

UNIVERSIDADE ESTADUAL DE CAMPINAS – UNICAMP

**FACULDADE DE ENGENHARIA ELÉTRICA E DA
COMPUTAÇÃO – FEEC**

DISSERTAÇÃO DE MESTRADO

Título:

Ambiente de Robótica para Aplicações Educacionais
com SuperLogo

Autor: Marco Túlio Chella

Orientador: Profº. Dr. Mauro Sérgio Miskulin

Banca:

Dr. João Vilhete Viegas D`Abreu

Dr. Miguel Gustavo Lizarraga Espinosa

Profº Dr. Furio Damiani

Profº Dr. José Raimundo de Oliveira

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DA ÁREA DE ENGENHARIA - BAE - UNICAMP

C418a	Chella, Marco Túlio Ambiente de robótica para aplicações educacionais com SuperLogo / Marco Túlio Chella.--Campinas, SP: [s.n.], 2002. Orientador: Mauro Sérgio Miskulin. Dissertação (mestrado) - Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação. 1. Logo (Linguagem de programação de computador). 2. Educação. 3. Robótica. 4. Linguagem. 5. Programação (Computadores). I. Miskulin, Mauro Sérgio. II. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. III. Título.
-------	--

*“Sábio não é o homem que fornece as verdadeiras respostas;
é o que formula as verdadeiras perguntas”*

Claude Lévi-Strauss

Dedicatória

A minha companheira Jane, pelo estímulo; a minha filha Tainá pela compreensão e paciência; e a meus pais.

Agradecimentos

Ao meu orientador, professor Mauro, pela oportunidade e confiança.

Ao pesquisador João Vilhete, sempre aberto para conversar e discutir sobre Logo, Robótica e Educação.

A toda equipe do NIED (Núcleo de Informática Aplicada a Educação) pelo apoio e suporte dados ao desenvolvimento deste trabalho.

Ao CNPQ pelo apoio financeiro.

Resumo

Este trabalho, intitulado “Ambiente para Robótica Educacional com SuperLogo” descreve a implementação de um conjunto de dispositivos de hardware e módulos de programa que integrados a linguagem de programação para propósitos educacionais SuperLogo propiciam condições para que o aprendiz possa desenvolver atividades que incluem a montagem e automação de dispositivos. Estas atividades podem ocorrer no modo presencial ou remotamente por meio da rede internet.

O Ambiente de Robótica Educacional desenvolvido oferece uma ferramenta para que atividades apoiadas por princípios de teorias como construtivismo e construcionismo, que valorizam o aprender pelo fazer, possam ser desenvolvidas nos mais diversos contextos de ensino-aprendizagem.

A concepção modular do projeto provê meios para que tecnologias disponíveis comercialmente e outras desenvolvidas no futuro possam ser agregadas ao ambiente, proporcionando opções de escolha quanto a recursos e materiais.

Abstract

This work, entitled “Environment for Educational Robotics with SuperLogo”, describe the development of a set of hardware devices and software modules which, together with the SuperLogo educational programming language, provide means for the learners to perform activities involving the construction and automation of devices. Such activities may take place in local mode or remotely, through the Internet.

The Environment for Educational Robotics is a tool for performing, in different learning-teaching contexts, activities based on pedagogic theories as constructivism and constructionism, which value learning-by-making.

The project’s modular conception makes possible to aggregate commercially available technologies with other technologies that may be developed in the future, providing a variety of possible choices of resources and materials.

Sumário

<u>Introdução</u>	13
<u>Objetivo</u>	13
<u>Organização do Trabalho</u>	14
<u>Capítulo 1</u>	15
<u>A Informática Aplicada à Educação</u>	15
1.1. <u>Histórico da Informática na Educação</u>	15
1.2. <u>Fundamentos da Aplicação do Computador na Educação</u>	18
1.3. <u>A linguagem de programação Logo</u>	20
1.3.1. <u>Aspecto pedagógico do Logo</u>	20
1.3.2. <u>Aspecto computacional do Logo</u>	21
1.4. <u>Robótica Educacional</u>	23
1.4.1. <u>Dispositivos eletromecânicos</u>	24
1.4.2. <u>Componentes eletrônicos</u>	24
1.4.3. <u>Programa</u>	24
1.5. <u>A Robótica Educacional em ambientes de ensino-aprendizagem</u>	25
1.5.1. <u>Robótica Educacional e as Tecnologias de Comunicação</u>	26
1.6. <u>Considerações sobre o capítulo</u>	27
<u>Capítulo 2</u>	29
<u>Ambientes para Robótica Educacional</u>	29
2.1. <u>Ambientes para Robótica Educacional com aplicação no modo presencial</u>	29
2.1.1. <u>Computador MSX com interface para Robótica</u>	29
2.1.2. <u>TcLogo</u>	31
2.1.3. <u>Lego Dacta Control Lab versão 1.2</u>	33
2.1.4. <u>Coach Júnior</u>	35
2.1.5. <u>Lego-MINDSTORMS</u>	35
2.2. <u>Ambientes de Telerobótica</u>	42
2.2.1. <u>RemoteBot.net</u>	43
2.2.2. <u>Telegarden</u>	44
2.2.3. <u>Bigsignal 2000</u>	46
2.2.4. <u>Projeto Redrover</u>	48
2.2.5. <u>RexLab</u>	49
2.3. <u>Considerações sobre o capítulo</u>	49
<u>Capítulo 3</u>	51
<u>Estrutura do Ambiente de Robótica implementado</u>	51
3.1. <u>Descrição dos componentes implementados</u>	53
3.1.1. <u>Interface de Hardware para Robótica (IHR)</u>	53
3.1.2. <u>Interface SuperLogo RCX (ISR)</u>	53
3.1.3. <u>Procedimentos SuperLogo</u>	54
3.1.4. <u>Programa CGI</u>	54
3.2. <u>Considerações sobre capítulo</u>	54
<u>Capítulo 4</u>	55
<u>Implementação da Interface de Hardware para Robótica</u>	55

4.1.	Hardware da Interface	55
4.1.1.	Estrutura Mecânica	66
4.1.2.	Módulo de comunicação infravermelho	67
4.1.3.	Módulo de comunicação por radiofrequência	72
4.1.4.	Servomotores	75
4.1.5.	Sensores	78
4.1.6.	Programa do microcontrolador	85
4.1.7.	Programa SuperLogo para comunicação com IHR	88
4.2.	Considerações sobre o capítulo	93
Capítulo 5		95
Implementação da Comunicação SuperLogo – Lego RCX		95
5.1.	Painel de controle da ISR (Interface SuperLogo RCX)	99
5.1.1.	Módulo de Diagnóstico	99
5.1.2.	Módulo de Configuração	101
5.1.3.	Módulo de Operação	102
5.2.	Utilizando o SuperLogo com o Lego RCX	103
5.3.	Considerações sobre o capítulo	104
Capítulo 6		105
Implementação do Ambiente de Telerobótica		105
6.1.	Envio e recebimento de comandos Remotos	105
6.2.	Modelo Cliente Servidor	105
6.3.	Interface CGI	106
6.3.1.	Recebendo entrada de dados pelo navegador WEB	110
6.3.2.	Enviando dados ao Navegador	112
6.3.3.	Servidores WEB	113
6.3.4.	O Servidor WEB Xitami	113
6.3.5.	Implementação do Programa CGI	114
6.4.	Monitoração do robô remoto	115
6.5.	Comunicação com o usuário	117
6.6.	Considerações sobre o capítulo	119
Capítulo 7		121
Aplicação do Ambiente de Robótica Educacional		121
7.1.	A utilização do ARE por aluno do curso de Mecatrônica	121
7.2.	Utilização do ARE com alunos do curso de Pós-Graduação em Informática na Educação	124
7.2.1.	Projeto Dinossauro	125
7.2.2.	Projeto Pauta Musical	126
7.2.3.	Projeto Emília	127
7.2.4.	Projeto Energia Alternativa	128
7.3.	Oficina de Robótica Educacional com alunos do ensino fundamental	129
7.4.	Utilização do ARE no modo remoto – Telerobótica	131
7.5.	Considerações sobre o capítulo	132
Capítulo 8		133
Conclusão		133
8.1.	Considerações Finais	133
8.2.	Perspectivas para futuros trabalhos	134
Glossário		137

<u>Referências</u>	139
Apêndice 1 – Lego Mindstorms.....	143
Apêndice 2 – Robolab.....	147
Apêndice 3 – Funções SuperLogo para controle do RCX.....	157
Apêndice 4 – Exemplos de utilização do SuperLogo com o RCX.....	163
Apêndice 5 – Projetos Lego RCX com SuperLogo.....	173

Índice de Figuras

Figura 1.1 Tartaruga gráfica	22
Figura 1.2 Gráfico obtido com o Logo	22
Figura 2.3 Computador MSX com interface	30
Figura 2.4 Cartucho do MSX	30
Figura 2.5 Interface para o MSX	31
Figura 2.6 Tela do TcLogo	32
Figura 2.7 Interface Lego 70288	32
Figura 2.8 Placa da interface Lego 70288	33
Figura 2.9 Tela do programa Control Lab	33
Figura 2.10 Interface gráfica do Control Lab	34
Figura 2.11 Interface Lego 70909	34
Figura 2.12 Interface gráfica do Coach Lab	35
Figura 2.13 Sensores e motor Lego	36
Figura 2.14 Comunicação RCX computador PC	37
Figura 2.15 Lego RCX	38
Figura 2.16 Arquitetura de Programa do RCX	39
Figura 2.17 Tela do programa Mindstorms	41
Figura 2.18 Tela do programa Robolab	42
Figura 2.19 Robô Khepera	43
Figura 2.20 Interface gráfica do Remotebot.net	44
Figura 2.21 Manipulador do Telegarden	45
Figura 2.22 Interface gráfica do Telegarden	46
Figura 2.23 Interface gráfica do projeto Bigsinal	47
Figura 2.24 Interface para exibição de vídeo	47
Figura 2.25 Robô do projeto Bigsinal	48
Figura 2.26 Interface gráfica do projeto Redrover	49
Figura 3.27 Estrutura do ambiente de robótica educacional ARE	52
Figura 4.28 Diagrama da interface de hardware	55
Figura 4.29 Diagrama do microcontrolador PIC16F84	58
Figura 4.30 Circuito esquemático do bloco do microcontrolador	64
Figura 4.31 Circuito integrado 74LS245	65
Figura 4.32 Circuito esquemático da interface de hardware	66
Figura 4.33 Fotografia da interface de hardware	67
Figura 4.34 Diagrama da comunicação por infravermelho	68
Figura 4.35 Circuito esquemático do transmissor de infravermelho	70
Figura 4.36 Receptor de infravermelho	71
Figura 4.37 Módulos para radiofrequência	73
Figura 4.38 Circuito do transmissor de RF	73
Figura 4.39 Circuito do receptor de RF	74
Figura 4.40 Servomotor	75
Figura 4.41 Conjunto de peças que constituem o servomotor	76
Figura 4.42 Temporização do servomotor	77
Figura 4.43 Engrenagem do servomotor	78
Figura 4.44 Amplificador operacional	80

Figura 4.45 Circuito integrado LM358	81
Figura 4.46 Circuito do sensor de temperatura	82
Figura 4.47 Curva típica de fotodiodo	83
Figura 4.48 Sensor de luminosidade	84
Figura 4.49 Sensor de umidade	85
Figura 4.50 Diagrama do programa do microcontrolador	86
Figura 4.51 Diagrama de acionamento dos servomotores	88
Figura 5.52 Esquema da comunicação do SuperLogo com o RCX	97
Figura 5.53 Fluxograma de mensagem entre o SuperLogo e a ISR	98
Figura 5.54 Fluxograma para sensores	98
Figura 5.55 Interface gráfica da ISR	100
Figura 5.56 Interface gráfica da ISR - diagnóstico	100
Figura 5.57 Interface gráfica ISR demonstração do diagnóstico	101
Figura 5.58 Interface gráfica ISR - configuração	102
Figura 5.59 Interface gráfica ISR - operação	103
Figura 5.60 Experimento com o RCX	103
Figura 6.61 Fluxograma da requisição de arquivo em servidor WEB	107
Figura 6.62 Fluxograma de programa CGI	108
Figura 6.63 Formulário em página html	110
Figura 6.64 Envio de informações ao navegador WEB	112
Figura 6.65 Instalação física do ambiente de telerobótica	116
Figura 6.66 Diagrama em blocos do programa de captura de vídeo	117
Figura 6.67 Interface gráfica do ambiente de telerobótica	118
Figura 6.68 Janela de chat (bate-papo) do ambiente de telerobótica	119
Figura 7.69 Carro Lego RCX	122
Figura 7.70 Projeto dinossauro	126
Figura 7.71 Projeto pauta musical	127
Figura 7.72 Projeto emília	128
Figura 7.73 Projeto energia alternativa	128
Figura 7.74 Grupo de alunos programando robô	129
Figura 7.75 Projeto máquina de refrigerante	130
Figura 7.76 Vista da máq. de refrigerante	130
Figura 7.77 Projeto Pega-Lixo	130
Figura 7.78 Projeto parque de diversões	131
Figura 7.79 Projeto mão robótica	131

Introdução

A rotina do homem atual é cercada de artefatos tecnológicos, todavia essa tecnologia a cada dia se torna mais opaca, entendendo-se por opaca, o fato de que os mecanismos, programas e microcircuitos dos modernos equipamentos estão embutidos e cada vez mais invisíveis, propiciando pouca ou nenhuma oportunidade para que o usuário possa explorar e compreender seus princípios técnicos e científicos. Provavelmente muitas pessoas se indagadas sobre o funcionamento de algo familiar como uma máquina automática para venda de refrigerantes, irão responder que ela é programada. Ainda que a resposta esteja correta é importante observar que a máquina foi programada de uma forma particular aplicando idéias e conceitos interconectados que abrangem vários domínios do conhecimento e cujo desenvolvimento está situado dentro de um contexto histórico e social.

A escola, sensível a esta realidade, tem se empenhado em atender aos anseios e necessidades resultantes das mudanças que ocorrem na sociedade. A educação baseada em teorias como o construtivismo, que valoriza o aprender pelo fazer, a compreensão e não apenas a assimilação de conteúdos e também o aparelhamento da sala de aula com as modernas tecnologias como computadores, Internet, vídeo, é a resposta que a escola tem dado a sociedade.

Objetivo

Este trabalho propõe a implementação de um ambiente para robótica educacional por meio do qual o aprendiz tenha a oportunidade de manusear concretamente idéias e conceitos, dentro de um contexto que estimule a multi e interdisciplinaridade, dando-lhe o controle sobre a elaboração do seu próprio conhecimento. Com suporte às diversas formas de comunicação eletrônica favoreça a cooperação em atividades, discussão, troca e comunicação de idéias, propiciando meios para a construção de um conhecimento coletivo.

Para atingir este propósito é preciso desenvolver um conjunto de aplicativos e componentes de hardware. O desenvolvimento deve adotar um modelo implementação que permita a integração de tecnologias disponíveis comercialmente no momento, e outras que surgirem no futuro.

Organização do Trabalho

No desenvolvimento dos capítulos desta dissertação de mestrado serão apresentados os fundamentos teóricos relacionados à utilização da informática como instrumento educacional, os aspectos técnicos relacionados à implementação de um ambiente de robótica para aplicações educacionais e o relato das experiências de utilização do mesmo. A dissertação está organizada em capítulos com a seguinte constituição:

Capítulo 1 - apresenta um breve histórico do uso do computador com propósitos educacionais e as teorias de aprendizado que fundamentam sua utilização. Os aspectos computacionais e pedagógicos da linguagem de programação Logo são discutidos. O capítulo também aborda o tema robótica educacional, conceituando e demonstrando experiências de sua inserção em ambientes de ensino aprendizagem.

Capítulo 2 - demonstra a evolução dos vários ambientes para robótica educacional ao longo do tempo e as tecnologias de programação e hardware empregadas, fazendo-se a distinção entre os que tem aplicação no modo presencial e no remoto.

Capítulo 3 - descreve a estrutura do ambiente de robótica educacional implementado, os diversos módulos que foram desenvolvidos e os recursos de hardware e programa empregados.

Capítulo 4 - apresenta a implementação de uma interface de hardware para utilização educacional e módulos acessórios como sensores e módulos de comunicação.

Capítulo 5 - discorre sobre a implementação dos recursos necessários para comunicação entre dispositivos para robótica da empresa Lego e o programa Logo.

Capítulo 6 - apresenta o desenvolvimento dos sistemas que permitirão a utilização do ambiente de robótica no modo remoto, via rede internet.

Capítulo 7 - trata da aplicação e avaliação do ambiente de robótica desenvolvido.

Capítulo 8 – apresenta a conclusão sobre este trabalho e as perspectivas para futuras pesquisas.

A seção para referências apresenta as bibliografias referenciadas na dissertação, incluindo aplicativos e documentos eletrônicos da Internet.

Na parte final encontram-se os apêndices relativos a listagem de programas, relatórios e material de apoio para os usuários do ambiente de robótica.

Capítulo 1

A Informática Aplicada à Educação

1.1. Histórico da Informática na Educação

A utilização de computadores pode ser dividida em três fases (Starr, 1996):

A primeira ocorreu da metade da década de 1950 até meados de 1980, concentrou-se principalmente nos sistemas CAI¹(Computer Aided Instruction), com sistemas desenvolvidos por empresas como RCA e IBM.

Os esforços e pesquisas para o desenvolvimento de sistemas CAI, em países como os Estados Unidos, originaram-se inicialmente fora das escolas, tinham os militares como seus principais financiadores, contribuindo até o início de 1980 com aproximadamente setenta e cinco por cento dos fundos para as pesquisas com tecnologia educacional.

O PLATO (Programed Logic for Automatic Teaching Operations), baseado na Universidade de Illinois e controlado pela Control Data Corporation, esperava estabelecer um modelo de negócio para educação que abrangesse todo o mundo a partir de sua central de computadores. Por volta de 1981, a Control Data chegou a ter 115 Centros de Aprendizado nos Estados Unidos, fazendo deste o maior sistema CAI implantado.

Entretanto, devido a seu custo, o sistema PLATO foi muito pouco utilizado pelas instituições de ensino. A empresa Control Data teve perdas próximas a um bilhão de dólares com o PLATO, prejuízo que levou à reavaliação de aprendizado pelo sistema CAI.

Paralelamente ao desenvolvimento dos sistemas CAI, em meados de 1960 Seymour Papert e pesquisadores do Massachusetts Institute of Technology (MIT) desenvolveram o Logo como uma linguagem de programação para crianças. A aplicação mais popular envolvia a "tartaruga de solo", um robô mecânico simples conectado ao computador por um "cordão umbilical". Comandos especiais que utilizavam a sintaxe LOGO permitiam o deslocamento da tartaruga no solo.

Inicialmente esta linguagem foi implementada em computadores de médio e grande porte (PDP 11 e PDP 10), isso fez com que até o surgimento dos microcomputadores, o uso

¹ Sistema no qual determinado conteúdo a ser transmitido é selecionado, organizado e armazenado para ser apresentado ao aprendiz. Geralmente, ao final de cada apresentação o aluno é submetido a um questionário e caso as respostas não correspondam ao especificado ele é impedido de continuar.

do Logo ficasse restrito às universidades e laboratórios de pesquisa, em razão desses computadores apresentarem grande volume e alto custo. Para atividades com o Logo as crianças e professores se deslocavam até esses centros. Nessas circunstâncias, os resultados se mostraram interessantes e promissores. Pode-se dizer que esta foi a única alternativa que surgiu para o uso do computador na educação com uma fundamentação teórica, passível de ser usada em diversos domínios do conhecimento e com muitos casos documentados, os quais mostravam a sua eficácia como meio para a construção do conhecimento (Valente, 1999).

A segunda fase de desenvolvimento ocorreu no início dos anos 80 com a expansão na oferta de computadores pessoais, interfaces gráficas, e programas de aplicação geral.

Nos Estados Unidos, em 1981, o número de estudantes por computador era de 125 sendo que em 1991 esse número passou para 18 (Starr, 1996). Nesse período, as principais aplicações dos computadores em sala de aula estavam focadas no aprendizado de conceitos sobre o funcionamento do computador seus aplicativos e noções de programação.

Os computadores, em sua maioria, ficavam localizados exclusivamente em laboratórios especializados, disponíveis para os estudantes em atividades por períodos de aproximadamente uma hora por semana, representando quatro por cento do tempo total de aula. No nível secundário os computadores tiveram aplicação principalmente em "computer literacy", isto é, eram utilizados para o aprendizado de linguagens de programação, processadores de texto, planilhas de cálculo e sistemas operacionais; no nível básico os computadores foram usados com os chamados "Sistemas Integrados de Aprendizagem" que ofereciam atividades de exercício-prática em conteúdos básicos. Nenhum desses sistemas interferia de maneira significativa no currículo escolar estabelecido.

Nesta segunda fase, o emprego do computador começou a desviar-se de maneira significativa do padrão seguido no início de sua utilização como tecnologia educacional. Aumentou o interesse de professores e alunos em explorar as potencialidades do computador como instrumento educacional e o questionamento em relação aos modelos impostos pela política educacional. O advento do computador pessoal e programas de aplicação geral fizeram a computação mais flexível e facilmente adaptada aos diferentes perfis e estilos de aprendizado. Diferentemente da TV, do rádio e demais recursos audiovisuais, os computadores tornavam disponíveis recursos que privilegiavam o modelo

de aprendizado centrado no aprendiz. Nesta fase iniciou-se a utilização do computador como ferramenta de comunicação, o que favoreceu o desenvolvimento de novas comunidades de aprendizagem, propiciando um ambiente muito diferente daquele oferecido com as tecnologias precursoras.

Na terceira fase, iniciada nos anos 90 ocorreu a expansão das possibilidades de uso do computador como ferramenta educacional, favorecida pelo advento da multimídia, do explosivo crescimento da Internet e do seu serviço mais popular a World Wide Web a qual tem sido utilizada principalmente como:

- recurso para a identificação, avaliação e integração de uma grande variedade de informação;
- meio para conversação, colaboração, discussão, troca e comunicação de idéias;
- meio para a participação em experiências simuladas, aprendizagem e parcerias cognitivas;
- plataforma internacional para expressão e construção coletiva de conceitos, significados artísticos e cognitivos (Lucena & Fucks, 2000).

O termo WEB é empregado para se referir ao serviço da Internet World Wide Web e será adotado neste trabalho.

A utilização da Internet em atividades educacionais diferencia-se das atividades desenvolvidas na sala de aula tradicional em muitos aspectos:

- Elimina as limitações físicas e estende fronteiras.
- Promove o aprendizado baseado em experimentos vivenciados por pessoas ou grupos.
- Favorece o aprendizado cooperativo que se estende para além da sala de aula.
- Muda a fonte predominante de conteúdo, que se desloca do livro texto e do professor para uma fonte variada de informação.
- Apresenta o conteúdo em formato hipertexto, onde as informações contendo textos e recursos multimídia podem ser acessados de modo não linear.
- Permite ao aprendiz aproveitar-se da flexibilidade de tempo e conteúdo, contribuindo para estimular a educação baseada em projetos.

Nesta fase inicia-se um processo de transformação da computação como uma atividade segregada para uma ferramenta que vai se incorporando ao dia a dia dos trabalhos escolares e do meio ambiente da escola.

1.2.Fundamentos da Aplicação do Computador na Educação

Para a implantação dos computadores na educação são necessários basicamente quatro ingredientes: o computador, o programa educativo, o professor capacitado para usar o computador como meio educacional e o aprendiz.

Espera-se que com a utilização de computadores o aprendiz possa adquirir conhecimentos sobre os mais variados assuntos.

A abordagem pedagógica de como o aprendiz adquire conceitos sobre diversos conteúdos é bastante variada, oscilando entre dois pólos que são caracterizados pelos mesmos componentes: computadores (hardware), programas de computador (software) e o aluno. O que estabelece a polaridade é a maneira como estes ingredientes são aplicados. Em um dos pólos o computador, por meio do programa, transmite informações ao aprendiz, enquanto no outro o aprendiz, também por meio do programa, elabora e expressa seu conhecimento (Valente, 1993).

A utilização do computador como máquina provedora de informações pode ser considerada como uma reprodução do método tradicional de ensino no qual o professor é o transmissor de conhecimento e o aprendiz um recipiente passivo a ser preenchido; neste caso, o professor é substituído pelo computador na tarefa de transmitir as informações. Os programas do tipo tutorial, exercício-e-prática, jogos e simulações enquadram-se na categoria de programas que utilizam o computador para ensinar.

Os tutoriais são versões computadorizadas do que ocorre em uma aula tradicional. A sua aplicação não envolve mudanças na rotina da escola, já que os papéis do professor e do aluno não são alterados.

Os programas de exercício-e-prática são utilizados de maneira geral para revisão de conteúdos que envolvem memorização e repetição.

Os jogos educacionais propiciam uma abordagem de exploração auto dirigida, ao invés da instrução explícita e direta.

A simulação envolve a criação de modelos dinâmicos e simplificados do mundo real, oferecendo ao aprendiz a possibilidade de desenvolver hipóteses, testá-las, analisar os resultados e refletir sobre os mesmos.

A disponibilidade de recursos multimídia e de ferramentas de autoria, que permitem o desenvolvimento de programas educacionais integrando sons, imagens, vídeos e textos, propiciam ao aprendiz, em um mesmo programa, atividades como tutoriais, exercício-e-prática, jogos e simulações. Ainda assim, o conceito de utilização do computador como máquina de transmitir conhecimentos é mantido.

A categoria de programa que suporta a abordagem pedagógica, onde o aprendiz elabora e expressa seu conhecimento, é representada pelas linguagens de programação.

Quando o aprendiz, por meio de uma linguagem, programa o computador, este pode ser visto como uma ferramenta para resolver problemas. O programa produzido utiliza conceitos, estratégias e um estilo de resolução de problemas. Nesse sentido, a elaboração de um programa exige que o aprendiz processe informação e a transforme em conhecimento que, de certa maneira, é explicitado no programa (Valente, 1999) .

A análise da atividade de programar o computador, usando uma linguagem de programação, permite identificar diversas ações, que acontecem em termos do ciclo descrição-execução-reflexão-depuração-descrição, que o aluno realiza, e são de extrema importância na aquisição de novos conhecimentos:

- Descrição da resolução do problema. Isso significa utilizar toda a estrutura do conhecimento para representar e explicitar os passos da resolução do problema a partir de uma linguagem de programação.
- Execução dessa descrição pelo computador. A execução fornece uma realimentação fiel e imediata, desprovida de qualquer parcialidade entre o computador e o aprendiz. O resultado obtido é fruto do que foi solicitado ao computador.
- Reflexão sobre o que foi produzido pelo computador. O processo de refletir sobre o resultado apresentado pelo computador pode levar o aprendiz às seguintes ações: ele não modifica o seu procedimento porque as idéias iniciais sobre a resolução daquele programa correspondem aos resultados apresentados pelo computador; ou depura o programa, quando o resultado é diferente do esperado.

- Depuração dos conhecimentos por intermédio da busca de novas informações. O aprendiz pode buscar informação sobre: conceitos de determinada área, alguma convenção da linguagem de programação. Essa informação é assimilada pela estrutura mental (passa a ser conhecimento) e aplicada no programa, para modificá-lo.

Sob a ótica do ciclo descrição-execução-reflexão-depuração-descrição, o programa criado pelo aprendiz pode ser visto como uma explicitação do seu raciocínio, em termos de uma linguagem precisa e formal. A execução deste programa pode ser interpretada como a execução do raciocínio do aprendiz.

1.3. A linguagem de programação Logo

Em meados de 1960 o professor Seymour Papert (Papert,1985) e outros pesquisadores do Massachusetts Institute of Technology (MIT) desenvolveram a linguagem Logo.

A linguagem Logo pode ser considerada como uma linguagem de programação de aplicação genérica, apresentando o mesmo grau de flexibilidade de linguagens de programação como Pascal ou C. Friendly (Friendly, 1988) considera o Logo como uma linguagem de limiar, isto é, apresenta facilidade de aprendizado, mesmo para crianças, e é potente e flexível o suficiente para atender programadores avançados.

Como a maioria das linguagens de programação convencionais, pode ser utilizada como um modelo para uma ampla variedade de fenômenos, tanto naturais quanto artificiais.

Diferentemente da maioria das linguagens de programação, o Logo foi projetado para prover facilidade de uso. Com o Logo se obtêm resultados significativos dentro de um curto período o que normalmente não ocorre no início do aprendizado de uma linguagem (Goodyear, 1986).

1.3.1. Aspecto pedagógico do Logo

O fundamento pedagógico da linguagem Logo é o construtivismo, teoria do aprendizado desenvolvida por Jean Piaget (Flavell, 1986). O construtivismo propõe que o conhecimento não é algo que possa ser transmitido, mas construído. Esta construção ocorre quando o aprendiz elabora internamente sua estrutura de conhecimento, o que ocorre a

partir das experiências propiciadas pelo seu ambiente material, a cultura e os valores presentes em sua comunidade.

O Logo enfatiza este processo de aprendizagem quando propicia ao aprendiz um ambiente onde o conhecimento não lhe é transmitido, mas no qual, a interação com os objetos favorece o desenvolvimento de conceitos.

Seymour Papert (Papert, 1994) propõe uma teoria derivada do construtivismo que ele denominou construcionismo. O construcionismo apresenta vários pontos em comum com a teoria de Piaget, ao propor que a forma mais eficiente de formar a estrutura de conhecimentos do aprendiz é por meio da construção de objetos externos que possam ser manuseados e compartilhados.

Este conhecimento é valorizado por ser útil, por ser um tipo que pode ser compartilhado com os outros e por combinar com o estilo individual do indivíduo. O construcionismo valoriza a pluralidade de definições, significados, e modos de conhecimento, considerando o aprendizado como algo pessoal e que deve ser controlado pelo aprendiz. Por isso a importância de se desenvolver ambientes que propiciem essa construção

1.3.2. Aspecto computacional do Logo

A maioria das linguagens de computador como as linguagens naturais, têm um vocabulário e uma gramática que devem ser conhecidos antes que se possa utilizá-las. Este também é o caso do Logo, mas com um pouco menos de ênfase, porque é possível chegar a resultados significativos conhecendo um número pequeno de comandos. Uma outra característica do Logo é a possibilidade que o usuário tem de criar suas próprias palavras. Estas novas palavras incorporadas ao vocabulário Logo podem ser utilizadas para formar as ações da máquina de maneira pessoal, ampliando o controle do usuário.

O objeto central do Logo é a tartaruga (Figura 1.1); nos ambientes gráficos a tartaruga é representada por uma figura que pode ser controlada e ter o formato alterado pelo usuário. Sua função é agir como um "objeto com o que pensar", e é particularmente útil para o tipo de raciocínio apresentado com o desenho, a geometria e as regras matemáticas do movimento.

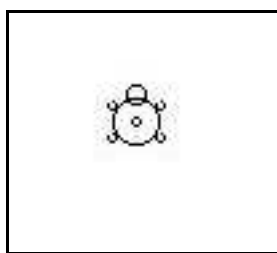


Figura 1.1 Tartaruga gráfica

Por exemplo, se o objetivo for construir na tela uma figura geométrica que represente um quadrado, é preciso instruir a tartaruga a se movimentar em um caminho, girar em um ângulo que corresponda ao quadrado e desenhar linhas neste percurso. Para fazer a tartaruga gráfica deslocar-se à frente 40 unidades a sintaxe é:

Para frente 40

Para que a tartaruga gire 90 graus à direita a sintaxe é:

Paradireita 90

Para realizar o quadrado é preciso executar quatro vezes os comandos anteriores; para fazer isso pode-se utilizar o comando *repita*:

Repita 4 [Para frente 40 Paradireita 90]

Portanto conhecendo apenas tres comandos em Logo e um pouco sobre sua gramática é possível produzir uma figura que representa um quadrado (Figura 1.2).

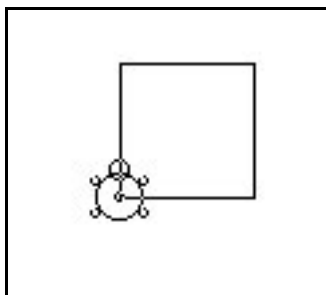


Figura 1.2 Gráfico obtido com o Logo

Também é possível dar-se um nome à seqüência de comandos que produziu o quadrado e a partir desta ação, a tartaruga sempre executará o procedimento que produz a

figura quando o nome que contém a seqüência de comandos for solicitado. Para "ensinar" a seqüência de comandos ao Logo é utilizado o comando *Aprenda*.

Aprenda quadrado

Repita 4 [Parafrente 40 Paradireita 90]

Fim

Acrescentando novos comandos ao vocabulário do Logo o usuário pode criar situações onde seja possível testar conjecturas sobre a natureza da geometria ou da matemática. Do mesmo modo que se trabalha com os conteúdos matemáticos, também se podem explorar outras áreas do conhecimento utilizando-se os diversos recursos da linguagem.

A capacidade de ampliar a linguagem Logo, ao acrescentar-se novos comandos ao vocabulário, agrega-lhe o valor de ser algo compartilhável. Por exemplo, a palavra quadrado, que desenha a respectiva figura, pode ser utilizada por outro aprendiz associada a um triângulo, criando a representação de uma casa.

Considerando a linguagem como algo que une a cultura, o Logo tem a capacidade de criar uma fértil cultura de computador. Neste contexto o professor não assume apenas o compromisso do aprendizado da linguagem, mas atua junto ao aprendiz na construção, utilização, refinamento e descarte de comandos Logo.

1.4. Robótica Educacional

Neste trabalho designou-se por Robótica Educacional como o ambiente constituído pelo computador, componentes eletrônicos, eletromecânicos e programa, onde o aprendiz, por meio da integração destes elementos, constrói e programa dispositivos automatizados com o objetivo de explorar conceitos das diversas áreas do conhecimento. Estes dispositivos automatizados passam a comportar-se como músculos do computador atuando no ambiente, ou seus “órgãos do sentido”, coletando dados e enviando-os ao computador onde será processado (D´Abreu, 1993) .

1.4.1. Dispositivos eletromecânicos

São constituídos de componentes elétricos como motores, responsáveis pela geração de movimentos, relés, alguns tipos de sensores e peças mecânicas como engrenagens, rodas, correias, entre outros. As peças mecânicas podem fazer parte de conjuntos para montagem comerciais ou ser construídas e adaptadas pelo aprendiz a partir de materiais disponíveis em seu meio como plástico, metal, madeira entre outros (Reshko et al., 2001).

1.4.2. Componentes eletrônicos

Para que seja possível, por meio do computador, controlar motores e receber dados provenientes de sensores, é necessário dispor de um elemento eletrônico que seja responsável pela conversão das instruções do formato digital, enviadas pelo computador, em sinais elétricos compatíveis com o acionamento de motores ou relés. Uma outra função desempenhada por ele é converter as grandezas físicas de um sensor em valores digitais que serão enviados ao computador para posterior processamento.

Este elemento eletrônico é usualmente denominado de Interface de Hardware e pode assumir várias configurações quanto a forma de se comunicar com o computador, a quantidade e tipos de motores que podem ser controlados, e os tipos de sensores que podem ser conectados.

1.4.3. Programa

Quando se desenvolve um dispositivo robótico, é planejada uma funcionalidade, que pode ser: acionar os motores, alterar o sentido de rotação, desligá-los, reconhecer o estado de um sensor, e, a partir deste estado, executar alguma ação. Para que isto seja possível, não basta que se tenha uma interface de hardware conectada ao computador e a estrutura mecânica com seus motores e sensores montados. É preciso que um conjunto de instruções seja executado em uma sequência pré-determinada; normalmente utiliza-se uma linguagem de programação para esta tarefa.

A priori, qualquer linguagem de programação, que ofereça recursos para a comunicação com as portas do hardware como USB, RS232, Paralela ou Centronics; pode ser utilizada para automatizar dispositivos robóticos. Contudo, a linguagem Logo, em razão de seu propósito educacional, é particularmente apropriada para ser o programa em um Ambiente de Robótica Educacional.

Originalmente a linguagem Logo não dispõe de comandos específicos para robótica. Esses comandos têm de ser criados para atender os protocolos de comunicações das interfaces de hardware. Como opção ao Logo, algumas linguagens de programação têm sido desenvolvidas, inclusive empregando conceitos de orientação a objeto e programação baseada em ícones, especificamente para ambientes de robótica educacional.

No capítulo 2 é apresentada a evolução histórica dos vários ambientes para robótica educacional, alguns disponíveis comercialmente, outros, desenvolvidos por instituições de pesquisa.

1.5. A Robótica Educacional em ambientes de ensino-aprendizagem

Estudos realizados por vários pesquisadores como Stager (2001), Jarvinen (1998), Oliveira (1993), Sidericoudes (1993), entre outros, têm demonstrado que atividades com robótica educacional, seguindo uma abordagem construtivista, propiciam meios para que o aprendiz formule hipóteses relacionadas ao seu objeto de investigação, explore idéias que o levem a discutir e colocar em prática a sua própria maneira de pensar, a validar resultados e construir argumentos que possam ser aplicados.

Para que um dispositivo, criado pelo aprendiz, funcione corretamente, é preciso que certos princípios sejam conhecidos e respeitados. Por exemplo, na construção de um “carrinho”, princípios da física como atrito e lei da gravidade, devem ser aplicados. Os motores usados na tração do carrinho podem girar numa velocidade alta. Neste caso, o carro se deslocará rapidamente, mas terá conseqüentemente uma potência baixa, onde a menor inclinação fará com que ele pare. A solução para este problema é a utilização de engrenagens ou polias. Contudo, para usá-las eficazmente é preciso conhecer os conceitos sobre proporções de engrenagens.

Conceitos que não estão relacionados às ciências exatas também têm sido explorados a partir de atividades com robótica. No programa de pesquisa “Con-science” (Bers & Urrea, 1999) desenvolvido no MIT (Massachusetts Institute of Technology) a construção e automação de dispositivos robóticos têm sido utilizadas para desenvolver temas como a representação de símbolos e valores, e evocar a reflexão e cooperação entre grupos de aprendizes.

Oficinas de robótica educacional, realizadas, pelo presente autor, com professores de ensino fundamental, resultaram no desenvolvimento de alguns projetos onde a

construção de maquetes automatizadas representando, por exemplo, um parque de diversões, um ambiente rural com roda d'água e monjolo, ou o sistema solar, criaram condições para que conceitos relacionados a história, geografia, ecologia e astronomia pudessem ser abordados (Chella, 2002).

Alunos do curso de ciência da computação têm utilizado a robótica para estudar de maneira concreta algoritmos de inteligência artificial, trabalho que envolve a aplicação de diferentes heurísticas para resolver problemas relacionados à montagem de protótipos, e identificar soluções para o programa (D'Abreu, 2002).

1.5.1. Robótica Educacional e as Tecnologias de Comunicação

As novas tecnologias de comunicação, particularmente a Internet, abrem novas possibilidades para o desenvolvimento de atividades com robótica.

Quando se integram aos ambientes de robótica a recursos da rede Internet, com o propósito de permitir que dispositivos possam ser controlados remotamente, caracteriza-se o que tem usualmente recebido a denominação de telerobótica. No contexto de uso pedagógico entende-se telerobótica como sendo um ambiente de aprendizagem à distância que, utilizando técnicas de teleoperação e telemonitoração, possibilita a pessoas que estejam afastadas geograficamente projetarem-se virtualmente em um espaço remoto, com poder de monitorar e atuar em tal espaço, controlando dispositivos robóticos.

Este ambiente de telerobótica deve possuir recursos para o controle e visualização de robôs que estejam conectados em um espaço físico diferente do local onde o aprendiz se encontra. Neste ambiente, ao invés da troca acontecer com colegas, numa sala de aula, como no caso presencial, ela se dá com pessoas do mundo inteiro que tenham acesso ao ambiente.

