



## TELEOPERAÇÃO ROBÓTICA: FOMENTO DE COMPETÊNCIAS ARTICULADAS PARA ENGENHARIA DE CONTROLE E AUTOMAÇÃO

**Rodrigo A. P. Andrade** - rodrigoapandrade@gmail.com

**André Del Bianco Giuffrida** - andre.giuffrida@usp.br

**Pedro Henrique Oliveira Queiroz** – phoq1@hotmail.com

**Fernando Olímpio Gomes Pereira** – fernando-mgbr@hotmail.com

**Marlon José do Carmo** – marloncarmo@ieee.org

Centro Federal de Educação Tecnológica de Minas Gerais, Campus III – Leopoldina  
Rua José Peres, 558

36700-000 – Leopoldina – Minas Gerais

**Resumo:** Este documento tem o intuito de criar e implementar um braço robótico capaz de ser controlado de qualquer parte do mundo através da internet, para este projeto de teleoperação robótica, vai ser usado um Robô Industrial da ABB, modelo IRB 2400/10 do Laboratório de Otimização de Processos de Fabricação (LOPF), pertencente ao Núcleo de Manufatura Avançada (NUMA) da USP de São Carlos – SP. O software vai ser desenvolvido em java, o que deve fazer a parte de programação mais fácil envolvendo acesso a porta serial, porta USB e transporte de dados através do protocolo TCP / IP

**Palavras-chave:** Teleoperação, Robótica, Sockets, Protocolo TCP/IP, JAVA

### 1. INTRODUÇÃO

Desde os tempos mais primórdios, o ser humano tem feito uso de ferramentas que estendam seu alcance com o objetivo de aumentar a sua destreza seja qual for a aplicação. Os homens do tempo das cavernas, por exemplo, utilizavam galhos de árvore para cavar covas, para fazer fogo, para extrair raízes que serviam como alimento, ou até para servir de lança, seja para caça ou pesca. Recentemente, temos os ferreiros como exemplo que utilizavam e utilizam de ferramentas que facilitam o manuseio de objetos quentes. É claro que existe certa limitação quanto ao manuseio desse objeto quando comparado com uma mão humana. Essa limitação é imposta pela natureza da ferramenta utilizada. A teleoperação é uma técnica particularmente interessante, pois ela permite que um determinado trabalho seja realizado remotamente, mas ainda sim controlado pelas próprias mãos, em circunstâncias onde a ação direta não é possível, inadequada ou em situações onde é impossível estabelecer rotinas para a operação, seja pela sua imprevisibilidade, grau de risco, ou complexidade (Vertut e Coiffet, 1985).

De acordo com a RIA – Robotic Industries Association, fonte oficial de estatísticas do

Realização:



Organização:



o ENGENHEIRO  
PROFESSOR E O  
DESAFIO DE EDUCAR



setor nos Estados Unidos, em 2008 havia cerca de 182 mil robôs, número superado apenas pelo Japão (Estatística Mundial de Robôs).

O Brasil tem apenas quatro mil robôs. “Nossa estatística pode estar subestimada, porque só temos dados completos do Brasil a partir de 2004, mas de qualquer forma, o grau de automação ainda é muito baixo” (Litzenberger 2009, p.57). Muitos desses robôs presentes no Brasil, não são teleoperados, ou seja, exige a presença de um operador próximo ao robô para ser devidamente operado. E mais do que óbvio que com o passar do tempo esses robôs vão ficando ultrapassados, e os preços para atualizá-los, são exorbitantes.

O Robô IRB 2400/10 da ABB, é um ótimo exemplo. Possuindo um sistema fechado, onde só é possível operá-lo fazendo o uso de seu Joystick denominado Teach Pendant, é um tipo de robô muito comum nas indústrias brasileiras.

Conforme a seção anterior, muitos desses robôs presentes no Brasil, não são teleoperados, ou seja, exige a presença de um operador próximo ao robô para ser devidamente operado. E mais do que óbvio que com o passar do tempo esses robôs vão ficando ultrapassados, e os preços para atualizá-los, são exorbitantes.

