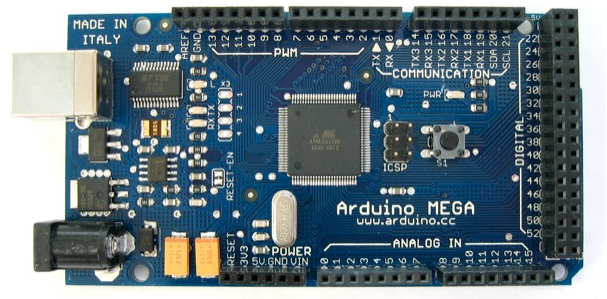


Como o servomecanismo das juntas é acionadas por motores de corrente contínua, seria necessária a construção de um circuito do tipo Ponte H, para fazê-lo movimentar-se em ambas as direções. Foi adquirida a ponte L298N, uma ponte H completa, com alimentação em 5V e pino de ativação e desativação, capaz de suprir dois motores dc com 2A ou um motor dc com 3A, ou um motor de passo. Na figura abaixo verificamos o desenho esquemático do componente.



A ponte integra duas saídas de potência. As duas saídas estão em configuração ponte. Cada ponte é conduzida através de dois gates na entrada e um pino de ativação e desativação. Quando este não tem tensão de alimentação a ponte está no modo desativada. Cada fonte deve ser ligada ao sinal de saída por um caminho bem curto. Para um controle bidirecional de um motor DC é necessário apenas uma das saídas, ou seja, é possível controlar dois motores com um componente. Além disso, foi necessário uma ponte externa de diodos, pois a carga indutiva do motor estava dificultando a própria movimentação.

Para fazer o controle de posição do servomecanismo, foi utilizando um Arduíno MEGA, uma placa de microcontrolador baseado no ATmega1280. Ele tem 54 pinos digitais de entrada/saída (dos quais 14 podem ser usados ​​como saídas PWM), 16 entradas analógicas, 4 UARTs (portas seriais de hardware), um cristal oscilador de 16 MHz, uma conexão USB, um entrada para alimentação e um botão de reset. Ele contém tudo o necessário para apoiar o microcontrolador; basta conectá-lo a um computador com um cabo USB ou ligá-lo com um adaptador AC para DC ou bateria.



Com o Arduíno foi possível ler os sinais de tensão dos potenciômetros das juntas, e acionar os circuitos de das entradas das pontes com PWM. Com a saída PWM foi possível controlar com precisão a quantidade de tensão que aciona cada motor e assim realizar o controle de posição e de fechamento da garra.

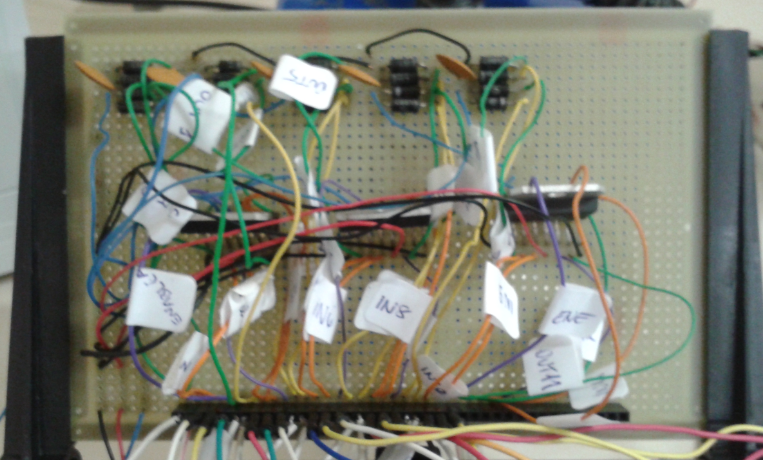
Além do material descrito, foi utilizada uma fonte de tensão variável para a alimentação dos motores, garra e circuito eletrônico.

**Execução**

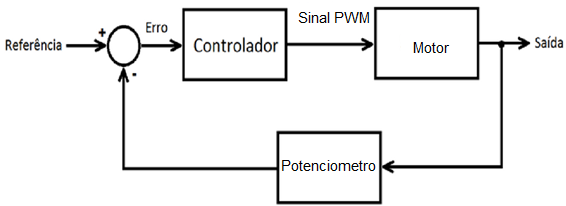
Para a execução do projeto, primeiro foi soldada uma conexão que facilita-se o acesso a porta do robô. Após, realizou-se a verificação das pontes e testou-se o acionamento dos motores direto por elas, verificando a necessidade de 3 pontes para o acionamento de todos os motores do braço antropomórfico.