Assim, ao se desenvolver uma atividade no laboratório, um usuário estará interagindo com todos os outros que estiverem conectados naquele instante. Isto faz com que a sua idéia seja compartilhada com os demais, permitindo que o diálogo e cooperação sejam praticados interdisciplinarmente de forma muito ampla (Garcia, 1995). Com isso, o número de sugestões para a resolução de um determinado problema também aumenta. Este fato em certa medida, pode ser muito interessante, caso as sugestões contribuam para que diferentes pontos de vista sejam apresentados na resolução do problema, o que consiste também em uma das características do ambiente interdisciplinar.

Projetos como o Roboteach (George & Despres, 1999) e KhepheraOnWeb (Remotbot.net, 2002) baseiam-se na implementação de ambientes com propósito educacional nos quais é possível controlar e visualizar robôs remotamente. O Roboteach permite a comunicação entre os aprendizes e os professores, dispondo ainda de ferramentas para a coordenação das atividades. No ambiente KhepheraOnWeb é possível controlar o movimento do robô e monitorar o estado dos seus sensores.

Alguns dos benefícios que podem ser obtidos por meio de Ambientes de Robótica Educacionais com suporte a telerobótica são:

- Maior utilização dos recursos disponíveis nos computadores.
- O acesso de professores e aprendizes a recursos não disponíveis em sala de aula.
- Proporcionar a um grande número de usuários a realização de experimentos praticamente sem custo.
- Atividades desenvolvidas podem ser compartilhadas favorecendo a cooperação no desenvolvimento de projetos.

1.6. Considerações sobre o capítulo

Os princípios derivados da teoria de Piaget (Piaget, 1966) sobre o desenvolvimento cognitivo, e revisados por Seymour Papert (Papert, 1985), sugerem que o centro do processo relacionado ao aprendizado é a participação ativa do aprendiz, que amplia seus conhecimentos por meio da construção e manipulação de objetos significativos para o próprio aprendiz e a comunidade que o cerca.

A utilização de atividades de robótica educacional, como as exemplificadas, fundamentadas nas teorias de Piaget e Papert, demonstra a possibilidade de se abordar concretamente e de forma contextualizada os diversos conceitos utilizados nas práticas da sala de aula, estabelecendo conexões entre os diversos conteúdos, promovendo, desta maneira, a interdisciplinaridade, e estimulando o trabalho cooperativo.

Capítulo 2

Ambientes para Robótica Educacional

A partir da década de 80 houve a popularização dos microcomputadores pessoais, principalmente das linhas Apple e MSX . Estes microcomputadores, dispondo de recursos para comunicação e barramentos acessíveis ao usuário, possibilitaram o desenvolvimento de interfaces para controle de motores, lâmpadas e sensores. Os ambientes de programação geralmente eram baseados na linguagem Logo e seus diversos dialetos, acrescidos de comandos para o controle dos recursos providos pela interface de hardware.

Com o lançamento do microcomputador padrão IBM PC, no final da década de 80, e desenvolvimento das interfaces gráficas, como o Windows, novos ambientes de robótica com propósitos educacionais e de entretenimento foram criados. O ambiente gráfico do Windows propiciou meios para o desenvolvimento de ferramentas de programação baseadas em ícones, incorporando algumas características da programação orientada a objetos.

A seguir são descritos ambientes e ferramentas de robótica que foram desenvolvidos a partir da década de 80.

2.1. Ambientes para Robótica Educacional com aplicação no modo presencial

Nestes ambientes é possível montar o dispositivo robótico, criar o programa de controle e executá-lo no próprio computador.

2.1.1. Computador MSX com interface para Robótica.

No ano de 1986 foi lançado no mercado brasileiro pelas empresas Sharp do Brasil e Gradiente, o microcomputador MSX (Figura 2.3).



Figura 2.3 Computador MSX com interface

Voltado principalmente para o mercado de vídeo-jogos, dispunha de inúmeros recursos de hardware que permitiam implementar animação, quatro canais para produção simultânea de som, 256 cores e usava, como monitor, uma televisão em cores. Essas facilidades permitiram o desenvolvimento de inúmeros programas educacionais, e também das versões da linguagem Logo disponíveis em cartucho com ROM² (Figura 2.4), Hot-Logo e Logo MSX, produzidos respectivamente pelas empresas Gradiente e Sharp.



Figura 2.4 Cartucho do MSX

Tendo como base estas versões do Logo, no ano de 1989, pesquisadores do NIED/UNICAMP desenvolveram uma interface de hardware (Figura 2.5) com as seguintes características:

² ROM é a abreviatura do inglês de Read Only Memory ou memória apenas de leitura. É um tipo de memória utilizada em aplicações onde o dado é gravado uma única vez e fica disponível para um número indefinido de leituras.

- 2 saídas para acionamento de motor de corrente contínua com reversão.
- 2 saídas para acionamento de motor ou luz.
- 1 entrada para sensores de toque ou luz.
- Comunicação com o computador por meio da porta da impressora.

O Logo MSX foi dotado de comandos para controlar as saídas e as entradas de sensor desta interface.



Figura 2.5 Interface para o MSX

2.1.2. TcLogo

O programa TcLogo (Figura 2.6) foi desenvolvido pela empresa LCSi (Logo Computer Systems Inc.) e apresentado ao mercado no ano de 1988. Baseava-se nos conceitos do Logo e era acrescido de comandos que possibilitavam o controle da interface Lego modelo 70288. O ambiente de trabalho e os comandos eram na língua inglesa.

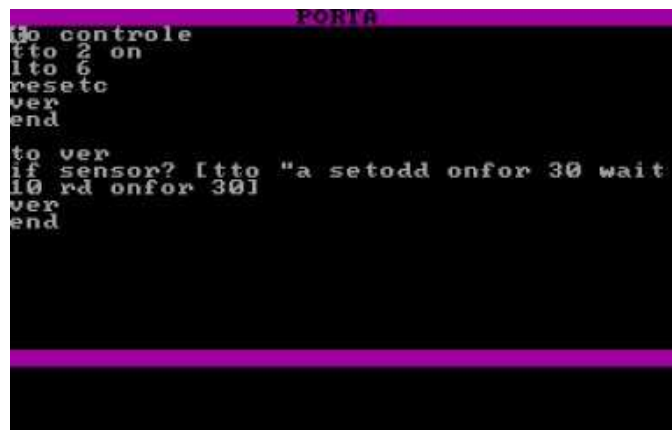


Figura 2.6 Tela do TcLogo

O TcLogo operava em microcomputadores padrão IBM PC com sistema operacional Microsoft DOS 2.1 ou superior.

A interface Lego código 70288 (Figura 2.7) dispunha de seis portas de saída, sendo 3 com inversão de sentido para motor de corrente contínua e duas entradas para sensor de toque ou luz. A comunicação com o computador ocorria por meio de uma placa de interface conectada ao barramento ISA do computador (Figura 2.8).

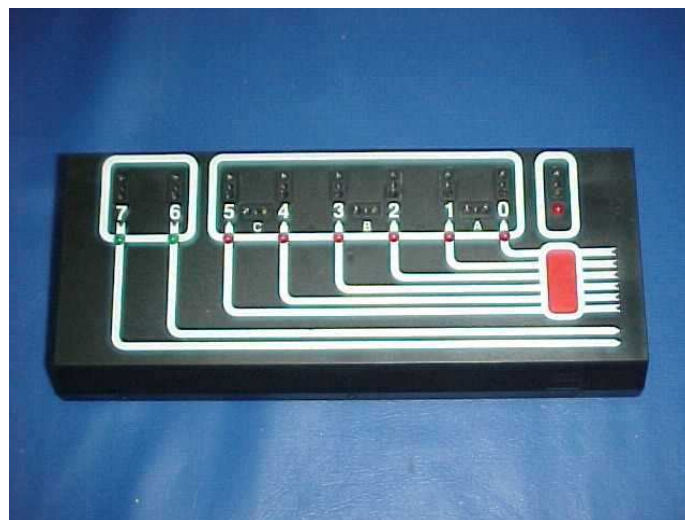


Figura 2.7 Interface Lego 70288



Figura 2.8 Placa da interface Lego 70288

2.1.3. Lego Dacta Control Lab versão 1.2

Apresentado em 1993 pela empresa norte americana LCSi, adotava o conceito de programação do Logo, dispondo de comandos para o controle de portas de entrada e de saída da interface Lego código 70909 como ilustrado na Figura 2.9.

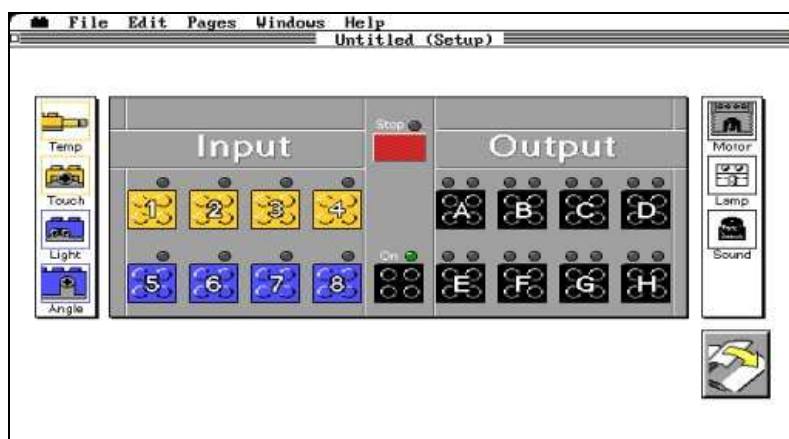


Figura 2.9 Tela do programa Control Lab

O Lego Control Lab diferencia-se das versões tradicionais do Logo por não possuir a tartaruga gráfica. Sua área de trabalho é constituída por caixas de ferramentas que representam instrumentos de um laboratório de ciências (Figura 2.10).

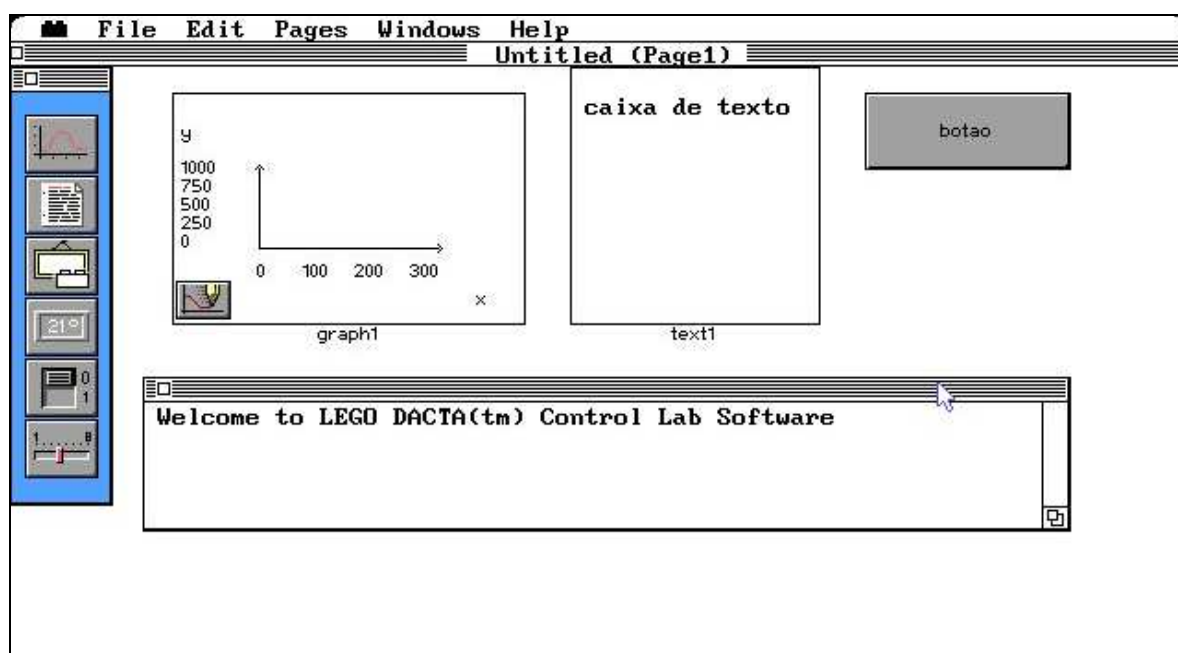


Figura 2.10 Interface gráfica do Control Lab

A interface Lego 70909 comunicava-se com o microcomputador por meio de interface serial; dispunha de seis saídas para controle de motores, lâmpadas e emissor sonoro, e seis entradas para controle de sensores de toque, luz, temperatura e posição (Figura 2.11).



Figura 2.11 Interface Lego 70909

2.1.4. Coach Júnior

Desenvolvido pela Foundation CMA - Centre for Microcomputer Applications e lançado no ano de 2000 o programa Coach Júnior (Figura 2.12) é um pacote de programa de autoria, com o qual é possível controlar as interfaces Lego 70909, Lego RCX e CMA Coachlab .

O ambiente de programação permite o acionamento das saídas para controle de motores atuadores e lâmpadas, a seleção das entradas para leitura dos sensores conectados à interface. Operando em microcomputadores padrão IBM PC, com sistema operacional Microsoft Windows 95/98, possui uma linguagem específica para o controle das interfaces.

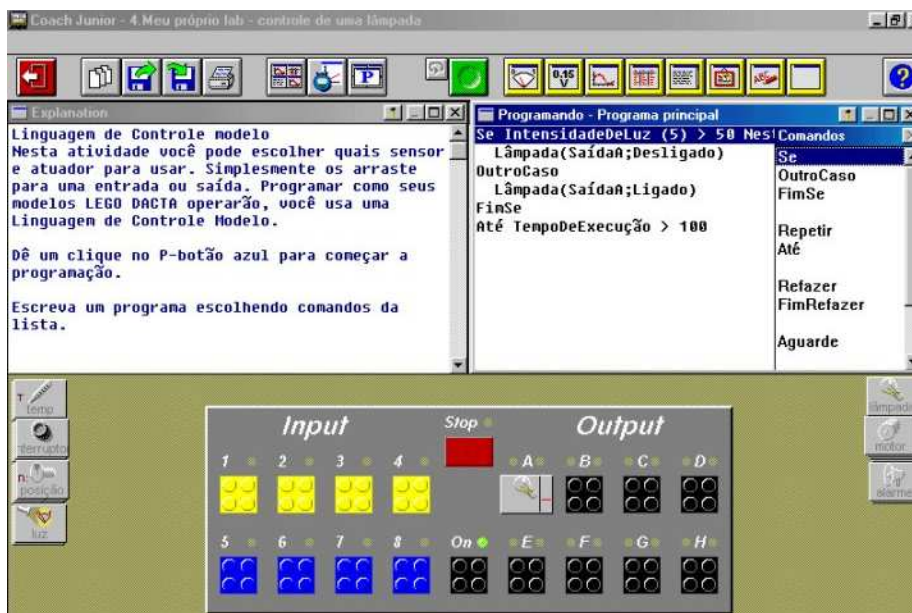


Figura 2.12 Interface gráfica do Coach Lab

2.1.5. Lego-MINDSTORMS

MINDSTORMS é o nome de uma das linhas de produtos comercializados pela empresa Lego. Um dos principais componentes dessa linha de produtos é o Robotic Invention System (RIS), um conjunto de peças que permite a construção de robôs. Com este conjunto é possível a construção de robôs de maneira simples e rápida.

O conjunto RIS possui um tijolo³ que pode ser considerado o cérebro do robô, o qual é chamado de Robotic Command Explorer (RCX). O RCX é um pequeno computador encapsulado em um tijolo Lego. O conjunto inclui também dois sensores de toque, um sensor de luz e dois motores (Figura 2.13). Além dessas peças, que em conjunto com o RCX permitem que o robô possa "sentir" e se locomover no meio em que se encontra, o RIS oferece 700 peças Lego que podem ser utilizadas para construir a estrutura física do robô. O RCX foi utilizado como um dos elementos do ambiente de robótica desenvolvido neste trabalho e por isso será estudado de forma mais detalhada.



Figura 2.13 Sensores e motor Lego

Para que os robôs construídos tenham funcionalidade, é preciso escrever programas que são executados no RCX, utilizando linguagem de alto nível. Estes programas uma vez escritos podem ser carregados no RCX através de uma conexão feita por luz infravermelha. Para tanto, o conjunto é provido de um dispositivo que, uma vez conectado à porta serial de um microcomputador, realiza essa tarefa. Este dispositivo é chamado de *Torre IR* (do inglês Infrared Tower). A figura 2.14 apresenta o esquema básico de comunicação entre um microcomputador e o RCX.

³ Os blocos para montagem Lego são denominados na documentação do fabricante e aceitos pelo usuários como tijolos, por esta razão neste trabalho foi mantida esta nomenclatura.

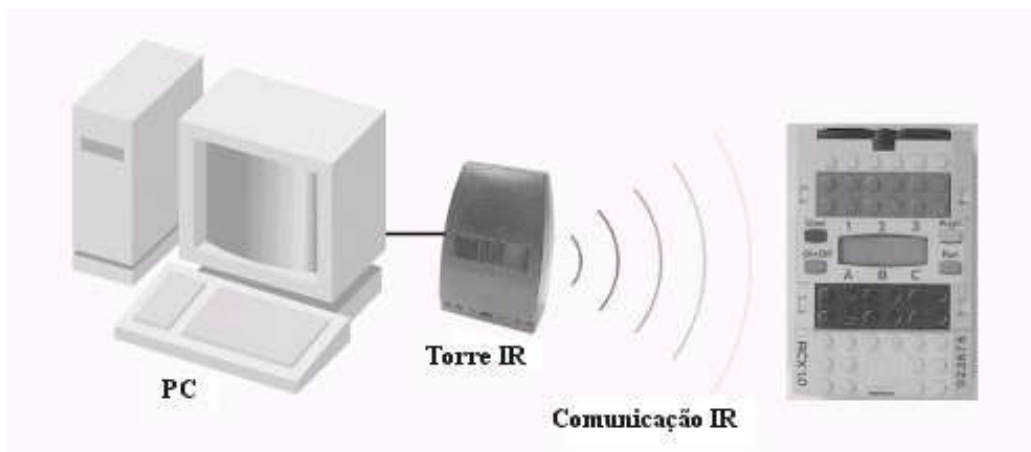


Figura 2.14 Comunicação RCX computador PC

A construção de um robô utilizando RCX consiste basicamente de quatro passos:

- Construir o corpo do robô.
- Escrever um programa utilizando as ferramentas de programa existentes no microcomputador.
- Carregar o programa no robô.
- Executar o programa.

O RCX (Figura 2.15) é um pequeno computador com as seguintes características:

- Três portas de saída, identificadas como A, B e C, localizadas próximas ao centro do tijolo. Os atuadores do robô, motores ou luzes, são conectados a essas portas.
- Três portas de entrada, identificadas com os números 1, 2 e 3, também estão presentes. Vários tipos de sensores podem ser conectados a essas portas, permitindo ao RCX capturar grandezas físicas do ambiente onde se encontra.
- O RCX inclui uma pequena tela de cristal líquido. Esta tela apresenta informações úteis tais como o valor da leitura dos sensores e o estado das portas de saída.
- O RCX é capaz de produzir sinais sonoros em diferentes frequências.
- Existem 4 botões no RCX, através dos quais é possível selecionar programas, iniciar e interromper seu funcionamento. Pode-se também visualizar o valor dos sensores conectados ou o estado das portas de saída.

- O RCX comunica-se com o microcomputador por meio de uma conexão feita por infravermelho. Ele também pode comunicar-se com outros tijolos RCX através da mesma conexão.



Figura 2.15 Lego RCX

2.1.5.1. Ambiente de Programação

O programa do RIS inclui um ambiente para desenvolvimento dos programas que serão executados no RCX. Esta técnica é chamada de *cross-compiling*, o que significa que o programa é escrito em um computador, porém este programa será executado em outro. Neste caso, o programa é escrito no microcomputador PC e executado no RCX.

Uma vez escrito o programa para o RCX, ele deve ser carregado no tijolo. O carregador de programas do RIS é um aplicativo que é executado a partir do microcomputador e que transmite, através da torre IR o programa que o RCX irá executar.

2.1.5.2. Arquitetura de Programa do RCX

Escrever programas para o RCX envolve a execução de várias camadas de programas, tanto no microcomputador como no próprio RCX. A Figura 2.16 apresenta um diagrama dessas camadas.

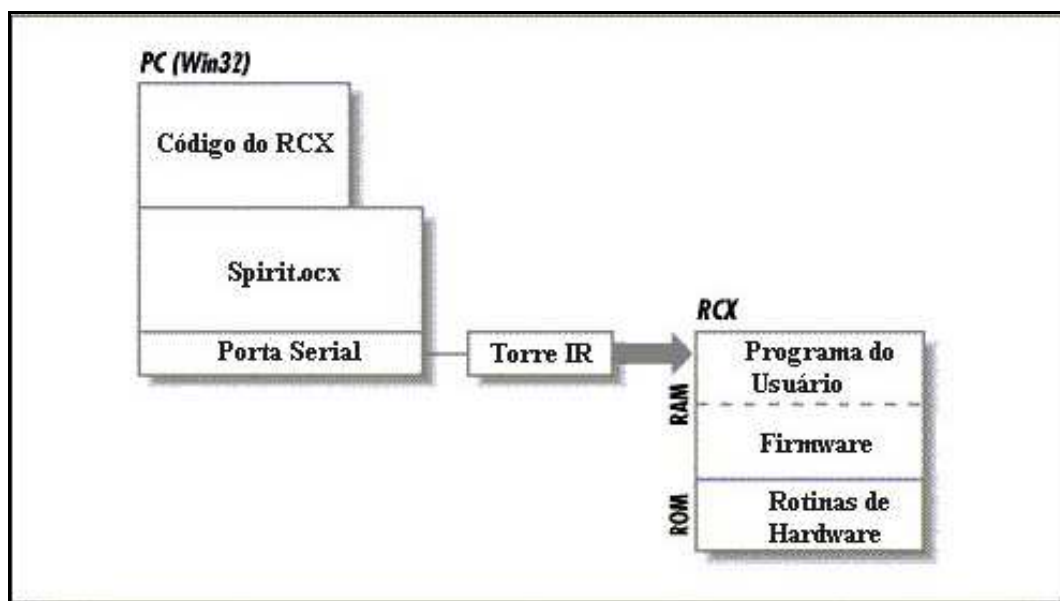


Figura 2.16 Arquitetura de Programa do RCX

O RCX é um pequeno computador baseado no microcontrolador Hitachi H8/3292. O RCX possui dois tipos de memórias: Memória Somente de Leitura (ROM) e Memória de Acesso Aleatório (RAM). Como o próprio nome indica, não pode se escrever na ROM, já vem programada de fábrica e não pode ser mudada. Por outro lado, pode se escrever e ler na RAM quantas vezes for necessário, todavia para preservar o conteúdo da RAM é necessária a presença de uma fonte de energia. Isto significa que se as baterias do RCX são retiradas, o conteúdo da RAM é apagado.

2.1.5.3. Firmware e Bytecode

O primeiro procedimento a ser realizada para se utilizar o RCX é carregar o firmware. O firmware é, essencialmente, o sistema operacional do RCX. As rotinas da ROM têm funções para carregar as rotinas do firmware recebidas da torre IR e armazená-las na RAM. As funções básicas do firmware são: chamada às rotinas da ROM, a interação com o teclado e o painel do RCX e a execução dos programas recebidos pela torre IR.

Embora à primeira vista o firmware e os programas do robô possam aparentar certa semelhança, eles são módulos distintos. O firmware é o código de máquina do Hitachi H8. Em conjunto com o código de máquina do H8 existente na ROM, o firmware define o sistema operacional do RCX. Este provê acesso às portas de entrada e saída do RCX e

também os meios pelos quais os programas do robô são carregados, armazenados, inicializados e interrompidos.

Os programas do robô que são carregados no RCX não são código de máquina do H8. Eles são definidos num nível mais alto de programação e são chamados de bytecodes. Desta forma, um código de máquina do H8 é bastante rudimentar, tal como "mova este valor para o registrador 1", por outro lado uma instrução em bytecode é muito poderosa, tal como "ligue o motor 2 a potência máxima". O firmware interpreta o bytecode e executa a ação apropriada.

2.1.5.4. Programando o RCX

As versões iniciais dos kits Lego Mindstorms incluíam um CD-ROM para microcomputador PC contendo a documentação, tutoriais e a ferramenta de programação baseada em ícones utilizada para criar os programas que a seguir seriam transferidos e executados no RCX. Atualmente o programa deve ser adquirido separadamente do kit Lego Mindstorms e a opção oferecida é o Robolab, também de programação baseada em ícones.

Desenvolvedores independentes criaram ambientes de programação alternativos sendo um dos mais conhecidos o NQC (Near Quite C) baseado na estrutura da linguagem de programação C. Esse programa pode ser encontrado na Internet e é distribuído como freeware.

A seguir são discutidas algumas das características destes programas.

2.1.5.5. Programando com o Lego Mindstorms

O programa que acompanhava o Kit do Lego Mindstorms era a única opção disponível na época para a programação do RCX. Este programa era de tal maneira associado ao kit que comumente se referia a ele com o nome do kit Mindstorms ou Lego Mindstorms.

O Lego Mindstorms dispunha de uma interface gráfica e os elementos de programação, sensores e motores eram representados por ícones (Figura 2.17). Programar, então, consistia em arrastar tais elementos para a área de trabalho e associá-los de maneira que propiciassem ao robô a funcionalidade projetada. Ao terminar a programação o programa deveria ser enviado para o RCX, estando disponível, a partir daí, para execução.

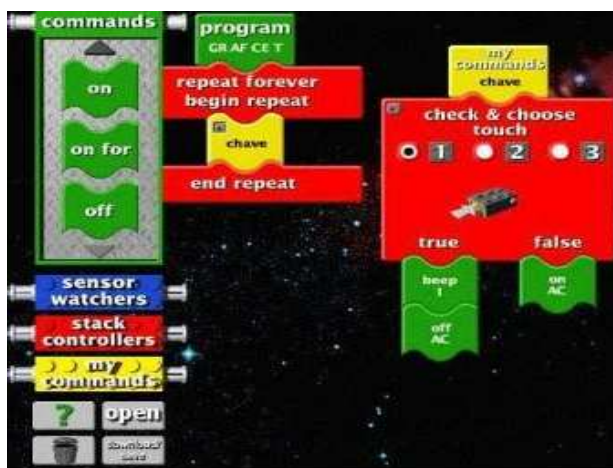


Figura 2.17 Tela do programa Mindstorms

Os elementos de programação do Lego Mindstorms se dividem em comandos, monitores de sensores, controle de fluxo e criação de procedimentos.

Os blocos de comandos (Commands) atuam sobre os motores controlando seu acionamento, a potência aplicada e o sentido de rotação dos mesmos.

Os blocos monitores de sensores (Sensor Monitor) permitem a configuração e leitura dos valores dos sensores conectados ao RCX.

Os controladores de fluxo (Stack Controller) permitem ao programador determinar o caminho de execução que o programa deve seguir a partir de informações obtidas de sensores, contadores e temporizadores entre outros.

O programador pode associar um conjunto de comandos e criar seu próprio procedimento por meio do elemento Meus Procedimentos (My Commands).

A descrição dos diversos comandos do Lego Mindstorms está no Apêndice 1.

2.1.5.6. Programa ROBOLAB

ROBOLAB é uma linguagem de programação, desenvolvida especificamente para ser utilizado na programação do RCX, sendo constituído de uma linguagem gráfica, baseada em ícones.

O ROBOLAB foi desenvolvido pela National Instruments em parceria com a Tufts University - College of Engineering. Este programa baseia-se no LabVIEW™, da National Instruments.

Os principais elementos de sua interface gráfica são a janela de funções e a janela de diagramas (Figura 2.18). Na janela de funções encontram-se os ícones que representam

motores, sensores e blocos de programação como estruturas condicionais e repetição entre outros. A janela de diagramas é o espaço onde os objetos da janela de funções são inseridos e conectados para se montar o programa que posteriormente será descarregado, no RCX .

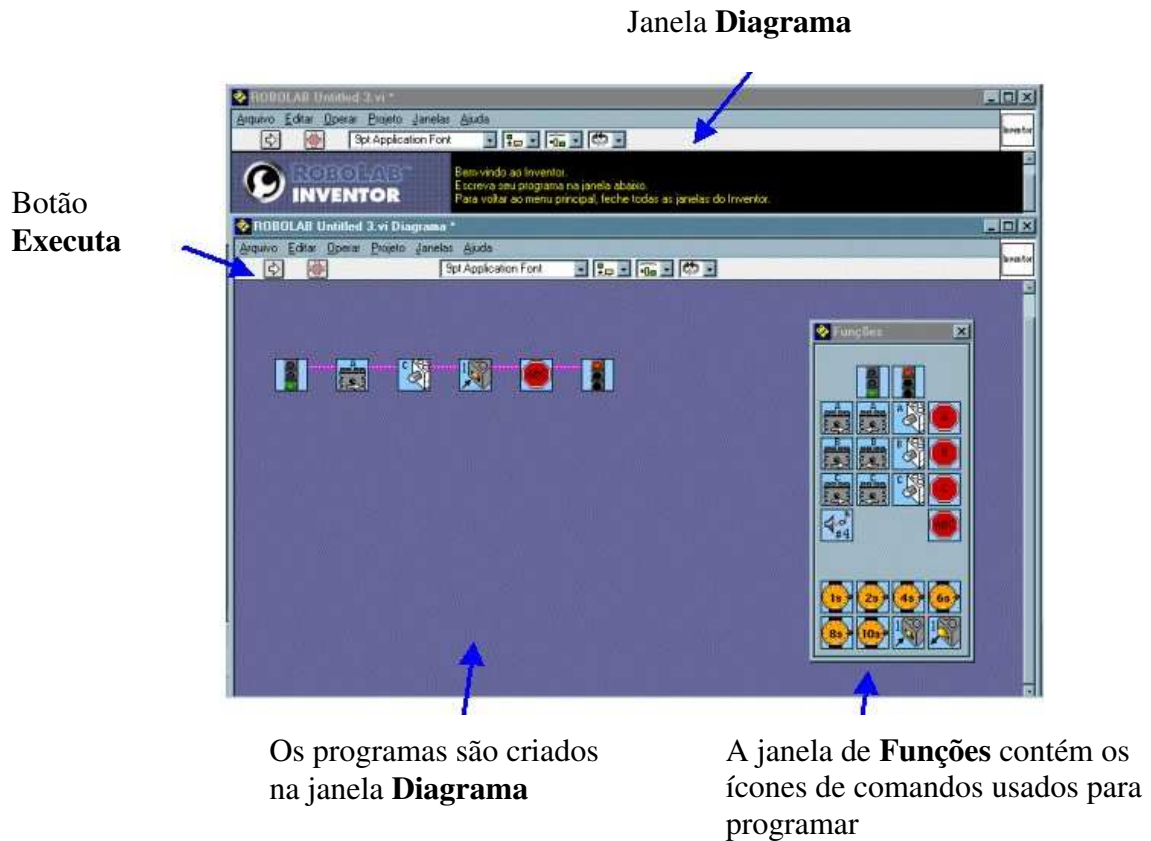


Figura 2.18 Tela do programa Robolab

No Apêndice 2 estão os diversos comandos do Robolab.

2.2. Ambientes de Telerobótica

Nos ambientes de Telerobótica o aprendiz programa e controla um dispositivo robótico que não está presente fisicamente no local onde ele se encontra.

Com o desenvolvimento da Internet é possível utilizar a sua infraestrutura para atuar e controlar dispositivos robóticos a partir de qualquer computador conectado à rede Internet. Ambientes implementados a partir do serviço World Wide Web permitem que o usuário com um navegador padrão como o Internet Explorer da empresa Microsoft ou Netscape tenha acesso aos recursos disponibilizados no ambiente de telerobótica, dispensando a instalação de programas específicos.

2.2.1. RemoteBot.net

No período de 1997 a 2000, pesquisadores do Swiss Federal Institute of Technology, desenvolveram trabalhos relacionados ao controle de robôs remotamente por meio da rede Internet. Os experimentos utilizavam robôs móveis Khepera (Figura 2.19) produzidos pela empresa Suíça K-Team. Os recursos de programa utilizados foram o Personal Web Server da Microsoft e aplicativos desenvolvidos em JAVA.



Figura 2.19 Robô Khepera

Esses experimentos deram origem a um pacote disponível comercialmente, constituído de programas e hardware que possibilitam a montagem de um sistema completo para controle e monitoramento de robôs via WEB.

O pacote é constituído de aplicativos em JAVA, os quais atuam no lado servidor e cliente, programa para captura de vídeo com suporte a dispositivos Matrox Meteor, robô móvel Khepera e outros da empresa K-Team.

No lado cliente, applets Java contidos em uma página WEB criam uma interface gráfica (Figura 2.20) onde é possível visualizar os robôs, o estado dos sensores, e enviar comandos para movimentar o robô. O website do projeto está disponível no endereço: <http://remotebot.k-team.com/museum/>.

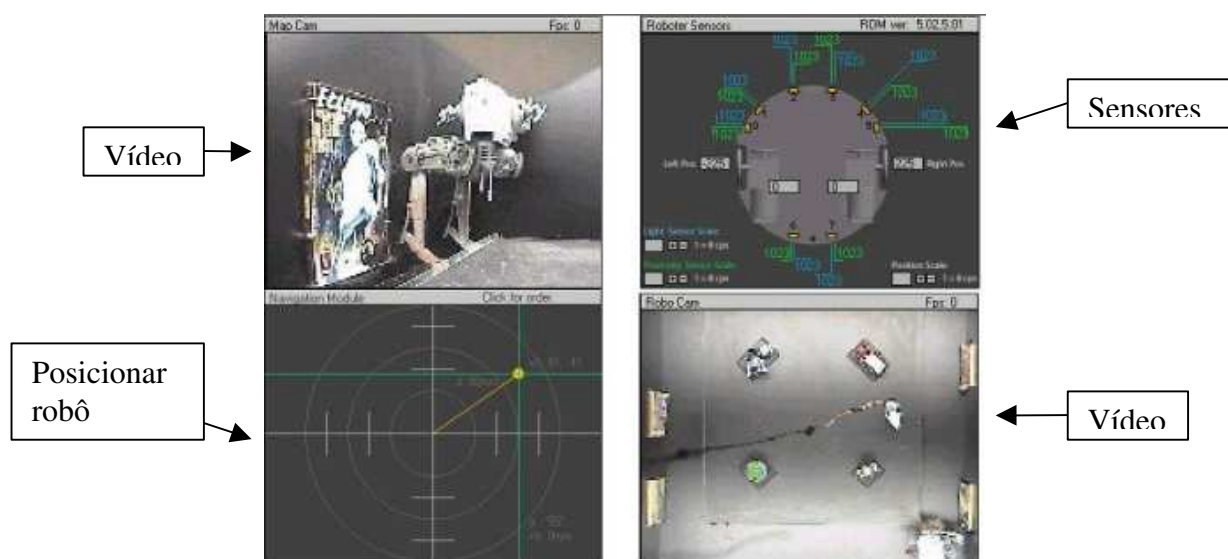


Figura 2.20 Interface gráfica do Remotebot.net

2.2.2. Telegarden

O projeto Telegarden foi desenvolvido pela University of Southern Califórnia e está disponível on-line desde 1995.

Este projeto de telerobótica permite aos usuários a visualização e a interação remota com um jardim constituído de plantas, um manipulador robótico industrial e câmeras de vídeo. É possível acessar o sistema de duas maneiras: como visitante ou como membro, neste último caso é preciso preencher um formulário e aguardar o recebimento de uma senha para que seja possível efetuar o login. Enquanto o usuário visitante pode apenas visualizar o jardim, o usuário membro pode plantar, irrigar e monitorar o progresso das plantas por meio da câmera de vídeo acoplada ao manipulador industrial (Figura 2.21).

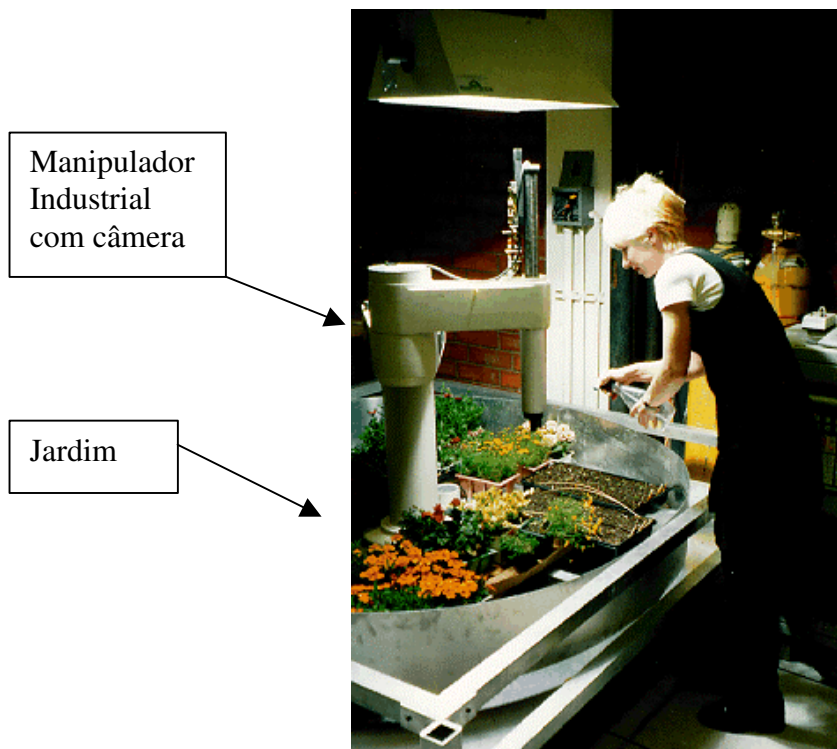


Figura 2.21 Manipulador do Telegarden

A interface com o usuário ocorre por meio de um navegador WEB e é composta por uma janela para visualização do jardim e por um gráfico que representa os movimentos do manipulador robótico (Figura 2.22). Clicando na imagem do manipulador robótico é possível atuar sobre ele alterando sua posição. Existem opções para alterar a composição da interface gráfica, possibilitando a personalização da mesma.



Figura 2.22 Interface gráfica do Telegarden

2.2.3. Bigsignal 2000

Este projeto foi baseado em um experimento científico coordenado pela NASA, denominado NOMAD, constituído por um robô móvel que se deslocava pelo continente Antártico em busca de meteoritos. O projeto Bigsignal propiciou que alunos de diversas escolas norte-americanas acompanhassem os dados provenientes do robô em tempo real, de forma semelhante ao que é feito pelos cientistas. Uma agenda de trabalho incluía as atividades de:

- Obtenção de informações sobre robótica, o continente Antártico, meteoritos e a missão NOMAD.
- Organização de grupos de trabalho e contato com o ambiente de controle do robô na WEB.
- Interação com os dados provenientes do robô como vídeo transmitido do local, informações dos sensores e instrumentos.
- Análise e elaboração de relatórios com as informações obtidas.

Os trabalhos eram realizados a partir de um site na WEB (Figura 2.23), acessível em <http://www.bigsignal.net/2000/index2.html>; por meio dele os alunos podiam comunicar-se e acessar a base de dados do projeto. Também por meio de uma interface gráfica WEB (Figura 2.24) era possível acessar o robô (Figura 2.25) que estava na Antártida, visualizar vídeos gerados por suas câmeras e acompanhar o estado dos diversos sensores.