O Robô IRB 2400/10 da ABB, é um ótimo exemplo. Possuindo um sistema fechado, onde só é possível operá-lo fazendo o uso de seu Joystick denominado Teach Pendant, é um tipo de robô muito comum nas indústrias brasileiras. Atualmente com a globalização em crescimento exponencial, a disputa pelo mercado consumidor, faz com que as empresas a cada dia produzam produtos novos e diferenciados. Para isso, é comum a mudança do layout do chão de fábrica, a calibração e reprogramação de robôs para se adaptar ao novo layout e consequentemente novos produtos. Um tipo comum de problema que ocorre, é com relação à reconfiguração realizada nesses robôs, pois é necessária a presença real do técnico/engenheiro. O que pode levar muito tempo e dinheiro, pois nem sempre poderá existir um profissional apto e disponível naquele local e momento. Portanto imaginemos a seguinte situação:

“Certa montadora no Brasil possui trinta robôs distribuídos em seis plantas, em locais distintos e afastados. A montadora dispõe de três engenheiros especializados nesses robôs industriais. É chegado o segundo trimestre, e os novos modelos já estão sendo produzidos, porém os 30 robôs precisam urgentemente ser reconfigurados em alguns parâmetros internos e realinhar os pontos de soldagem, pois a configuração anterior fez com que as peças tivessem problemas no setor de qualidade. Acontece que a distância física que existe entre as plantas, impede que os engenheiros reconfigurem todos os robôs a tempo, fazendo com que a montadora pare a produção e perca dinheiro”.

E se fosse possível realizar essas alterações, sem sair do escritório? O trabalho proposto é direcionado para esses tipos de robôs e soluciona esse e outros tipos de problemas, pois em muitas das vezes, a presença de um operador no local de atuação do robô, nem sempre é possível, seja por insalubridade, ou periculosidade.

No âmbito do ensino, a teleoperação é extremamente útil, pois o mesmo robô, uma vez feito teleoperado, poderá servir para ensinar a alunos de qualquer lugar do mundo, sem a necessidade de sair de sua própria sala de aula. É o ensino à distância ganhando forças no ramo da robótica.

A principal motivação deste trabalho é o desenvolvimento de um sistema (baseado em hardware, firmware e software) que permita a teleoperação do Robô IRB 2400/10 da ABB.

A teleoperação é definida como o “controle remoto do movimento realizado em um modo contínuo e direto de um operador sobre uma máquina teleoperada, que está situada a uma



distância do operador com o propósito de executar diversas tarefas” (Álvares e Romariz 2002, pp. 122-126).

O trabalho proposto e realizado é o início de um projeto maior, que terá sequência no Laboratório de Otimização de Processos de Fabricação, da USP de São Carlos. Uma vez que o projeto tenha sucesso com o Robô descrito acima, o mesmo também será aplicado em outros robôs e centros de usinagem.

Esse trabalho é composto por:

- Estudo da possibilidade de teleoperação do robô descrito;
- Projeto e construção de duas placas de teleoperação baseadas em placas de aquisição de dados USB, utilizando PIC 18F4550, sendo uma placa para o Controlador e outra para o Teach Pendant (TP);
- Desenvolvimento dos firmwares das placas do Controlador e do TP de acordo com o projeto acima citado.
- Projeto e construção do software em JAVA, com acesso USB, Serial e Internet para transmissão dos dados.
- Desenvolvimento de interface para leitura e transmissão dos dados do Robô via porta serial.

## 2. DESCRIÇÃO DO ROBÔ

“Um robô industrial é um manipulador multifuncional reprogramável projetado para mover materiais, peças ferramentas ou dispositivos especiais, através de movimentos variáveis programados, possibilitando a realização de uma variedade de tarefas.” (Estatística Mundial de Robôs). O que distingue um robô de outros dispositivos de movimentação é sua habilidade de ser reprogramado para o cumprimento de tarefas distintas. A maioria dos robôs modernos possui a capacidade de armazenar diversos programas na memória permitindo, com um simples apertar de um botão, selecionar o programa desejado para a realização de uma determinada tarefa.

O robô é composto por duas partes principais: caixa de controle (controlador) e manipulador.