Foram verificados os limites de movimentação de cada motor e associados a tensão associada fornecida por cada potenciômetro, de modo que para a posição inferior de cada junta haveria um limite de tensão inferior e para cada última posição um limite superior de tensão, sendo possível realizar uma interpolação linear entre os valores. Este mesmo mapeamento foi traduzido numa escala digital para comparação direta com o nível de tensão produzido pelo PWM do Arduíno.

As pontes foram conectadas ao Arduíno, com os pinos de entrada ligados aos pinos de PWM e os pinos de ativação ligados às saídas lógicas do microcontrolador. Após a primeira automação com mudança de direção de movimentação, verificou-se que seria necessária uma ponte de diodos para bloquear a energia acumulada nas bobinas do motor e permitir uma mudança de direção mais rápida. A foto abaixo mostra o estado final do protótipo funcional.



A partir deste ponto o trabalho tornou-se basicamente de programação. Dentro da unidade Arduíno foi construído um programa que realiza o controle de posição das juntas. Determinamos a posição desejada e aplicamos tensão nos motores até que a posição alcançada esteja de acordo com o solicitado. A posição atual é medida pelo nível de tensão do potenciômetro que é comparada com o nível equivalente à posição de referência. Depois disso o motor é acionado para modificar a sua posição girando para o lado necessário. A posição da junta é novamente calculada e o processo se repete até que a posição solicitada seja alcançada. O diagrama de blocos abaixo retrata o algoritmo utilizado.



O algoritmo de controle implementado foi do tipo PID, para o qual apenas a parte proporcional foi utilizada. A estrutura básica de cálculo é reproduzida abaixo. A estrutura na íntegra pode ser visualizada no anexo 1.

error = Setpoint[l] - Input[l];

     errSum[l] += (error \* timeChange);

     dErr = (error - lastErr[l])/timeChange;

     Output[l] = (k[l][0])\*error + (k[l][1])\*errSum[l] + (k[l][2])\*dErr;

As etapas do algoritmo descrito acima realizam os seguintes procedimentos:

- calcule a diferença entre posição desejada e posição atual do loop

- calcule o erro acumulado na execução de um loop

- calcule o diferencial do erro em um loop

- calcule e aplique a tensão de saída do loop com base nos parâmetros acima.

Com isso conseguiu-se restaurar a função básica do manipulador, que seria a função de posicionamento. O algoritmo foi implementado para cada uma das juntas rotativas, obtendo sucesso no posicionamento da estrutura em todas elas.

Além do mostrado, foi introduzida uma adaptação técnica à garra, com fins de indicar o fechamento da mesma. A garra só possuía indicador de abertura máxima, realizada por uma chave de fim de curso. A adaptação constiui-se de um circuito de toque, normalmente aberto.

**Próximos passos**

Há dois pontos cruciais que não foram visitados durante a execução deste projeto. O primeiro é a modelagem da estrutura dinâmica do robô, incluindo as juntas e os elos. Sem essa modelagem não é possível realizar a sintonização da estrutura PID desenvolvida. Sem uma estrutura de PID sintonizada não é possível obter a melhor dinâmica do robô, e deslocamentos em tempos mínimos, além de que, a máxima precisão de posicionamento não é se torna alcançável. É possível o desengripamento das juntas de rotação e substituição das molas de contrabalanceamento.

O segundo ponto crucial é o de determinação da geometria do robô. Sem isso o espaço de trabalho do mesmo é desconhecido, não sendo possível entender em qual zona deste há maior agilidade para a movimentação e reposicionamento de objetos.

Outras ações para o melhoramento futuro da programação são: desenvolvimento da comunicação do Arduíno com o Matlab, aumentando o acesso a ferramentas de identificação e controle para o sistema; Desenvolvimento de uma estrutura de cinemática inversa, para determinação automatizada das posições das juntas para uma dada posição de ação no espaço de trabalho; Desenvolvimento de um código para seguimento de instruções de programação de trajetória em modo manual, a qual tem se tornado de uso constante na indústria automotiva, com as funções de soldagem de ponto elétrico.