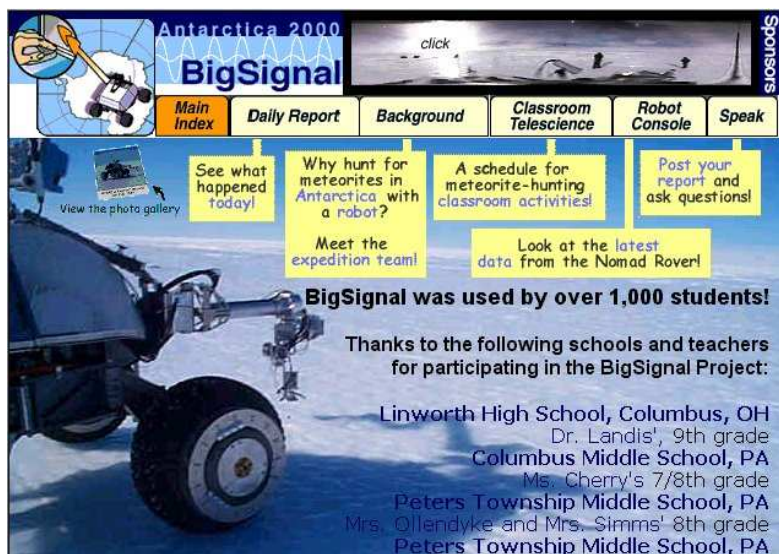


Figura 2.23 Interface gráfica do projeto Bigsinal

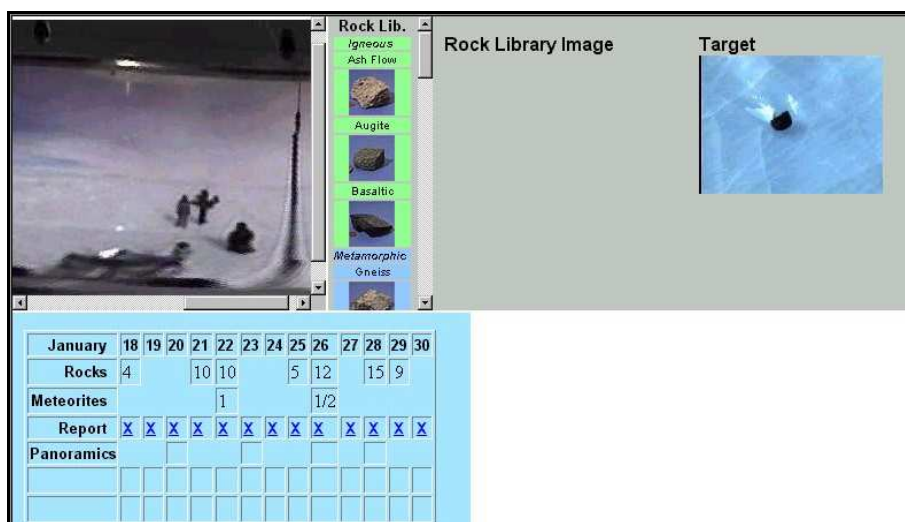


Figura 2.24 Interface para exibição de vídeo

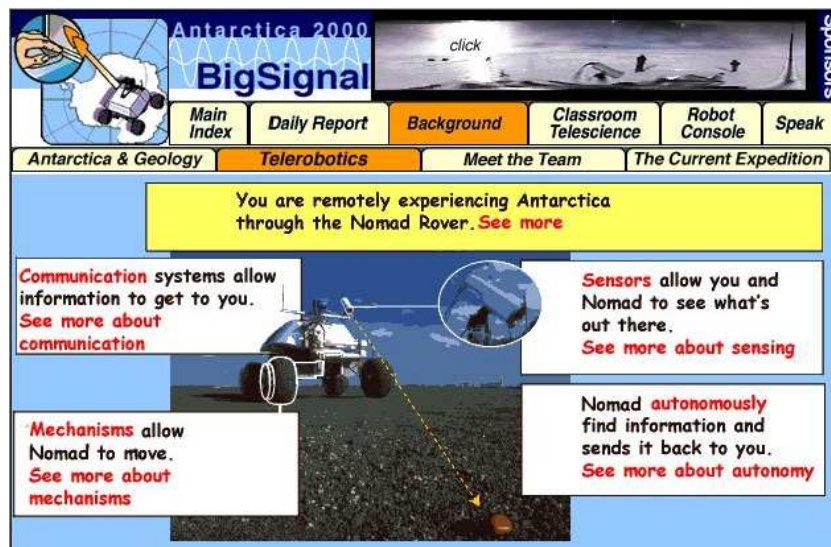


Figura 2.25 Robô do projeto Bigsignal

2.2.4. Projeto Redrover

Este sistema de operação remota foi desenvolvido inicialmente para simular as operações do European Mars Rover MIDD. Neste contexto a interface de usuário para teleoperação foi implementada, para realizar avaliações nos sistemas de sensores e aspectos operacionais. Após o final dos testes os sistemas foram transferidos para a Universidade FH Ravensburg-Weingarten para serem usados com fins educacionais tendo sido utilizado em cursos de telemática. Uma interface de usuário simplificada (Figura 2.26) foi desenvolvida e é acessível a partir do endereço <http://redrover.ars.fh-weingarten.de/index.htm>, permitindo ao usuário controlar a direção e o deslocamento de um robô móvel. Uma janela apresenta o vídeo com a imagem do robô em tempo real.



Figura 2.26 Interface gráfica do projeto Redrover

A cooperação entre diversas universidades, e em particular a Utah State University, tem permitido o desenvolvimento de atividades no projeto Redrover.

2.2.5. RexLab

O RexLab (Laboratório de Experimentação Remota) é um projeto desenvolvido pela Universidade Federal de Santa Catarina. Trata-se de um sistema combinado de um computador atuando como servidor e dispositivos como osciloscópios, geradores de função e multímetros, entre outros, que podem ser monitorados e controlados remotamente. Para que o usuário possa utilizar o sistema é preciso de um programa cliente que deve ser obtido no website do projeto. O acesso ao RexLab é feito pelo endereço: <http://www.inf.ufsc.br/~jbosco/lexrem3p.htm>.

2.3. Considerações sobre o capítulo

Dos conjuntos para robótica apresentados os que podem ser encontrados no mercado atualmente são os produtos da empresa Lego e alguns produtos do CMA, como o Coach Junior.

Os conjuntos de robótica disponíveis no mercado têm privilegiado a facilidade na utilização em razão de atenderem a um público consumidor interessado apenas no entretenimento.

Nenhuma dos conjuntos estudados dispõe de ferramentas que permita o controle dos dispositivos robóticos pela internet, sendo esta uma característica que será implementada neste trabalho.

Os ambientes de telerobótica descritos podem ser acessados pela Internet ainda que não exista regularidade no funcionamento de alguns recursos principalmente os relacionados ao controle e visualização dos robôs.

Nesses ambientes de telerobótica se observa que é possível executar controles por meio de comandos simples, mas sem a oportunidade de programar seqüências de ações, limitando as aplicações educacionais que poderiam ser mais ricas caso o aprendiz dispusesse de meios para programar por meio de seqüências de comandos as ações a serem executadas pelo dispositivo robótico.

Capítulo 3

Estrutura do Ambiente de Robótica implementado

Neste capítulo são apresentados a estrutura e os elementos constituintes do Ambiente de Robótica Educacional (ARE) desenvolvido neste trabalho. Nos capítulos seguintes é apresentado em detalhes a implementação de cada um destes elementos.

O ARE desenvolvido neste trabalho é constituído por um conjunto de aplicativos e pelo hardware. Alguns programas, como a linguagem Logo na versão SuperLogo 3.0, servidor WEB e servidor de Chat por estarem disponíveis não precisaram ser desenvolvidos. Outros, como o programa CGI para telerobótica, o driver de programa RCX/SuperLogo, e o programa para o microcontrolador da Interface de Hardware, tiveram que ser implementados.

Componentes de hardware desenvolvidos:

- Interface de Hardware para Robótica (IHR) para controle de servomotores e com entradas para sensores
- Sensores de temperatura, luminosidade e umidade.
- Módulos receptor/transmissor de infravermelho.
- Módulos receptor/transmissor de radiofrequência.

Elementos de programa implementados:

- Procedimentos SuperLogo para controle de servomotores da IHR.
- Procedimentos SuperLogo para capturar o estado dos sensores conectados à IHR.
- Interface de programa RCX/SuperLogo que implementa a comunicação entre o Lego RCX e o SuperLogo.
- Procedimentos SuperLogo para controle de sensores e motores do RCX.
- Programa CGI, utilizado no ambiente de telerobótica.
- Procedimentos SuperLogo para operação com o programa CGI.

Produtos disponíveis que foram integrados ao ARE:

- Linguagem de programação SuperLogo 3.0, versão em português do Logo, traduzida e adaptada pelo NIED (Núcleo de Informática Aplicada a Educação) com base no Microsoft Windows Logo 6.4.
- Servidor Xitami, programa servidor WEB da empresa IMATIX, distribuído em regime de freeware.
- Programa para captura e servidor de vídeo WEBCAM 2.2 da empresa TrueTech, distribuído em regime de freeware.
- Servidor de chat Infochat da empresa de mesmo nome, distribuído em versão demonstração.
- Lego RCX, conjunto composto pelo tijolo RCX e acessórios de montagem.

Esses recursos foram integrados e tornados acessíveis a partir de comandos do SuperLogo, para que tanto no modo presencial quanto no remoto, via Internet, o aprendiz possa montar e programar dispositivos robóticos.

A figura 3.27 ilustra os vários elementos do ARE e a forma como eles se inter-relacionam.

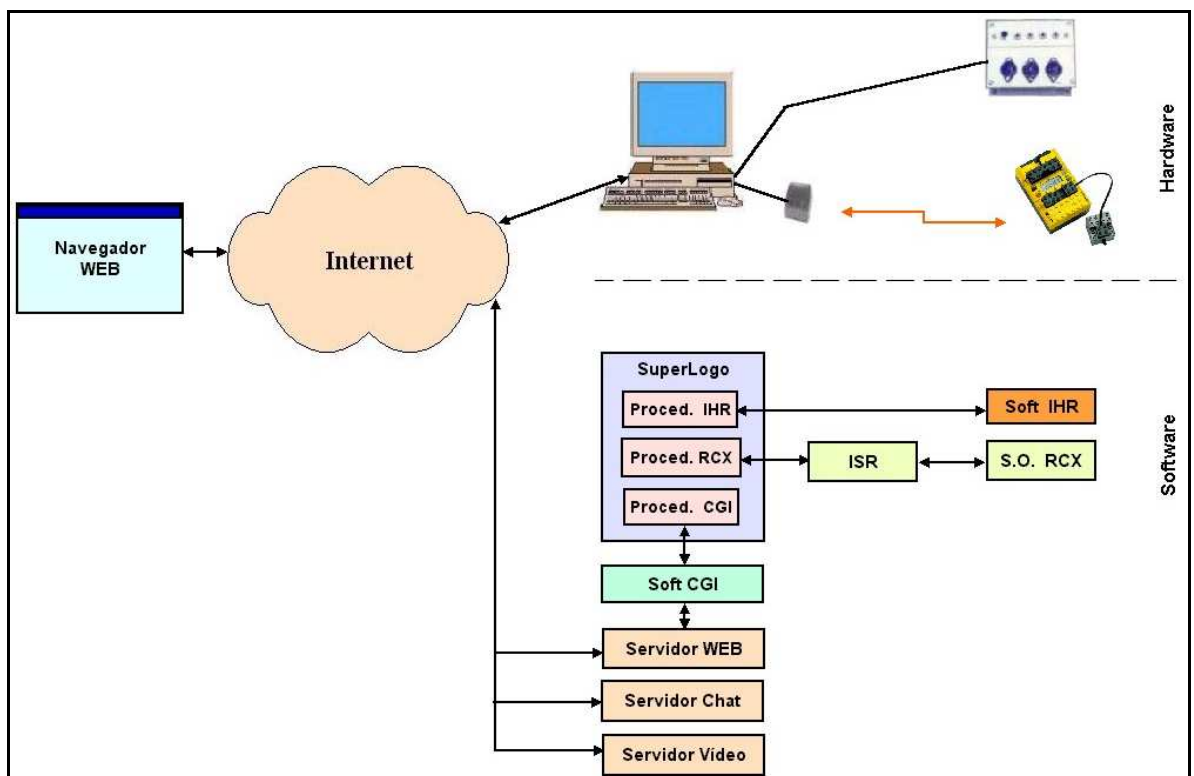


Figura 3.27 Estrutura do ambiente de robótica educacional ARE

3.1. Descrição dos componentes implementados

A seguir trazemos uma breve descrição dos programas e do hardware desenvolvidos. Os detalhes de projeto e de implementação são apresentados nos capítulos subseqüentes.

3.1.1. Interface de Hardware para Robótica (IHR)

O propósito da IHR é possibilitar que comandos enviados pelo computador atuem no acionamento de motores e possam receber dados provenientes de sensores. A IHR comunica-se com o computador PC por intermédio da porta paralela (padrão Centronics) e dispõe de recursos para controlar três servomotores e receber dados provenientes de quatro sensores digitais.

A alimentação da IHR, e dos sensores e servomotores é feita por uma fonte de alimentação ou baterias no caso de operação com os módulos de radiofrequência ou infravermelho.

A fim de prover os recursos citados é utilizado o microcontrolador PIC16F84 produzido pela empresa Microchip. Fazendo o microcontrolador responsável por todas as funções da IHR, acrescentando-se apenas circuitos para reforço (buffer) de sinal, obteve-se uma montagem compacta e pelo pequeno número de componentes, mais confiável.

Circuitos para sensores que necessitam de amplificação e de condicionamento como os de temperatura e luminosidade foram projetados para serem conectados a esta interface.

A fim de possibilitar que a comunicação entre a interface e o computador pudesse ocorrer sem cabos de conexão, permitindo maior grau de liberdade ao dispositivo criado pelo aprendiz, foram desenvolvidos circuitos para comunicação por radiofrequência e por infravermelho.

3.1.2. Interface SuperLogo RCX (ISR)

Em razão do SuperLogo não dispor de recursos para comunicação com o protocolo do Lego RCX, foi preciso desenvolver um programa que intermediasse esta comunicação.

A abordagem utilizada na implementação foi a de troca de mensagens. Assim, quando o aprendiz envia um comando por meio de um procedimento SuperLogo, é gerada uma mensagem que será interpretada pela ISR e posteriormente enviada ao Lego RCX. O mesmo acontece no sentido inverso: informações oriundas do Lego RCX e recebidas

pela ISR são convertidas em mensagens que podem ser recuperadas e utilizadas pelo aprendiz. Desta maneira é possível acionar motores e obter dados de sensores conectados ao Lego RCX a partir do SuperLogo.

3.1.3. Procedimentos SuperLogo

O controle dos diversos recursos disponíveis no Lego RCX e IHR, quer seja no modo presencial ou no modo à distância, é feito a partir do Superlogo. Para tanto foram implementados procedimentos específicos para cada recurso de hardware e para operação remota via Internet. No desenvolvimento destes procedimentos foi adotado o critério de que seus nomes e sua forma de utilização obedeceriam à estrutura da estética Logo facilitando a utilização por aprendizes que tenham familiaridade com esta linguagem.

Para controle dos recursos da IHR e do Lego RCX foram desenvolvidos procedimentos específicos, com funções para controlar o acionamento de motores e obter dados de sensores conectados às suas entradas.

Para operações remotas via Internet foram implementados procedimentos SuperLogo, com a função de receber e executar os comandos que o usuário elaborou no navegador e que foram processados pelo programa CGI.

3.1.4. Programa CGI

Operações remotas via Internet são constituídas de comandos que o aprendiz digita numa área de edição do navegador, e que a seguir são enviadas ao servidor WEB. Ao receber esses dados o servidor os encaminha ao programa CGI, o qual executa um processamento para decodificar essas informações e torná-las disponíveis para utilização pelo procedimento SuperLogo para telerobótica.

3.2. Considerações sobre capítulo

Neste capítulo foram apresentados os vários módulos desenvolvidos e que, integrados, constituem o ambiente de robótica educacional. Os capítulos seguintes 4, 5 e 6 trazem os detalhes relacionados à implementação desses módulos de hardware e dos programas.

Capítulo 4

Implementação da Interface de Hardware para Robótica

A Interface de Hardware para Robótica (IHR) é um dos elementos do ambiente de robótica e tem como funções converter os bytes de instruções enviados pelo computador em sinais elétricos adequados ao acionamento de servomotores e converter as grandezas físicas presentes nos sensores em instruções que serão enviados ao computador através da porta centronics.

A IHR dispõe de três saídas para acionamento de servomotores e quatro entradas digitais para utilização com sensores. A comunicação com o computador ocorre por meio de conexão à porta Centronics. Circuitos auxiliares conectados à IHR possibilitam que a comunicação entre computador e a IHR ocorra por Radiofrequência ou Infravermelho, permitindo o desenvolvimento de dispositivos, como robôs móveis, e outros, nos quais seja conveniente a operação sem cabos de conexão (Figura 4.28).

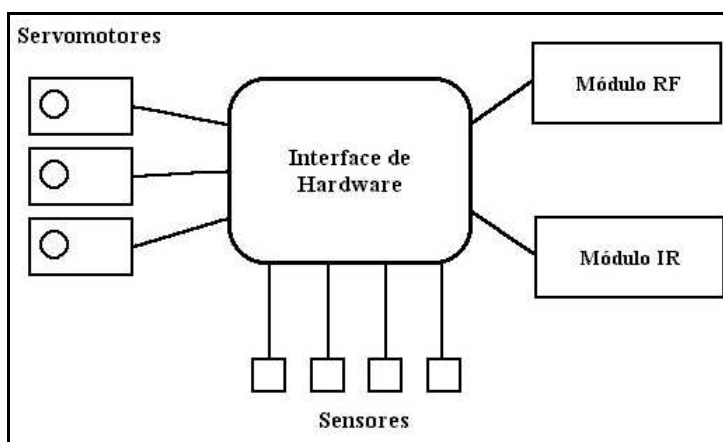


Figura 4.28 Diagrama da interface de hardware

4.1. Hardware da Interface

A IHR tem como principal componente o microcontrolador PIC16F84. Por ser um componente programável necessita ter um programa instalado em sua memória de programa para que possa desempenhar alguma função.

O PIC16F84 (Microchip, 2002) é um microcontrolador de 18 pinos fabricado pela empresa MICROCHIP empregando tecnologia de fabricação CMOS. Dispõe de memória FLASH para programação o que permite a reprogramação sem a necessidade de utilizar luz ultravioleta. Necessita de poucos componentes externos, o que reduz custos e o consumo de energia, aumentando a confiabilidade.

Características do PIC16F84:

- 13 portas configuráveis como entrada ou saída.
- 4 interrupções disponíveis (TMRO, Externa, Mudança de Estado e EEPROM).
Memória de programação FLASH, que permite a gravação do programa diversas vezes no mesmo chip.
- RAM de 68 bytes.
- Memória EEPROM (não volátil) interna de 64 bytes.
- Via de programação com 14 bits e 35 instruções.

Os 13 pinos de I/O do PIC16F84 são separados em dois grupos denominados PORTAS. Desta forma temos a Porta A e a Porta B as quais nos referimos como PORTA e PORTB respectivamente.

PORTA possui cinco pinos que podem ser configurados como entrada ou saída, e seus nomes são definidos como RA0, RA1, RA2, RA3 e RA4. O pino referente ao RA4 pode ser configurado também para incrementar o contador TMR0, que será descrito mais adiante.

PORTB possui oito pinos configuráveis que podem atuar como entrada ou saída, sendo seus nomes RB0, RB1, RB2, RB3, RB4, RB5, RB6 e RB7. O RB0 pode ser utilizado também para gerar a interrupção externa; os pinos de RB4 a RB7 podem gerar a interrupção por mudança de estado.

A alimentação é aplicada nos pinos Vss (GND) e Vdd (+5Vcc). A tensão de alimentação nominal dos PIC16F84 é de 5Vcc, mas a faixa de variação desta tensão vai de 2.0 a 6.0 Vcc.

O oscilador deve ser ligado aos pinos OSC1 e OSC2.

O pino denominado MCLR, é o Master Clear externo. Sempre que este pino for colocado em nível lógico baixo (GND), o processamento do programa em execução será paralisado. Ao ser colocado em nível alto (+5V), a execução do programa será retomada do ponto inicial.

Os microcontroladores PIC apresentam uma estrutura de máquina interna do tipo Harvard, enquanto grande parte dos microcontroladores tradicionais apresenta uma arquitetura tipo Von Neumann. A diferença está na forma como dados e programas são processados pelo microcontrolador. Na arquitetura tradicional, tipo Von Neumann, existe apenas um barramento (bus) interno (geralmente de 8 bits), por onde passam as instruções e os dados. Já na arquitetura tipo Harvard existem dois barramentos internos, sendo um de dados e outro de instruções (Tanenbaum, 1999). No caso dos microcontroladores PIC, o barramento de dados é sempre de 8 bits e o de instruções pode ser de 12, 14 ou 16 bits, dependendo do microcontrolador. Esse tipo de arquitetura permite, que enquanto uma instrução está sendo executada outra instrução seja "buscada" da memória, o que torna o processamento mais rápido. Além disso, como o barramento de instruções é maior do que 8 bits, o OP CODE da instrução já inclui o dado e o local onde ela vai operar (quando necessário), o que significa que apenas uma posição de memória é utilizada por instrução, economizando assim memória de programa.

Desta forma, observa-se que dentro da palavra do OP CODE, que pode ser de 12, 14 ou 16 bits, não sobra muito espaço para o código da instrução propriamente dito. Por isso, os PICs utilizam uma tecnologia chamada RISC, que significa Reduced Instruction Set Computer (Computador com Conjunto de Instruções Reduzido). Desta forma, o PIC16F84 possui cerca de 35 instruções, muito menos que os microcontroladores convencionais (CISC), que chegam a possuir mais de 100 instruções.

ESTRUTURA INTERNA

No diagrama de blocos (Figura 4.29), podem ser visualizados os diversos elementos que compõe o microcontrolador PIC16F84.

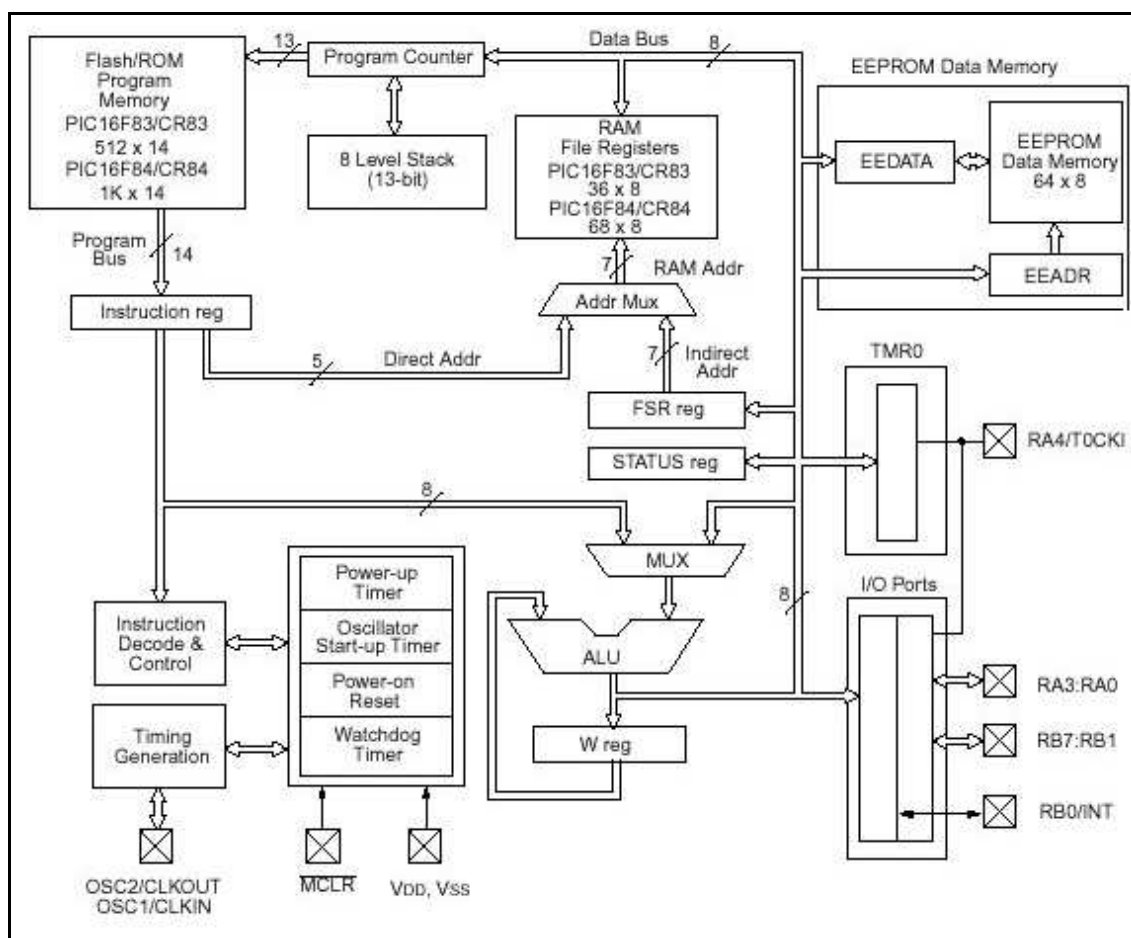


Figura 4.29 Diagrama do microcontrolador PIC16F84

A ULA (Unidade Lógico Aritmética) é diretamente ligada ao registrador W (work reg). No canto superior esquerdo encontra-se a memória de programa, e saindo deste bloco tem-se um barramento de 14 bits (Program Bus 14). Mais ao centro está a memória de dados (RAM); que já possui um barramento de 8 bits (Data Bus 8), conforme explicado na definição da arquitetura Harvard. Ao lado direito encontra-se o primeiro periférico, a EEPROM (memória de dados não volátil). Outros periféricos também podem ser observados, tais como: o timer (TMR0) e as portas paralelas (I/O Ports). Observa-se que ligado ao timer está conectado o pino RA4/TOCKI. Este pino, em conjunto com o timer forma o contador externo. Mais ao centro, há o registrador de status (STATUS reg); algumas informações importantes sobre as operações aritméticas da ULA ficam armazenadas neste registrador. Na parte superior está o contador de linha de programa (Program Counter) e a pilha de 8 níveis (Stack). A comunicação entre todos os periféricos é

feita por meio de um barramento de 8 vias. Existem ainda os circuitos internos de reset, osciladores e Watchdog Timer (WDT).

OS CICLOS DE MÁQUINA

Nos microcontroladores PIC, o sinal do clock é internamente dividido por quatro. Portanto, para um clock externo de 4MHz, temos um clock interno de 1 MHz, e conseqüentemente, cada ciclo de máquina dura 1 microssegundo.

A divisão do clock por quatro forma as fases Q1, Q2, Q3 e Q4. O contador de programa é incrementado automaticamente na fase Q1 do ciclo de máquina e a instrução seguinte é buscada na memória de programa e armazenada no registrador de instruções no ciclo Q4. Ela é decodificada e executada no próximo ciclo, no intervalo de Q1 a Q4. Esta característica de buscar a informação em um ciclo de máquina e executá-la no próximo ciclo é conhecida como PIPELINE. Ela permite que quase todas as instruções sejam executadas em apenas um ciclo, gastando assim 1 microssegundo (para um clock de 4MHz) o que torna o sistema muito mais rápido. As únicas exceções referem-se às instruções que geram "saltos" no contador de programas, como chamadas a rotinas e retornos. Ao executar essas instruções, o PIPELINE deve ser primeiramente limpo para depois poder ser carregado novamente com o endereço correto, consumindo para isso 2 ciclos de máquina. Esse PIPELINE é facilmente implementado, devido à arquitetura Havard.

Como já foi dito durante a explicação da arquitetura Havard, o PIC16F84 possui barramentos diferenciados para as memórias de programa e de dados. Desta forma, podemos concluir também que essas memórias são totalmente separadas. No caso do PIC 16F84, existe ainda uma terceira memória: a memória não volátil - EEPROM.

MEMÓRIA DE PROGRAMA

Devido à estruturação Havard, a memória de programa do PIC pode ser de 12, 14 ou 16 bits, variando de modelo para modelo. Na maioria dos modelos, essa memória é do tipo EPROM, que pode ser gravada somente uma vez para PICs normais, ou várias vezes, no caso de PICs janelados (que podem ser apagados por meio de luz ultravioleta). Existem ainda modelos que possuem memória de programa do tipo FLASH, que pode ser gravada várias vezes sem necessidade de apagar a gravação anterior, tecnologia empregada no

PIC16F84. A possibilidade de fazer a gravação sem a necessidade de expor o microcontrolador à luz ultravioleta facilita o trabalho para o desenvolvedor. Por outro lado os microcontroladores com tecnologia FLASH apresentam um custo maior em relação aos que utilizam tecnologia de EPROM.

VETOR DE RESET

Trata-se do primeiro endereço da memória de programa que será executado quando o PIC começar a operar (ao fornecer alimentação ou um reset). No PIC16F84 o reset aponta para o endereço 0x00.

VETOR DE INTERRUPÇÃO

As rotinas de interrupções são armazenadas na área de programação, juntamente com todo o resto do programa. No entanto, existe um endereço que é reservado para o início do tratamento de todas as interrupções. Esse endereço é denominado vetor de interrupção e encontra-se sempre na posição 0x04.

PILHA (STACK)

A pilha é um local, totalmente separado da memória de programação, no qual são armazenados os endereços de retorno, quando são utilizadas instruções de chamadas de rotinas. Quando o programa é desviado para o início de uma rotina por meio da instrução correta, o endereço seguinte ao ponto que estava sendo rodado é armazenado na pilha para que, ao final da rotina, o programa possa retornar ao ponto onde ocorreu o desvio. Caso se tente chamar um número de rotinas maior que o tamanho da pilha, o endereço de retorno mais antigo será perdido.

A MEMÓRIA DE DADOS

A memória de dados do sistema é a RAM, utilizada para guardar todas as variáveis e registradores utilizados pelo programa. Essa memória armazena dados de 8 bits e é volátil, ou seja, quando o PIC16F84 é desligado ela é automaticamente perdida. Pode-se dividi-la em dois grupos: registradores especiais e registradores de uso geral. A memória de

dados é dividida em mais de um banco e o acesso aos endereços é feito com o auxílio de chaves que controlam o banco que está sendo utilizado no momento.

REGISTRADORES ESPECIAIS

Nessa região da memória encontram-se todos os registradores especiais, denominados SFRs, utilizados pelo microcontrolador para a execução do programa e processamentos da ULA. Esses registradores ocupam espaço na RAM e podem ser acessados da mesma maneira que as variáveis do sistema, com mudança somente do endereço de acesso. Podem ser escritos ou lidos tanto pelo usuário quanto pelo hardware, são armazenados na parte baixa da memória (início dos endereços) e estão distribuídos em mais de um banco de memória.

REGISTRADORES DE USO GERAL

Trata-se de uma área destinada ao armazenamento de variáveis definidas pelo usuário para serem escritas e lidas pelo programa. O tamanho dessa memória varia de acordo com o modelo de PIC e também pode ocupar mais de um banco.

EEPROM

O PIC16F84 possui ainda uma terceira memória que também pode ser utilizada pelo usuário para guardar dados. Entretanto, ao contrário da memória de dados estudada anteriormente, esta é uma EEPROM, isto é uma memória não volátil, que mantém as informações mesmo sem alimentação.

INTERRUPÇÕES

Como o próprio nome diz, uma interrupção serve para interromper o programa imediatamente. Desta maneira, é possível programar ações que serão executadas ao ocorrer uma interrupção. As interrupções são ações tratadas diretamente pelo hardware, o que as torna muito rápidas e disponíveis em qualquer ponto do sistema. Assim, quando uma interrupção acontece, o programa é paralisado, uma função específica (definida pelo programador) é executada, e depois o programa continua a ser executado a partir do ponto em que estava.

INTERRUPÇÃO DO TIMER 0

Essa interrupção acontece sempre que um contador de tempo interno, denominado TMR0 (Timer 0), estoura, ou seja, como ele é um contador de 8 bits, sempre que ele passar de 0xFF para 0x00. Ela é utilizada normalmente para a contagem de tempo. Como pode acontecer a qualquer momento, a contagem de tempo fica precisa, não dependendo de análises constantes durante o programa para garantir que o tempo seja contado. O TMR0 pode tanto ser incrementado internamente pelo clock da máquina, como por um sinal externo. Neste último caso, ele passa a ser um contador de pulsos, podendo ser utilizado para outras finalidades.

INTERRUPÇÃO EXTERNA

Essa interrupção é gerada por um sinal externo ligado a uma porta específica do PIC, que no caso é a porta RB0, caso ela esteja configurada como entrada. Desta maneira, podemos identificar e processar imediatamente um sinal externo. Ela é utilizada para diversas finalidades, como, por exemplo, para comunicação entre microcontroladores, garantindo o sincronismo, o reconhecimento de um sensor ou outro sinal do sistema que necessite de uma ação imediata.

INTERRUPÇÃO POR MUDANÇA DE ESTADO

A interrupção externa estudada acima, funciona somente na borda de subida ou na borda de descida (quando o sinal lógico sobe ou desce), dependendo de como ela foi configurada. Já a interrupção por mudança de estado acontece em ambos os casos. Essa interrupção, por sua vez, está ligada às portas RB4, RB5, RB6 e RB7 simultaneamente. Por isso, se essas portas forem configuradas como entradas, a mudança de estado em qualquer uma delas irá gerar a interrupção.

INTERRUPÇÃO DE FIM DA ESCRITA NA EEPROM

Essa interrupção serve para detectar o final de uma rotina de escrita nessa memória. A utilização da interrupção não é obrigatória para que a escrita funcione, mas como o processo de escrita na EEPROM é lento, em alguns sistemas a sua utilização pode ser necessária para evitar uma parada durante a escrita.

TRATAMENTO DE INTERRUPÇÕES

Sempre que uma interrupção ocorre (qualquer uma delas), o programa guarda o endereço da próxima linha a ser executada na pilha, e desvia para um endereço fixo da memória de programação. Basta então, nesse endereço (0x04), escrever a rotina que irá reconhecer e tratar a interrupção ocorrida. Quando a rotina de interrupção for finalizada o programa automaticamente voltará para o ponto em que estava antes da interrupção ocorrer.

O HARDWARE DO PIC16F84

O hardware do PIC consiste apenas de um oscilador e um circuito de power-on reset(inicializa ao ser ligado).

O oscilador do microcontrolador é externo; existem basicamente quatro tipos de osciladores que podem ser utilizados com o PIC16F84:

- RC (resistor-capacitor); é o mais simples e o mais barato, porém menos preciso. Variações na frequência podem ocorrer, em função da tolerância dos componentes e das condições ambientais como, por exemplo, alteração na temperatura.
- Ressonador cerâmico; o custo é superior ao oscilador RC, mas apresenta maior estabilidade e precisão.
- Cristal; é a solução de maior precisão, porém de maior custo; sua utilização é recomendada quando se deseja extrema precisão. Por possuir internamente os circuitos para oscilação, o cristal pode ser ligado diretamente ao pinos OSC1 e OSC2.
- Híbrido; são cristais híbridos ou circuitos para oscilação, normalmente empregados quando se utiliza mais de um microcontrolador e é necessário manter o sincronismo entre eles.

A solução de oscilador escolhida para o projeto foi a RC (resistor-capacitor). Utilizando-se capacitores cerâmicos obtêm-se a estabilidade e precisão que o projeto necessita. O valor do capacitor utilizado é 22pF e o resistor 4,7 KOhms ficando neste caso a frequência de oscilação em 10 MHz. O Power-On Reset implementado é a solução sugerida pelo fabricante e consiste na conexão do pino MCLR ao Vcc por meio de uma

resistência de 1 KOhms. A Figura 4.30 apresenta o circuito esquemático para o microcontrolador.

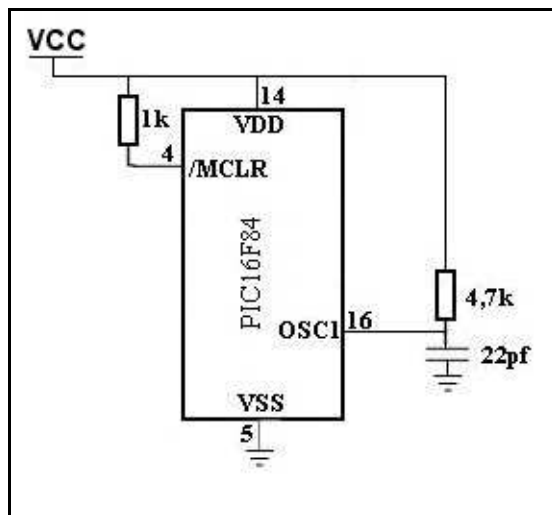


Figura 4.30 Circuito esquemático do bloco do microcontrolador

Além do PIC16F84, dois Circuitos Integrados 74LS245 são utilizados atuando no isolamento elétrico e reforço (buffer) dos sinais de entrada e saída garantindo um funcionamento estável e seguro da IHR.

O circuito integrado 74LS245 (Figura 4.31) é um transceiver de barramento octal com tecnologia TTL. Com este dispositivo é possível a transmissão de dados do barramento A ao B e também o inverso; o sentido é determinado pelo sinal lógico presente no pino DIR. O pino G pode ser usado para desabilitar o dispositivo, propiciando o isolamento do barramento. A tabela 4.1 apresenta as configurações possíveis para o dispositivo.

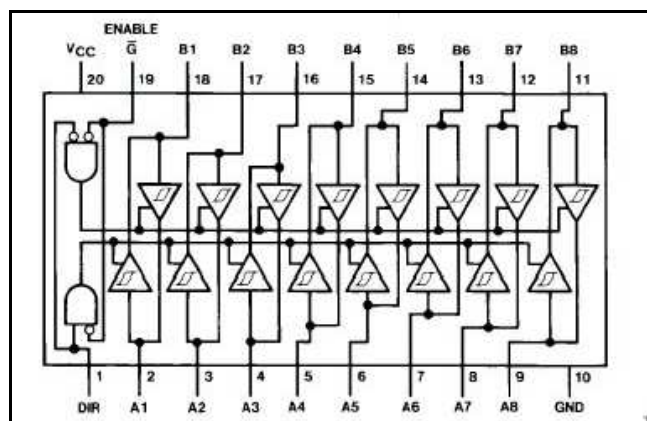


Figura 4.31 Circuito integrado 74LS245

Enable $\overline{G}$	Direction Control DIR	Operation
L	L	B data to A bus
L	H	A data to B bus
H	X	Isolation

H = High Level, L = Low Level, X = Irrelevant

Tabela 4.1 Configurações para o circuito integrado 74LS245

As instruções de 4 bits para controle dos servomotores conectados à IHR, são enviadas aos pinos de dados (D0, D1, D2, D3) da porta Centronics do computador. Na IHR estes sinais são recebidos pelo circuito integrado 74LS245 e a seguir vão para o PIC16F84. Os níveis lógicos que indicam o estado dos sensores passam por outro 74LS245 e são encaminhados para os pinos de entrada da porta Centronics ACK, BUSY, PE, ERR.

No circuito da IHR, servomotores e sensores são alimentados por uma tensão de 5 Vcc proveniente de fonte de alimentação do tipo utilizada em microcomputadores PC AT. Para utilização deste tipo de fonte foram necessárias algumas adaptações que envolveram retirar cabos e conectores originais, inserir uma chave liga/desliga e montar um cabo com conector adequado. A Figura 4.32 apresenta o circuito eletrônico completo da IHR.