A Interface Homem-Máquina é realizada através do teach pendant e/ou do painel de operação, localizado na parte frontal do controlador.

O controlador modelo S4C contém toda a eletrônica responsável pela operação do manipulador e dos demais equipamentos periféricos. A Fig. 1 indica a localização dos vários componentes da caixa de controle (Manual do Produto – IRB2400).

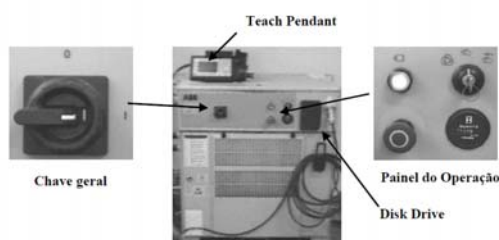


Figura 1. Componentes do controlador S4C



Dentre os componentes da figura acima, destacam-se os dois dispositivos para comunicação entre o robô e o operador:

- Teach pendant
- Painel de operação

A Figura 2 apresenta a principal interface entre o operador e o robô. Ela é responsável por enviar alguns sinais de segurança e também por permitir a manipulação manual e automática, dentre outras características citadas a seguir.

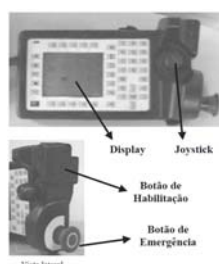


Figura 2. Teach Pendant

### 3. A COMUNICAÇÃO ENTRE O ROBÔ E O TEACH PENDANT

A comunicação do Robô IRB 2400 com o computador é extremamente limitada, pois a versão S4C do controlador, não possui um protocolo de comunicação aberto. O que permitiria algumas funções básicas através da comunicação serial EIA-232 como, por exemplo, a movimentação do braço robótico, salvar, editar e carregar arquivos através de um computador. Sendo assim, a primeira vista, é inviável a operação à distância de um robô com essas características. Como a proposta é deslocar o Teach Pendant, resta saber quais são as ações tomadas por ambas às partes quando a comunicação física é interrompida.

O ato de deslocar o Teach Pendant é fazê-lo se comunicar com o controlador, sem estar conectado fisicamente, utilizando para isso, conversores, placas de aquisição, computadores e outros periféricos.

Em um projeto de teleoperação envolvendo um robô com protocolo de comunicação fechado, existem grandes riscos de a teleoperação não ser bem sucedida, seja pela limitação natural do sistema, ou imposta pelos projetistas. Portanto faz-se necessário realizar testes passo a passo para assegurar a viabilidade do projeto.

#### 3.1. Entendendo a comunicação

Para entender como funciona a comunicação do robô, vários testes foram realizados. O primeiro teste foi executado com o robô ligado e funcionando no modo manual. O cabo que liga o TP ao controlador foi desconectado. O resultado foi uma parada imediata do robô. Ao reconectar o cabo, percebeu-se que apareceu um erro na tela do TP informando que a conexão havia sido perdida e a parada de emergência ativada. Após religar os motores, o robô



continuou funcionando normalmente. Esse teste indica que não é necessário reinicializar o controlador, caso o cabo se desconecte.

O cabo de conexão do Teach Pendant é formado por 10 fios:

- 4 fios para liberar os freios
- 4 fios para parada de emergência
- 2 fios de comunicação com padrão EIA 485

Os computadores convencionais, não possuem suporte ao tipo de padrão EIA 485, portanto dois conversores EIA-485/EIA-232 serão utilizados para a comunicação com os PCs. Um teste extremamente importante, parte do princípio em conectar os conversores entre si para ver se o robô funciona.

O esquema de ligação segue na Figura 3.