Construção de uma base com esteiras e sistema de câmera para a implementação de um simulador de movimentação de carga, permitindo o uso contínuo do robô, em várias disciplinas da instituição.

Pode-se realizar a construção de uma placa apropriada para o sistema de potência do robô, deixando-o independente de fonte externa, com alimentação própria.

Referências

<http://www.tech.dmu.ac.uk/~mgongora/Resources/L298N.pdf>

<http://arduino.cc/en/Main/arduinoBoardMega>

ANEXO 1

\*  Configurando array para medida dos ultimos tempos medidos

\*/

unsigned long lastTime[4];

//-------------------------------------------------------------------------------------//

/\*

  Definindo variavel para controle de inputs e outputs e setpoints

  cada linha representa o elo

\*/

  double Input[4], Output[4], Setpoint[4];

//-------------------------------------------------------------------------------------//

/\*

  Definido arrays para somatorio dos erros e ultimos erros

\*/

  double errSum[4]={0,0,0,0},lastErr[4]={0,0,0,0};

//-------------------------------------------------------------------------------------//

/\*

  configuração do array de ganho dos elos

  o numero da linha corresponde o do numero do elo a ser controlado

  a coluna corresponde a função de ganho na ordem

    1)proporcional

    2)integral

    3)derivativo

\*/

double k[4][3]={{5,0,0},{5,0,0},{5,0,0},{5,0,0}};//arrays com os valores de kp,kd,ki

//----------------------------------------------------------------------------------------//

/\*configuração do array de pinos para controle dos elo

    linhas representam o elo a ser controlado e colunas as a funcionalidade do tipo

    controle do elo 1 (linha 0) o pulso:

      pino potenciometro 1 será 1

      pino do input 1 do motor 2 será 6

      pino do input 2 do motor 2 será 7

      pino do enable motor 2 será 33

    controle do elo 2 (linha 1)o cotovelo:

      pino potenciometro 2 será 2

      pino do input 1 do motor 3 será 8

      pino do input 2 do motor 3 será 9

      pino do enable motor 3 será 35

    controle do elo 3 (linha 2)o ombro:

      pino potenciometro 3 será 3

      pino do input 1 do motor 4 será 10

      pino do input 2 do motor 4 será 11

      pino do enable motor 4 será 37

    controle do elo 4 (linha 3)o ombro:

      pino potenciometro 4 será 4

      pino do input 1 do motor 5 será 12

      pino do input 2 do motor 5 será 13

      pino do enable motor 5 será 39

\*/

int pino[4][4] = {{1,6,7,33},{2,8,9,35},{3,10,11,37},{4,12,13,39}};

//---------------------------------------------------------------------------------------//

/\*configuração de valores medidos

  para leitura minima e maxima da entrada analogica

  Alimentação do fonte Vs=4.9V

  Alimentação da fonte Vss=4.9V

  vetor chamado de limites

  1)linhas indicam o potenciometro

  2)1ª Coluna indica valor maximo de deslocamento angular em relação a referencia

  3)2ª Coluna indica valor minimo de setpoint medido pelo arduino

  4)3ª COluna indica valor maximo de setpoint medido pelo arduino

\*/

double limites[4][3]={{360,0,1023},{293,66,900},{96,376,648},{291,101,928}};

//----------------------------------------------------------------------------------//

unsigned long now;

double timeChange;

double error;

double dErr;

double lasttime;

int condicao;

void setup(){

  Serial.end(); //eliminando qualquer configuração serial anterior

  //configurando pinos de entrada

  for(int i = 0;i<4;i++){

    pinMode(pino[i][0],INPUT);

    //configurando pinos de saída

    for (int j=1;j<4;j++){

      pinMode(pino[i][j],OUTPUT);

    }

  }

  pinMode(41,OUTPUT);   //Enable do motor 1 - garra

  pinMode(43,OUTPUT);   //Input 2 do motor 1

  pinMode(45,OUTPUT);   //Input 1 do motor 1

  Serial.begin(57600); //configurando entrada serial

  //Verificar estado inicial que se quer da garra

  condicaogarra();

  //Impondo a codição angular desejada usando a funcao converte angulo detalhes abaixo

  for(int i=0;i<4;i++){

    Setpoint[i] = converteangulo(i);

  }

  //iniciando os valores do tempo dos lastTime medidos

  for(int j=0;j<4;j++){

    lastTime[j]=millis();