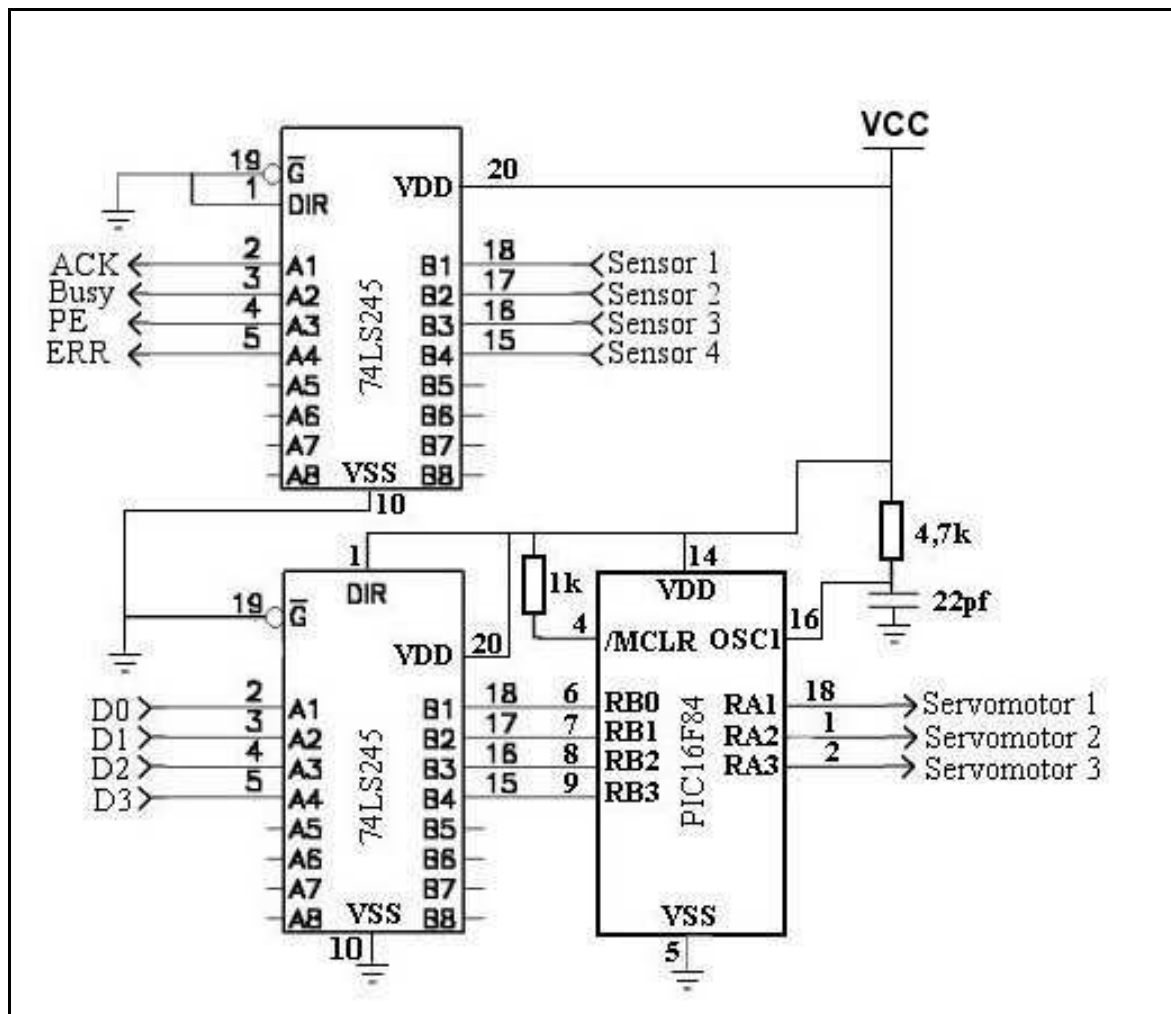


Figura 4.32 Circuito esquemático da interface de hardware

4.1.1. Estrutura Mecânica

Os circuito eletrônico foi montado em uma placa de circuito impresso, a qual é fixada em uma base de poliestireno. Sobre ela, sustentada por quatro parafusos, está um painel onde estão montados os conectores de saída de servomotores, de entrada dos sensores e da fonte de alimentação (Figura 4.33).



Figura 4.33 Fotografia da interface de hardware

Com o objetivo de facilitar a utilização pelo usuário e evitar a possibilidade de conexões incorretas foram empregados conectores diferenciados para os servomotores, sensores e fonte de alimentação.

Nas saídas para servomotores foram utilizados conectores DIN de cinco pinos, em razão de os servomotores necessitarem de um condutor para alimentação, outro para o terra e um terceiro para o sinal de controle.

4.1.2. Módulo de comunicação infravermelho

Para alguns projetos de robótica, como por exemplo, um robô móvel, a utilização da IHR com o cabo conectado ao computador pode limitar os movimentos e a distância em que o mesmo poderá se deslocar. Uma outra situação é quando se deseja coletar dados de um sensor distante do computador.

Para contornar esta limitação é preciso dotar a IHR de um sistema de comunicação com o computador onde não se utilize conexão física por meio de cabo. Uma das soluções possíveis é por meio do emprego de comunicação por luz no espectro de infravermelho.

A fim de implementar a comunicação por infravermelho entre a IHR e o computador PC, foram desenvolvidos circuitos para transmissão e recepção das instruções.

Na transmissão o circuito deve executar as seguintes funções:

- Codificar as instruções de entrada.
- Converter as instruções codificadas para o formato serial.
- Efetuar a modulação do sinal para transmissão pelo emissor infravermelho.

- Fornecer os níveis de corrente e de tensão necessários à excitação do LED (Light Emitter Diode) emissor de infravermelho.

Na recepção o circuito é responsável por executar as seguintes funções:

- Detectar e amplificar o sinal modulado contendo as instruções.
- Decodificar as instruções
- Apresentar as instruções no formato em que foram enviadas.

O diagrama da Figura 4.34 demonstra o processo de transmissão e recepção de instruções por infravermelho entre a IHR e o computador.

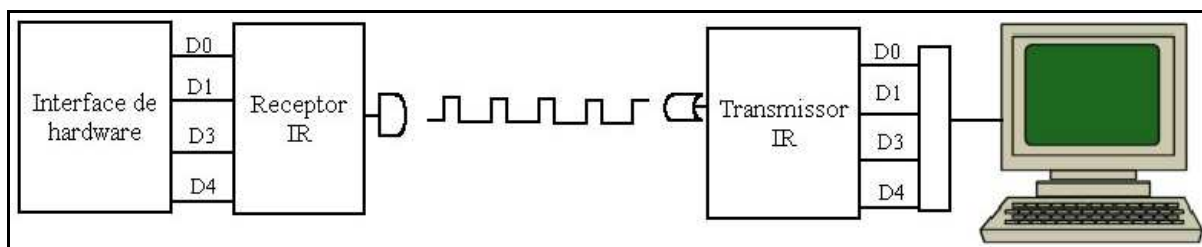


Figura 4.34 Diagrama da comunicação por infravermelho

4.1.2.1. Transmissor Infravermelho

No circuito para transmissão dos sinais por infravermelho foram utilizados: o circuito integrado MC145026, responsável pela codificação e conversão dos dados com as instruções para o formato serial; o circuito integrado CD4011 para implementação do oscilador usado no modulador; e um transistor BD435 que fornece os níveis de tensão e de corrente necessários à excitação do LED emissor de Infravermelho.

O circuito integrado MC145026 codifica nove bits de informação e os transmite no formato serial assim que o sinal transmissão habilitada TE (Transmit Enabled) esteja ativo. As nove entradas podem ser codificadas com dados trinários (0, 1, aberto), permitindo 19683 combinações.

A sequência de transmissão é iniciada ao ocorrer um nível baixo no pino TE. Cada vez que o pino TE é levado ao nível zero o encoder gera uma saída com duas palavras de dados idênticas. Se o pulso TE for mantido em zero, o codificador transmite continuamente as palavras.

Cada bit transmitido é codificado de acordo com os três estados possíveis: zero lógico, um lógico e aberto. A lógica zero é codificada como dois pulsos curtos, a lógica um por dois pulsos longos e consecutivos, e o estado aberto é codificado por um pulso longo seguido de um pulso curto. Quando TE é levado a zero, o oscilador é iniciado e um reset é gerado para inicializar a sequência de transmissão. No circuito desenvolvido os valores dos capacitores e resistores associados aos pinos CTC, RTC e RS do MC145026 propiciam uma frequência de operação de 1.71 KHz.

Os dados seriais disponíveis no pino Data Out seguem para um oscilador, implementado com duas portas NAND de um circuito integrado CD4011, a frequência de oscilação é de 38 KHZ e é determinada pelo capacitor de 10nF e pelos resistores de 220 KOhms e 13 KOhms.

A frequência de oscilação de 38 KHz foi definida por ser esta a frequência de operação do sensor IR utilizado no módulo receptor.

O sinal modulado passa por duas outras portas do CD4011, que atuam como buffer, e segue para o transistor BD435 que fornece os níveis de tensão e de corrente ao LED emissor de infravermelho conectado ao seu coletor.

Os dados podem ter origem no computador PC e ser enviados para controle da IHR ou podem ter origem na mesma, representando o estado de saída de sensores, para então serem encaminhados ao computador PC. O circuito completo do emissor de infravermelho está representado no desenho esquemático da Figura 4.35.

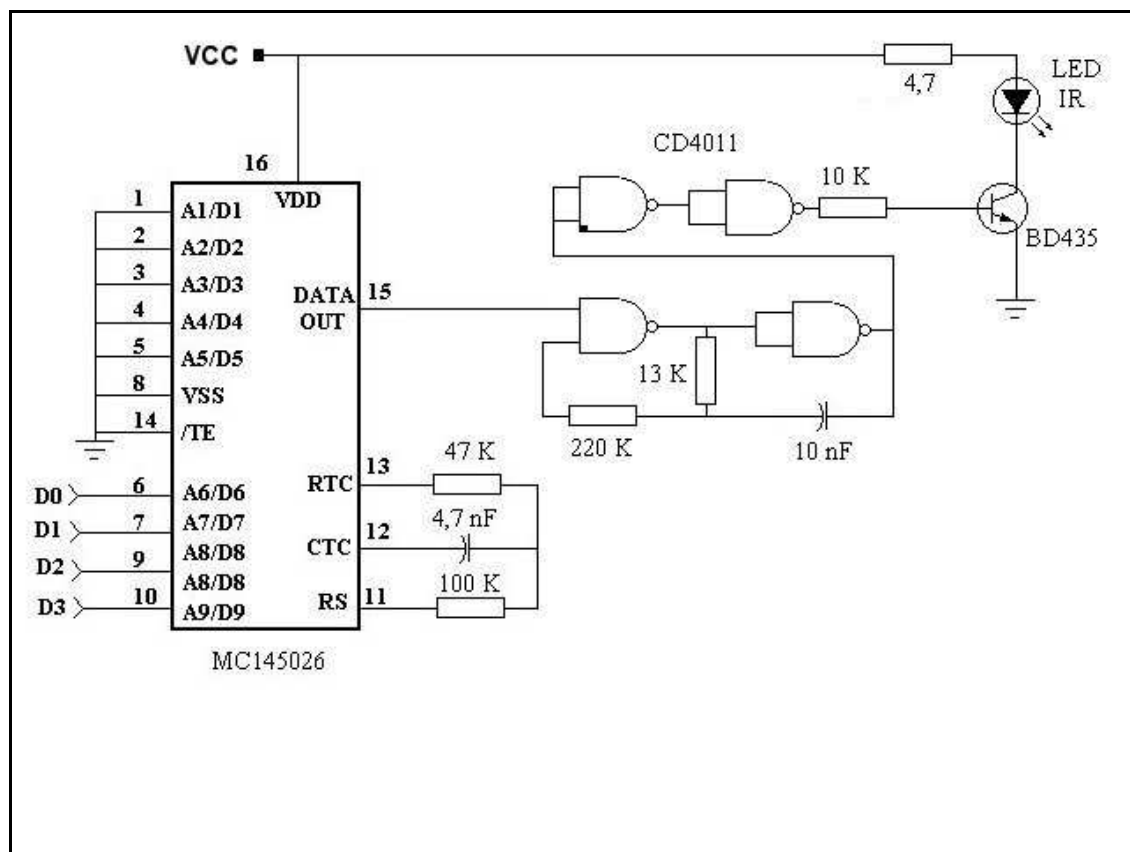


Figura 4.35 Circuito esquemático do transmissor de infravermelho

4.1.2.2. Receptor Infravermelho

O receptor de Infravermelho opera em conjunto com o transmissor de infravermelho; sua função é receber o sinal de infravermelho, amplificá-lo, demodulá-lo e decodificar os dados recebidos, que serão enviados à IHR .

Como sensor de infravermelho é utilizado o receptor monolítico PIC26043SM produzido pela empresa Kodenshi. Este receptor consiste de um fotodiodo, pré-amplificador e processador de sinais. Sua operação ocorre na frequência de 38 kHz e tem o sinal de saída invertido em relação à entrada.

O sinal infravermelho, após ser processado pelo receptor PIC26043SM, é entregue ao circuito formado pelo transistor BC557, cuja função é inverter o nível lógico do sinal; a seguir, o sinal é enviado ao circuito integrado MC145527.

O circuito integrado MC145527 é o decodificador que recebe o sinal com os dados no formato serial, efetua uma verificação para detecção de erros e apresenta os dados na

saída caso eles sejam válidos. Os dados transmitidos consistem de duas palavras de dados que são examinadas bit a bit, assim que são recebidas. Os primeiros cinco bits são considerados endereços e devem ter a mesma codificação do receptor. Se os bits de endereços forem iguais, os próximos 4 bits são armazenados e comparados com os últimos dados válidos armazenados.

Entre duas palavras de dados não são enviados sinais pelo período de três bits de dados. Assim que a segunda palavra codificada é recebida, o endereço deve combinar novamente; se isso ocorrer, os bits de dados são comparados novamente com os bits previamente armazenados. Se os dois conjuntos de bits forem iguais, os dados são transferidos para o latch de saída do circuito integrado, sendo mantidos até que novos dados válidos sejam recebidos. Nesse mesmo instante o pino Valid Transmission (VT) é colocado em nível lógico um, sendo mantido nesta condição até que uma situação de erro ocorra ou que não sejam recebidos sinais de entrada por um período correspondente a quatro bits de dados.

Um LED conectado ao pino VT permite o monitoramento do estado da comunicação entre os módulos receptor e transmissor de infravermelho. Os dados presentes nos pinos D6, D7, D8 e D9 são conectados diretamente à entrada de dados da IHR. O circuito do receptor infravermelho está ilustrado na Figura 4.36.

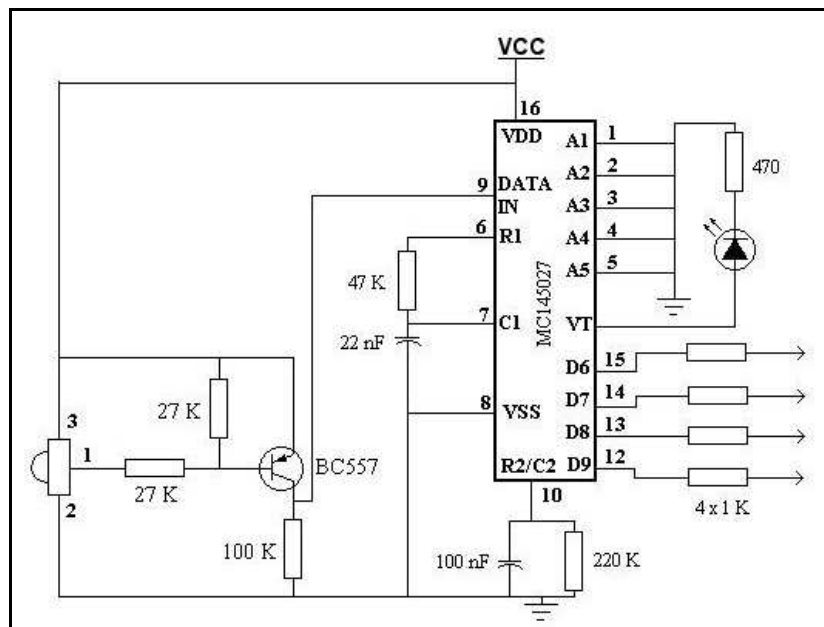


Figura 4.36 Receptor de infravermelho

4.1.3. Módulo de comunicação por radiofrequência

Para alguns projetos, o alcance de operação proporcionado pela comunicação por infravermelho pode ser insuficiente, para tais casos existe a opção de se utilizar a comunicação por radiofrequência.

Da mesma maneira que o infravermelho, para a comunicação por radiofrequência são necessários um transmissor e um receptor. Atualmente, é possível encontrar no mercado pequenos módulos de radiofrequência de baixa potência, que operam na faixa de UHF (Ultra High Frequency) e se mostraram adequados para utilização com a IHR.

No Brasil são encontrados com certa facilidade os módulos de radiofrequência produzidos pela empresa italiana Telecontrolli. Os modelos utilizados para implementação de comunicação por radiofrequência são os módulos receptor RR3 e transmissor RT4 (Figura 4.37), produzidos pela empresa Telecontrolli, cujas características são:

RR3

- Receptor super-regenerativo.
- Sensibilidade típica de -100 dBm 2.2 μ Vrms com carga de 50 Ohms.
- Frequência de operação 315 MHz.
- Tensão de alimentação de 5 V.
- Frequência de operação ajustada a laser.

RT4

- Potência de saída 7 dBm (alimentação =12V e tensão de entrada= 5 V).
- Consumo de corrente 4 mA.
- Tensão de alimentação de 2V a 14V.
- Frequência de operação 315 MHz.

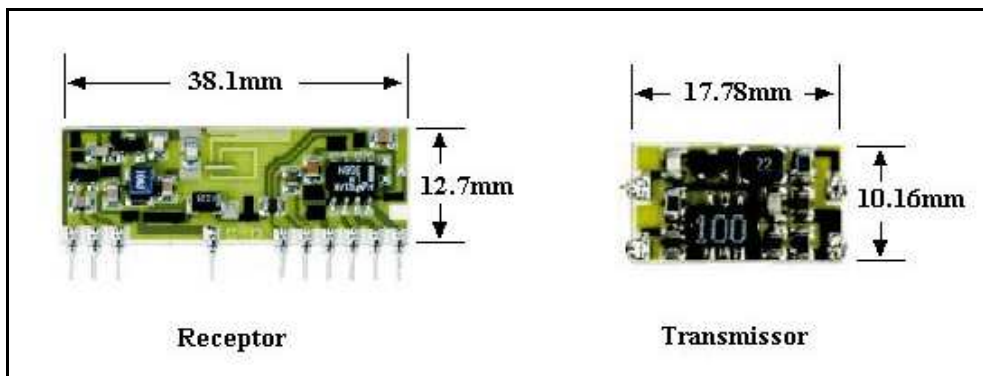


Figura 4.37 Módulos para radiofrequência

O sistema de comunicação por radiofrequência (Figura 4.38) é constituído dos módulos RR3 e RT4 e dos circuitos integrados para codificação e decodificação MC145026 e MC145027.

O circuito do transmissor é constituído pelo circuito integrado MC145026 com a mesma configuração utilizada no transmissor de infravermelho, dispensando a etapa de modulação do sinal. Os dados de saída são aplicados à entrada do módulo RT4.

O módulo RT4, por ser um transmissor completo dispensa, componentes adicionais requerendo apenas a conexão da linha de alimentação, do terra e da entrada de dados. A antena é constituída por um cabo rígido com a dimensão de 15 cm.

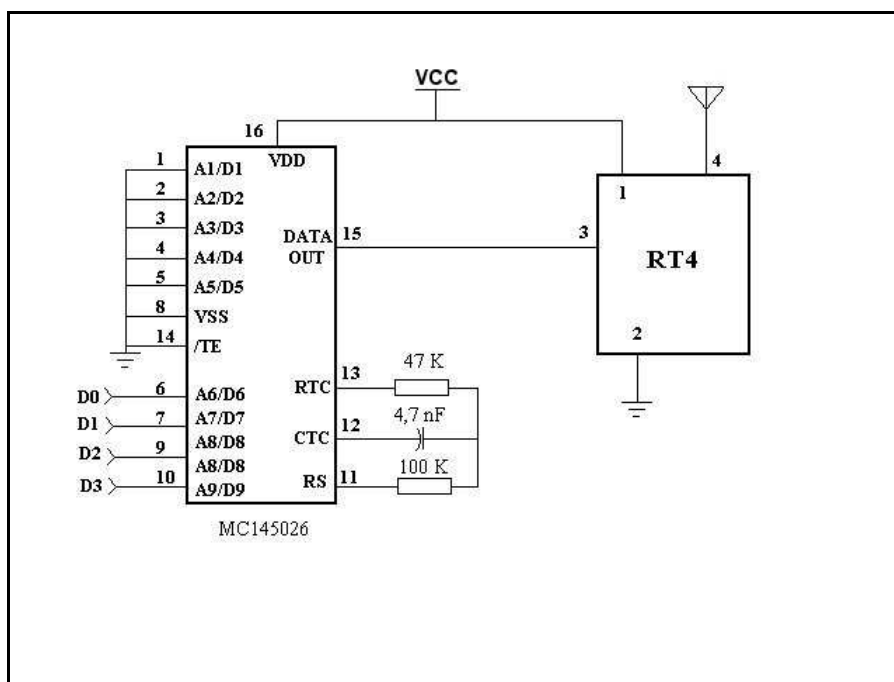


Figura 4.38 Circuito do transmissor de RF

Para o sistema de recepção (Figura 4.39) é empregado o circuito integrado decodificador MC145027 e o módulo de recepção RR3. A configuração do circuito integrado MC145027 é a mesma que foi utilizada no receptor de infravermelho. O módulo dispensa componentes externos para seu funcionamento, sendo necessário apenas a conexão às linhas de alimentação, de terra e da saída de dados. A antena é constituída de um cabo rígido com a dimensão de 15 cm. O sinal de saída do módulo RR3 é entregue diretamente à entrada de dados do circuito integrado MC145027. O resistor de 100 Ohms e os capacitores de 220 μ F e 100 nF presentes na entrada de alimentação do RR3 constituem um filtro contra ruídos.

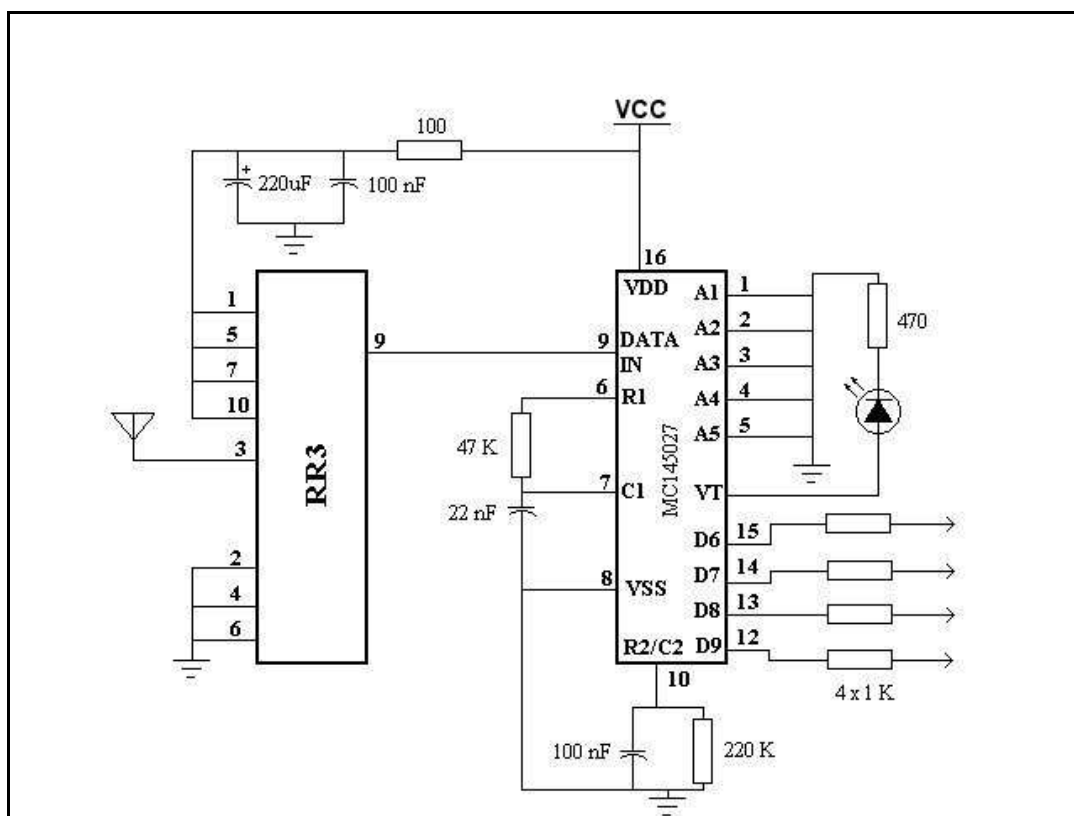


Figura 4.39 Circuito do receptor de RF

4.1.4. Servomotores

O servomotor é um atuador que dispõe de um eixo móvel, o qual pode ser posicionado com deslocamentos angulares específicos a partir do envio de um sinal elétrico codificado. Mantendo-se o sinal codificado, o servomotor manterá a posição angular do eixo. Alterações nas características deste sinal alteram o posicionamento do eixo.

Os servomotores utilizados neste projeto são do tipo empregado em aeromodelos e outros equipamentos rádio controlados (Figura 4.40).

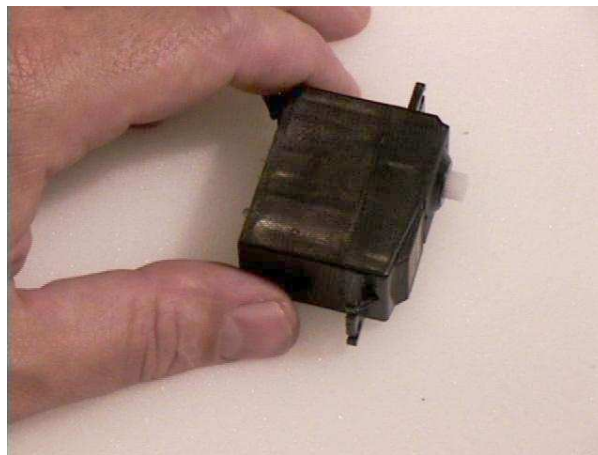


Figura 4.40 Servomotor

Algumas de suas características listadas a seguir justificaram a escolha deste tipo de servomotor:

- A dimensão: 4 cm de altura, 5cm de comprimento e largura de 2 cm.
- Torque de aproximadamente 480 gramas por centímetro.
- Peso aproximado de 40 gramas.
- Robustez e confiabilidade. Alguns dos servomotores utilizados neste projeto estão em uso intensivo há dois anos sem apresentar problemas.
- Custo, variando de 18 a 25 dólares, dependendo do fabricante e do modelo.

O servomotor é constituído fisicamente de engrenagens, de um motor DC, de um circuito eletrônico e uma caixa plástica que suporta esses componentes (Figura 4.41). A sua conexão elétrica ocorre por meio de um cabo de três vias onde são conectados à alimentação de 5V, o terra e o sinal de controle.



Figura 4.41 Conjunto de peças que constituem o servomotor

Conectado eletricamente ao circuito eletrônico e mecanicamente ao eixo de saída, existe um potenciômetro. Sua função é monitorar o ângulo em que o eixo está posicionado. Se o eixo estiver no ângulo correto determinado pelo sinal codificado, o servomotor permanece parado; se o ângulo estiver em desacordo com o determinado pelo sinal codificado, o motor irá deslocar-se até encontrar a posição correta.

Dependendo do fabricante, os servomotores podem realizar a rotação com variações de 0 a 270 graus. Contudo com algumas modificações elétricas e mecânicas é possível obter rotação completa; na seção 4.1.4.1 está descrito o processo de modificação.

Para controlar o posicionamento devem ser aplicados à entrada de controle do servomotor pulsos com largura variável ou PWM (Pulse With Modulation). O servomotor aguarda que ocorra um pulso a cada 20 milisegundos. A dimensão do pulso irá determinar o deslocamento do servomotor.

Em um servomotor com deslocamento máximo de 180 graus, um pulso de 1,5 ms irá posicionar o servomotor a 90 graus, também denominada posição neutra. Se o pulso for menor que 1,5 ms, o sentido de rotação será em direção ao ângulo 0, se maior que 1,5 ms, a rotação ocorrerá na direção do ângulo de 180 graus (Figura 4.42).

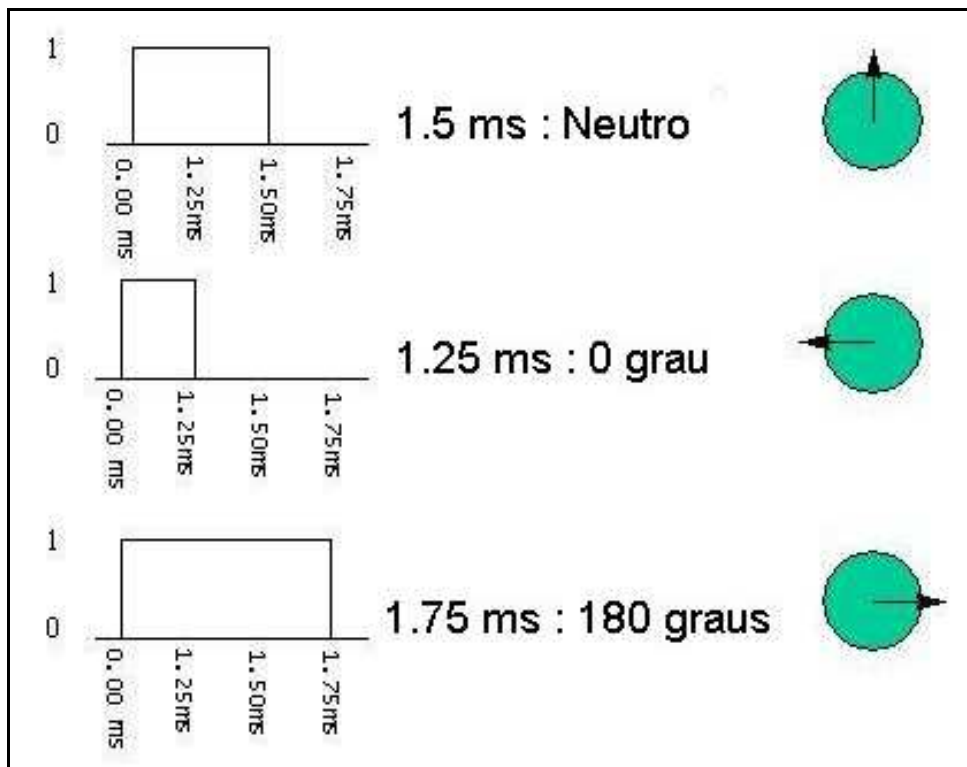


Figura 4.42 Temporização do servomotor

Portanto, controlar o posicionamento do eixo de um servomotor significa enviar pulsos com largura variável à entrada de controle, de acordo com o deslocamento desejado.

4.1.4.1. Alterando o servomotor para rotação completa

Em alguns projetos pode ser necessário o movimento com rotação contínua do servomotor, em tais casos o servomotor original não será útil, pois existe um limite para seu deslocamento. Todavia, é possível efetuar alterações para contornar esta limitação, nas partes mecânica e elétrica do servomotor. Na parte mecânica é preciso verificar qual engrenagem tem um limitador de giro e eliminá-lo (Figura 4.43). A modificação elétrica envolve a retirada do potenciômetro do circuito e a substituição por dois resistores com resistência correspondentes à metade da resistência do potenciômetro.



Figura 4.43 Engrenagem do servomotor

Com as modificações apontadas o servomotor passa a operar com rotação contínua. Se for aplicado um pulso menor que 1.5 ms o giro ocorre no sentido anti-horário, se maior que 1.5ms ele gira no sentido horário, a ausência de pulso interrompe o movimento.

4.1.5. Sensores

O sensor é normalmente um dispositivo baseado em um transdutor, capaz de converter uma grandeza de entrada física ou elétrica em um sinal de saída elétrico que, após processado por determinado algoritmo, torna-se adequado para ser interfaceado com um controlador de processo (Ruocco,1987). O elemento transdutor é o componente do sensor capaz de converter uma entrada física e não elétrica em determinada quantidade de energia elétrica. O transdutor, isoladamente não realiza nenhum processamento além da conversão de energia.

Os sensores e transdutores podem ser classificados de acordo com suas características de entrada e saída. Em relação à entrada estes dispositivos podem ser classificados como:

- Absolutos, quando, definindo-se um valor fixo de referência, o sinal elétrico de saída pode representar todos os valores possíveis de entrada sem ambigüidade.
- Incremental, quando uma referência não pode ser fixada para todos os pontos do campo de medida e cada ponto passa a ser a referência para o próximo.
- Analógicos, quando o sinal de saída é contínuo e proporcional a quantidade do sinal de entrada.
- Digital, se para uma variação contínua da grandeza de entrada o sinal de saída assume um número discreto de valores.

Quanto à saída os sensores e transdutores podem ser classificados como:

- Estáticos, quando operam em ambientes onde as condições ambientais como pressão, temperatura e umidade são controladas para que não ocorram variações significativas.
- Dinâmicos, relacionando a resposta das medições em função do tempo e do ambiente.

Os sensores implementados para operar com IHR estão incluídos na categoria de entrada digital, com saída dinâmica.

4.1.5.1. Sensor de temperatura

O transdutor utilizado na implementação do sensor de temperatura é o termistor.

Os termistores são sensores fabricados com materiais semicondutores, como óxido de magnésio ou cobalto. Em aplicações que exigem alta precisão, o semicondutor utilizado pode ser o silício ou o germânio, dopados com algum outro material como o latão ou determinadas ligas de cobre. Por serem construídos de material semicondutor, possuem a grande vantagem de poderem ser fabricados em pequenas dimensões. O termistor de coeficiente negativo de temperatura (NTC) é um sensor muito conhecido e encontrado no mercado com uma variedade muito grande no tipo construtivo e nos valores de resistência. Já o termistor de coeficiente *positivo* (PTC), é mais raro, em razão da complexidade no aspecto construtivo.

Os termistores do tipo NTC podem ser classificados sob quatro tipos principais. O primeiro, de pequenas dimensões físicas, serve para operação em baixa potência, usado para controle de circuitos através de algum sistema amplificador de sinal. O segundo, de grandes dimensões físicas, operando em altas potências, controla diretamente um determinado circuito, sem sistema amplificador. O terceiro tipo engloba os termistores em bloco, nos quais as correntes transportadas praticamente não afetam os valores de sua resistência elétrica; são usados no braço de uma ponte, para proporcionar uma compensação da temperatura ambiente para um outro termistor que é usado para leitura. O último tipo é o termistor aquecido indiretamente, onde o material semicondutor é aquecido por meio de um filamento que tem valor desprezível de coeficiente de resistência à temperatura.

Os termistores NTC, ao contrário dos PTC, diminuem sua resistência elétrica com o aumento da temperatura.

Injetando uma corrente constante de alguns miliampéres no termistor, ao ocorrer uma determinada variação de temperatura junto ao sensor (com a conseqüente variação de resistência), haverá uma correspondente variação de tensão na saída do sensor, que será proporcional à variação de temperatura sofrida.

O circuito do sensor de temperatura é implementado utilizando-se o termistor e amplificador operacional configurado como comparador.

Um comparador é um circuito que compara duas tensões de entrada e produz na saída uma tensão relativa ao estado dos dois terminais de entrada. A Figura 4.44 apresenta um comparador básico. Por não haver uma impedância na realimentação, o amplificador operacional atua em laço aberto (open-loop). Uma das entradas é definida como referência de tensão e a outra como a entrada que contém a tensão desconhecida. A saída do comparador indica quando a saída desconhecida está acima ou abaixo da tensão de referência. Na figura 4.17 a tensão de referência V_r é aplicada à entrada não inversora enquanto V_i , tensão desconhecida, é aplicada à entrada inversora. Quando V_i apresenta um valor maior que V_r a tensão de saída V_o vai a $-V$ (tensão negativa de saída). Para V_i menor que V_r , V_o será igual a $+V$. Mudando-se as entradas há a inversão da saída.

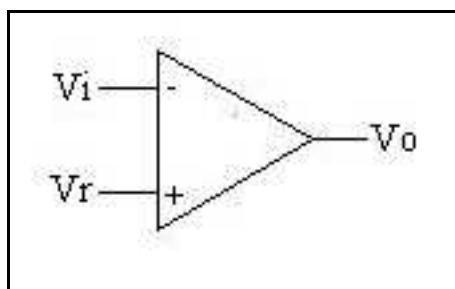


Figura 4.44 Amplificador operacional

O amplificador operacional utilizado na implementação do sensor de temperatura é o LM358.

O LM358 é constituído de dois amplificadores independentes, de alto ganho, com compensação interna de frequência, projetado especialmente para operar com fonte de alimentação simples em uma extensa faixa de tensão. A operação com fonte de alimentação simétrica também é possível e a potência consumida é independente da tensão de

alimentação. As áreas de aplicação incluem amplificadores para transdutores, blocos para ganho DC e demais circuitos convencionais que utilizam amplificadores operacionais.

A figura 4.45 apresenta o LM358 e sua pinagem.

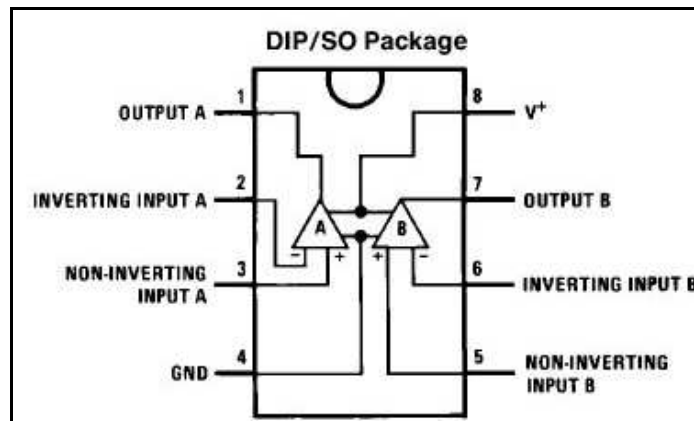


Figura 4.45 Circuito integrado LM358

As principais características do amplificador operacional LM358 são:

- Frequência compensada internamente para ganho unitário
- Ganho DC de 100 dB.
- Largura de banda de 1 MHz .
- Faixa de alimentação de 3 a 32 V com fonte simples.
- Consumo de corrente de 500 μ A independente da tensão de alimentação.
- Tensão do offset de entrada de 2 mV.
- Faixa da tensão de entrada diferencial igual à da tensão de alimentação

A Figura 4.46 apresenta o circuito do sensor de temperatura. O termistor utilizado é um NTC com resistência de 470 Ohms a 25 graus centígrados. O resistor de 470 Ohms conectado ao termistor fornece a corrente de carga do mesmo. Alterações na temperatura do ambiente onde se encontra o termistor ocasionam a alteração de sua resistência causando a variação da tensão no ponto onde está conectada a entrada não inversora do amplificador operacional. Caso a tensão na entrada não inversora seja maior que a tensão de referência encontrada na entrada inversora a saída do amplificador operacional é levada à tensão de alimentação. O potenciômetro na entrada inversora permite alterar a tensão de referência,

possibilitando ajustar a temperatura na qual o sensor será ativado. A tensão de alimentação do sensor de temperatura é de 5 V e utiliza a mesma fonte da IHR.

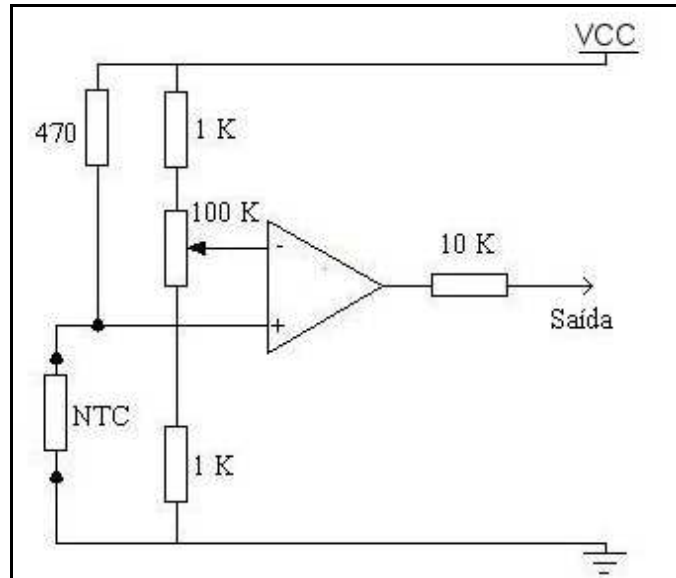


Figura 4.46 Circuito do sensor de temperatura

A saída do sensor é conectada à entrada digital da IHR. A leitura do estado do sensor por um programa no computador, permite que se desenvolvam projetos que irão atuar em função da temperatura. Uma aplicação possível é o controle de uma estufa de plantas onde a partir de determinada temperatura, selecionada pelo aprendiz, é possível acionar um sistema de refrigeração e fornecer um aviso sonoro, ou gravar em arquivo a data e hora em que o sensor foi ativado. Os dados contidos nesse arquivo podem ser tratados por um programa, como uma planilha de cálculo no qual possa criar gráficos que mostrarão os períodos em que houve alterações da temperatura.