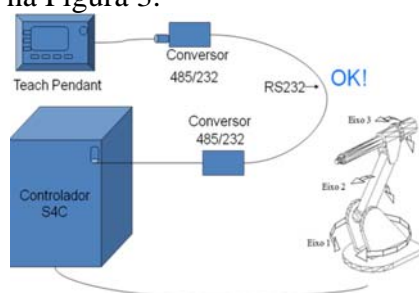


Figura 3. Teste utilizando os conversores EIA-485

O Sucesso da ligação dos dois conversores EIA485/EIA232 entre si, mostra que é viável trabalhar com o padrão EIA232 que será utilizado para levar o sinal até o computador. Vale lembrar que os outros oito fios de segurança, não trabalham com o mesmo padrão, portanto terão que ser transportados de maneira diferente, e no início dos testes eles foram mantidos conectados fisicamente entre o Teach Pendant e o Controlador, e somente os dois fios de comunicação passaram pelos testes. O tratamento dos sinais dos oito fios será discutido mais a frente, e por convenção os fios não estão sendo mostrados nas figuras.

### 3.2. Inserindo um PC no caminho

O próximo passo é ligar cada um dos dois conversores em um PC com duas portas seriais como segue na Figura 4:

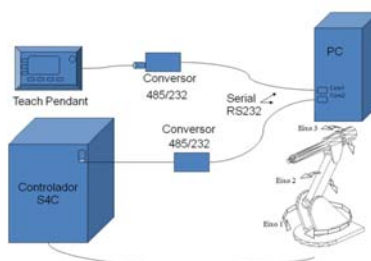


Figura 4. Teste utilizando um PC com duas conexões seriais

O resultado desse teste, se executado com sucesso, demonstra se é possível utilizar um PC para transportar os dados. O Computador inicialmente servirá apenas como uma bridge para os sinais.





Utilizando o software Labview® (Site de apresentação do Software Labview), famoso programa para criação de instrumentos virtuais diversos, uma Virtual Instrument (VI) foi criada, a fim de ler os dados de uma porta serial e escrever na outra e vice-versa. Porém quando se adiciona um computador no meio da transmissão, alguns conceitos devem ser observados. Um software para ler e reproduzir fielmente um dado conectado a porta serial, deverá estar configurado exatamente com o mesmo Baud Rate utilizado pelo TP. O Baud Rate é a velocidade na qual um equipamento consegue transmitir os dados, e a configuração incorreta desse valor, impossibilita a comunicação.

Realizando uma pesquisa no User Guide (Guia do usuário do Robô – IRB2400) e no Product Manual (Manual do Produto – IRB2400) do robô, esse valor em nenhum momento é citado. Porém em um determinado capítulo e como foi citado no Capítulo 2 desse trabalho, o autor cita que a máxima velocidade que as Inputs e Outputs (IO) seriais do controlador trabalham, é 19200 bauds. Portanto, na base da tentativa e erro, configurou-se o VI com valores de 19200, 9600, 4800, entre outros, porém sem sucesso.

Pensou-se até que o sinal poderia ser criptografado para proteger contra alterações, mas logo essa ideia foi descartada, porém acreditava-se que não seria possível ler e reproduzir os sinais por alguma limitação imposta pelos desenvolvedores. No entanto, fazendo uso de um osciloscópio, pôde-se diagnosticar o valor correto do Baud Rate. E o valor aferido foi de 38400 bauds, o que indica que a informação da frequência dada pelo Product Manual, não se enquadrava a conexão do Teach Pendant.

Utilizando o VI dito acima, pela primeira vez a mensagem de “conexão perdida” que sempre era mostrada na tela, deu lugar as mensagens de inicialização do robô. Nesse momento a teleoperação começava a tomar forma. Mas em um determinado momento, a comunicação simplesmente travou e o robô teve de ser reiniciado.

Esse problema curioso intrigou bastante o grupo e após certo tempo foi solucionado. O momento no qual a comunicação trava, é o momento cujo ambos trocam uma grande quantidade de informações. E o buffer da serial do PC é limitado e acabou “transbordando” e os dados enviados de seus remetentes, não foram entregues ao destinatário. E como nesse caso não existe sinal de aviso de chegada de mensagem, ambos ficaram esperando a resposta um do outro.

Como esse problema é causado pelo teste em um PC com duas seriais, a solução foi pular o teste com apenas um PC e testar diretamente a comunicação entre dois PCs em rede.

### **3.3. Teleoperação usando dois PCs em rede**

Utilizando de um software chamado TCPCom (Site do desenvolvedor do Software TCP-COM), (que tem a propriedade de ler/enviar os dados de/para uma porta serial, e enviar e receber via TCP/IP para/de outro computador que faz uso do mesmo software) dois computadores ligados em uma rede local, foram configurados como mostra a Figura 5.