  }

}

//----------------------------------------------------------------------------------//

void loop()

{

  //laço para leitura de todos os potenciometros

  Input[0] = analogRead(A1); //potenciometro 1

  Input[1] = analogRead(A2); //potenciometro 2

  Input[2] = analogRead(A3); //potenciometro 3

  Input[3] = analogRead(A4); //potenciometro 4

  Serial.println();Serial.println("Diferenca de setpoint");

  Serial.print(Setpoint[0]-Input[0]);Serial.print(" |  ");

  Serial.print(Setpoint[1]-Input[1]);Serial.print(" |  ");Serial.print(Setpoint[2]-Input[2]);

  Serial.print(" |  ");Serial.print(Setpoint[3]-Input[3]);Serial.println(" |  ");Serial.println();

  //

  for(int l=0;l<4;l++){

     now = millis();

     timeChange = (double)(now - lastTime[l]);

     error = Setpoint[l] - Input[l];

     errSum[l] += (error \* timeChange);

     dErr = (error - lastErr[l])/timeChange;

     Output[l] = (k[l][0])\*error + (k[l][1])\*errSum[l] + (k[l][2])\*dErr;

     //Output[elo] = abs(Output[elo]);

     lastErr[l] = error;

     lastTime[l] = now;

     if(Output[l]>0){

       Output[l]=abs(Output[l]);

       Output[l]=constrain(Output[l],0,1023);

       Output[l]=map(Output[l],0,1023,0,255);

       analogWrite(pino[l][1], 0); //acionando input 1 desde que condição de sentido favoravel

       analogWrite(pino[l][2], Output[l]); //acionando input 2 desde que condição de sentido favoravel

       digitalWrite(pino[l][3],HIGH); //habilitando enable

     }

     else{

       Output[l]=abs(Output[l]);

       Output[l]=constrain(Output[l],0,1023);

       Output[l]=map(Output[l],0,1023,0,255);

       analogWrite(pino[l][1], Output[l]); //acionando input 1 desde que condição de sentido favoravel

       analogWrite(pino[l][2], 0); //acionando input 2 desde que condição de sentido favoravel

       digitalWrite(pino[l][3],HIGH); //habilitando enable

     }

  }

  Serial.println();Serial.println("Valores do PID");

  Serial.print(Output[0]);Serial.print(" |  ");

  Serial.print(Output[1]);Serial.print(" |  ");Serial.print(Output[2]);

  Serial.print(" | ");Serial.print(Output[3]);Serial.println(" |  ");

  Serial.println();

  if (Serial.available())

  {

    //Impondo a codição angular desejada usando a funcao converte angulo detalhes abaixo

    for(int i=0;i<4;i++)

      {

      Setpoint[i] = converteangulo(i);

      }

  }

}

//----------------------------------------------------------------------------------//

/\*

  Funcao que fornece o deslocamento angular desejado

\*/

double converteangulo(int elo){

  Serial.print("Digite o deslocamento para ");

  switch(elo){

   case 0:

     Serial.print("o Pulso 0-");

     break;

   case 1:

     Serial.print("o Cotovelo 0-");

     break;

   case 2:

     Serial.print("o Ombro 0-");

     break;

   case 3:

     Serial.print("a Base 0-");

     break;

  }

  Serial.print(limites[elo][0]);Serial.println(" em graus:");

  while(!Serial.available());

  double x =Serial.parseInt();

  double deslocamento = map(x,0,limites[elo][0],limites[elo][1],limites[elo][2]);

  return(deslocamento);

}

void condicaogarra(){

  Serial.print("Deseja-se garra aberta( digite 1) ou fechada(digite 2): ");

  while(!Serial.available());

    condicao=Serial.parseInt();

    if(condicao==1){

        abrirgarra();

    }

    else{

        fechargarra();

    }

  Serial.println();

}

void abrirgarra(){

  digitalWrite(41,HIGH);//Enable do motor 1

  digitalWrite(43,HIGH);//Input1 do motor 1 alto

  digitalWrite(45,LOW);//Input2 do motor 2 baixo

}

void fechargarra(){

  digitalWrite(41,HIGH);//Enable do motor 1

  digitalWrite(43,LOW);//Input1 do motor 1 baixo

  digitalWrite(45,HIGH);//Input2 do motor 2 alto

}

//----------------------------------------------------------------------------------//