4.1.5.2. Sensor de luminosidade

Para implementação do sensor de luminosidade foi utilizado como transdutor um fotodiodo.

O fotodiodo é um diodo semiconductor com a junção exposta à luz. A corrente reversa de um fotodiodo aumenta linearmente com a incidência de luz, como pode ser visto nas curvas características I/V da figura 4.47. No escuro a corrente reversa é muito pequena, geralmente da ordem de nA. Pode-se observar que as curvas características passam pelo

quarto quadrante ($V \times I < 0$), de modo que o fotodiodo pode produzir energia elétrica. Os valores mostrados são apenas um exemplo, pois dependendo do tipo e tecnologia de implementação os fotodiodos apresentam diferentes sensibilidades.

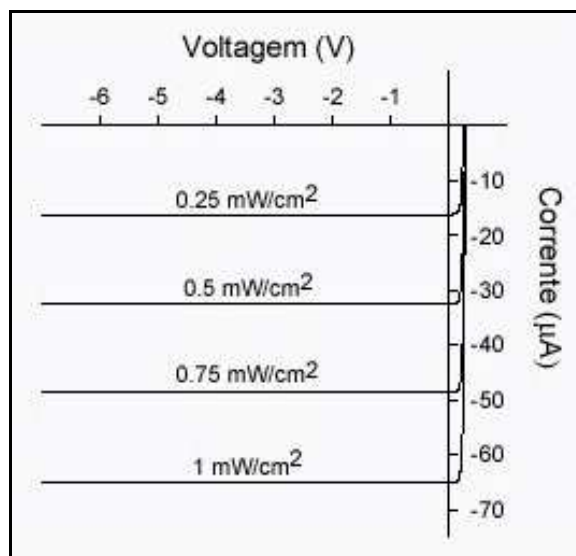


Figura 4.47 Curva típica de fotodiodo

Os fotodiodos podem ser aplicados como sensores de radiação num espectro que vai da luz visível ao infravermelho. A resposta espectral (sensibilidade a diferentes comprimentos de onda) de um fotodiodo de silício atinge o máximo em torno de 900 nanômetros (infravermelho). Alguns fotodiodos têm filtros que filtram a luz visível ou a radiação infravermelha. As correntes obtidas com um fotodiodo são geralmente baixas, menores que 100 µA para uma incidência luminosa de 1 mW/cm².

Para o sensor de luminosidade é utilizado um circuito semelhante ao do sensor de temperatura (Figura 4.48). Neste caso é utilizado um fotodiodo polarizado reversamente por um resistor de 120 KOhms. Se a tensão na entrada não inversora for maior que a tensão de referência encontrada na entrada inversora, a saída do amplificador operacional é levada à tensão de alimentação. O potenciômetro na entrada inversora serve para alterar a tensão de referência, permitindo ajustar o nível de luminosidade que ativará o sensor.

Uma aplicação possível para o sensor de luz é a construção de um robô móvel cujos motores são acionados de modo a mover-se em direção a uma fonte de luz ou afastar-se dela (Braitenberg, 1989).

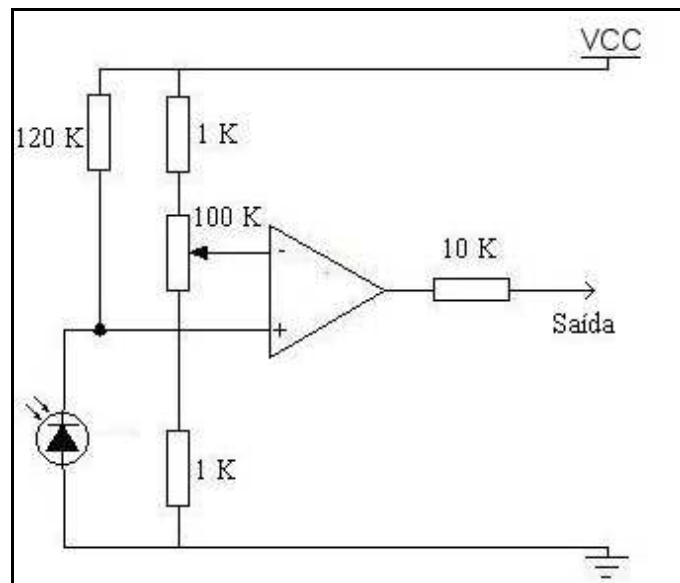


Figura 4.48 Sensor de luminosidade

4.1.5.3. Sensor de umidade

O sensor de umidade foi desenvolvido para permitir a detecção de água em estado líquido. Este sensor pode ser utilizado para determinar o nível de água em um recipiente ou a umidade do solo, por exemplo.

O princípio empregado para detecção da água é a resistividade da mesma. O circuito utiliza um amplificador operacional na configuração de comparador. Da entrada não inversora e do ponto de terra saem condutores que entrarão em contato com o local onde se deseja determinar o nível de umidade.

O elemento transdutor é constituído de dois eletrodos que estão conectados ao ponto de terra e à entrada não inversora do amplificador operacional. Estes eletrodos ao entrar em contato com um meio onde exista água ocasionam uma diminuição da resistência entre o terra e a entrada não inversora, o que reduz a tensão presente na entrada. O circuito do sensor pode ser visto na Figura 4.49. Dependendo da tensão de referência presente na entrada inversora, e que pode ser ajustada pelo potenciômetro de 100 KOhms, o nível de saída será levado a zero, indicando que foi detectada umidade.

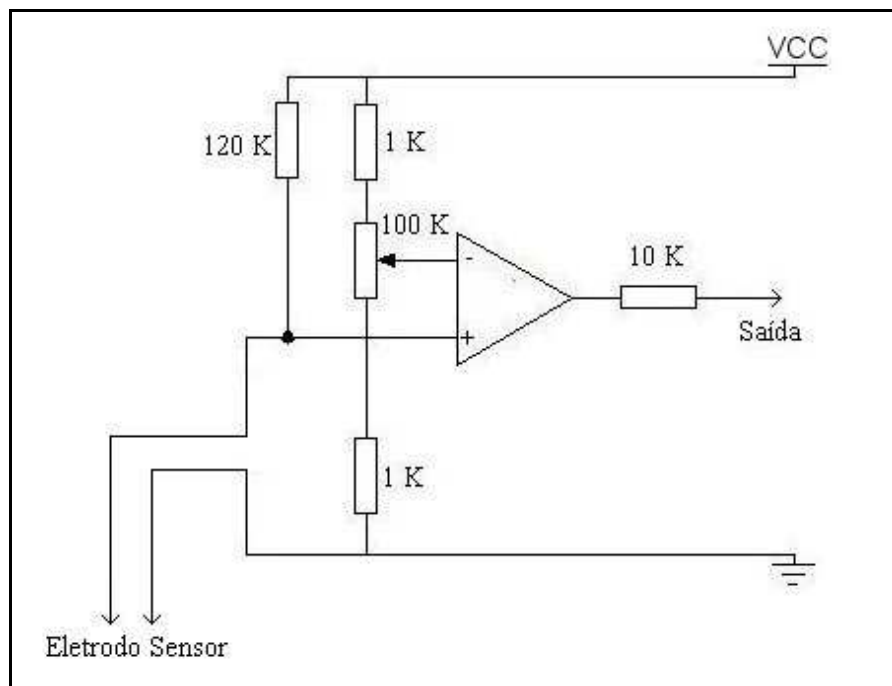


Figura 4.49 Sensor de umidade

Um exemplo de projeto que foi desenvolvido é uma estufa de plantas automatizada. Os eletrodos do sensor de umidade são inseridos no solo e caso a umidade diminua o sensor enviará esta informação ao computador onde um programa ativa uma bomba que irriga o solo restaurando a umidade.

4.1.6. Programa do microcontrolador

O programa implementado para o microcontrolador PIC16F84 foi desenvolvido com o compilador Hitech C, sendo responsável pelas tarefas relacionadas ao recebimento das instruções do computador e geração do sinal de controle PWM (Pulse With Modulation) para o servomotor.

O programa pode ser dividido em 4 módulos, segundo a funcionalidade de cada um:

- Inicialização: responsável por inicializar os vetores de interrupção, selecionar e configurar os pinos que atuarão como entrada e saída e colocar o microcontrolador em estado de espera para receber as instruções.
- Leitura de instruções: os pinos configurados como entrada são lidos e seus valores são armazenados para utilização posterior.

- Decodificação: as instruções são decodificadas, gerando informações que selecionam o servomotor e a ação a ser executada sobre ele.
- Geração de PWM: neste bloco são gerados os sinais de controle PWM, que atuarão sobre os servomotores segundo as instruções recebidas do módulo de decodificação.

O diagrama da Figura 4.50 representa a operação do programa e a tabela 4.2 mostra o conjunto de instruções para a IHR.

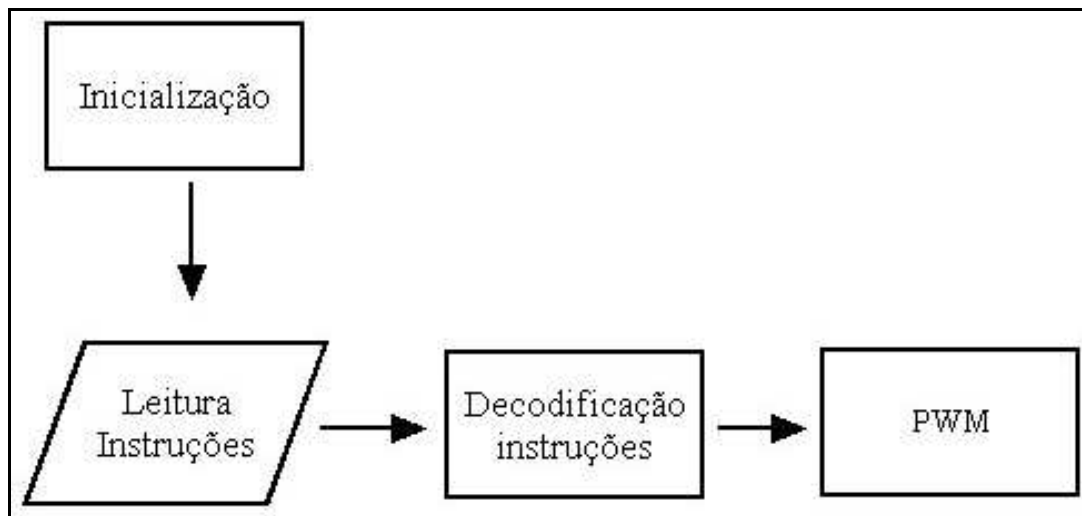


Figura 4.50 Diagrama do programa do microcontrolador

Cada instrução com comandos para controle de servomotores conectados à IHR é composta de 4 bits, com funções para :

- Selecionar qual dos três servomotores será acionado.
- Indicar o sentido de rotação do servomotor.
- Indicar se o controle é para atuar sobre servomotor de rotação contínua ou de passo.
- Inicializar servomotores.
- Interromper o movimento do servomotor sob controle. Neste caso, a IHR continua enviando o pulso que mantém o servomotor posicionado.
- Desligar servomotor sob controle. Neste caso a IHR deixa de enviar os pulsos de comando, causando efeito mecânico semelhante ao da retirada da alimentação.

Instrução	Ação no Servomotor
0	Interrompe comandos enviados aos servomotores
1	Aciona servomotor de passo conectado à saída 1 no sentido horário
2	Aciona servomotor de passo conectado à saída 1 no sentido anti-horário
3	Aciona servomotor de passo conectado à saída 2 no sentido horário
4	Aciona servomotor de passo conectado à saída 2 no sentido anti-horário
5	Aciona servomotor de passo conectado à saída 3 no sentido horário
6	Aciona servomotor de passo conectado à saída 3 no sentido anti-horário
7	Coloca todos servomotores na posição central
8	Desligar servomotor de movimento contínuo conectado à saída 1
9	Desligar servomotor de movimento contínuo conectado à saída 3
10	Desligar servomotor de passo conectado à saída 3
11	Aciona servomotor de movimento contínuo conectado à saída 1 no sentido horário
12	Aciona servomotor de movimento contínuo conectado à saída 1 no sentido anti-horário
13	Aciona servomotor de movimento contínuo conectado à saída 3 no sentido horário
14	Aciona servomotor de movimento contínuo conectado à saída 3 no sentido anti-horário

Tabela 4.2 Instruções para acionamento de servomotores

Quando uma instrução válida é enviada o servomotor executa esta instrução até receber a instrução de interrupção de comando (0). Como o servomotor mantém o movimento até receber a instrução de interrupção, o seu deslocamento está condicionado ao tempo decorrido entre um comando e outro.

Por exemplo, para efetuar um deslocamento no servomotor 2 no sentido anti-horário deve-se proceder com os seguintes passos:

- Enviar a instrução que comanda o servomotor no sentido anti-horário “4”
- Criar um intervalo de tempo.
- Enviar a instrução que cancela comandos do servomotor “0”.

Este algoritmo deve ser seguido ao se implementar programas para controle da IHR no computador PC, o diagrama da Figura 4.51 representa este procedimento.

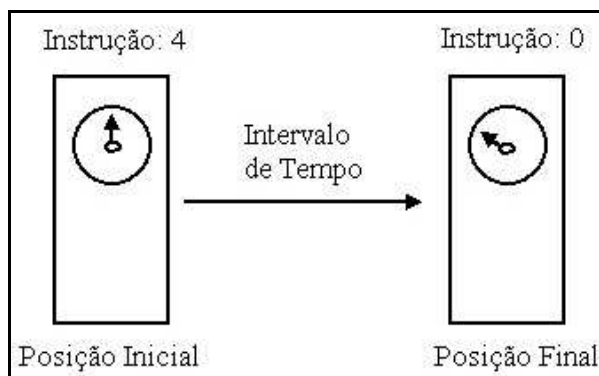


Figura 4.51 Diagrama de acionamento dos servomotores

4.1.7. Programa SuperLogo para comunicação com IHR

Originalmente o SuperLogo não dispõe de comandos e funções específicos para o controle de robôs, portanto tais comandos precisaram ser desenvolvidos. Os comandos para robótica implementados no SuperLogo mantêm a estética Logo, sendo, por esta razão, facilmente compreendidos por usuários que tenham alguma familiaridade com a linguagem, dispensando a necessidade de conhecimentos técnicos relacionados à comunicação com o hardware.

4.1.7.1. Procedimentos SuperLogo para a IHR

Com a interface de hardware é possível controlar três servomotores, que podem ter movimento contínuo ou de passo, e receber dados provenientes de quatro sensores.

Os comandos e funções implementados no SuperLogo permitem o controle dos servomotores e a leitura dos dados dos sensores.

4.1.7.1.1. Procedimentos SuperLogo para controle de servomotores

A comunicação com a Interface de Hardware ocorre por meio da porta Centronics. Para enviar dados a portas de hardware do computador PC o SuperLogo dispõe do comando *portasaídab* que deve ser utilizado com os parâmetros que indicam o endereço da porta e o byte de dados. Assim, para enviar à porta Centronics LPT1, que tem o endereço decimal 888 (decimal), o byte 128, o comando será:

portasaídab 888 128

O valor 128 permanecerá na porta Centronics até que um novo dado seja enviado.

Para leitura das portas de hardware o SuperLogo dispõe do comando *portaentrab*.

Na implementação dos comandos para controle dos servomotores foi utilizado o comando *portasaídab* que envia os dados conforme especificado na tabela 4.2.

Para controlar os servomotores com movimento de passo é preciso fornecer como parâmetro valores que indicam o servomotor que se vai acionar, o sentido da rotação e o deslocamento. A implementação do comando é mostrada na tabela 4.3. Tal comando recebeu a denominação de *motorp* e deve ser utilizado com os parâmetros conforme indicado:

motorp *servomotor direção deslocamento*

Os seguintes valores são admitidos para os parâmetros

Servomotor : 1, 2 ou 3

Direção : D para sentido horário e E para sentido anti-horário

Deslocamento: > 0

A implementação do procedimento *motorp* é apresentada a seguir:

```
aprenda motorp :motor :direc :desloc
se :motor=1[
    se :direc = "d [portasaídab 888 1]
    se :direc = "e [portasaídab 888 2]
]
se :motor=2[
    se :direc = "d [portasaídab 888 5]
    se :direc = "e [portasaídab 888 6]
]
se :motor=3[
    se :direc = "d [portasaídab 888 3]
    se :direc = "e [portasaídab 888 4]
]
espere :desloc
portasaídab 888 0
fim
```

Tabela 4.3 Implementação do comando motorp

Para os servomotores com movimento contínuo foi implementado o comando *motorc* (tabela 4.4) que deve ser utilizado com os parâmetros que permitem selecionar o

motor que se vai acionar e a ação a ser executada: movimento à direita, à esquerda e interrupção do movimento.

Motorc servomotor ação

Os seguintes valores são admitidos para os parâmetros:

Servomotor : 1 ou 3

Ação: D para sentido horário, E para sentido anti-horário e P para interromper o movimento.

A implementação do procedimento *motorc* é apresentada a seguir:

```
aprenda motorc :motor :direc
se :motor=1[
  se :direc = "d [portasaídab 888 11]
  se :direc = "e [portasaídab 888 12]
  se :direc = "p [portasaídab 888 8]
]
se :motor=3[
  se :direc = "d [portasaídab 888 13]
  se :direc = "e [portasaídab 888 14]
  se :direc = "p [portasaídab 888 9]
]

espere 5
portasaídab 888 0
fim
```

Tabela 4.4 Implementação do comando motorc

Imotor

Este procedimento ao ser executado coloca os servomotores com movimento de passo na sua posição inicial. Este procedimento não tem parâmetros. A implementação do procedimento *imotor* esta apresentada na tabela 4.5.

```
aprenda imotor
portasaídab 888 7
espere 5
portasaídab 888 0
fim
```

Tabela 4.5 Implementação do comando imotor

4.1.7.1.2. Utilizando os procedimentos SuperLogo para controlar servomotores

A seguir apresentamos uma seqüência de comandos que aciona um servomotor de passo conectado a saída 1 da IHR e um servomotor de movimento contínuo conectado a saída 3.

imotor

motorp 1 “d 200

motorc 3 “d

espere 100

motorc 3 “p

4.1.7.1.3. Procedimentos SuperLogo para sensores

Os níveis lógicos provenientes da IHR com informações do estado dos sensores são conectados aos pinos ERR, PE, ACK e BUSY da porta Centronics. Estes pinos pertencem ao registrador de status da porta Centronics e o valor correspondente a cada sinal pode ser obtido a partir da leitura do byte retornado pelo registrador no endereço 889 (decimal).

O comando disponível para leitura de portas de hardware no SuperLogo é o *portaentradab*, sendo a sintaxe:

portaentradab <endereço da porta de hardware>

Como o endereço do registrador de status da porta Centronics é 889 (decimal), para obter os dados disponíveis no registrador a sintaxe é:

Mostre portaentradab 889

O comando *portaentradab* retorna um byte, portanto é preciso conhecer a posição que os sinais ERR, PE, ACK e BUSY ocupam dentro desse byte e efetuar uma operação de bit com o operador E para recuperar o nível desses bits e, por conseguinte, o estado dos sensores.

A configuração do registrador de status da porta Centronics está mostrada na tabela 4.6.

Bit	Sinal
00	não usado
01	não usado
02	não usado
03	-ERR
04	SLCT
05	PE
06	-ACK
07	-BUSY

Tabela 4.6 Registro de Status da porta Centronics

O procedimento desenvolvido com o SuperLogo para leitura do estado dos sensores conectados à IHR denomina-se *sensor* e tem como parâmetro o valor numérico do sensor cujo estado se deseja conhecer. Se o sensor estiver ativado, a função *sensor* retorna verdadeiro, caso contrário retorna falso. A sintaxe do procedimento é:

Sensor <número do sensor>

Para conhecer o estado do sensor conectado à entrada 1 da interface de hardware a sintaxe é:

Mostre sensor 1

Na implementação do procedimento *sensor* inicialmente é atribuída a uma variável o valor retornado por *portaentradas* e que contém o valor disponível no registrador de status. A seguir é verificado qual sensor foi passado como parâmetro. Com essa informação é realizada uma operação de bit E, por meio do comando *bite* entre a variável que contém o valor do registrador e o bit que corresponde à posição do sinal dentro do byte. Se a operação de bit retornar um valor maior que 0, o sensor estará ativo e o procedimento retorna verdadeiro, do contrário retorna falso.

A implementação da função está apresentada na tabela 4.7.

```

aprenda sensor :valor
atr "entrada portaentrada 889

se :valor=1 [
    atr "sinal bite :entrada 8
    se :sinal>0[envie "verd][envie "falso]
]

se :valor=2[
    atr "sinal bite :entrada 32
    se :sinal>0[envie "verd][envie "falso]
]

se :valor=3[
    atr "sinal bite :entrada 128
    se :sinal>0[envie "falso][envie "verd]
]

se :valor=4[
    atr "sinal bite :entrada 64
    se :sinal>0[envie "verd][envie "falso]
]

fim

```

Tabela 4.7 Implementação do comando sensor

4.1.7.1.4. Utilizando os procedimentos SuperLogo para sensor

Um exemplo de utilização do procedimento *sensor* é apresentado abaixo:

mostre "sensor:

se sensor 1 [mostre "ativado] [mostre "desativado]

Para realizar este exemplo é preciso ter um sensor conectado à entrada 1 da IHR. Ao executar esta seqüência de comandos será exibida a mensagem “sensor: ativado” se o sensor estiver em nível lógico 1. A mensagem “sensor: desativado” será apresentada caso o sensor esteja em nível lógico 0.

4.2. Considerações sobre o capítulo

Neste capítulo foram descritos a implementação da interface para robótica IHR, seu hardware, o programa do microcontrolador, os sensores e os procedimentos no SuperLogo para acesso a seus recursos. No capítulo 7 são apresentadas experiências de aplicação do ambiente de robótica, podendo-se observar que durante a utilização da IHR em condições reais de operação, como oficinas e aulas, com aprendizes de vários níveis a mesma mostrou-se confiável e segura.

Capítulo 5

Implementação da Comunicação SuperLogo – Lego RCX

No capítulo 2 foi apresentado o tijolo Lego RCX e os programas distribuídos nos kits comercializados pela empresa Lego; o Mindstorms e o Robolab. Conforme exposto, estes programas não permitem integração com outras ferramentas de programação e hardware, e aplicam-se exclusivamente na elaboração de projetos com o tijolo RCX.

Como um dos objetivos do Ambiente de Robótica Educacional desenvolvido, é integrar a partir do SuperLogo, o controle de dispositivos robóticos que empregam tecnologias diversas, incluindo o Lego RCX, foi preciso desenvolver meios para se estabelecer a comunicação entre o SuperLogo e o hardware do RCX.

A comunicação com o Lego RCX ocorre por meio do envio de bytes à porta serial, contendo os bytecodes que são reconhecidos e executados no tijolo RCX. Todavia, a empresa Lego não fornece documentação sobre o protocolo de comunicação serial empregado e a funcionalidade dos bytecodes, o que dificulta a utilização do RCX com outras ferramentas.

Com o objetivo de minimizar esta dificuldade a empresa Lego fornece um componente de programa chamado SPIRIT.OCX que emprega a tecnologia ActiveX⁴. Quando é realizada a instalação do CD-ROM com o programa Mindstorms, este componente é instalado e torna-se disponível para ser utilizado com ferramentas de programação com suporte a ActiveX.

As funções disponíveis no SPIRIT.OCX podem ser classificadas como:

- Controle de comunicação: responsável por iniciar e encerrar a comunicação com o RCX, detectar e corrigir falhas na comunicação.
- Controle de firmware: efetua o download e habilita o firmware.
- Diagnóstico: testar os diversos elementos da comunicação como cabo serial, torre e link de infravermelho com o RCX.

⁴ Activex é um conjunto de programas desenvolvidos pela empresa Microsoft baseada em duas tecnologias denominadas OLE (Object Linking and Embedding) e COM (Component Object Model). O seu objetivo é permitir o encapsulamento de softwares dentro de um padrão de interface tornando-o compatível com as ferramentas de programação que suportam ActiveX.

- Sistema: atua no RCX para selecionar display, ajustar data e hora, tempo de desligamento quando inativo, limpar temporizador, enviar e limpar mensagens de sistema.
- Controle de saída: atua sobre os motores permitindo a seleção, acionamento, alteração da direção, da potência aplicada e do tempo de acionamento.
- Controle de entrada: opera com os sensores, possibilitando selecionar, determinar o tipo e forma de leitura (digital ou analógica).
- Execução de programas: controla a execução de programas disponíveis no RCX permitindo a seleção do programa, seu início e a interrupção de sua execução.
- Fluxo de programa, comandos para controle de fluxos das tarefas elaboradas pelo usuário.
- Lógica e aritmética: comandos com os operadores lógicos e aritméticos.
- Requisição de dados: comandos que efetuam requisição de dados ao RCX. Estes dados podem ser provenientes de sensores, indicar o estado da bateria do RCX, ou mostrar o mapa de utilização da memória.
- Aquisição de dados: efetua leitura dos dados dos sensores.

O conjunto de comandos pode ser classificado quanto ao modo de execução, podendo ser imediato ou apenas por download. Os comandos de execução imediata são aqueles que, quando enviados ao RCX, executam a função no momento que ela é recebida. A grande maioria dos comandos se enquadra nesta categoria, e entre eles estão os comandos para controle de sensor, motores e diagnóstico.

Os comandos para execução por download precisam ser descarregados no RCX antes de serem executados. Nesta categoria estão os comandos para controle de fluxo de programa, operadores lógico/aritméticos e alguns comandos de controle de sistema.

Uma maneira de estabelecer comunicação com o RCX sem utilizar os programas fornecidos pela empresa Lego é desenvolver um aplicativo com uma ferramenta de programação que suporte a tecnologia ActiveX, como o Visual Basic ou Delphi, e usar o componente SPIRIT.OCX. Porém o SuperLogo por não oferecer suporte a tecnologia Activex não pode utilizar diretamente o componente SPIRIT.OCX em seu ambiente de programação, constituindo uma limitação que precisou ser contornada.

A solução adotada para que fosse possível estabelecer a comunicação com o RCX a partir do SuperLogo e permitir a utilização de seus recursos, foi implementar um aplicativo que atua entre o SuperLogo e o RCX, funcionando como uma interface de programa (Figura 5.52).

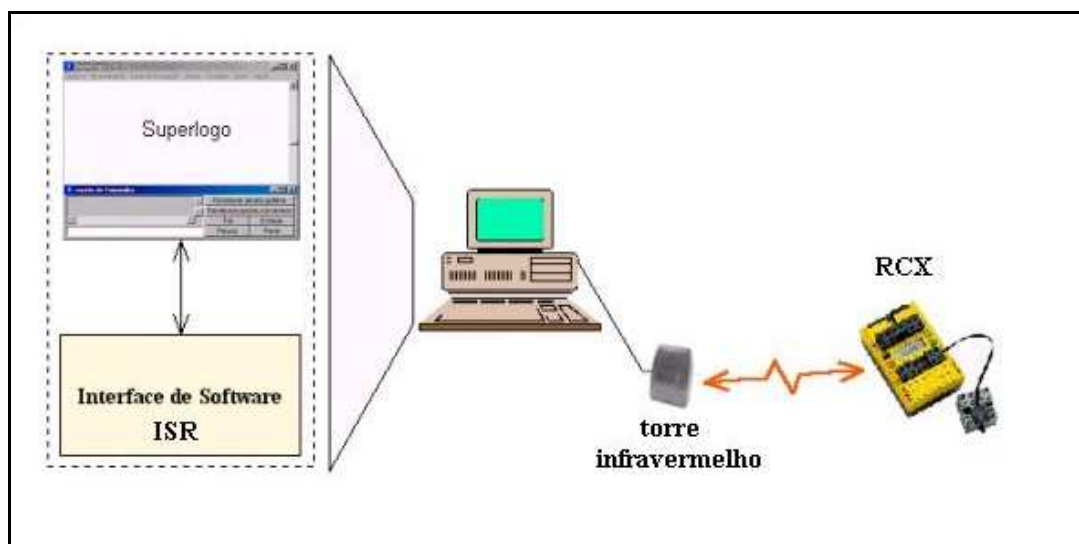


Figura 5.52 Esquema da comunicação do SuperLogo com o RCX

Por proporcionar a comunicação entre o SuperLogo e o RCX a interface de programa foi denominada de Interface SuperLogo RCX (ISR) .

A ISR comunica-se com o SuperLogo por meio da troca de mensagens que são classificadas como de dados e de controle. As mensagens de dados contêm informações sobre seleção e controle dos motores e sensores conectados ao RCX. As mensagens de controle, coordenam e sincronizam o fluxo entre a ISR e o SuperLogo.

As mensagens são arquivos de texto ASCII, tipo de arquivo compatível com o SuperLogo e o Visual Basic. O arquivo com a mensagem de dados indica o comando a ser executado e os parâmetros, que variam conforme a atuação ocorra sobre um motor ou sensor. Por exemplo, para acionar um motor a mensagem deve conter o comando que indique ligar o motor e o parâmetro que informa qual dos três motores será acionado. O diagrama da figura 5.53 ilustra esse processo.

Ao receber esta mensagem a ISR irá interpretá-la e executará a função correspondente à ação solicitada.

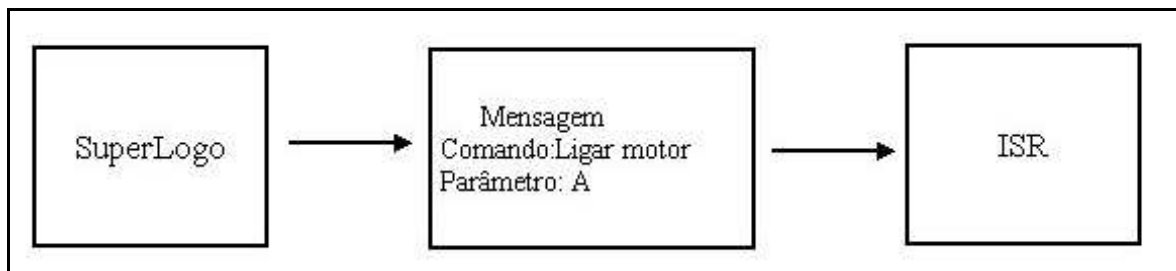


Figura 5.53 Fluxograma de mensagem entre o SuperLogo e a ISR

Para efetuar a leitura de um sensor conectado é preciso efetuar uma solicitação ao RCX, o qual capturará os dados do sensor e os enviará ao computador, para que possam ser lidos pela ISR.

Para que o SuperLogo possa ler os dados de um sensor, são gerados três arquivos com mensagens. O primeiro, enviado pelo usuário a partir do SuperLogo, “lersensor”, indica o sensor que será lido; o segundo “sensordisponivel” é uma mensagem de controle, que indica que o dado do sensor já está disponível e pode ser lido no terceiro arquivo, “dadosensor”. O diagrama da figura 5.54 ilustra o processo de aquisição de dados dos sensores do RCX.

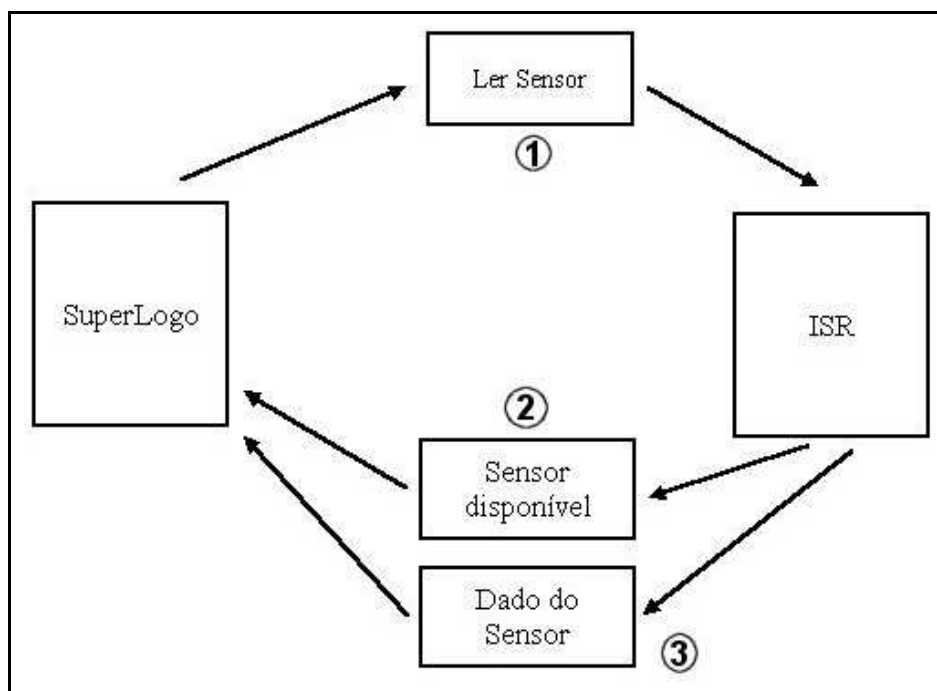


Figura 5.54 Fluxograma para sensores

5.1. Painel de controle da ISR (Interface SuperLogo RCX)

Além das funções de comunicação entre o SuperLogo e o RCX, a ISR dispõe de recursos para ativação da comunicação, e para diagnóstico e configuração do RCX. Estas funções são acessíveis ao usuário por meio de uma interface gráfica.

5.1.1. Módulo de Diagnóstico

O conjunto de hardware do RCX compreende um cabo que deve ser conectado à porta serial RS232, à torre de infravermelho e o próprio RCX. Todos estes itens devem estar funcionando corretamente, e caso algum deles apresente alguma falha a comunicação é interrompida. Na utilização do RCX constatou-se a possibilidade de ocorrência de falhas na porta RS232 do computador, no cabo serial, na torre de infravermelho ou no RCX. Solucionar este tipo de problema pode ser uma tarefa difícil caso não existam meios para detectar os componentes que não estão operando corretamente. Para auxiliar na resolução de falhas foi desenvolvido para a ISR um módulo de diagnóstico. Dispondo de uma interface gráfica (Figura 5.55) onde uma imagem representa cada um dos componentes de hardware do RCX, um conjunto de botões permite testar cada um destes elementos indicando com uma mensagem o estado dele. Desta maneira é possível identificar e corrigir rapidamente qualquer problema.

O teste pode ser realizado separadamente para cada componente ou para todos de uma única vez, com a opção do *Teste Geral*. Também é possível obter o valor da tensão da bateria do RCX clicando no botão *Bateria RCX*.



Figura 5.55 Interface gráfica da ISR

Se todos os componentes estiverem operando adequadamente e for realizado o teste, a mensagem OK será apresentada (Figura 5.56).



Figura 5.56 Interface gráfica da ISR - diagnóstico

Se algum dos componentes, apresentar algum problema de operação, é apresentada a mensagem de *Falha*. Na figura 5.57 esta ilustrada a tela da situação onde existe problema com a torre de infravermelho.



Figura 5.57 Interface gráfica ISR demonstração do diagnóstico

5.1.2. Módulo de Configuração

Para que o RCX possa operar, ele precisa ter o seu firmware instalado. A função de transferência é ativada quando se clica no botão *Download Firmware* (Figura 5.58). Uma caixa de diálogo é aberta, solicitando que se abra o arquivo que contém o firmware. Esta função também é utilizada para efetuar atualizações. Após efetuar o download é preciso enviar um código ao RCX para habilitar o firmware, permitindo que o mesmo entre em operação, este procedimento ocorre ao clicar no botão *Habilitar Firmware*.

Uma outra função do módulo de configuração é selecionar a porta serial RS232, à qual se encontra conectada a torre de infravermelho. Após a seleção o usuário deve clicar no botão *Aplicar*, para que a configuração seja atualizada. O botão *Salvar*, grava a configuração para que nas próximas vezes em que se utilizar o aplicativo não seja necessário efetuar as configurações.



Figura 5.58 Interface gráfica ISR - configuração

5.1.3. Módulo de Operação

Este módulo (Figura 5.59) contém os botões *Iniciar* e *Finalizar*. O primeiro ativa a comunicação para troca de mensagens com o SuperLogo. O botão *Finalizar* finaliza o hardware (porta serial e o RCX), a comunicação com o SuperLogo, e encerra o aplicativo.



Figura 5.59 Interface gráfica ISR - operação

5.2. Utilizando o SuperLogo com o Lego RCX

Para utilizar o SuperLogo com o RCX o aprendiz deve carregar o SuperLogo e a ISR. Para ativar a comunicação, ele deve clicar no botão *Iniciar*. A partir deste instante os comandos específicos para o RCX estão operantes e, caso ele esteja ligado, as instruções digitadas no SuperLogo serão imediatamente executadas.

Um exemplo de utilização é ter um RCX com um motor conectado à porta C e um sensor de toque na porta 3 (Figura 5.60). O comportamento desejado é que toda vez que o sensor de toque for pressionado o motor seja ligado e permaneça neste estado desligando-se quando o sensor não estiver pressionado.



Figura 5.60 Experimento com o RCX

Uma solução de implementação de um programa que cumpra esta função é:

```
aprenda exemplo1  
se rcxlersensortoque "3 [rcxligamotor "c] [rcxdesligamotor "c]  
exemplo1  
fim
```

No procedimento *exemplo 1* a função *rcxlersensordetorque* retorna verdadeiro se o sensor de toque conectado a entrada 3 do RCX estiver pressionado, neste caso é executada a função *rcxligamotor*, que aciona o motor conectado a saída C do RCX, se o sensor de

toque não estiver ativado *rcxlersensordetoque* retorna falso e a função *rcxdesligamotor* é executada.

No Apêndice 3 estão apresentados em detalhes os comandos implementados no SuperLogo para operação com o Lego RCX.

A fim de oferecer apoio aos usuários da ISR foi desenvolvido material com vários exemplos de utilização do Lego RCX e do SuperLogo. São apresentados vários procedimentos que exploram os comandos para sensores e motores, acompanhados de ilustrações, facilitando a montagem do experimento. O Apêndice 4 apresenta este material.

5.3. Considerações sobre o capítulo

Neste capítulo foi apresentada a implementação de um aplicativo que habilita o SuperLogo a operar as funcionalidades do Lego RCX. Cria-se assim uma nova alternativa para utilização do tijolo RCX independente das ferramentas de programação originais fornecidas pelo fabricante. Deste modo o usuário do SuperLogo não necessitará aprender uma nova ferramenta para elaborar e controlar seu robô, além de poder integrar em seu projeto rotinas que ele já tenha desenvolvido e empregar os diversos recursos do SuperLogo, incluindo comandos gráficos e multimídia, entre outros.

Capítulo 6

Implementação do Ambiente de Telerobótica

Este capítulo trata da implementação dos módulos de programa necessários para que o aprendiz possa desenvolver atividades de robótica com a IHR e o RCX, remotamente, por meio da rede Internet.

O Ambiente de Telerobótica é constituído de três programas que são responsáveis pelo envio e recebimento de dados do servidor, monitoração do robô sob controle e comunicação entre usuários. Do ponto de vista de hardware, é composto pelo computador PC, que irá atuar como servidor, uma câmera de vídeo e as interfaces e dispositivos robóticos que se deseja controlar.

6.1. Envio e recebimento de comandos Remotos

O programa responsável pelo envio e recebimento dos dados contendo os comandos na sintaxe SuperLogo adota arquitetura cliente-servidor, e foi implementado seguindo o padrão de interface CGI (Common Gateway Interface).

Um servidor WEB é empregado para dar suporte ao programa CGI.