Configurando o software do PC do controlador como Servidor e o software do PC do TP como Cliente, tudo estava pronto para o teste de teleoperação. E pela primeira vez foi possível teleoperar o robô com sucesso. O problema de buffer não existe neste caso, pois apenas uma porta serial é utilizada em cada PC.

Com o sucesso desse último teste, veio finalmente a certeza que era possível teleoperar o robô de qualquer lugar do mundo. Porém faz-se necessário também, transportar os sinais dos outros oito fios descritos no início deste capítulo. Para isso, duas placas de aquisição foram projetadas e desenvolvidas. Uma placa acompanhará o Teach Pendant, junto com um



conversor EIA-485. A outra placa será conectada ao controlador juntamente com o conversor restante.

#### 4. PLACAS DE TELEOPERAÇÃO

As placas de aquisição são responsáveis por ler os dados de segurança entre o TP e o Controlador. Para transportar esses dados fielmente, um estudo acerca dos sinais nos dez fios foi feito.

A Figura 5 representa o esquema elétrico de sinais do TP. Como se pode perceber, dez fios são responsáveis por todas as informações trocadas entre o Teach Pendant e o Controlador. Os sinais TP e TP-N, são responsáveis pela troca de informações entre o controlador e o robô, onde os dados que chegam ao TP, indicam as posições dos eixos do robô que serão exibidas na tela, ou alguma mensagem. Todos os testes feitos anteriormente, se referiam ao transporte desse sinal. Os outros oito fios são responsáveis por transportar os sinais de segurança, Enable Device e Emergency Stop.

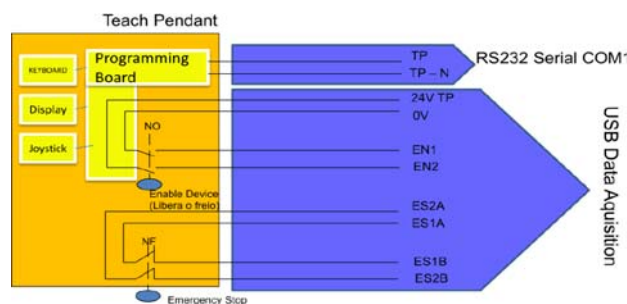


Figura 5. Esquema elétrico de comunicação do Teach Pendant

Como visto em II, o botão de Enable Device (Botão de Habilitação) serve para liberar os freios dos motores quando o robô estiver no modo de operação manual <250mm/s ou 100%. E quando acionado, o robô poderá ser levado para a condição MOTORS ON. A liberação deste botão provocará a parada do robô e o acionamento dos freios. Já o botão de Emergency Stop (parada de emergência), pára imediatamente o robô quando é pressionado, independente do modo ou estado do sistema. Este botão permanece apertado e deve ser retornado à posição original para que o robô volte à condição MOTORS ON.

Os oito fios restantes são responsáveis por transportar esses sinais de segurança. Os sinais de 24V TP e 0V são responsáveis pela alimentação do TP. A diferença de potencial entre os sinais EN1 e EN2 indicam para o controlador, se o botão de Enable Device foi (ou não) acionado. Os sinais ES2A e ES1A, possuem entre si, ddp de 24V que são devolvidas para ES2B e ES1B respectivamente, caso o Emergency Stop não esteja pressionado, pois o botão é normalmente fechado.

Levando em consideração que esses sinais devem ser levados para lugares distantes, faz-se a necessidade de fazer a aquisição dos mesmos, através de uma placa. Essas placas de aquisição precisam ler os dados e enviar ao PC, e receber os dados do PC e escrever na saída da placa. A partir dessa indigência, começou-se a projetar duas placas de aquisição de dados. Uma para o Teach Pendant, e outra para o Controlador. Essas placas, foram projetadas utilizando como base um microcontrolador da Microchip, modelo PIC18f4550. A característica principal desse PIC é a comunicação via USB com o Computador (DataSheet do PIC 18F4550).



Essas placas serão responsáveis por ler os sinais de ambos os lados e enviar para o computador ao qual estiverem conectadas. Esses computadores por sua vez, através do software de teleoperação, empacotarão esses dados, junto com os sinais lidos da porta serial, e fará o envio via TCP/IP para o servidor, que encaminhará essas mensagens recebidas para o destinatário.

O projeto da placa de aquisição foi feito baseado em uma placa feita pela própria Microchip (PICDEM™ FS USB Demo Board) (Especificações da Placa Microchip (PICDEM™ FS USB Demo Board)), para aquisição de dados.

#### **4.1. Placa de Teleoperação – Teach Pendant**

A placa de Teleoperação do TP será responsável pelas seguintes tarefas:

- Fornecer alimentação de 24 v para o Teach Pendant
- Comunicar com o PC Via USB
- Ler, empacotar e enviar ao PC o estado de cada um dos seguintes Botões: Enable Device, Emergency Stop, Motors On
- Receber e desempacotar os sinais do PC e acionar os LEDs que indicarão o estado de cada uma das Ações: Enable Device, Emergency Stop, Motors On

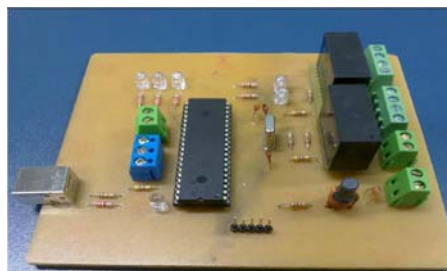


Figura 6. Placa de Teleoperação - Teach Pendant

#### **4.2. Inserindo um PC no caminho**

O projeto da Placa de Teleoperação do Controlador é um pouco mais simples que a do TP, pois além de não necessitar de fonte externa ela funciona exatamente como descrita na Figura 21 que representa o esquema elétrico de comunicação do TP.

Portanto a Placa de Teleoperação do Controlador será responsável pelas seguintes tarefas:

- Comunicar com o PC Via USB
- Ler, empacotar e enviar ao PC o estado da lâmpada sinalizadora do Motors On.
- Receber e desempacotar os sinais do PC e chavear os relés referentes aos seguintes sinais do Teach Pendant: Enable Device, Emergency Stop, Motors On
- Esses valores recebidos, do Enable Device e Emergency Stop, também deverão retornar ao PC do TP para acionar os LEDs indicadores.







Figura 7. Placa de Teleoperação - Controlador

## 5. SOFTWARES E FIRMWARES DE TELEOPERAÇÃO

O projeto de teleoperação robótica, utiliza de placas microcontroladas para fazer a aquisição dos dados, e de computadores para enviar e receber os dados via internet, portanto é extremamente necessário, o desenvolvimento de Softwares e Firmwares específicos para o projeto.

Como se pode perceber, as palavras software e firmware estão no plural. Isso quer dizer que estão sendo desenvolvidas duas versões de cada. O software será dividido entre duas versões de Clientes e um Servidor. Utilizando a API Java.net (Descrição da API Java.net). Um dos clientes ficará no computador onde o Teach Pendant está conectado e o outro ficará no computador cujo Controlador se encontra. O servidor também ficará no PC do Controlador.

Os firmwares por pertencerem a placas fisicamente diferentes, também são distintos e compatíveis com seus propósitos.

Os Softwares responsáveis pela teleoperação foram desenvolvidos utilizando a linguagem Java.

A linguagem Java possui um grande benefício quando comparado a outras plataformas de desenvolvimento. O Software uma vez desenvolvido poderá ser executado em diversas plataformas sob uma mesma compilação, ou seja, não é necessário reescrever ou nem mesmo recompilar o código fonte como é comum em outras linguagens.

Porém, existem algumas desvantagens. Pois fazer comunicação diretamente com o hardware, torna a vida do programador um pouco mais árdua.