6.2. Modelo Cliente Servidor

O termo "cliente-servidor" pode adquirir vários significados, dependendo do contexto em que é aplicado. Nos ambientes de computadores mainframe, foi usado para indicar a relação entre o computador host e seus terminais associados. Em uma rede local ou intranet, é empregado para designar a associação entre um computador pessoal que atua como "servidor" de dados e arquivos de aplicações e o "cliente" microcomputador pessoal, que requisita os arquivos e aplicações necessários.

O desenvolvimento de sistemas que adotam o modelo cliente-servidor é feito aproveitando-se características dos modelos centralizado (computadores mainframe e terminais) e descentralizado (computadores pessoais). Por esta razão, os sistemas cliente-servidor combinam benefícios de ambos os modelos. Do modelo centralizado, vem o armazenamento de dados em computadores que atuam como servidores de dados, o que favorece o gerenciamento e segurança das informações. Do modelo descentralizado vem a

utilização do computador pessoal com poder de computação suficiente para executar aplicações poderosas e com interfaces gráficas amigáveis.

No modelo cliente-servidor o desenvolvedor separa a aplicação em dois componentes, "front end" e "back end", que compartilham as demandas de processamento de acordo com a tarefa a ser realizada. Esta separação de atribuições permite a utilização mais eficiente da capacidade de computação e da largura de banda da rede.

O componente "front end", ou parte cliente da aplicação, provê a interface de usuário, isto é, o ambiente gráfico por meio do qual o usuário interage com o aplicativo, podendo também executar algum tipo de processamento, dependendo das características do sistema.

Quando o componente "front end" faz uma requisição, o componente "back end", residente no servidor, executa o processamento solicitado, e o resultado é retornado ao "front end" .

Pode-se dizer que um sistema cliente-servidor é constituído de três componentes básicos:

1. O cliente, com o seu aplicativo específico, que lhe possibilita comunicar-se com o servidor.
2. O servidor, com o seu aplicativo específico, que lhe permite receber a mensagem, interpretá-la e retornar uma resposta para o cliente.
3. A rede, conjunto de programas e hardware, por meio da qual se dá comunicação entre o cliente e o servidor.

Servidor e Cliente são capazes de comunicar-se porque usam as mesmas regras ou protocolos, definidos pelo modelo Cliente/Servidor.

6.3. Interface CGI

Quando um usuário requisita o endereço de uma página WEB por meio de um navegador, estabelece-se uma conexão com uma máquina que também está conectada à Internet. A requisição deste documento é passada a um programa que está em execução, aguardando a chegada de requisições e respondendo de forma apropriada. Este programa é conhecido como servidor WEB.

Ao receber a solicitação de um arquivo, o servidor WEB verifica a existência do mesmo. Se o arquivo existir o servidor identifica qual é o tipo do arquivo (html, gif, entre

outros); a seguir, indica ao navegador a existência do arquivo e envia informações sobre o tipo de dado e o próprio dado, que pode ser, por exemplo, uma página HTML ou um arquivo gráfico em formato JPEG, GIF. A figura 6.61 ilustra este processo.

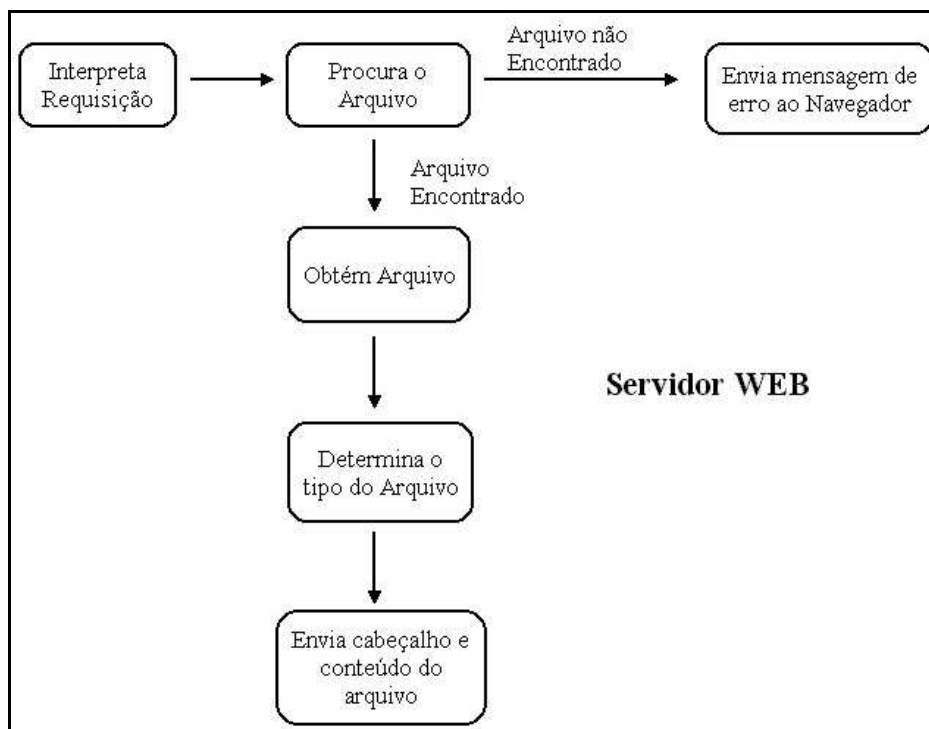


Figura 6.61 Fluxograma da requisição de arquivo em servidor WEB

Ainda que o servidor WEB seja capaz de enviar e receber dados, o servidor tem a funcionalidade limitada para aplicações que exigem mais do que apenas troca de arquivos.

Caso haja a necessidade de operações mais complexas do que recuperar e enviar arquivos por meio do navegador Web é preciso recorrer a meios de ampliar a funcionalidade do servidor Web.

A CGI (Common Gateway Interface) é uma interface que permite expandir as capacidades de um servidor WEB. A principal motivação para utilização de CGI é estabelecer interação com os usuários que acessam o site. A expansão da capacidade do WEB server ocorre quando a CGI interpreta as entradas do usuário, realizadas por meio do navegador, e retorna informações baseadas nessas entradas (Kim,1996) (Figura 6.62). Para o desenvolvedor, CGI é um recurso que propicia a implementação de programa que se

comunica com o servidor sem que seja necessário conhecer as particularidades de cada programa servidor WEB.

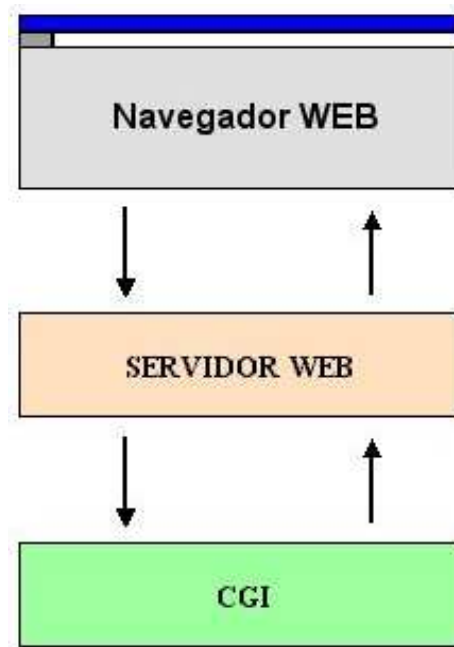


Figura 6.62 Fluxograma de programa CGI

Sem a utilização da CGI, qualquer expansão da capacidade de um servidor WEB envolveria mudar o código do próprio servidor. Isto requer que o código fonte do servidor seja editado, alterado e recompilado, ou que um servidor dedicado seja desenvolvido para cada tarefa. Esta pode ser considerada uma opção não muito eficiente, pois exige do desenvolvedor conhecimentos de programação nas camadas de protocolo de rede, Internet e WEB.

Pode-se supor, como exemplo, o desenvolvimento de uma extensão para o servidor WEB com o propósito de adicionar o recurso que o habilita a atuar como servidor de Webmail. A principal função de um servidor de Webmail é receber os dados de entrada do usuário por meio do navegador e enviar o e-mail para o usuário destino. Para prover esta expansão é necessário incluir no servidor código capaz de interpretar a entrada de dados do usuário, enviar a mensagem de email para o destino e enviar uma resposta ao browser do usuário indicando o êxito, a falha ou qualquer outra mensagem significativa.

Para este tipo de solução, é necessário que o código fonte do servidor esteja disponível, o que nem sempre é possível quando se trabalha com servidores comerciais. Mesmo que o código fonte esteja disponível, a extensão desenvolvida irá funcionar especificamente no servidor escolhido. Caso seja necessário migrar para outro servidor Web, todo o programa terá de ser desenvolvido novamente ocasionando gastos de tempo e recursos financeiros.

A utilização de CGI é uma solução para estes problemas. O protocolo CGI define um padrão para que programas se comuniquem com o servidor WEB. Sem que seja necessário conhecimentos específicos do servidor WEB, é possível escrever programas nas linguagens de programação mais comuns disponíveis. Este programa será capaz de trabalhar com praticamente todos os servidores WEB que atendam o protocolo CGI.

A comunicação nos processos CGI ocorre sobre os dispositivos padrões de entrada e saída. Isto significa que, por meio dos recursos de leitura e de escrita das linguagens de programação, é possível escrever uma aplicação que irá operar a partir de um servidor WEB. Programar uma aplicação CGI é equivalente ao trabalho realizado no desenvolvimento de qualquer outra aplicação.

Em razão da CGI ser uma "interface comum", não há restrição quanto às linguagens de programação que podem ser utilizadas no desenvolvimento de aplicativos. Pode ser utilizada qualquer linguagem de programação que atenda os seguintes requisitos:

- Escrever no dispositivo de saída padrão.
- Ler a partir do dispositivo de entrada padrão.
- Ler as variáveis de ambiente do sistema operacional.

Pode-se dizer que a grande maioria das linguagens de programação e muitas linguagens de script atendem estes três requisitos e, portanto, podem ser utilizadas para o desenvolvimento de aplicações CGI.

Um aspecto a ser considerado na escolha de uma linguagem de programação para o desenvolvimento de aplicações CGI está no fato de esta linguagem ser interpretada ou compilada. Um aplicativo criado com uma linguagem compilada como C, ou C++ tende a ser menor e mais rápido que o aplicativo desenvolvido com linguagens interpretadas como Perl ou PHP, que requerem na maioria das vezes o carregamento de um interpretador.

Ao se escolher uma linguagem de programação, é importante considerar os requisitos e características desejadas no projeto e balancear os ganhos de eficiência obtidos e facilidade de programação.

A programação CGI pode ser descrita como a realização de duas tarefas básicas: obter informação de um navegador para Web, processar esta informação, e retornar o resultado para o navegador. Normalmente é apresentado ao usuário um formulário que deve ser preenchido, como mostrado na Figura 6.63. Ao preencher o formulário e clicar no botão *enviar*, os dados são encaminhados ao servidor WEB que a seguir os envia para o aplicativo CGI. O aplicativo CGI deve decodificar esta informação, processá-la adequadamente, e enviar uma resposta ao navegador, que pode ser apenas a informação de que o dado foi recebido e processado, ou pode ser o resultado do processamento como, por exemplo, os registros de uma consulta a determinada base de dados.

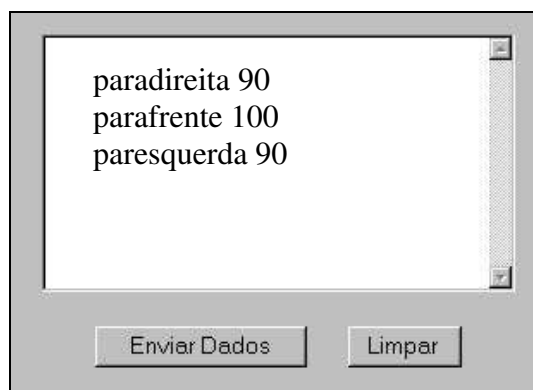
A imagem mostra uma janela de navegador com um formulário HTML. O formulário contém três campos de texto empilhados verticalmente, cada um com o seguinte texto: "paradireita 90", "para frente 100" e "para esquerda 90". Abaixo dos campos, há dois botões: "Enviar Dados" e "Limpar".

Figura 6.63 Formulário em página html

6.3.1. Recebendo entrada de dados pelo navegador WEB

Um programa CGI recebe dois tipos de informação do navegador WEB:

- Primeiro, é recebido um conjunto de informações sobre o navegador WEB (tipo, o que ele pode apresentar, o nome do host, entre outros), o servidor (o nome e a versão, o endereço da porta na qual ele está ativo) e do próprio programa CGI (seu nome, local onde está localizado, entre outros).

- Em segundo lugar, o programa CGI recebe as informações enviadas pelo usuário. Essa informação, após ser codificada, é enviada por meio de uma variável de ambiente (método GET) ou por meio da entrada padrão (stdin - método POST).

As informações, tanto do sistema quanto de usuário, são recebidas pelo programa por meio de variáveis de sistema. A tabela 6.1 apresenta algumas das variáveis de ambiente que são normalmente utilizadas no desenvolvimento de programas CGI.

Variável	Conteúdo
REMOTE_ADDR	endereço IP da máquina cliente
REMOTE_HOST	o nome do host da máquina cliente
HTTP_ACCEPT	lista dos tipos MIME de dados que o navegador reconhece
HTTP_USER_AGENT	informação sobre o navegador (nome, versão, sistema operacional)
REQUEST_METHOD	GET ou POST
CONTENT_LENGTH	a dimensão da entrada se for enviada via POST. Se não houver entrada ou se o método GET for utilizado, não é definida
QUERY_STRING	contém a informação de entrada quando ela é passada utilizando-se o método GET

Tabela 6.1 Variáveis de Ambiente usadas pelo aplicativo CGI

Quando o usuário envia os dados inseridos no formulário, o navegador codifica a informação para então enviá-la ao servidor e, em seguida, ao programa CGI. Ao ser utilizada a tag da linguagem HTML <input> é atribuído um nome que simbolicamente irá representar uma variável. O dado inserido pelo usuário será considerado o valor desta variável.

Com o propósito de efetuar esta especificação, o navegador faz uma chamada ao que é denominado URL Encoding Especification (BERNERS-LEE et al. 1994), a qual executa os seguintes procedimentos.

1. Separa diferentes campos com o símbolo ampersand (&).

2. Separa nomes e valores com o símbolo igual (=) com o nome à esquerda e o valor à direita.

3. Substitui espaços com símbolo de mais (+).

4. Substitui caracteres desconhecidos com o símbolo de porcentagem (%) seguido por dígitos em hexadecimal com o código do caractere.

Para que as informações recebidas possam ser úteis é preciso decodificá-las.

6.3.2. Enviando dados ao Navegador

O programa CGI deve enviar ao navegador dois conjuntos de dados: um cabeçalho que contém informações relacionadas ao tipo de informação que será apresentada e a informação a ser mostrada (Figura 6.64). Estes dois blocos de informação devem ser separados por uma linha em branco.

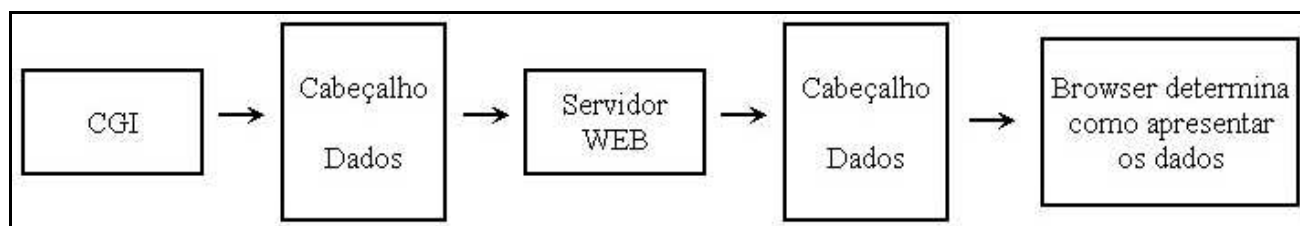


Figura 6.64 Envio de informações ao navegador WEB

O cabeçalho é denominado cabeçalho HTTP, e fornece informações sobre os dados que o navegador irá receber. Existem vários tipos de cabeçalho HTTP; o mais comum, e que será utilizado nos aplicativos desenvolvidos para o ambiente de telerobótica, é o “Content-Type”.

Diferentes combinações de cabeçalhos HTTP podem ser usadas separando-os com um *carriage return* seguido de um *newline* (`\r\n`). A linha em branco que separa o cabeçalho dos dados também consiste do *carriage return* e *newline* (`\r\n`).

O cabeçalho “Content-Type” descreve o tipo de dado que o programa CGI está retornando. Este tipo de cabeçalho tem o seguinte formato:

Content-Type: subtipo/tipo

onde subtipo/tipo é um MIME (Multipurpose Internet Mail Extension) válido.

Os tipos mais comuns são apresentados na Tabela 6.2.

Tipo/Subtipo	Arquivo
text/html	HTML
text/plain	arquivos texto
image/gif	arquivos gráficos *.GIF
image/jpeg	arquivos gráficos *.JPEG
audio/basic	arquivo de áudio da SUN *.au
audio/x-wav	arquivo de áudio do Windows *.wav

Tabela 6.2 – Cabeçalho Content-Type

Os dados que se seguem ao cabeçalho e a linha em branco serão apresentados no navegador. Por exemplo, se o cabeçalho indicar o tipo/subtipo HTML então devem ser enviados à saída padrão (stdout) tags HTML.

6.3.3. Servidores WEB

Uma característica dos aplicativos que adotam o padrão de Interface CGI é a portabilidade. O programa desenvolvido para o ambiente de telerobótica por atender aos requisitos do padrão de Interface CGI está apto a executar nos vários servidores WEB disponíveis.

O mercado de informática oferece várias opções de servidores WEB, capaz de atender os mais diversos requisitos como volume no tráfego de dados, plataformas de sistemas operacionais e hardware, consumo de recursos computacionais e preço. Entre os programas servidores WEB disponíveis destacam-se o Apache, o IIS (Internet Information Server) da Microsoft e o Xitami.

6.3.4. O Servidor WEB Xitami

Entre as várias opções disponíveis para executar o sistema do ambiente de telerobótica foi adotado o servidor Xitami (Imatix, 2002).

As principais características que influenciaram a seleção deste servidor foram:

- Baixo consumo de recursos computacionais como memória e disco, o mesmo ocupa 547kb de espaço em disco.
- Dispõe de versões para várias plataformas de sistemas operacionais e hardware.
- Desempenho adequado e compatível para o volume de dados da aplicação.
- Documentação completa.
- Facilidade para instalação, configuração e gerenciamento.
- Distribuído como freeware, portanto sem qualquer custo financeiro.

6.3.5. Implementação do Programa CGI

Para controlar o robô conectado ao servidor remoto é necessário enviar os comandos a partir do navegador WEB. Esses comandos são inseridos em um campo de formulário existente em uma página HTML. Esta página é um dos elementos constituintes da interface gráfica. Os comandos digitados são enviados ao servidor, e devem ser processados a fim de que o SuperLogo possa interpretá-los e efetuar a atuação no dispositivo robótico sob controle. Para a implementação do aplicativo foi utilizada a linguagem de programação Microsoft Visual Basic versão 6.0.

Os comandos que chegam ao servidor, enviados pelo navegador WEB, são processados pelo aplicativo CGI, que executa as seguintes ações:

- Capturar dados: Para o envio dos dados pelo navegador foi utilizado o método GET; desta maneira a variável de ambiente QUERY_STRING contém os dados que são recuperados por meio de uma função capaz de ler o conteúdo das variáveis de ambiente.
- Decodificar dados: Ao enviar os dados o navegador faz a codificação de acordo com a URL Encoding Specification. Apresenta-se abaixo um conjunto de comandos na forma como devem ser digitados para serem reconhecidos e interpretados pelo SuperLogo, e os mesmos comandos após a codificação pelo navegador WEB.

```
rcxsetasensorluz "1  
rcxligamotor "c  
rcxlrsensorluz "1  
se :valorsensorluz < 700 [rcxdesligamotor "c ] [rcxligamotor "c ]
```

Comandos digitados no formulário

```
S1=rcxsetasensorluz+%221+%0D%0Arcxligamotor+%22c%0D%0Arcxlers  
ensorluz+%221+%0D%0Ase+%3Avalorsensorluz+%3C+700+%5Brcxdeslig  
amotor+%22c+%5D+%5Brcxligamotor+%22c+%5D&B1=Submeter
```

Comandos após codificação

Portanto o aplicativo CGI deve efetuar a decodificação para que os dados retornem ao formato original. Este processo é executado por um algoritmo que busca as ocorrências dos caracteres codificados e os substitui pelo caractere original além de eliminar os caracteres de controle.

- Gravar dados: um arquivo é atualizado com os dados recebidos remotamente e já decodificados, este arquivo contém a sequência de comandos que será executada pelo SuperLogo.
- Sinalizar SuperLogo: é gerado um arquivo contendo uma mensagem que sinaliza ao SuperLogo a existência de dados recebidos remotamente, e que os mesmos devem ser executados.
- Retornar mensagem ao navegador: uma mensagem é retornada ao servidor que trata de encaminhá-la ao navegador; esta mensagem fornece ao usuário a indicação do sucesso ou não do processo.

6.4. Monitoração do robô remoto

Normalmente os robôs sob controle no ambiente de Telererobótica executam movimentos a partir dos comandos enviados pelo aprendiz. Por exemplo, um robô com a função de um braço mecânico, para deslocar uma peça de um local a outro, deverá receber comandos para abrir e fechar sua garra, levantar a peça e efetuar o deslocamento até a posição desejada. Para que o aprendiz possa acompanhar essas ações, verificando se as instruções enviadas corresponderam à ação desejada é necessário prover meios para visualização em tempo real do que está ocorrendo com o robô sob controle.

No ambiente de Telerobótica uma câmera de vídeo captura a imagem do robô sob controle e a transmite, via internet, para o aprendiz que está realizando a interação (Figura 6.65).



Figura 6.65 Instalação física do ambiente de telerobótica

Para captura e apresentação do vídeo é utilizada uma câmera conectada ao PC, compatível com o padrão VFW(Vídeo For Windows) e o programa Webcam da empresa Truetech .

Algumas características justificam a escolha do programa WebCam como programa para a apresentação de vídeo, sendo as mais relevantes a estabilidade de operação, a compactação e transmissão do vídeo capturado em tempo real, a dispensa de instalação de programa pelo usuário, além de apresentar diversos recursos de configuração e a distribuição em regime de freeware.

O programa WebCam dispõe de módulos para captura e compactação de vídeo e também de um servidor de vídeo (Figura 6.66). O acesso ao vídeo disponibilizado ocorre por meio de uma página HTML com um applet Java embutido.

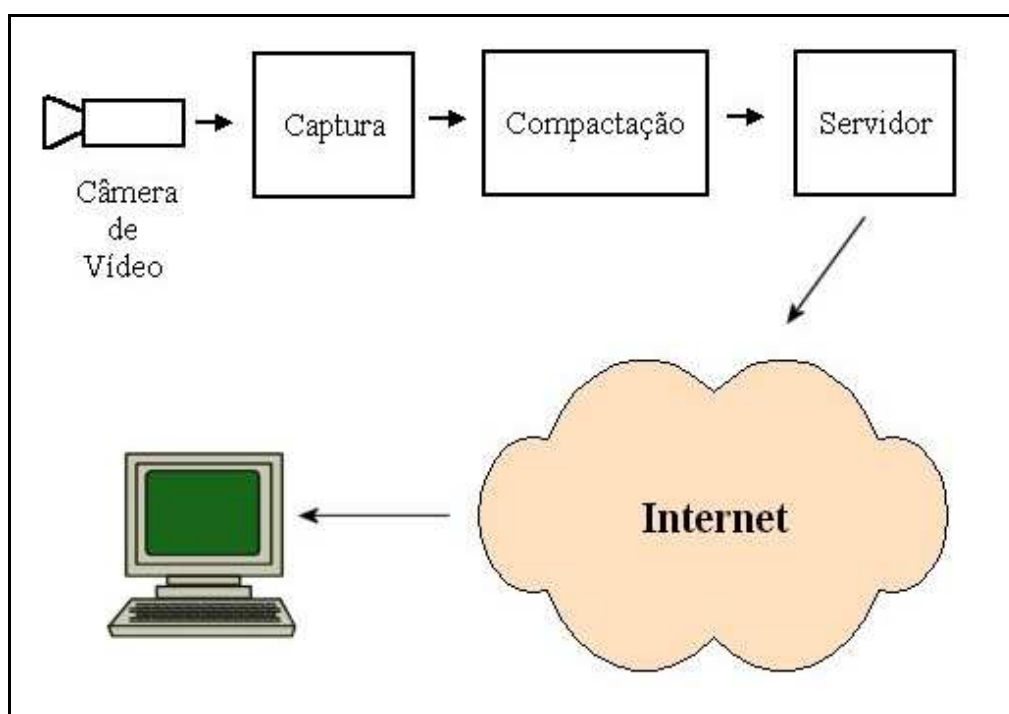


Figura 6.66 Diagrama em blocos do programa de captura de vídeo

O vídeo é transmitido a uma taxa de 5 frames por segundo. Testes realizados no protótipo instalado na rede do NIED, com acesso por linha discada, comprovaram que com esta taxa de transmissão é possível uma visualização adequada, mesmo com largura de banda típica de conexão discada com modem de 56 kbps.

6.5. Comunicação com o usuário

A fim de que a interação do aprendiz com o Ambiente de Telerobótica possa estimular atividades colaborativas entre aprendizes dispersos geograficamente, foi incluído o recurso de comunicação por chat (bate-papo). Assim, aprendizes que estejam participando de sessões de trabalho no Ambiente de Telerobótica podem comunicar-se entre si para sugerir ou discutir atividades a serem realizadas com o dispositivo robótico sob controle. Também é possível a comunicação com o operador que se encontra no mesmo local do robô, caso em que podem ser solicitadas alterações ou ajustes na configuração física dos dispositivos.

Para implementação do recurso de chat foi utilizado o programa Infocchat (12planet, 2002). Este programa é composto de um servidor para chat com a tecnologia Java e um applet que é carregado pelo computador cliente.

A fim de promover a integração entre os recursos de controle do robô, apresentação de imagens e comunicação, foi desenvolvida uma interface gráfica que agrega esses recursos e permite que sejam utilizados de forma simultânea (Figura 6.67).



Figura 6.67 Interface gráfica do ambiente de telerobótica

A interface gráfica foi desenvolvida em HTML, e dispõe de um formulário onde os comandos SuperLogo são digitados. Para que os comandos sejam processados pelo servidor e o dispositivo sob controle execute a ação correspondente, o usuário deve clicar no botão “Executar Comandos”.

Uma janela com o applet Java, lado cliente do programa WebCam, é responsável por exibir o vídeo com imagens em tempo real do dispositivo robótico que está sendo controlado.

Se o aprendiz desejar comunicar-se por meio de chat com o operador, que está no mesmo local do dispositivo robótico, ou com outros aprendizes que estejam acompanhando as atividades, ele deve clicar no botão “chat”, que abrirá uma pequena janela onde o applet

Java, lado cliente do servidor Infochat, apresentará uma interface onde é possível digitar a comunicação e ler o que os outros estão escrevendo (Figura 6.68).



Figura 6.68 Janela de chat (bate-papo) do ambiente de telerobótica

Uma janela na parte inferior, contém informações com a sintaxe dos comandos SuperLogo para robótica.

Para o desenvolvimento de atividades de telerobótica com alunos, foi implementado o projeto SIROS (Sistemas Robóticos com SuperLogo). O sistema foi montado no NIED é composto era constituído do hardware, aplicativos, dispositivos robóticos e o site do projeto acessível no endereço <http://www.nied.unicamp.br/~siros>.

No site encontra-se a interface gráfica para controle dos dispositivos robóticos e material de referência com informações detalhadas sobre os dispositivos robóticos disponíveis, a documentação dos comandos de robótica, do SuperLogo e exemplos de utilização.

6.6. Considerações sobre o capítulo

A rede Internet tem sido usada principalmente como ferramenta de comunicação e disseminação de informações. O ambiente de telerobótica implementado propõe uma opção de utilização da Internet ao permitir que dispositivos robóticos possam ser controlados remotamente propiciando, a pessoas e grupos que não disponham de laboratórios equipados com conjuntos para robótica a possibilidade de programar e controlar dispositivos robóticos.

Capítulo 7

Aplicação do Ambiente de Robótica Educacional

Uma atividade que acompanhou o processo de desenvolvimento do ARE foi a utilização deste ambiente com aprendizes. O objetivo desta atividade foi avaliar aspectos como facilidade de utilização, funcionalidade dos diversos recursos, confiabilidade em condições reais de uso e a receptividade do aprendiz, representada pelo grau de interesse demonstrado em explorar e aprender sobre as possibilidades oferecidas pelo ambiente.

O ARE foi aplicado em ambientes formais de educação como escolas, por meio de oficinas e também em locais como feiras de ciências e tecnologia, palestras, e um evento em parque de diversões.

7.1. A utilização do ARE por aluno do curso de Mecatrônica

A primeira versão do ARE foi utilizada por um aluno do curso de Mecatrônica da Unicamp, que à época fazia estágio no NIED e estava elaborando um projeto de iniciação científica, no qual se propunha o desenvolvimento de um dispositivo automatizado, para efetuar a classificação de produtos por meio do reconhecimento de cores.

O plano de trabalho do aluno incluía o desenvolvimento de alguns protótipos com o Lego RCX, a fim de determinar se com os recursos disponíveis no RCX seria possível desenvolver o projeto. Ele já tinha experiência de programação com o Lego RCX nos ambientes Robolab e Mindstorms, e a princípio descartou estes ambientes em razão de os mesmos não oferecerem recursos para integração com outras ferramentas, como banco de dados, e outras plataformas de robótica empregadas em automação industrial.

Naquela época, a Interface SuperLogo RCX (ISR) e os procedimentos SuperLogo relacionados ao controle do Lego RCX já estavam implementados. Assim, foi sugerida a utilização do ARE na elaboração desses protótipos.

Com um kit composto pelo RCX, sensores de luz e toque, motores, e blocos de montagem, o aluno iniciou algumas montagens e testes para avaliar o comportamento do ARE.

Inicialmente o aluno desenvolveu uma série de procedimentos, que incluíam entre outros o acionamento dos motores, alteração no sentido de rotação dos mesmos,

acionamento e desligamento em função do estado dos sensores, verificação dos valores obtidos pelo sensor de luz e a sua variação, dependendo da iluminação.

A seguir foi montado um carro motorizado (Figura 7.69), que deveria ser capaz de desviar-se de obstáculos e buscar um caminho alternativo. Constituído de dois motores e sensores de toque na parte frontal, um do lado direito e outro do lado esquerdo, era controlado por um programa que testava constantemente o estado dos sensores e acionava os dois motores, com movimentos que alteravam o sentido do carrinho quando o sensor tocava algum obstáculo.

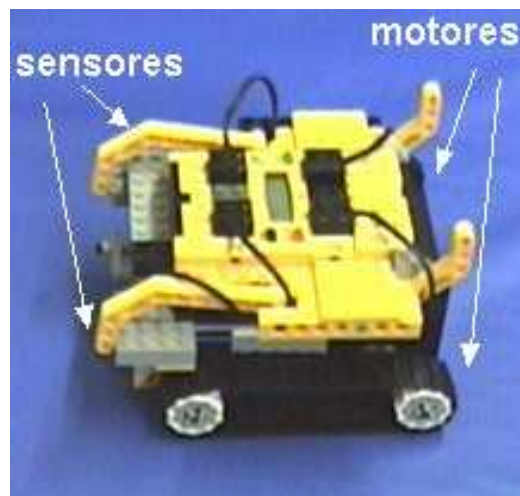


Figura 7.69 Carro Lego RCX

O código em SuperLogo criado pelo aluno para controle do carro é apresentado abaixo:

aprenda carro

rcxsetasensortoque [1 3]

rcxligamotor "ac

atr "lcarro 0

façaenquanto [

atr "s0 rcxlersensor 1

atr "s1 rcxlersensor 3

mostre :s0

se e :s0=0 :s1=0[

rcxmotorfrente "ac

```

]
se e :s0=1 :s1=1[
    rcxrevertemotor "ac
    espere 30
    rcxmotorfrente "a
]
se e :s0=1 :s1=0 [
    rcxrevertemotor "ac
    espere 30
    rcxmotorfrente "a
]
se e :s0=0 :s1=1 [
    rcxrevertemotor "ac
    espere 30
    rcxmotorfrente "c
]
]
[:lcarro=0]
fim

```

Outros protótipos foram desenvolvidos incluindo um traçador gráfico (plotter) que, acoplado a um lápis, realizava desenhos no papel, conforme eram enviados comandos; uma porta automática com funcionamento semelhante ao das portas encontradas em locais públicos, que se abria com a aproximação de um objeto e, finalmente, o leitor de cor. Para descrever essas trabalhos o aluno elaborou um relatório apresentado no Apêndice 5.

Com este trabalho o aluno foi capaz de testar hipóteses, projetar, desenvolver algoritmos e elaborar protótipos que posteriormente foram aplicados no seu trabalho de iniciação científica (Takiguthi, 2001), neste caso aplicando ferramentas específicas de automação industrial.

7.2. Utilização do ARE com alunos do curso de Pós-Graduação em Informática na Educação.

A FAI – Faculdade de Administração e Informática de Santa Rita do Sapucaí, Minas Gerais, oferece o curso de Pós-Graduação em Informática Aplicada à Educação. Neste curso os alunos recebem formação teórica em disciplinas que tratam dos fundamentos da pedagogia e da psicologia entre outros e formação técnica na qual são apresentadas ferramentas computacionais com enfoque educacional, com destaque para programas de autoria para multimídia e Internet. Os alunos, contudo, não têm contato com linguagens de programação.

Entre as disciplinas técnicas existe a de Robótica Educacional, com duração de 30 horas. A proposta desta disciplina é discutir com os alunos os fundamentos da aplicação da robótica em contextos educacionais e propiciar meios para que eles desenvolvam atividades práticas, o que inclui a montagem e programação de dispositivos robóticos.

Este curso de Pós-Graduação atende professores de ensino fundamental das redes pública e particular, diretores de escola, coordenadores pedagógicos e professores que trabalham com alunos portadores de necessidades especiais.

A direção da FAI em contato como o autor deste trabalho, solicitou a elaboração de uma proposta para o desenvolvimento da disciplina de Robótica Educacional. Esta disciplina é a última do curso, de forma que os alunos já dispõem de conhecimentos sobre aplicativos mais comuns como editores de texto, planilhas e em programas para criação de multimídia e páginas para Internet.

Com base nessa informação foi proposta uma ementa para a disciplina de Robótica Educacional com a abordagem dos seguintes tópicos.

- Fundamentos e conceitos da robótica aplicada à educação.
- A linguagem Logo, aspectos pedagógicos e computacionais.
- Apresentação do SuperLogo.
- Prática na programação do SuperLogo no modo gráfico.
- Controle de motores e sensores com o SuperLogo.
- Desenvolvimento de projeto prático com robótica.

As diretrizes propostas por esta ementa foram aplicadas em duas turmas: uma nos meses de agosto e setembro de 2001 e outra nos meses de fevereiro e março de 2002, perfazendo um total de sessenta horas aula.

No primeiro contato com os alunos foi constatada a falta de conhecimento da maioria deles sobre o que é robótica educacional e quais suas aplicações em um ambiente de ensino-aprendizagem. Alguns se mostraram receosos quanto a sua capacidade para desenvolver projetos de robótica, outros vislumbravam sua utilização apenas em áreas relacionadas às disciplinas da área de exatas, como física e matemática, mas havia dúvidas quanto à utilização para disciplinas como história, línguas, geografia, e outras.

Nas etapas de fundamentação, apresentação e programação do SuperLogo no modo gráfico, os alunos puderam perceber que os recursos disponíveis propiciam meios para o desenvolvimento das mais variadas aplicações. Com a atividade de programação e controle de motores e sensores os alunos adquiriram fluência e segurança que facilitaram a concepção do projeto prático, o qual consistia em montar e programar um dispositivo robótico baseado em um tema que, de alguma forma estivesse ligado a sua prática de sala de aula.

Para o desenvolvimento do projeto prático foram disponibilizados conjuntos constituídos pelo SuperLogo com os comandos de robótica, Interface de Hardware para Robótica (IHR), servomotores e sensores. Material de apoio impresso, contendo uma referência do ambiente SuperLogo e dos comandos gráficos e de robótica, serviu de fonte de consulta para sanar dúvidas na elaboração do programa para controle do dispositivo robótico.

A seguir são apresentados alguns projetos desenvolvidos pelos alunos.

7.2.1. Projeto Dinossauro

O objetivo deste projeto foi explorar conceitos de ecologia, cadeia alimentar e a relação entre predadores e presas. Para tanto foi montado um cenário, contendo várias peças plásticas que representavam dinossauros de diversas espécies e características (Figura 7.70). Os dinossauros “presa” foram distribuídos sobre uma base e posicionados sobre sensores de toque. O dinossauro “predador” foi montado em uma base com dois servomotores e rodas, podendo locomover-se. A operação desejada pela aluna era que, quando um dinossauro “presa” se movimentasse, isto é, quando ele fosse tirado da sua posição o dinossauro predador se movesse até essa posição e simulasse um ataque. Para que essa funcionalidade fosse obtida a aluna elaborou procedimentos que efetuavam a leitura do estado dos sensores e, ao detectar a alteração, acionavam o funcionamento dos

servomotores do dinossauro “predador”, de modo que ele se encaminhasse na direção adequada. Vários ajustes foram realizados nos procedimentos que acionavam os servomotores até que os trajetos correspondessem ao desejado.



Figura 7.70 Projeto dinossauro

7.2.2. Projeto Pauta Musical

A aluna que desenvolveu este projeto atua em arte educação, especificamente no ensino de música, aplicando uma metodologia que envolve a utilização de desenhos e gráficos que representam notas musicais. No seu trabalho prático de robótica ela propôs acrescentar aos materiais empregados em sua metodologia os recursos tecnológicos propiciados pela robótica. Para tanto foi idealizado um teclado semelhante a um piano com sensores de toque para cada nota musical (Figura 7.71). O procedimento desenvolvido com o SuperLogo obtinha o estado dos sensores e tocava a nota correspondente, apresentando no monitor do computador uma imagem que representava a nota e um pequeno texto . Para que o conteúdo multimídia (sons e imagens) fosse apresentado, foi preciso que a aluna pesquisasse os comandos do SuperLogo para estas funções.



Figura 7.71 Projeto pauta musical

7.2.3. Projeto Emília

Associar literatura portuguesa a robótica foi o projeto idealizado por um grupo de alunas. Os livros e histórias de Monteiro Lobato eram o assunto que o grupo estava trabalhando na sala de aula, portanto foi o escolhido. Tomando como personagem central a Emília, o projeto consistia em uma maquete onde esta personagem, representada por um boneco de pano, faria movimentos com o corpo e por meio do computador contaria histórias (Figura 7.72). A implementação envolveu a instalação de servomotores no interior do boneco e sensores para que pudesse haver interação com o usuário. Neste projeto foram utilizados os comandos multimídia do SuperLogo para reproduzir arquivos de sons e para apresentar imagens no monitor. Estes comandos, associados aos de robótica, foram utilizados para a criação de procedimentos que contavam uma história com o boneco da Emília executando movimentos. Os procedimentos para sensores não foram desenvolvidos, pois as alunas usaram parte substancial do tempo disponível na preparação das mídias e na montagem da maquete comprometendo o tempo que seria aplicado à programação dos sensores.



Figura 7.72 Projeto emília

7.2.4. Projeto Energia Alternativa

A fim de demonstrar o funcionamento de sistema de geração de energia por meio da força do vento, um aluno desenvolveu um maquete (Figura 7.73) constituída de elementos que representavam um catavento , uma linha de transmissão de energia e uma televisão que executava alguns movimentos representando seu funcionamento. O procedimento SuperLogo desenvolvido demonstrava que somente quando o catavento funcionasse é que haveria energia para que a televisão funcionasse.