A teleoperação de um sistema legacy como o do robô, só é possível se reproduzirmos exatamente aquilo que for lido, ou seja, não é possível acessar o controlador e operar o robô somente utilizando um Computador, pois para essa versão do robô, não é disponibilizado um protocolo de comunicação com as funções básicas do robô. É necessário fazer a aquisição dos sinais do cabo do TP e através de dois computadores conectados na internet, reproduzir os sinais (respeitando o padrão de comunicação de cada sinal) para o controlador e vice-versa. Portanto o software de teleoperação é um fator fundamental nesse processo.

Para que a teleoperação ocorra com sucesso, é necessário que o operador, enxergue exatamente o que está acontecendo e com o menor atraso possível, para aumentar a sensação de presença.

O primeiro passo, para o projeto do software de teleoperação é especificar exatamente o papel a ser desempenhado por ele.

O cabo que liga o Teach Pendant e o Controlador, possui 10 fios, sendo dois deles, responsáveis pela troca de dados entre o Teach Pendant e o controlador, e os outros oito, são responsáveis por levar os sinais de segurança Enable Device e Emergency Stop, além de alimentar o TP com 0 e 24V.

O Padrão de comunicação dos dois primeiros sinais é o RS485, que fazendo uso de um conversor RS485/RS232 é possível conectarmos a um computador que possua uma conexão serial. Portanto o primeiro desafio no campo do desenvolvimento do software foi trabalhar com a comunicação serial em Java.



## 5.1. Comunicação Serial em JAVA

A Sun e demais empresas envolvidas no desenvolvimento Java desenvolveram algumas API's para facilitar o trabalho, como é o caso da API RXTX (Descrição da API RXTX) para a comunicação tanto serial quanto paralela, podendo também trabalhar via USB como será também o caso nesse projeto.

A API RXTX é baseada na API Javacomm (Java Communications API) distribuída pela própria Sun, com a seguinte vantagem: Essa API é portátil para Linux, Windows e Mac, o que garante a vantagem de construir um software que roda em qualquer plataforma.

A placa de aquisição de dados, que transporta os sinais do Enable Device, Emergency Stop e Motors On, possui comunicação USB, utilizando no firmware a classe CDC, faz com que a placa de teleoperação seja vista como uma conexão serial pelo PC ao qual estiver conectada. Essa tarefa facilita a vida do desenvolvedor do software, uma vez que o ele já faz leitura serial utilizando a API RXTX.

## 5.2. A comunicação entre os Softwares

As informações, tanto das placas de teleoperação quanto dos sinais de comunicação, precisam ser enviadas do TP para o Controlador e vice-versa. O elo entre os processos do servidor e dos clientes (PC-TP e PC-Controlador) é o socket. Ele é a “porta” na qual os processos enviam e recebem mensagens. Portanto, socket é a interface entre a camada de aplicação e a de transporte dentro de uma máquina. Inicialmente o cliente deve se comunicar com o servidor da seguinte maneira: O servidor determina previamente uma determinada porta e com um socket, fica aguardando conexões nesta. Já o cliente, através do IP do servidor e da porta previamente estabelecida, solicita uma conexão com este servidor.

Por sua vez, o servidor aceita a conexão gerando um novo socket em uma determinada porta do lado do servidor criando um canal de comunicação entre o cliente e o servidor, permitindo assim a liberação do socket inicial para que este possa ser chamado por outros clientes.

A API Java.net possui um conjunto de funções bem conhecidas que realizam essas tarefas de acesso. Essas bibliotecas implementam o código necessário internamente, e disponibilizam uma interface padrão para que as aplicações Java possam executar estas tarefas.

Para identificação, os Clientes recebem um nome de Remetente e do seu respectivo Destinatário.



Figura 8. GUI inicial do Software de Teleoperação



### 5.3. Firmwares de Teleoperação

Os firmwares desenvolvidos para as placas de teleoperação foram projetados na linguagem C utilizando a IDE MPLAB. O compilador utilizado é o C18 da Microchip (mesma desenvolvedora do MPLAB), próprio para programação C em microcontroladores PIC da série 18. Este firmware utiliza a CDC (Class Device Communication), que apesar da conexão ser USB, emula a conexão como serial, portanto ao conectar ao PC, a placa é vista como se estivesse conectada a porta serial.

Com a utilização do software ISIS, foi possível simular essa conexão USB do PIC com o PC. E a simulação foi bem sucedida para ambas as placas, seguidas de seus respectivos firmwares.