Figura 7.73 Projeto energia alternativa

7.3. Oficina de Robótica Educacional com alunos do ensino fundamental.

A fim de validar o ARE, o autor deste trabalho desenvolveu duas oficinas com duração de doze horas cada, nas quais os alunos participantes apresentavam idades entre 11 e 15 anos. As oficinas ocorreram nas escolas EE José Arantes Terra e EE Gilbert de Carvalho, ambas na cidade de Araçatuba. Cada turma foi formada por 15 alunos e a participação era voluntária.

Foram utilizados os laboratórios de informática dessas escolas; e para desenvolver os trabalhos, foram colocados à disposição dos alunos conjuntos contendo a IHR, servomotores e o programa SuperLogo.

Aos alunos, foi apresentado o ambiente do SuperLogo, os comandos básicos para programação gráfica e os comandos para robótica. A seguir eles tiveram a oportunidade de elaborar procedimentos para controlar um manipulador robótico (Figura 7.74).



Figura 7.74 Grupo de alunos programando robô

A atividade seguinte foi a proposta de desenvolvimento de um pequeno projeto que envolvesse a construção e programação de um dispositivo robótico. Os materiais utilizados seriam plástico, papelão de embalagens, brinquedos usados e cola entre outros. Estes projetos não incluíram a utilização de sensores, pois em razão da limitação de tempo, não foi possível trabalhar os conceitos necessários à sua aplicação.

Um dos projetos desenvolvido foi uma máquina automática de refrigerantes (Figura 7.75), constituída por três servomotores, que ao se deslocarem empurravam uma

lata (Figura 7.76). Um procedimento no SuperLogo permitia ao usuário selecionar o refrigerante desejado.



Figura 7.75 Projeto máquina de refrigerante

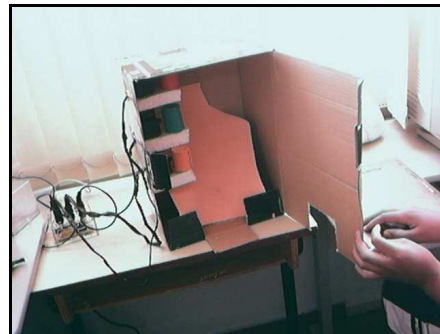


Figura 7.76 Vista da máq. de refrigerante

Alguns outros projetos foram o robô “Pega-Lixo” (Figura 7.77), construído com dois servomotores, que era capaz de segurar objetos abrindo e fechando um braço; o Parque de Diversões (Figura 7.78) com servomotores que controlavam uma roda gigante e um carrossel; e a Mão Robótica (Figura 7.79) com dois servomotores possibilitando movimentos na vertical e horizontal.

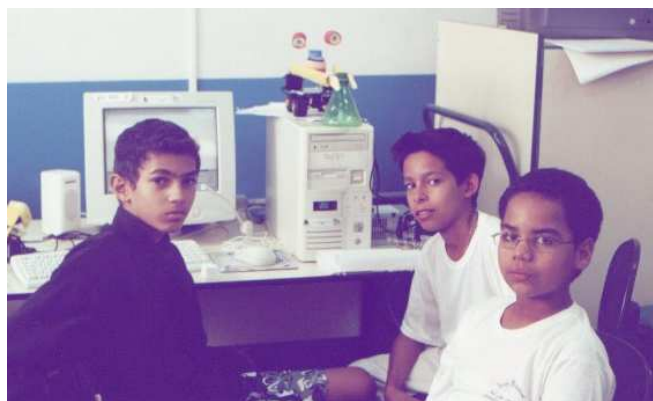


Figura 7.77 Projeto Pega-Lixo



Figura 7.78 Projeto parque de diversões



Figura 7.79 Projeto mão robótica

O que se observou em ambos os grupos é que todos conseguiram compreender os princípios básicos da programação no SuperLogo e com esse conhecimento foram capazes de elaborar o seu projeto.

7.4. Utilização do ARE no modo remoto – Telerobótica

O Ambiente de Robótica Educacional (ARE) no modo remoto foi implantado em um microcomputador PC, do NIED equipado com câmera de vídeo, conexão à rede Internet, manipulador robótico conectado a uma IHR e o Lego RCX.

Na fase inicial de testes foi utilizada a intranet do NIED para avaliar a funcionalidade no controle e visualização do robô. A seguir foram realizados os mesmos testes utilizando uma conexão por linha discada, nesta situação, caracterizada pela baixa velocidade de comunicação foi possível verificar o tempo de resposta para execução dos comandos e a atualização do vídeo, que se mostraram satisfatórios.

Constatado o funcionamento de todo o sistema, foi preparada uma oficina de telerobótica, com a participação do parceiro de projeto do NIED na OEA (Organização dos Estados Americanos) da Argentina.

Durante o mês de outubro de 2001 foram realizadas atividades no ambiente de telerobótica, com um grupo de alunos da Argentina controlando os robôs que estavam disponíveis no laboratório do NIED. As sessões de controle eram realizadas com acompanhamento por bate-papo (Chat) entre os participantes onde eram sugeridas atividades e os alunos solicitavam alterações nos robôs ou enviavam questionamentos sobre os comandos e/ou o robô.

Em maio de 2002 foi realizada uma nova oficina de telerobótica com, parceiros do NIED no projeto da OEA da Argentina, Chile, Costa Rica e Brasil. Durante um mês o sistema esteve disponível para o desenvolvimento de atividades de controle.

7.5. Considerações sobre o capítulo

As várias atividades desenvolvidas com todos os elementos do ARE permitiram avalia-lo quanto a sua funcionalidade, confiabilidade e facilidade de utilização. Todo o sistema operou adequadamente sem apresentar problemas. Alterações foram realizadas nos procedimentos SuperLogo para o Lego RCX, com o objetivo de facilitar sua utilização, assim alguns comandos foram renomeados e comandos simplificados para sensores foram implementados.

A IHR mostrou-se confiável e robusta, além de ser utilizada nas oficinas foi apresentada em eventos onde chegou a operar por dez horas seguidas durante cinco dias.

A visualização bidimensional propiciada pela câmera do ambiente de Telerobótica dificulta algumas atividades, como, por exemplo, pegar uma peça com o manipulador robótico. O acréscimo de uma segunda câmera, pode melhorar esta condição.

Capítulo 8

Conclusão

8.1.Considerações Finais

Os trabalhos e estudos realizados, e que resultaram no ambiente de robótica para aplicações educacionais, envolveram o desenvolvimento de um conjunto de programas e equipamentos eletrônicos que, integrados, propiciam ao usuário meios para construir dispositivos robóticos, atividade que envolve a montagem da estrutura mecânica, a adição de motores responsáveis pelos movimentos, de sensores para a conversão de grandezas físicas do meio ambiente em instruções que podem ser lidas pelo computador, e a elaboração de um programa para dar funcionalidade ao projeto.

As opções de projeto para a Interface de Hardware para Robótica (IHR) propiciaram um equipamento compacto e que nas atividades realizadas com aprendizes se mostrou confiável, não apresentando nenhum problema eletrônico ou mecânico. Com o emprego do microcontrolador, atualizações ou alterações para o acréscimo de novas funcionalidades podem ser aplicadas com a modificação do programa do microcontrolador, evitando alterações no hardware o que normalmente envolve a elaboração de novos circuitos e confecção de placas, encarecendo e reduzindo a flexibilidade da IHR.

Uma característica a ser considerado no ARE é que o mesmo não foi desenvolvido apenas para atividades de robótica a partir de tecnologias específicas, diferentemente de produtos analisados como Robolab e Mindstorms, que têm o seu conjunto de instruções dedicados a controle de interfaces da empresa Lego, de forma que caso o usuário tenha a intenção de elaborar projetos com outros recursos, necessitará também de outro programa.

A utilização do SuperLogo, uma linguagem de programação genérica, permite que uma abordagem interdisciplinar seja adotada no desenvolvimento de projetos. Elementos multimídia como sons, imagens, vídeo e textos, explorados nas diversas disciplinas podem ser acrescidos dos recursos de robótica criando, condições para que o aprendiz estabeleça conexões entre os diversos conteúdos.

A concepção modular do ARE permite que outras tecnologias e equipamentos para robótica possam ser integradas agregando recursos ao ambiente.

A extensão geográfica de um país como o Brasil e o custo dos equipamentos pode restringir o acesso dos aprendizes a ambientes para robótica; neste sentido, a utilização

remota do ambiente de robótica, que necessita unicamente de um computador conectado à Internet, permite que um número maior de pessoas tenha acesso e possa acompanhar as atividades de robótica.

Adotando-se uma metodologia pedagógica adequada é possível abordar concretamente e de forma contextualizada os diversos conceitos utilizados nas práticas da sala de aula, estabelecendo conexões entre os diversos conteúdos, promovendo a interdisciplinaridade e estimulando o trabalho cooperativo.

O desenvolvimento remoto de projetos de robótica pode desencadear a formação de comunidades de aprendizagem, que, embora dispersas geograficamente, comunicam-se para discutir, trocar informações e atuar no controle de dispositivos robóticos.

Os conhecimentos adquiridos no desenvolvimento deste trabalho criam condições para que, em função dos objetivos e necessidades, alterações nos programas e no hardware possam ser implementadas para adequá-lo a novos contextos de utilização.

8.2. Perspectivas para futuros trabalhos

A oferta, pelo mercado de computadores pessoais com poder de processamento e armazenamento crescentes, o acesso mais fácil à internet com banda larga e a oferta de componentes eletrônicos, particularmente os microcontroladores, com mais recursos e a um custo reduzido apresentam um vasto campo para o desenvolvimento de recursos e ferramentas para aplicações educacionais.

Os computadores podem ser convertidos por meio de programas adequados, em verdadeiros laboratórios, obtendo informações do mundo exterior por meio de interfaces e apresentando-os em gráficos para análise e manipulação. Se o computador estiver conectado à Internet essas informações poderão ser visualizadas e o usuário poderá atuar diretamente no experimento, mesmo estando distante do laboratório, criando-se assim verdadeiros laboratórios on-line. As interfaces de hardware desenvolvidas a partir de microcontroladores com mais recursos e capacidade de processamento permitirão o desenvolvimento de atividades e dispositivos robóticos com maior nível de complexidade.

Todo esse processo deve ser acompanhado de metodologias adequadas de aplicação capazes valorizar o senso crítico do aprendiz e os diversos estilos de aprendizado.

Esta dissertação de mestrado acredita ter contribuído para o desenvolvimento pesquisas que levem à elaboração de metodologias e instrumentos tecnológicos capazes de fazer frente aos desafios educacionais da nossa sociedade.

Glossário

Centronics - Padrão de interface para conexão de impressoras e outros dispositivos com comunicação paralela.

HTML (HyperText Markup Language) - linguagem de autoria usada para criação de documentos na World Wide Web.

HTTP (HyperText Transfer Protocol) - protocolo utilizado pela World Wide Web que define como as mensagens são formatadas e transmitidas, e quais ações os servidores e navegadores devem executar em resposta aos vários comandos.

IIS (Internet Information Server) - programa servidor de arquivos da empresa Microsoft

ISA - arquitetura de barramento utilizada em microcomputadores IBM PC/XT e PC/AT.

MIME (Multipurpose Internet Mail Extensions) - especificação para formatação de mensagens não texto para que possam ser enviadas e reconhecidas na rede Internet.

NIED - Núcleo de Informática Aplicada a Educação da Universidade de Campinas Unicamp.

RS-232 (Recommended Standard-232) - padrão de interface aprovado pela EIA (Electronic Industries Association) para conexão de dispositivos com comunicação serial.

USB (Universal Serial Bus) - padrão de barramento que suporta taxas de transferência de até 12 Mbps. Uma única porta suporta a conexão de até 127 periféricos com suporte a Plug-and-Play.

Referências

12PLANET. **Infochat** : 12planet, 2002. <<http://www.12planet.com/en/index.html>>. Servidor de Chat.

BERNERS-LEE, T., MASINTER, L., MCCAHERILL, M. **RFC1738**,1994. Disponível em: <<http://www.w3.org/Addressing/rfc1738.txt>> Acesso em: 23/04/2002.

BERS, M.U., URREA, C. **Con-Science Parents and Children Exploring Robotics and Values, 1994.** Disponível em: < <http://www-iaa.fmi.uni-sofia.bg/eurologo99/Contents/Conddoc.htm> > , 1999. Acesso em: 23/04/2002.

BRAINTENBERG, V. **Vehicles Experiments in Synthetic Psychology.** 3ª ed. Cambridge: MIT Press, 1989. 152 p.

CHELLA, M.T. Ambiente de Robótica Educacional com Logo In: **XXII Congresso da Sociedade Brasileira de Computação – SBC2002**, Florianópolis, SC, 2002.

D´ABREU, J.V.V. **Computadores e Conhecimento.** Campinas: UNICAMP/NIED, 1993. Cap. 17: Uso do computador para controle de dispositivos: O traçador gráfico (Plotter). p. 350-366.

D´ABREU, J.V.V., CHELLA, M.T. Desenvolvimento de Ambientes Colaborativos de Aprendizagem à Distância Baseado no Controle de Dispositivos Robóticos In: **XII Simpósio Brasileiro de Informática na Educação – SBIE2001**, Vitória, ES, p. 393-400, 2001.

D´ABREU, J.V.V., **Integração de Dispositivos Mecânicos para Ensino-Aprendizagem de Conceitos na área de Controle e Automação.** Tese de Doutorado, Universidade Estadual de Campinas – UNICAMP, Faculdade de Engenharia Mecânica – FEM, Campinas, 2002.

FLAVELL, J. H. **A Psicologia do Desenvolvimento de Jean Piaget.** 2ª ed. São Paulo: Pioneira, 1986. 479 p.

FRIENDLY, M. **Advanced Logo a Language for Learning.** New Jersey: Lawrence Erlbaum Associates, 1988. 420 p.

GARCIA, M. F., **Ambiente Logo e Interdisciplinaridade: A Concepção dos Professores.** Dissertação de mestrado Faculdade da Educação FE/UNICAMP, 1995.

GEORGE, S., DESPRES, C. A Multi-Agent System For a Distance Support in Educational Robotics. **International Conference on Telecommunication for Education and Training**, Gjøvik, Norway, p. 344-353, 1999.

GOODYEAR, P. **LOGO Introdução ao poder do ensino através da programação.** Rio de Janeiro: Campus, 1986. 204 p.

IMATIX. **Xitami:** Imatix, 2002. <<http://www.imatix.com>>. Servidor WEB.

JARVINEN, E. The Lego/Logo Learning Environment in Technology Education: An Experiment in a Finnish Context **Journal of Technology Education**, v. 9, n. 2, p. 47-59, maio de 1998. <http://scholar.lib.vt.edu/ejournals/JTE/v9n2/pdf/jarvinen.pdf>. Acesso em: 08/05/2002.

KIM, E. E. **CGI Developer's Guide.** 1ª ed. Indianapolis: Sams.net, 1996. 471 p.

LUCENA, C., FUKS, H. **A Educação na Era da Internet.** Rio de Janeiro: Clube do Futuro, 2000. 160 p.

MICROCHIP. **PIC16F84** Disponível em:
<<http://www.microchip.com/1010/pline/picmicro/category/digictrl/8kbytes/devices/16f84a/index.htm>> Acesso em: 23/04/2002.

OLIVEIRA, C.D. **Computadores e Conhecimento.** Campinas: UNICAMP/NIED, 1993. Cap. 19: O sistema Lego-LOGO no ensino da Física. 379-384.

PAPERT, S. **A máquina das Crianças Repensando a Escola na Era da Informática.** Porto Alegre: Artes Médicas, 1994. 210 p.

PAPERT, S. **LOGO: Computadores e Educação.** São Paulo: Editora Brasiliense, 1985. 184 p.

Remotobot.net Disponível em: <<http://remotobot.k-team.com/museum/>> Acesso em: 10/05/2002.

RESHKO, G. B., MASON, M.T., NOURBAKHASH, I.R. **Rapid Prototyping of Small Robots Draft**, 2001. Disponível em:
<<http://www.cs.cmu.edu/~reshko/Publications/prototyping.pdf>> Acesso em: 24/06/2002.

RUOCCO, S. R. **Robot Sensor and Transducers.** 1ª ed. New York: Halsted Press, 1987. 166 p.

SIDERICOUEDES, A. **Computadores e Conhecimento.** Campinas: UNICAMP/NIED, 1993. Cap. 18: Uma atividade Lego-LOGO em trigonometria. p. 367-378.

STAGER, G.S. Constructionism as a High-Tech Intervention Strategy for At-Risk Learners. **National Educational Computing Conference, "Building on the Future"**, p. 1-11, julho de 2001. http://confreg.uoregon.edu/necc2001/program/research_pdf/Stager.pdf. Acesso em: 12/11/2001.

STARR, P. Computing Our Way to Educational Reform. **The American Prospect**, v. 7, agosto de 1996. <http://www.prospect.org/print/V7/27/starr-p.html>. Acesso em: 05/02/2002.

TAKIGUTHI, G., **Implementação de Um Sistema Integrado para separação de produtos Numa Linha Automatizada de Produção a Partir do reconhecimento de Cores**. Relatório de Iniciação Científica, Universidade Estadual de Campinas – UNICAMP, Faculdade de Engenharia Mecânica – FEM, Departamento de Projeto Mecânico, Laboratório de Automação Integrada e Robótica – LAR/UNICAMP, Campinas, 2001.

TANENBAUM, A. S. **Structured Computer Organization**. 4ª ed. New Jersey: Prentice Hall, 1999. 651 p.

VALENTE, J.A. **Computadores e Conhecimento**. Campinas: UNICAMP/NIED, 1993. Cap. 1:Diferentes usos do computador na educação. p. 1-23.

VALENTE, J.A. **O computador na Sociedade do Conhecimento**. Campinas: UNICAMP/NIED, 1999. Cap. 1: Informática na Educação no Brasil: Análise e Contextualização Histórica. p. 1-27.

Apêndice 1 – Lego Mindstorms

Comandos do RCX

Os blocos de comando são aqueles blocos que informam ao RCX o que fazer; eles possuem a coloração esverdeada. Alguns desses comandos, como por exemplo *on*, *off*, *set power*, *set direction* e *reverse direction*, controlam as portas de saída **A**, **B** e **C**, onde os atuadores são conectados no RCX.

Nas portas de entrada **1**, **2** e **3** poderão ser conectados sensores de toque, de rotação, de temperatura e de luz.



BEEP: permite a emissão sonora de um dos seis tipos de sons existentes.



TONE: permite ao RCX emitir um tom, com uma determinada frequência, por um certo intervalo de tempo.



ON: aciona as portas **A**, **B** e/ou **C** do RCX.



ON FOR: aciona a porta **A**, **B** e/ou **C** por um determinado intervalo de tempo.



OFF: desliga as portas **A**, **B** e/ou **C**.



SET POWER: permite o ajuste da potência do motor, sendo o valor mínimo 1 e o máximo 8. A porta é sempre ativada na potência máxima (8), a menos que seja definido, anteriormente, outro valor.



SET DIRECTION: define o sentido em que o motor deve girar. Sentido horário ou anti-horário.



REVERSE DIRECTION: permite a inversão da rotação do motor.



WAIT: faz uma pausa no RCX por um certo intervalo de tempo antes que ele inicie o próximo comando. O tempo é dado em décimos de segundos.



RESET ROTATION: redefine a posição do ponto **0** do sensor de rotação. Este sensor sempre define como sendo o ponto **0** a posição na qual o RCX iniciou a execução do programa.



RESET MESSAGE: atualiza a mensagem no RCX receptor, permitindo a execução de comandos referentes à mensagem mais atual.



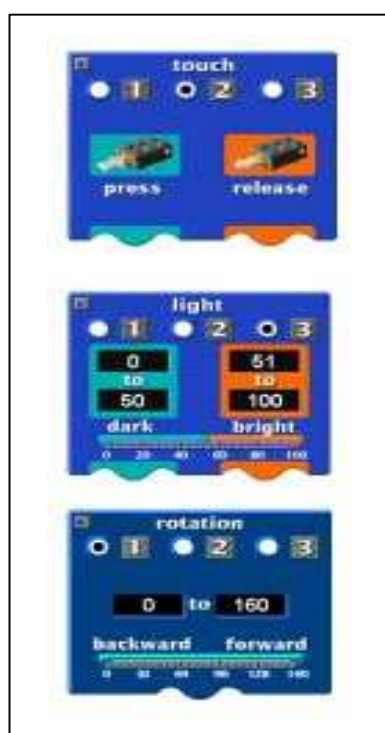
RESET TIMER: inicia a contagem a partir do **0** toda vez que o RCX começa a executar o programa. Usando-se este comando, é possível reiniciar a contagem em qualquer outra parte do programa.



SEND TO RCX: envia mensagem de um RCX para outro RCX para a execução de um determinado comando ou uma sequência deles.

Monitores de Sensores

Por intermédio de sensores, o RCX consegue “perceber” o ambiente onde se encontra. Por exemplo, através do sensor de luz, o RCX “percebe” a variação de luminosidade e, assim, permite ao programador que associe os fenômenos *escuro* e *claro* com acionamentos de motores, beeps ou acendimento de lâmpadas. Os blocos referentes aos sensores possuem coloração azulada. Usam-se sensores de toque, de temperatura, de ângulo ou de rotação quando o programador deseja que, devido à mudança de estado do sensor, o sistema execute determinados comandos.



SENSOR DE TOQUE: possui dois estados: pressionado (press) e não pressionado (release).

SENSOR DE LUZ: possui dois estados, definidos pelo programador, “escuro” e “claro”. Por exemplo, luminosidade 30% (dark) e 70% (bright).

SENSOR DE ROTAÇÃO: permite que o eixo acoplado ao sensor mantenha uma certa rotação dentro de um intervalo, com incremento de 16 unidades de contagem por rotação.



SENSOR DE TEMPERATURA: possui duas faixas de temperatura definidas pelo programador: “frio” e “quente”.



CONTADOR: executa os comandos anexados ao bloco “counter” (azul) quando o contador registra um número dentro do intervalo definido pelo programador. Para se utilizar o contador adequadamente, deve-se utilizar o bloco de comando “add to counter” (verde), que incrementará ao contador de uma unidade de contagem.



TIMER: inicia a contagem assim que o programa carregado no RCX, começa a ser executado. Os comandos anexados ao bloco “timer” serão iniciados dentro do intervalo estabelecido pelo programador.



RCX: estabelece a comunicação entre dois ou mais RCX. Com isso, um RCX(transmissor) pode enviar mensagem para outro RCX (receptor) para que este inicie um certo comando ou sequência deles. Os comandos anexados a este bloco serão executados assim que o transmissor enviar uma mensagem (“send to RCX”) para o receptor.

Stack Controllers

Estes blocos permitem que o programador escolha diferentes caminhos, que devem ocorrer durante a execução, para as partes do programa, usando como auxílio sensores, contadores, *timers* e mensagens para o controle do RCX. Estes blocos possuem coloração avermelhada.



CHECK & CHOOSE: segue um dos caminhos determinados, de acordo com a condição estabelecida pelo programador.



REPEAT: executa um determinado número de vezes os comandos anexados entre “ *begin repeat*” e “*end repeat*” . Este número pode ser estabelecido pelo programador ou configurado no modo aleatório “*random*”.



REPEAT FOREVER: executa dentro de um laço infinito os comandos anexados entre “ *begin repeat*” e “*end repeat*”.

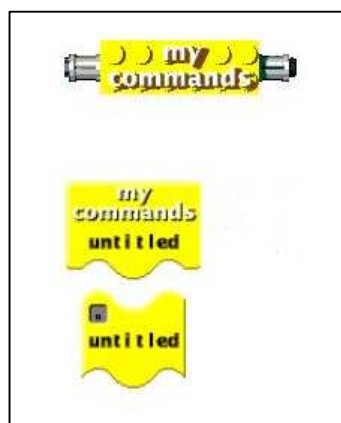


REPEAT WHILE: executa os comandos anexados entre “ *begin repeat*” e “*end repeat*” enquanto a condição para o sensor selecionado for verdadeira



WAIT UNTIL: faz uma pausa na execução do programa até que a condição estabelecida pelo programador seja satisfeita.

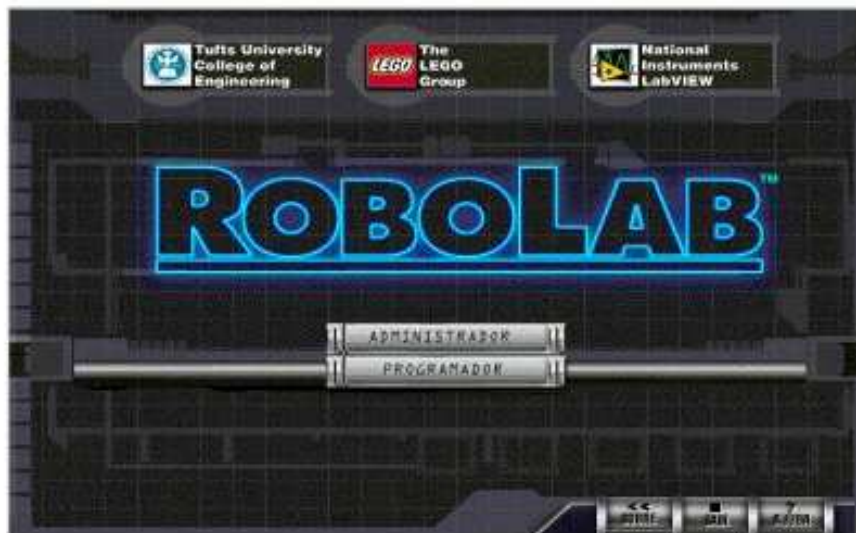
Comandos (SubProcedimentos)



MY COMMANDS: permite ao programador criar seus próprios comandos. Estes comandos são subprocedimentos que, uma vez criados passarão a ser incorporados ao conjunto de comandos do RCX.

Apêndice 2 - Robolab

Ao iniciar o Robolab, uma tela de introdução é aberta, apresentando duas opções: **Administrador** e **Programador**.



A opção **Administrador** serve para fazer as configurações do RCX e do software Robolab.

A opção **Programador** serve para abrir a tela do Menu Principal, a partir da qual pode-se optar em programar no modo Piloto ou no modo Inventor. Também faz parte da



tela principal os seguintes botões: **Sobre** (*About*), **Sair** (*Quit*) e **Ajuda** (*Help*).

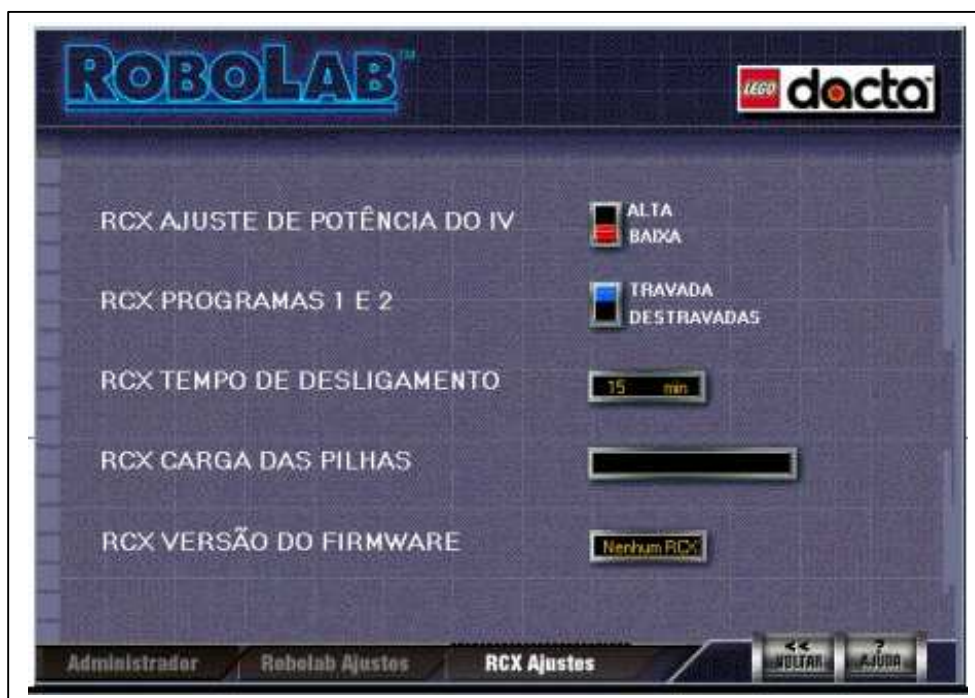
Na seção administrador existem opções para selecionar a porta serial a qual está conectada a torre de infravermelho, transferir o Firmware para o RCX e testar os componentes de comunicação incluindo porta serial, cabo, torre de infravermelho e resposta do RCX. Também a partir da seção administrador é possível esconder o botão administrador e executar um vídeo com demonstração dos recursos do Robolab.

A seção Ajustes do Robolab atua como um gerenciador de arquivos e projetos possibilitando configurar os locais dos arquivos de programa e a criação ou exclusão de projetos.

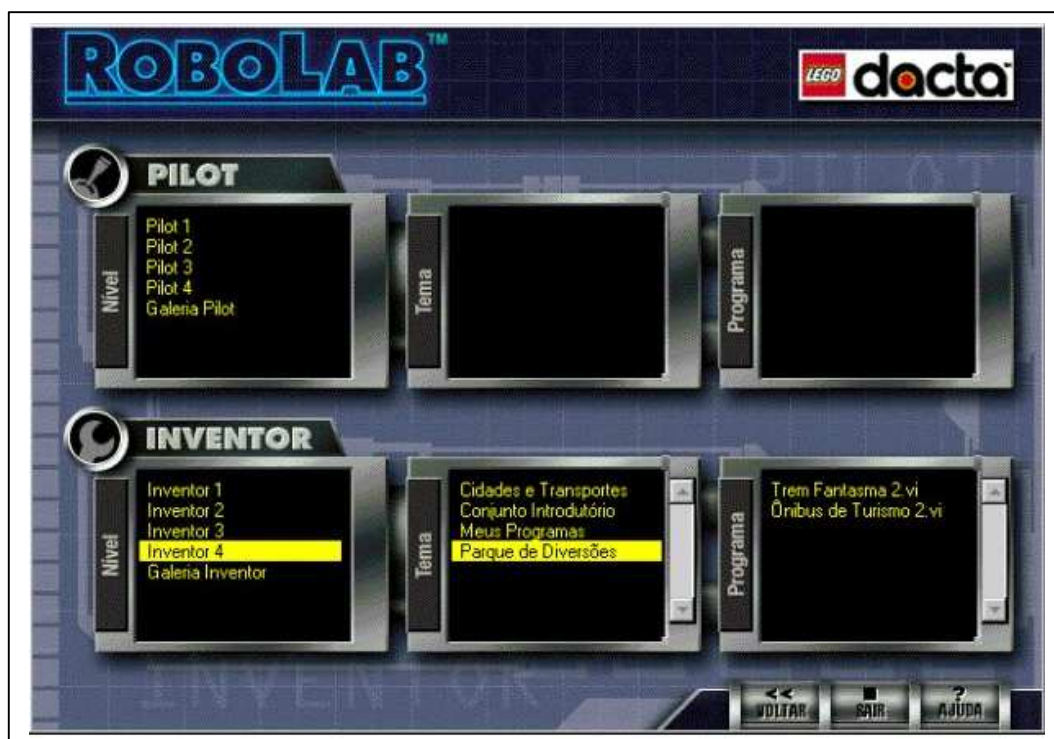
Ao selecionar-se a seção Ajustes do RCX o Robolab automaticamente estabelece a comunicação com o RCX. Caso esteja desligado, ou fora do alcance do sinal emitido pela torre de infravermelho, aparece a mensagem de erro de comunicação. Se a comunicação for possível, o Robolab automaticamente configura os dados presentes nesta tela. Ao sair desta tela, o Robolab ajusta o RCX para a nova configuração.

É possível alterar as seguintes configurações do RCX:

- Ajuste de potência do Infravermelho: o que altera a distância máxima possível para comunicação entre o RCX e o computador.
- Opções RCX Programs 1 e 2 permite bloquear as posições de memória 1 e 2 para evitar que programas sejam descarregados nestas posições.
- Tempo de Desligamento RCX define o tempo que o RCX ficará ligado quando estiver inativo. A variação é de **0** a **255** minutos. Por definição, o valor **0** significa que o RCX ficará sempre ligado, neste caso ele só se desligará ao ser pressionado o botão On-Off do RCX.
- Carga das Pilhas RCX: mostra o nível atual da bateria do RCX.
- Versão Firmware RCX: exibe versão do firmware em uso no RCX.



O software de programação do RCX no Robolab apresenta duas opções chamadas de **Pilot** e **Inventor** cada uma com 4 níveis.



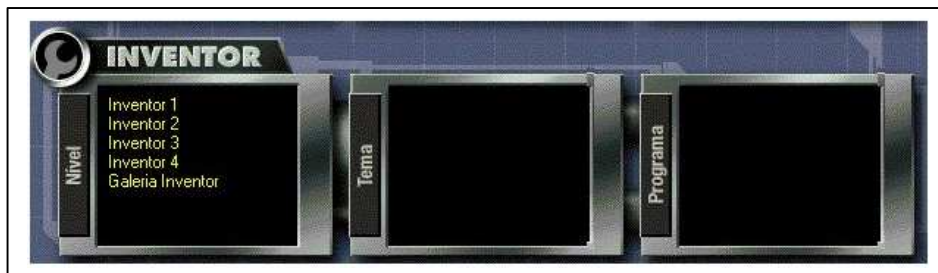
Cada seção apresenta 3 janelas: Nível, Tema e Programa. Em Nível pode-se selecionar o nível de trabalho. Um clique em Nível faz a janela Tema mostrar os temas disponíveis. Um clique num tema faz a janela Programa mostrar o conjunto de programas disponíveis para o tema e nível selecionados.

A programação Pilot é a fase introdutória e apresenta **4** níveis, sendo o Pilot **1** o mais simples e o Pilot **4** o mais flexível.



Inventor

A programação Inventor é uma seqüência da fase Pilot e também apresenta 4 níveis, sendo o Inventor **1** o mais simples e o Inventor **4** o que apresenta total flexibilidade e possibilidades ilimitadas.

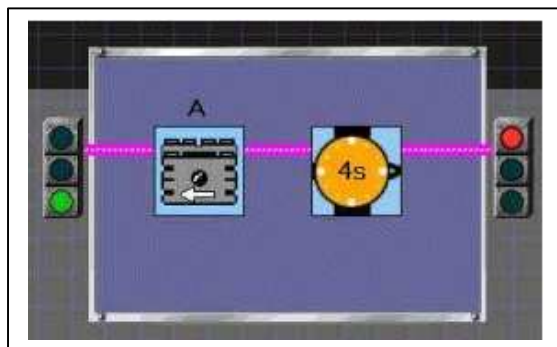


A fase Pilot compreende uma série de modelos com um formato fixo associado a eles. Trata-se de um modo eficiente de apresentar a seqüência lógica, pois é impossível modificar qualquer dos modelos de modo a criar um programa que falhe. O programa pode não fazer o que se espera, mas funcionará sempre e realizará exatamente a seqüência de comandos listados.

A programação Pilot usa uma interface fácil, do tipo Clique & Escolha, num modelo que pode ser alterado conforme suas necessidades de aprendizado.

Abaixo segue uma descrição do programa padrão (modelo) encontrado em cada um dos níveis.

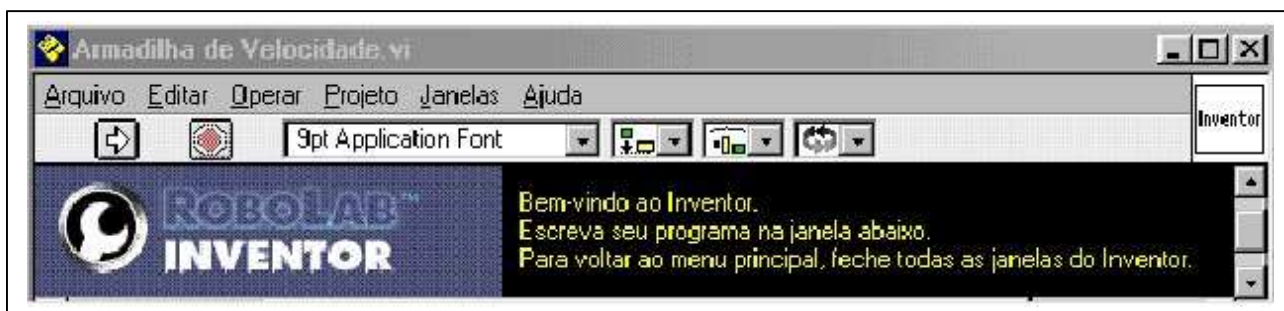
O programa Pilot Nível 1 (Figura) é uma tarefa simples que fornece potência máxima à Porta de Saída A do RCX durante um determinado tempo.



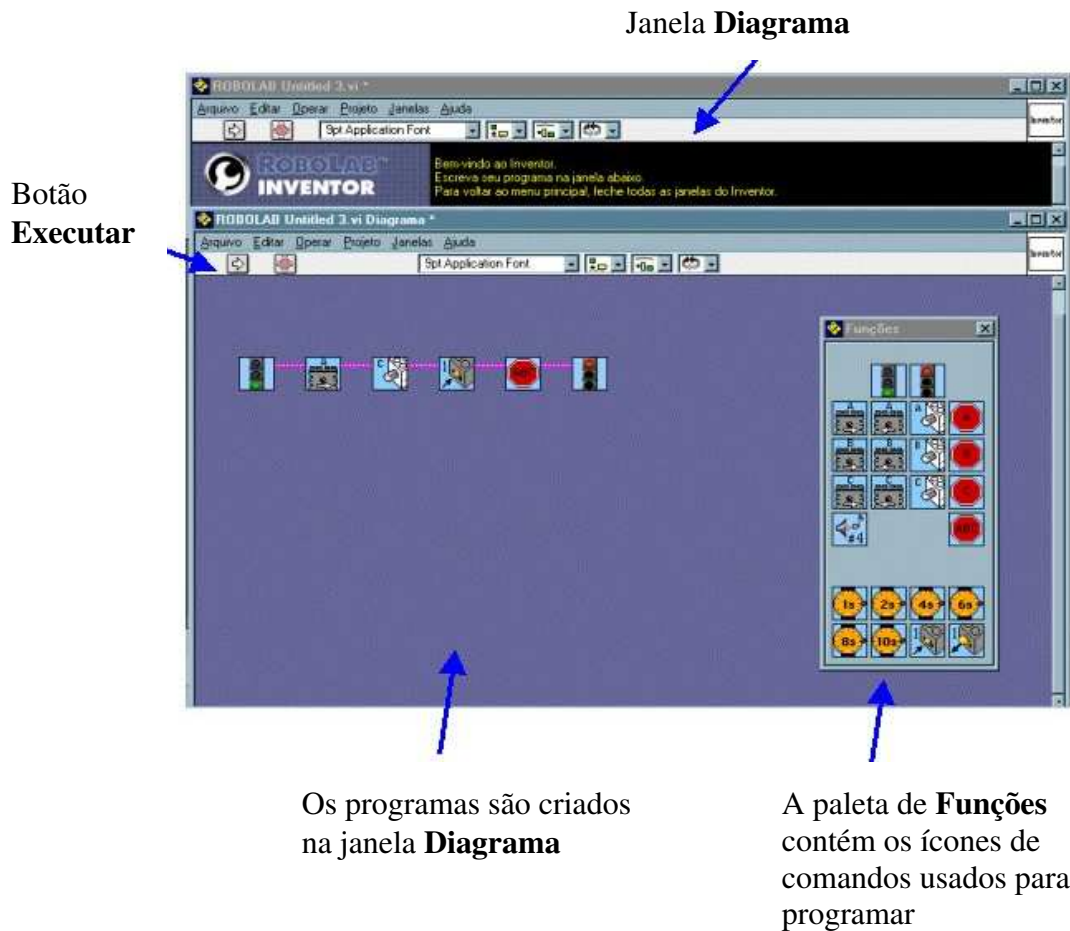
Os outros níveis do programa Pilot progressivamente incrementam a utilização dos vários recursos do Robolab e RCX com a inclusão de sensores, vários motores e a possibilidade de executar o programa em seqüências de passos.

A fase Inventor usa os mesmos ícones de comandos da fase Pilot. Além disso, são acrescentadas várias opções de comandos, conforme o usuário avança pelos níveis. A interface de programação do Inventor é diferente da utilizada no Pilot. A interface é do tipo Pegar & Colocar.

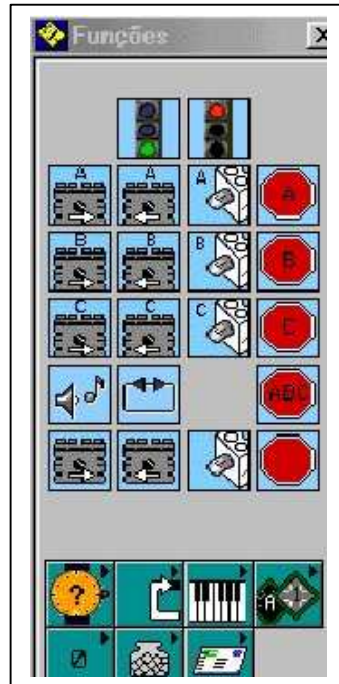
Nesta interface pode-se trabalhar com pelo menos 3 janelas. A janela Painel é usada na programação do Inventor. Ela precisa estar aberta para que o programa possa ser executado.



A janela utilizada para programar é chamada de Diagrama e é nela que se cria as tarefas a serem executadas pelo RCX.



A outra janela é chamada a Paleta de funções (Figura), a qual apresenta todos os comandos ou funções disponíveis para a montagem do programa. Esta paleta também contém os comandos de Espera, os de Estrutura, os Modificadores e outros mais.



A seguir são apresentados os principais elementos do Robolab, utilizados na construção de programas para o RCX.

Início & Fim do Programa



Início

No início do programa, exigido como o primeiro comando em qualquer programa Inventor.



Fim

No fim do programa, exigido como o último comando de cada tarefa no programa Inventor.



Parar A

Interrompe a potência da Porta **A** do RCX.



Parar tudo

Interrompe a potência das portas **A**, **B** e **C** do RCX.

Saídas Simples



Motor **A** (sentido horário)

Liga a Porta **A** do RCX no sentido horário com potência total.



Motor **A** (sentido anti-horário)

Liga a Porta **A** do RCX no sentido anti-horário com potência total.



Lâmpada **A**

Acende a Lâmpada conectada na Porta **A** do RCX com potência total.

Saídas Gerais



Lâmpada

Acende a lâmpada.

Padrão: Potência total nas Portas **A**, **B** e **C**.



Motor (sentido horário)

Liga o motor no sentido horário.

Padrão: Potência total nas Portas **A**, **B** e **C**.



Motor (sentido anti-horário)

Liga o motor no sentido anti-horário.

Padrão: Potência total nas Portas **A**, **B** e **C**.



Direcionar

Direciona a potência para as portas especificadas, invertendo a direção anterior.

Padrão: Portas **A**, **B** e **C**.



Tocar Som

Toca um som no RCX. Os sons disponíveis são:

1-Clique de chave

4-Silvo ascendente (Padrão)

2-BeepBeep

5-Bzzzz

3-Silvo descendente

6-Silvo ascendente rápido

Esperar por? (Wait For)



Espera 1 segundo

Espera 1 segundo antes de continuar o programa.



Espera por tempo

Espera por um tempo especificado.

Padrão: 1 segundo.



Espera tempo aleatório

Espera por um tempo aleatório. Tempo entre 1 e 8 segundos.



Espera empurrar

Espera até que o sensor de toque seja pressionado.

Padrão: Porta de Entrada 1.



Espera não pressionado

Espera até que o sensor de toque não seja pressionado.

Padrão: Porta 1.



Espera por luz

Espera até que o sensor de luz leia um valor mais brilhante que o número especificado.

Padrão = 55.



Espera por escuro

Espera até que o sensor de luz leia um valor mais escuro que o número especificado.

Padrão = 55.



Mais claro

Espera até que o sensor de luz leia um valor maior que o atual.

Padrão: Porta 1, diferença de valor = 5.



Mais escuro

Espera até que o sensor de luz leia um valor menor que o atual.

Padrão: Porta 1, diferença de valor = 5.

Modificadores



Entrada 1

Anexe este modificador a um comando para selecionar a Porta de Entrada 1.



Saída A

Anexe este modificador a um comando para selecionar a Porta de Saída A.



Potência Nível 3

Anexe este modificador a um motor ou lâmpada para ajustar a potência no nível 3.



Constante Numérica

Anexe este modificador a um sensor ou temporizador para ajustar o valor da constante.



Valor do *Container* vermelho

Retorna o valor do *container* vermelho



Container vermelho

Anexe-o a um comando *container*, para selecionar o *container* vermelho.



Número ?

Um número aleatório entre 1 e 8.



Valor da Porta 1

Retorna o valor da Porta de Entrada 1.



Temporizador vermelho

Anexe-o a um comando de temporizador para selecionar o temporizador vermelho.



Valor do Temporizador

Retorna o valor do temporizador vermelho.

vermelho



Valor da Carta

Retorna o valor da carta (variável).

Música



Nota musical

Toca notas musicais no RCX.

Padrão: semínimas na escala normal.



Pausa musical

Insere uma pausa na música.



Duração musical

Especifica quanto tempo uma nota deve tocar.



Uma oitava acima

Anexado a um comando musical, eleva o tom em uma oitava ou mais se mais de um for anexado.

Estruturas



Condição do Sensor
de Toque

Faz o programa escolher entre dois caminhos, dependendo dos dados do sensor de toque.

Padrão: Porta 1.



Junção

Usada sempre com uma condição para unir os dois caminhos da condição.



Começar Tarefa

Começa uma nova tarefa. Permite realizar tarefas múltiplas simultaneamente. O RCX permite um máximo de 10 tarefas.



Começar *Loop*

Começa um estrutura de "*loop*".

Padrão: 1 vez.



Fim do *Loop*

Pula de volta ao início do "*loop*".



Jump (pulo)

Faz o programa pular para um lugar específico da sequência.



Land (pouso)

Indica o lugar da sequência onde o comando *Jump* vai pular.

Apêndice 3 - Funções SuperLogo para controle do RCX

Funções SuperLogo para comandar o RCX

Controle de Motores

rcxligamotor <lista de motores>

rcxdesligamotor <lista de motores>

rcxligapor <lista de motores> <tempo>

rcxmotorfrente <lista de motores>

rcxrevertemotor <lista de motores>

Controle de Sensores

rcxsetatiposensor <lista de sensores > <tipo>

rcxsetamodosensor <lista de sensores > <modo>

rcxlersensor < sensor>

Funções simplificadas para sensores

rcxsetasensorluz <lista de sensores>

rcxsetasensortoque <lista de sensores>

rcxlersensorluz_<lista de sensores>

rcxlersensortoque <lista de sensores>

Som

rcxsom <frequência> < tempo>

rcxligamotor <lista de motores>

lg <lista de motores>

Liga motores conectados às saídas do RCX .

Parâmetros:

lista de motores : lista quais os motores que devem ser ligados.

Exemplo:

Os motores conectados às saídas A e B são ligados

rcxligamotor [A B]

rcxdesligamotor <lista de motores>

dlg <lista de motores>

Desliga motores conectados às saídas do RCX .

Parâmetros:

lista de motores : lista quais os motores que devem ser desligados.

Exemplo:

Os motores conectados às saídas A e B são desligados

rcxdesligamotor [A B]

rcxligapor <lista de motores> <tempo>

lgpor <lista de motores>

Liga motores conectados às saídas do RCX por um período determinado.

Parâmetros:

lista de motores : lista quais os motores que devem ser ligados.

tempo : tempo em que os motores permanecerão ligados . Cada unidade é equivalente a 1/60 segundos.

Exemplo:

Os motores conectados as saídas A e B são ligados por 1 segundo

rcxligapor [A B] 60

rcxmotorfrente <lista de motores>

mf <lista de motores>

Determina que a rotação dos motores conectados às saídas do RCX ocorra no sentido horário.

Parâmetros:

lista de motores : lista quais os motores que devem ter o sentido determinado.

Exemplo:

Os motores conectados as saídas A e B tem o sentido de rotação determinado como horário.

rcxmotorfrente [A B]

rcxrevertemotor <lista de motores>

rm <lista de motores>

Determina que a rotação dos motores conectados às saídas do RCX ocorra no sentido anti-horário.

Parâmetros:

lista de motores : lista quais os motores que devem ter o sentido determinado.

Exemplo:

Os motores conectados às saídas A e B tem o sentido de rotação determinado como anti-horário.

rcxrevertemotor [A B]

rcxsetatiposensor <lista de sensores > <tipo>

Determina o tipo dos sensores conectados às saídas do RCX .

Parâmetros:

lista de sensores : lista quais os sensores .

tipo do sensor:

0: Nenhum sensor

1: Chave

2: Temperatura

3: Reflexão (sensor de luz)

4: Ângulo

Exemplo:

Define o sensor ligado às entradas 1 e 2 como chave.

rcxsetatiposensor [1 2] 1

rcxsetamodosensor <lista de sensores > <modo>

Determina o modo como os sinais fornecidos pelos sensores serão representados .

Parâmetros:

lista de sensores : lista com os sensores .

modo do sensor:

0: Analógico

1: Booleano (verdadeiro / falso)

2: Contador de transições (positivas e negativas)

3: Contador de período

4: Percentagem

5: Celsius

6: Fahrenheit

7: Ângulo

Exemplo:

Define o sensor ligado às entradas 1 e 2 como analógico .

rcxsetamodosensor [1 2] 0

rcxlersensor < sensor>

Is <sensor>

retorna o valor do sensor conectado a entrada do RCX indicada em <sensor> .

Parâmetros:

sensor : lista com os sensor .

Exemplo:

Lê o valor do sensor ligado na entrada 1 .

rcxlersensor 1

rcxsetasensorluz <lista de sensores>

ssl <lista de sensores>

Define os sensores definidos em lista de sensores como sendo sensores de luz.

Parâmetros:

lista de sensores : lista com os sensores .

Exemplo:

Define o sensor ligado em 1 e 3.

rcxsetasensorluz [1 3]

rcxsetasensortoque <lista de sensores>

sst <lista de sensores>

Define os sensores definidos em lista de sensores como sendo sensores de toque.

Parâmetros:

lista de sensores : lista com os sensores .

Exemplo:

Define o sensor ligado em 1 e 3.

rcxsetasensortoque [1 3]

rcxlersensorluz <sensor>

lsl <lista de sensores>

Retorna o valor do sensor indicado na lista sensor

Parâmetros:

sensor : lista com o sensor .

Exemplo1 :

mo rcxlersensorluz 1

Exemplo 2 :

Com comandos condicionais (se , enquanto) utiliza o valor retornado em valorsensorluz
se :valorsensor > 100 [mo [maior que 100]] [mo [menor que 100]]

rcxlersensortoque <lista de sensores>

lst <lista de sensores>

Retorna o valor do sensor indicado na lista sensor.

Parâmetros:

sensor : lista com o sensor .

Exemplo:

rcxlersensorluz 1

rcxsom <frequência> < tempo>

tsom <frequência> < tempo>

Toca um tom no RCX.

Parâmetros:

frequência : frequência do tom emitido pode variar de 1 a 20000 Hz

tempo : determina a duração do tom emitido em passos de 10 ms , pode variar de 1 a 255

Exemplo:

Emite um tom com a frequência de 2000 Hz durante 1 segundo

rcxsom 2000 100

Apêndice 4 – Exemplo de utilização do SuperLogo com RCX

Este apêndice contém um conjunto de exemplos da utilização do RCX com o SuperLogo, que foi preparado para utilização com alunos em oficinas de Robótica.

Utilizando as funções Superlogo para controle do Lego RCX

Objetivo:

Acionar e desligar um motor conectado ao RCX.

Funções utilizadas:

rcxligamotor , rcxdesligamotor

Material necessário:

RCX e um motor.

Executando:

1. Conecte à saída **C** do RCX um motor.
A figura abaixo mostra como deve ficar a montagem.



2. Execute na linha de comando do Superlogo :
rcxligamotor "c ;liga motor conectado em C
espere 120 ;aguarda 2 segundos
rcxdesligamotor "c ;desliga motor conectado em C

Utilizando as funções SuperLogo para controle do Lego RCX

Objetivo:

Acionar por um determinado tempo e desligar um motor conectado ao RCX.

Funções utilizadas:

rcxligapor

Material necessário:

RCX e um motor.

Executando:

1. Conecte à saída **C** do RCX um motor.
A figura abaixo mostra como deve ficar a montagem.



2. Execute na linha de comando do Superlogo:
rcxligapor "c 120 ;liga o motor conectado a saída C por 2 segundos

Utilizando as funções SuperLogo para controle do Lego RCX

Objetivo:

Acionar, selecionar o sentido de rotação e desligar um motor conectado ao RCX.

Funções utilizadas:

rcxligamotor , rcxdesligamotor , rcxmotorfrente , rcxrevertemotor

Material necessário:

RCX e um motor.

Executando:

1. Conecte à saída **C** do RCX um motor.
A figura abaixo mostra como deve ficar a montagem.



2. Execute na linha de comando do Superlogo :

```
rcxligamotor "c ;liga o motor conectado em C
espere 120 ;aguarda 2 segundos
rcxrevertemotor "c ;muda o sentido de rotação do motor
espere 120 ;aguarda 2 segundos
rcxmotorfrente "c ;altera o sentido de rotação do motor
espere 120 ;aguarda 2 segundos
rcxdesligamotor "c ;desliga motor conectado em C
```

Utilizando as funções Superlogo para controle do Lego RCX

Objetivo:

Selecionar e ler o estado de um sensor de toque conectado ao RCX

Funções utilizadas:

rcxsetatiposensor , rcxsetamodosensor , rcxlensensor

Material necessário:

RCX e um sensor de toque.

Executando:

1. Conecte à saída **3** do RCX o sensor.
A figura abaixo mostra como deve ficar a montagem.



2. Execute na linha de comando do Superlogo :

rcxsetatiposensor 3 1 ;seta o sensor conectado em 3 como toque (chave)
rcxsetamodosensor 3 1 0 ;seta o modo do sensor como booleano (1 ou 0)
mostre rcxlensensor 13 3 ;mostra o valor do sensor no modo booleano

Utilizando as funções Superlogo para controle do Lego RCX

Objetivo:

Selecionar e ler o estado de um sensor de luz conectado ao RCX

Funções utilizadas:

rcxsetatiposensor, rcxsetamodosensor, rcxlersensor

Material necessário:

RCX e um sensor de luz.

Executando:

1. Conecte à saída **1** do RCX o sensor de luz.
A figura abaixo mostra como deve ficar a montagem.



2. Execute na linha de comando do Superlogo :

rcxsetatiposensor 1 3 ;seta o sensor conectado em 1 como luz
rcxsetamodosensor 1 0 31 ;seta o modo do sensor como analógico
mostre rcxlersensor 12 1 ;mostra o valor do sensor no modo analógico

Utilizando as funções Superlogo para controle do Lego RCX

Objetivo:

Selecionar e ler o estado de um sensor de luz conectado ao RCX utilizando as funções simplificadas para sensores

Funções utilizadas:

rcxsetasensorluz, rcxlersensor

Material necessário:

RCX e um sensor de luz.

Executando:

1. Conecte à saída **1** do RCX o sensor de luz.
A figura abaixo mostra como deve ficar a montagem.



2. Execute na linha de comando do Superlogo :

rcxsetasensorluz 1 ;seta o sensor conectado na entrada 1 como luz
mostre rcxlersensor 1 ;mostra o valor do sensor conectado a entrada 1

Utilizando as funções Superlogo para controle do Lego RCX

Objetivo:

Selecionar e ler o estado de um sensor de toque conectado ao RCX utilizando funções simplificadas.

Funções utilizadas:

rcxsetasensotoque , rcxlersensor

Material necessário:

RCX e um sensor de luz.

Executando:

1. Conecte à saída **3** do RCX o sensor.
A figura abaixo mostra como deve ficar a montagem.



2. Execute na linha de comando do Superlogo :
rcxsetasensortoque [3] ;seta o sensor conectado em 3 como toque (chave)
mostre rcxlersensor 3 ;mostra o valor do sensor

Utilizando as funções SuperLogo para controle do Lego RCX

Objetivo:

Desligar o motor se o sensor de toque for pressionado , caso contrário ligar o motor.

Funções utilizadas:

rcxsetasensortoque , rcxlensensortoque , rcxligamotor , rcxdesligamotor

Material necessário:

RCX , um sensor de toque , motor elétrico.

Executando:

1. Conectar à entrada **3** do RCX o sensor de toque
2. Conectar à saída **C** do RCX o motor.
A figura abaixo mostra como deve ficar a montagem.



3. Crie os procedimentos abaixo:

```
aprenda demo1
```

```
rcxsetasensortoque "3
```

```
rcxligamotor "c
```

```
controle1
```

```
fim
```

```
aprenda controle1
```

```
se rcxlensensortoque 3 [rcxdesligamotor "c] [rcxligamotor "c]
```

```
controle1
```

```
fim
```

Utilizando as funções SuperLogo para controle do Lego RCX

Objetivo:

Desligar e ligar o motor quando houver variação de luz sobre o sensor.

Funções utilizadas:

rcxsetasensorluz , rcxlersensorluz , rcxligamotor , rcxdesligamotor.

Material necessário:

RCX , um sensor de luz , um motor elétrico.

Executando:

1. Conectar à entrada **1** do RCX o sensor de luz
2. Conectar à saída **C** do RCX o motor.
A figura abaixo mostra como deve ficar a montagem.



3. Crie os procedimentos abaixo:

```
aprenda demo2
```

```
rcxsetasensorluz "1
```

```
rcxligamotor "c
```

```
controle2
```

```
fim
```

```
aprenda controle2
```

```
rcxlersensorluz "1
```

```
se :valorsensorluz < 700 [rcxdesligamotor "c ] [ rcxligamotor "c ]
```

```
controle2
```

```
fim
```


Apêndice 5 – Projetos Lego RCX com SuperLogo

A seguir é apresentado o relatório desenvolvido pelo aluno do curso de Mecatrônica, Guilherme Takiguthi, onde são descritos os experimentos desenvolvidos com o ARE.

Projeto: Carro de Combate

Introdução

Um dispositivo robótico, que consegue desviar-se de um obstáculo, pode ser uma tarefa interessante quando se pretende construir um robô capaz entrar e sair de um labirinto. Um dos modelos deste robô pode ter o formato de um Carro de Combate (Figura).



Objetivo

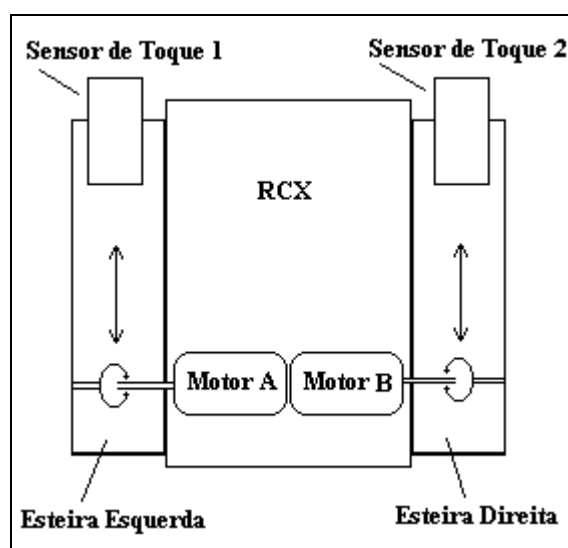
Construção de um veículo capaz de desviar de obstáculos utilizando sensores de toque.

Montagem:

Para a montagem deste dispositivo, além do RCX propriamente dito, foram utilizados os seguintes componentes elétricos:

1. 2 sensores de toque
2. 2 motores

Esquema de montagem do Carro de Combate:



A conexão dos componentes elétricos as entradas e saídas do RCX é a seguinte:

Entradas

	Entrada do RCX	Função
Sensor de Toque 1	Entrada 1	Detectar o contato com algum obstáculo no lado esquerdo do Carro de Combate
Sensor de Toque 2	Entrada 3	Detectar o contato com algum obstáculo no lado direito do Carro de Combate

Saídas

	Saída do RCX	Função
Motor A	Saída A	Movimentar a esteira esquerda
Motor C	Saída C	Movimentar a esteira direita

Funcionamento

O Carro de Combate possui dois motores que, individualmente, movimentam cada uma das esteiras relativas ao lado esquerdo e direito do veículo.

O Carro de Combate inicia seu movimento para a frente, ligando os dois motores no sentido horário. Ao encontrar um obstáculo (verificado através do contato do sensor de toque com o objeto), como uma parede, o veículo recua, invertendo o sentido dos motores, e, posteriormente, gira em torno do próprio eixo no sentido oposto ao lado em que se encontra o sensor pressionado. Este movimento de rotação ocorre devido ao fato dos motores se movimentarem em sentidos opostos.

Automação e Controle do dispositivo

Um possível programa para controle deste dispositivo foi escrito em linguagem SuperLogo. O código utilizado encontra-se abaixo. Os comandos referentes ao controle do RCX podem ser encontrados em www.nied.unicamp.br/~siros/

Procedimento em SuperLogo

```
aprenda carro
  rcxsetasensortoque [1 3]
  rcxligamotor "ac
  atr "lcarro 0
  façaenquanto [
    atr "s0 rcxlersensor 1
    atr "s1 rcxlersensor 3
    mostre :s0
    se e :s0=0 :s1=0[
      rcxmotorfrente "ac
    ]
    se e :s0=1 :s1=1[
      rcxrevertemotor "ac
      espere 30
      rcxmotorfrente "a
    ]
    se e :s0=1 :s1=0 [
      rcxrevertemotor "ac
      espere 30
      rcxmotorfrente "a
    ]
    se e :s0=0 :s1=1 [
      rcxrevertemotor "ac
      espere 30
      rcxmotorfrente "c
    ]
  ]
  [:lcarro=0]
fim
```

Projeto: Leitor de Cores

Introdução

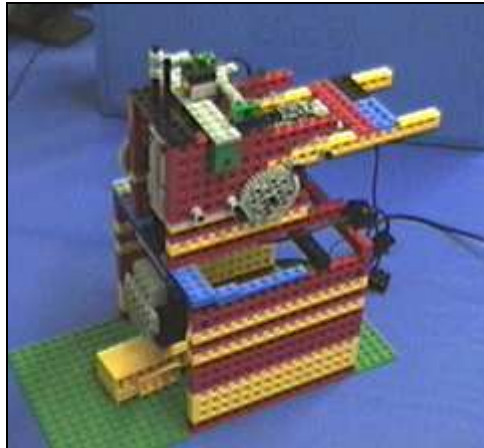
Em determinadas situações é importante que um dispositivo robótico diferencie uma cor da outra para, a partir desta diferenciação, executar uma tarefa especificada como, por exemplo, parar uma esteira; acionar um alarme; desligar uma luz etc..

Objetivo

Utilizar o sensor de luz para diferenciar/reconhecer a cor de alguns tijolos LEGO.

Descrição

O leitor de cor (Figura) é um dispositivo robótico cujo sistema mecânico é constituído por componentes Lego™ (jogos de engrenagens, rosca sem-fim, motor, sensor de toque, sensor de luz e tijolos comuns).



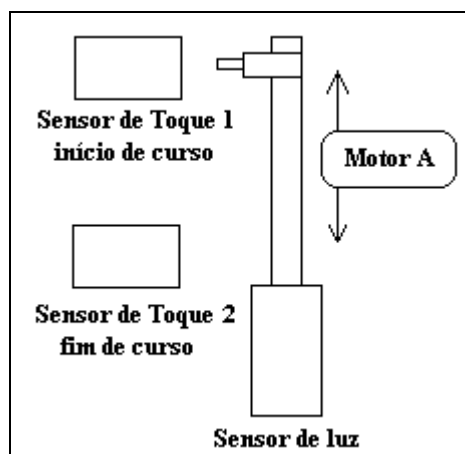
Figura

Montagem:

Para a montagem deste dispositivo, além do RCX propriamente dito, foram utilizados os seguintes componentes elétricos:

1. 2 sensores de toque
2. 1 sensor de luz
3. 1 motor

O esquema abaixo mostra a configuração do sistema:



A conexão dos componentes elétricos as entradas e saídas do RCX é a seguinte:

Entradas

	Entrada do RCX	Função
Sensor de Toque 1	Entrada 1	Determinar a posição inicial do Leitor
Sensor de Toque 2	Entrada 2	Determinar a posição final do Leitor
Sensor de Luz	Entrada 3	Reconhecer a variação de luminosidade refletida pelas peças

Saídas

	Saída do RCX	Função
Motor A	Saída A	Movimentar o Leitor verticalmente

Funcionamento

O tijolo LEGO cuja cor será reconhecida é posicionado abaixo do leitor. Por intermédio do acionamento de um motor, o leitor é abaixado até que este fique aproximadamente 0,5 mm do tijolo. Em seguida, é iniciado o processo de reconhecimento da cor, que se estende por aproximadamente 10 segundos. A cor lida é escrita na tela do computador. Após o reconhecimento da cor, o motor do sistema mecânico de movimentação do leitor é acionado no sentido contrário, ao primeiro acionamento, fazendo com que o leitor se afaste da peça e se posicione para a próxima leitura.

Automação e Controle do Dispositivo

Um possível programa para controle deste dispositivo foi escrito em linguagem SuperLogo. O código utilizado encontra-se abaixo. Os comandos referentes ao controle do RCX podem ser encontrados em www.nied.unicamp.br/~siros/

Fases do Programa:

Fase de inicialização

Inicialmente verifica-se o estado do Sensor 1 (início de curso do leitor) . Se o Sensor 1 estiver pressionado significa que o leitor se encontra pronto para iniciar o reconhecimento,

caso contrário, o motor será acionado no sentido anti-horário até que o Sensor 1 seja pressionado.

Fase de reconhecimento

Concluído a primeira fase, inicializa-se a fase de reconhecimento. O motor é acionado no sentido horário até que o Sensor 2 (fim de curso do leitor) seja pressionado, significando que o leitor de cor se encontra próximo da peça. Isso favorece um grau de precisão maior na leitura. Durante alguns segundos o Sensor de luz lê a intensidade de luz refletida pela peça. Em uma variável armazena-se o menor valor. Através deste menor valor é possível reconhecer as cores. São quatro as possibilidades: amarela, verde, preta e cinza. Terminada esta fase, retorna-se à primeira fase.

Procedimento em SuperLogo

```
aprenda escgrande :vpal
```

```
tat
```

```
pd 90
```

```
dt
```

```
mudefontejanelagráfica [[arial] 72 0 0 600 0 0 0 0 3 2 1 18]
```

```
rotule :vpal
```

```
at
```

```
fim
```

```
aprenda iniciar
```

```
  rcxsetasensortoque [1 2]
```

```
  rcxsetasensorluz [3]
```

```
  atr "s1 rcxlersensor [1]
```

```
  Se (:s1=0)[
```

```
    rcxrevertemotor [a]
```

```
    rcxligamotor [a]
```

```
    enquanto [:s1=0][
```

```
      atr "s1 rcxlersensor 1
```

```
    ]
```

```
    rcxdesligamotor [a]
```

```
  ]
```

```
  mostre [Inicializado]
```

```
fim
```

```
aprenda lecor
```

```
  atr "s1 rcxlersensor 1
```

```
  Se :s1=0 [
```

```
    iniciar
```

```
    mostre [Inicializando...]
```

```
  ]
```

```
  Se :s1=1 [
```

```
    mostre [Inicializando a Leitura de Cor...]
```

```

rcxmotorfrente [a]
rcxligamotor [a]
atr "s2 rcxlersensor 2
Enquanto [:s2=0][
  atr "s2 rcxlersensor 2
]
rcxdesligamotor [a]
atr "cont 0
atr "min 2000
Enquanto [:cont<30][
  atr "cont :cont+1
  atr "sluz rcxlersensor 3
  Se :sluz<:min [atr "min :sluz]
]
Se e (:min>750)(:min<1000)[escgrande [Preto]]
Se e (:min>700)(:min<750)[escgrande [Verde]]
Se e (:min>670)(:min<700)[escgrande [Cinza]]
Se e (:min>630)(:min<670)[escgrande [Amarelo]]
]
iniciar
fim

```

Projeto Porta Automática

Introdução

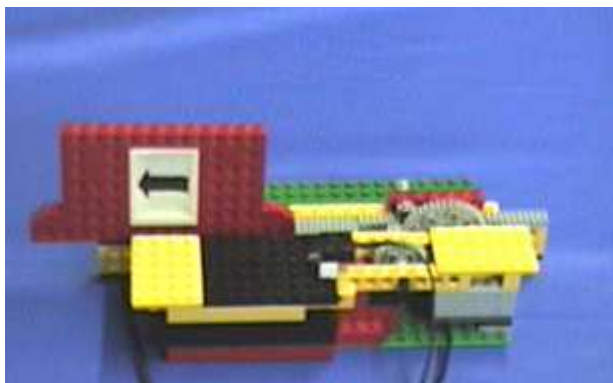
Portas automáticas estão presentes nos diferentes espaços que frequentamos no dia-a-dia tais como aeroportos, shopping centers. Construir um dispositivo que simula o princípio de funcionamento deste tipo de porta pode ser uma atividade de robótica que auxilia a compreensão da tecnologia empregada na sua implementação.

Objetivo

Construir uma porta que detecta a presença de pessoas/objetos por meio de um sensor de luz acionando um motor para abertura e fechamento da mesma.

Descrição

A porta automática (Figura) é um dispositivo robótico cujo sistema mecânico é constituído basicamente de jogos de engrenagens, rosca sem-fim, motor, sensores de toque, sensor de luz, lâmpada e tijolos comuns.



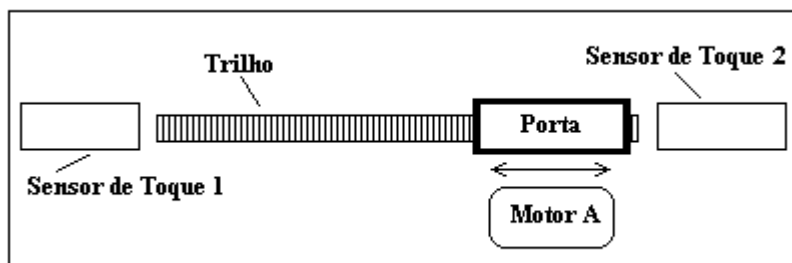
Uma vez detectada a presença de pessoas/objetos a mesma se abre e após um tempo determinado se fecha. Durante o fechamento caso o sensor seja ativado por pessoas /objetos a porta deve interromper o fechamento e abrir .

Montagem:

Para a montagem deste dispositivo, além do RCX propriamente dito, foram utilizados os seguintes componentes elétricos:

1. 2 sensores de toque
2. 1 sensor de luz
3. 1 motor
4. 1 lâmpada

O esquema abaixo mostra a configuração da montagem da porta:



A conexão dos componentes elétricos as entradas e saídas do RCX é a seguinte:

Entradas

	Entrada do RCX	Função
Sensor de Toque 1	Entrada 1	Determinar a posição inicial da Porta (porta fechada)
Sensor de Luz	Entrada 2	Detectar a presença de pessoas
Sensor de Toque 2	Entrada 3	Determinar a posição final da Porta (porta aberta)

Saídas

	Saída do RCX	Função
Motor A	Saída A	Movimentar a porta, abrindo-a ou fechando-a
Lâmpada C	Saída C	Iluminar o Sensor de Luz

Funcionamento

Inicialmente a lâmpada é ligada. Caso o sensor de luz detecte uma luminosidade alta, significa que não há pessoas obstruindo a passagem de luz. A porta automática, ao detectar a presença de uma "pessoa" (ou seja, sensor de luz detectar baixa luminosidade – menor do que 750) será aberta acionando o Motor A no sentido horário até que o sensor de fim de curso (sensor de toque 2) seja pressionado. Ao detectar luminosidade alta – maior do que 750 – a porta se fecha, acionando o Motor A no sentido anti-horário até que o sensor de fim de curso (sensor de toque 1) seja pressionado.

A porta permanece aberta por alguns segundos, fechando-se posteriormente e preparando-se para a próxima detecção.

Procedimento em SuperLogo

aprenda porta

rcxsetasensortoque [1 3]

rcxsetasensorluz [2]

rcxligamotor [c]

atr "s1 rcxlersensor [1]

atr "s2 rcxlersensor [2]

atr "s3 rcxlersensor [3]

testeluz

fim

aprenda testeluz

Senão :s2>750 [

Enquanto [:s1=0][

rcxrevertomotor [a]

rcxligamotor [a]

atr "s1 rcxlersensor [1]

]

][Enquanto [:s3=0][

rcxmotorfrente [a]

```
    rcxligamotor [a]
    atr "s3 rcxlersensor [3]
  ]
]
rcxdesligamotor [a]
testeluz
fim
```

Traçador Gráfico

Introdução

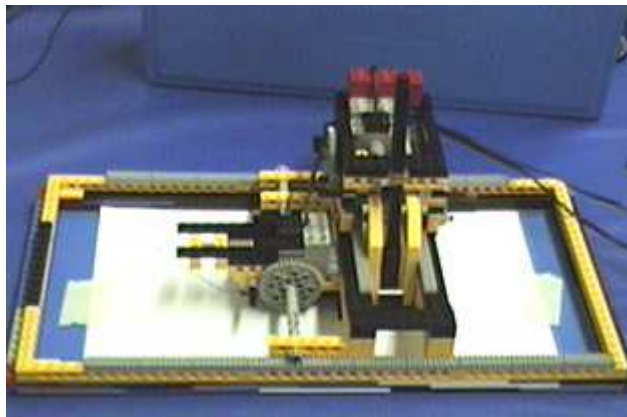
Para determinadas situações pode ser interessante utilizar um outro meio para reproduzir o desenho feito na tela do computador utilizando um dispositivo robótico. Um traçador X Y pode ser usado com esta finalidade.

Objetivo

Criar um traçador gráfico que reproduz no papel os traços produzidos na tela pela Tartaruga do SuperLogo.

Descrição

O Traçador X Y é um dispositivo robótico (Figura) cujo sistema mecânico é constituído por componentes LEGO™ (jogos de engrenagens, rosca sem-fim, cremalheiras, motores e tijolos comuns). Além disso, uma caneta é utilizada para a impressão.

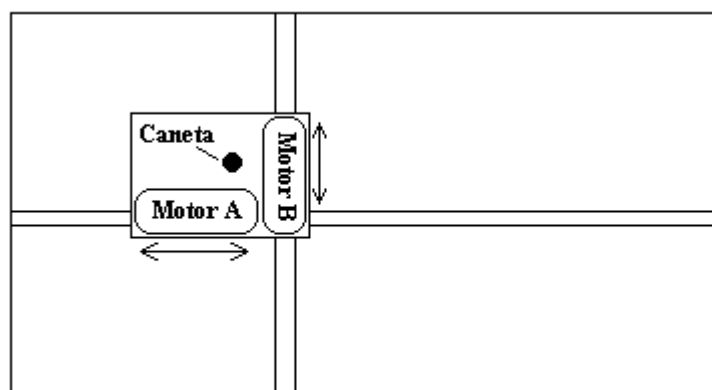


Montagem:

Para a montagem deste dispositivo, além do RCX propriamente dito, foram utilizados os seguintes componentes elétricos:

1. 2 motores

Esquema da montagem



A conexão dos componentes elétricos as entradas e saídas do RCX é a seguinte:

Entradas

Não foram utilizados sensores

Saídas

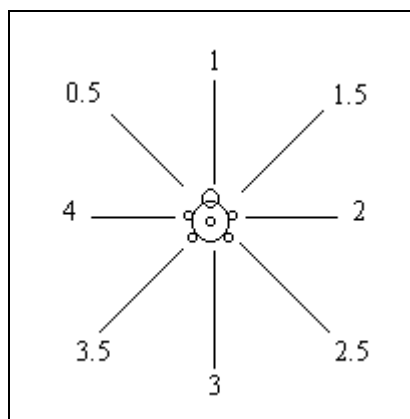
	Saída do RCX	Função
Motor A	Saída A	Movimentar a caneta no sentido horizontal no plano do papel

Motor B	Saída B	Movimentar a caneta no sentido vertical no plano do papel
---------	---------	---

Funcionamento

O traçador gráfico, através do comando da Tartaruga no SuperLogo faz a impressão, no papel, da trajetória realizada por ela. Foi criada uma variável denominada *direção* que relaciona a cada direção e sentido da Tartaruga um número. Como foram utilizados motores LEGO, que são de corrente contínua (cc), foi possível apenas obter trajetórias retas e perpendiculares entre si.. Com este dispositivo é possível também obter trajetórias com ângulo de 45°, pois a velocidade de deslocamento horizontal e o vertical são praticamente iguais.

A relação entre a direção e sentido da Tartaruga pode ser interpretada conforme a figura a seguir.



Controle do dispositivo

Um possível programa para controle deste dispositivo foi escrito em SuperLogo. Os procedimentos utilizados serão apresentados a seguir. Os comandos referentes ao controle do RCX podem ser encontrados em www.nied.unicamp.br/~siros/

Comando	Função
Iniciar	Inicia a variável de controle da direção da tartaruga
Pdireita90	Tartaruga gira em torno do próprio eixo 90° para direita
Pesquerda90	Tartaruga gira em torno do próprio eixo 90° para esquerda
Pdireita45	Tartaruga gira em torno do próprio eixo 45° para direita
Pesquerda45	Tartaruga gira em torno do próprio eixo 45° para esquerda
Pfrente [valor]	Tartaruga se movimenta na tela do SuperLogo e o plotter imprime no papel a trajetória equivalente à realizada pela tartaruga.

Procedimentos em SuperLogo

```
aprenda iniciar
atr "direcao 1
fim
```

```
aprenda pdireita45
Se :direcao<5 [atr "direcao :direcao+0.5]
Se :direcao=4.5 [atr "direcao 0.5]
pd 45
fim
```

```
aprenda pdireita90
Se :direcao<5 [atr "direcao :direcao+1]
Se :direcao=5 [atr "direcao 1]
pd 90
fim
```

```
aprenda pesquerda45
Se :direcao>0 [atr "direcao :direcao-0.5]
Se :direcao=0 [atr "direcao 4]
pe 45
fim
```

```
aprenda pesquerda90
Se :direcao>0 [atr "direcao :direcao-1]
Se :direcao=0 [atr "direcao 4]
pe 90
fim
```

```
aprenda pfrente :unid
atr "tempo 21
pf :unid
Se :direcao=0.5 [
  rcxmotorfrente [a]
  rcxrevertemotor [b]
  rcxligamotor [a b]
  Espere :tempo*:unid
  rcxdesligamotor [a b]
]
Se :direcao=1 [
  rcxrevertemotor [b]
  rcxligamotor [b]
  Espere :tempo*:unid
  rcxdesligamotor [b]
]
Se :direcao=1.5 [
  rcxrevertemotor [a b]
  rcxligamotor [a b]
  Espere :tempo*:unid
```

```

    rcxdesligamotor [a b]
]
Se :direcao=2 [
    rcxrevertemotor [a]
    rcxligamotor [a]
    Espere :tempo*:unid
    rcxdesligamotor [a]
]
Se :direcao=2.5 [
    rcxmotorfrente [b]
    rcxrevertemotor [a]
    rcxligamotor [a b]
    Espere :tempo*:unid
    rcxdesligamotor [a b]
]
Se :direcao=3 [
    rcxmotorfrente [b]
    rcxligamotor [b]
    Espere :tempo*:unid
    rcxdesligamotor [b]
]
Se :direcao=3.5 [
    rcxmotorfrente [a b]
    rcxligamotor [a b]
    Espere :tempo*:unid
    rcxdesligamotor [a b]
]
Se :direcao=4 [
    rcxmotorfrente [a]
    rcxligamotor [a]
    Espere :tempo*:unid
    rcxdesligamotor [a]
]
fim

```