As mensagens enviadas variam de acordo com cada placa. A placa do TP envia as informações das novas posições dos relés que reproduzem no controlador, o sinal do Teach Pendant. Abaixo seguem algumas possíveis mensagens a serem enviadas do TP para o Controlador:

```
||SED0||SES0||SMO0||
```

```
||SED1||SES0||SMO0||
```

```
||SED0||SES0||SMO1||
```

Onde:

- SED – Switch Enable Device
- SES – Switch Emergency Stop
- SMO – Switch Motors ON

Essas mensagens são armazenadas no buffer output\_buffer e como as mensagens sempre possuem o mesmo tamanho, as posições 5, 11 e 17, indicam os valores respectivos dos estados dos Switchs acima indicados. Quando enviadas, essas informações ao percorrerem todo o caminho, e chegarem ao PIC, a mesma ficará armazenada na variável input\_buffer, onde a mesma numeração (5, 11 e 17), serão utilizadas para chavear os relés que substituem os Switchs do TP.

A placa do Controlador, após chavear os relés, devolve ao PC, uma mensagem contendo os valores alterados, tanto do ED e do ES, e também o valor do LED indicador do MO.

Abaixo seguem algumas possíveis mensagens a serem enviadas do Controlador para o TP:

```
||LED0||LES0||LMO0||
```

```
||LED1||LES0||LMO0||
```

```
||LED0||LES0||LMO1||
```

Onde:

- LED – Led Enable Device
- LES – Led Emergency Stop
- LMO – Led Motors ON

Como visto, as posições das variáveis que alteram os valores dos LED, são as mesmas que as do Firmware do TP.



## 6. CONCLUSÃO

Durante a execução desse trabalho, foi possível determinar a possibilidade de teleoperar o Robô IRB 2400/10 da ABB. Este robô que possui protocolo de comunicação fechado, onde a solução encontrada para o seu manuseio foi deslocar o TP utilizando placas de aquisição, conversores EIA-485/EIA-232, e computadores para enviar através da rede, as mensagens para o outro PC.

Essa nova metodologia para interface de sistemas robóticos, é extremamente interessante para o pátio tecnológico brasileiro e mundial, visto que muitos robôs em atividade, possuem arquitetura fechada e a atualização desses controladores são de alto custo.

No âmbito educacional, a idéia é utilizar o sistema para dar aulas à distância, possibilitando assim, o acesso de mais usuários como meio de incentivo à área de robótica.

## REFERÊNCIAS BIBLIOGRÁFICAS

ABB Flexible Automation , BaseWare OS 3.0 , 3HAC 0930-1 , User Guide (Guia do usuário do Robô – IRB2400).

ABB Flexible Automation , M97A , 3HAC 0818-1 , Product Manual IRB 2400 (Manual do Produto – IRB2400).

Vertut, J. e Coiffet, P. Teleoperation and Robotics, Volume 3A, Robot Technology, Kluwer Academic Publishers Group, 1985.

Litzenberger G. (2009) Robôs made in Brasil. Revista Inovação em Pauta, 7ª Edição, p. 57. Java Communications API

Álvares, A. J., Romariz, L. J., "Telerobotics: Methodology for the Development of a Through the - Internet Robotic Teleoperated System", Journal of the Brazilian Society of Mechanical Sciences, v.24, n. 2, pp. 122-126, 2002.

## TELEOPERATION ROBOTICS: PROMOTION OF SKILLS FOR ARTICULATED CONTROL AND AUTOMATION ENGINEERING

**Abstract:** *This document has the purpose to create and implement a robotic arm that would be able to be controlled from any place in the world via internet, for this teleoperation robotics project, will be used an Industrial Robot ABB IRB 2400/10 Model Lab Manufacturing Process Optimization (LOPF) belonging to the Center for Advanced Technologic Manufacturing (NUMA), USP São Carlos – SP. The software is going to be developed in java, which might make the programming part easier involving access to the serial port, USB port, and data transport over TCP / IP.*

**Key-words:** *Teleoperation, Robotics, Sockets, TCP / IP, JAVA*